

**Panos Pardalos
Michael Khachay
Vladimir Mazalov (Eds.)**

LNCS 13367

Mathematical Optimization Theory and Operations Research

**21st International Conference, MOTOR 2022
Petrozavodsk, Russia, July 2–6, 2022
Proceedings**



Springer

Founding Editors

Gerhard Goos

Karlsruhe Institute of Technology, Karlsruhe, Germany

Juris Hartmanis

Cornell University, Ithaca, NY, USA

Editorial Board Members

Elisa Bertino

Purdue University, West Lafayette, IN, USA

Wen Gao

Peking University, Beijing, China

Bernhard Steffen 

TU Dortmund University, Dortmund, Germany

Moti Yung 

Columbia University, New York, NY, USA

More information about this series at <https://link.springer.com/bookseries/558>

Panos Pardalos · Michael Khachay ·
Vladimir Mazalov (Eds.)

Mathematical Optimization Theory and Operations Research

21st International Conference, MOTOR 2022
Petrozavodsk, Russia, July 2–6, 2022
Proceedings

Editors

Panos Pardalos 
University of Florida
Gainesville, FL, USA

Vladimir Mazalov 
Karelia Research Centre RAS
Institute of Applied Mathematical Research
Petrozavodsk, Karelian Republic, Russia

Michael Khachay 
Krasovsky Institute of Mathematics
and Mechanics
Ekaterinburg, Russia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-09606-8 ISBN 978-3-031-09607-5 (eBook)
<https://doi.org/10.1007/978-3-031-09607-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the refereed proceedings of the 21st International Conference on Mathematical Optimization Theory and Operations Research (MOTOR 2022)¹ held during July 2–6, 2022, in the Karelia region, near Petrozavodsk, Russia.

MOTOR 2022 was the forth joint scientific event unifying a number of well-known that have been held in Ural, Siberia, and the Far East of Russia for a long time:

- The Baikal International Triennial School Seminar on Methods of Optimization and Their Applications (BITSS MOPT) established in 1969 by academician N. N. Moiseev, with 17 events held up to 2017,
- The All-Russian Conference on Mathematical Programming and Applications (MPA) established in 1972 by academician I. I. Eremin, with 15 events held up to 2015,
- The International Conference on Discrete Optimization and Operations Research (DOOR), which was organized nine times between 1996 and 2016,
- The International Conference on Optimization Problems and Their Applications (OPTA), which was organized seven time in Omsk between 1997 and 2018.

First three events of this series, MOTOR 2019², MOTOR 2020³, and MOTOR 2021⁴ were held in Ekaterinburg, Novosibirsk, and Irkutsk, Russia, respectively.

As per tradition, the main conference scope included, but was not limited to, mathematical programming, bi-level and global optimization, integer programming and combinatorial optimization, approximation algorithms with theoretical guarantees and approximation schemes, heuristics and meta-heuristics, game theory, optimal control, optimization in machine learning and data analysis, and their valuable applications in operations research and economics.

In response to the call for papers, MOTOR 2022 received 161 submissions. Out of 88 full papers considered for review (73 abstracts and short communications were excluded for formal reasons) only 21 papers were selected by the Program Committee (PC) for publication in this volume. Each submission was reviewed by at least three PC members or invited reviewers, experts in their fields, in order to supply detailed and helpful comments. In addition, the PC recommended the inclusion of 22 papers in the supplementary volume after their presentation and discussion during the conference and subsequent revision with respect to the reviewers' comments.

¹ <http://motor2022.krc.karelia.ru/en/section/1>.

² <http://motor2019.uran.ru>.

³ <http://math.nsc.ru/conference/motor/2020/>.

⁴ <https://conference.icc.ru/event/3/>.

The conference featured six invited lectures:

- Rentsen Enkhbat (Institute of Mathematics and Digital Technology, Mongolia), “Recent Advances in Sphere Packing Problem”
- Vladimir Marianov (Instituto Sistemas Complejos de Ingenieria, Universidad Católica de Chile, Chile), “Store Location and Agglomeration in Competitive and Non-Competitive Retail”
- Alexander S. Nesterov (HSE, St. Petersburg, Russia), “Matching Market Design Theory and Applications”
- Yaroslav D. Sergeev (University of Calabria, Italy), “Numerical Infinities and Infinitesimals in Optimization”
- Sergey Sevastyanov (Sobolev Institute of Mathematics, SB RAS, Russia) “Three Efficient Methods of Finding Near-Optimal Solution for NP-hard Discrete Optimization Problems (Illustrated by Their Application to Scheduling Problems)”
- Georges Zaccour (GERAD, HEC Montréal, Canada) “Coordination in Closed-Loop Supply Chains: a Dynamic Games Perspective”

The following two tutorials were given by outstanding scientists:

- Alexander Gasnikov (Moscow Institute of Physics and Technology, Russia), “Markov Decision Process and Convex Optimization”
- Evgenii Sopov (A.N. Antamoshkin Siberian Institute of Applied System Analysis, Russia), “Hyperheuristics for Automated Synthesis and Control of Evolutionary Optimization Algorithms”

We thank the authors for their submissions, members of the Program Committee, and all the external reviewers for their efforts in providing exhaustive reviews. We thank our sponsors and partners: the Institute of Applied Mathematical Research (IAMR) of the Karelian Research Centre, the Sobolev Institute of Mathematics, the Krasovsky Institute of Mathematics and Mechanics, the Ural Mathematical Center, the Center for Research and Education in Mathematics, the Higher School of Economics (Nizhny Novgorod), and the Matrosov Institute for System Dynamics and Control Theory. We are grateful to the colleagues from the Springer LNCS and CCIS editorial boards for their kind and helpful support.

July 2022

Panos Pardalos
Michael Khachay
Vladimir Mazalov

Organization

General Chair

Vladimir Mazalov
Institute of Applied Mathematical Research,
Russia

Honorary Chair

Panos Pardalos
University of Florida, USA

Program Committee Chairs

Michael Khachay
Krasovsky Institute of Mathematics and
Mechanics, Russia
Oleg Khamisov
Melentiev Institute of Energy Systems, Russia
Yury Kochetov
Sobolev Institute of Mathematics, Russia
Anton Eremeev
Omsk Branch of Sobolev Institute of
Mathematics, Russia

Program Committee

Anatoly Antipin
René van Bevern
Maxim Buzdalov
Igor Bykadorov
Dorodnicyn Computing Centre, RAS, Russia
Huawei Cloud Technologies Co., Ltd., Russia
ITMO University, Russia
Sobolev Institute of Mathematics, SB RAS,
Russia
Tatjana Davidović
Stephan Dempe
Adil Erzin
Mathematical Institute SANU, Serbia
TU Bergakademie Freiberg, Germany
Sobolev Institute of Mathematics, SB RAS,
Russia
Yuri G. Evtushenko
Stefka Fidanova
Dorodnicyn Computing Centre, RAS, Russia
Institute of Information and Communication
Technologies, BAS, Bulgaria
Fedor Fomin
Eduard Gimadi
University of Bergen, Norway
Sobolev Institute of Mathematics, SB RAS,
Russia
Evgeny Gurevsky
Feng-Jang Hwang
Sergey Ivanov
Université de Nantes, France
University of Technology Sydney, Australia
Moscow Aviation Institute, Russia

Milojica Jaćimović	Montenegrin Academy of Sciences and Arts, Montenegro
Valeriy Kalyagin	Higher School of Economics, Russia
Vadim Kartak	Ufa State Aviation Technical University, Russia
Alexander Kazakov	Matrosov Institute for System Dynamics and Control Theory, SB RAS, Russia
Lev Kazakovtsev	Siberian State Aerospace University, Russia
Igor Konnov	Kazan Federal University, Russia
Alexander Kononov	Sobolev Institute of Mathematics, SB RAS, Russia
Dmitri Kvasov	University of Calabria, Italy
Bertrand M. T. Lin	National Yang Ming Chiao Tung University, Taiwan
Vittorio Maniezzo	University of Bologna, Italy
Nenad Mladenović	Khalifa University, UAE
Yury Nikulin	University of Turku, Finland
Evgeni Nurminski	Far Eastern Federal University, Russia
Nicholas Olenev	Doronicyn Computing Centre, RAS, Russia
Leon Petrosyan	Saint Petersburg State University, Russia
Alex Petunin	Ural Federal University, Russia
Leonid Popov	Krasovsky Institute of Mathematics and Mechanics, Russia
Mikhail Posypkin	Dorodnicyn Computing Centre, RAS, Russia
Artem Pyatkin	Sobolev Institute of Mathematics, SB RAS, Russia
Soumyendu Raha	Indian Institute of Science, India
Yaroslav Sergeev	University of Calabria, Italy
Sergey Sevastyanov	Sobolev Institute of Mathematics, SB RAS, Russia
Natalia Shakhlevich	University of Leeds, UK
Aleksandr Shanenin	Moscow Institute Physics and Technology, Russia
Angelo Sifaleras	University of Macedonia, Macedonia
Vladimir Skarin	Krasovsky Institute of Mathematics and Mechanics, Russia
Alexander Strelakovski	Matrosov Institute for System Dynamics and Control Theory, SB RAS, Russia
Tatiana Tchemisova	University of Aveiro, Portugal
Raca Todosijevic	Université Polytechnique Hauts-de-France, France
Alexey Tret'yakov	Dorodnicyn Computing Centre, RAS, Russia

Additional Reviewers

Artemyeva, Liudmila	Konovalchikova, Elena	Sedakov, Artem
Baklanov, Artem	Kulachenko, Igor	Shkaberina, Guzel
Buzdalov, Maxim	Lebedev, Pavel	Simanchev, Ruslan
Chernykh, Ilya	Lempert, Anna	Sopov, Evgenii
Chirkova, Julia	Levanova, Tatyana	Stanimirovic, Zorica
Dang, Duc-Cuong	Lushchakova, Irina	Stanovov, Vladimir
Gasnikov, Alexander	Melnikov, Andrey	Takhonov, Ivan
Gluschenko, Konstantin	Nikitina, Natalia	Tarashev, Alexander
Gribanov, Dmitriy	Ogorodnikov, Yuri	Tovbis, Elena
Gusev, Mikhail	Pankratova, Yaroslavna	Tur, Anna
Il'Ev, Victor	Parilina, Elena	Ushakov, Anton
Ivashko, Anna	Plotnikov, Roman	Vasilyev, Igor
Jelisavcic, Vladisav	Plyasunov, Alexander	Vasin, Alexandr
Khachay, Daniel	Potapov, Mikhail	Yanovskaya, Elena
Khlopin, Dmitry	Pyatkin, Artem	Zakharova, Yulia
Khutoretskii, Alexander	Rettieva, Anna	Zubov, Vladimir
Kononova, Polina	Rybalov, Alexander	

Industry Section Chair

Vasilyev Igor	Matrosov Institute for System Dynamics and Control Theory, SB RAS, Russia
---------------	--

Organizing Committee

Vladimir Mazalov (Chair)	IAMR, Russia
Anna Rettieva (Deputy Chair)	IAMR, Russia
Yulia Chirkova (Scientific Secretary)	IAMR, Russia
Anna Ivashko	IAMR, Russia
Elena Parilina	Saint Petersburg State University, Russia
Polina Kononova	Sobolev Institute of Mathematics, SB RAS, Russia
Timur Medvedev	HSE, Nizhny Novgorod, Russia
Yuri Ogorodnikov	Krasovsky Institute of Mathematics and Mechanics, Russia

Organizers

Institute of Applied Mathematical Research (IAMR), Russia
 Sobolev Institute of Mathematics, SB RAS, Russia
 Mathematical Center in Akademgorodok, Novosibirsk, Russia

Krasovsky Institute of Mathematics and Mechanics, Russia
Ural Mathematical Center, Russia
Higher School of Economics, Nizhny Novgorod, Russia

Sponsors

Higher School of Economics, Nizhny Novgorod, Russia
Mathematical Center in Akademgorodok, Novosibirsk, Russia
Ural Mathematical Center, Russia

Abstracts of Invited Talks

On the Design of Matheuristics that make Use of Learning

Rentsen Enkhbat 

Institute of Mathematics and Digital Technology, Mongolia, Ulaanbaatar
renkhbat46@yahoo.com

Abstract. We consider a general sphere packing problem which is to pack non-overlapping spheres with the maximum volume into a convex set. This problem has important applications in science and technology and belongs to a class of global optimization.

In two dimensional case, the sphere packing problem is a classical circle packing problem. It has been shown that 200 years old Malfatti's problem [3] is a particular case of the circle packing problem [1, 2].

We survey existing theories and algorithms on general sphere packing problems. We also discuss their applications in economics and a mining industry.

Keywords: 2D packing · Global optimization · Malfatti's problem

References

1. Enhbat, R.: Global optimization approach to malfatti's problem. *J. Glob. Optim.* **65**, 33–39 (2016). <https://doi.org/10.1007/s10898-015-0372-6>
2. Enkhbat, R.: Convex Maximization Formulation of General Sphere Packing Problem, the Bulletin of Irkutsk State University'. Series 'Mathematics', vol. 31, pp.142–149 (2020)
3. Malfatti, G.: "Memoria sopra un problema stereotomico." *Memorie di matematica e fisica della Società Italiana delle Scienze* **10**(1), 235–244 (1803)

Store Location and Agglomeration in Competitive and non-Competitive Retail

Vladimir Marianov 

Pontificia Universidad Católica de Chile, Santiago, Chile
marianov@ing.puc.cl

Abstract. Agglomeration or clustering of stores can be observed in practice in the location of both competitive and non-competitive stores. What makes several shoe stores locate beside each other, or a shoe store locate close to a pants store? Hotelling in 1929 proposed an explanation; however, it only works under very special circumstances. Economists, transport and market researchers have found better explanations to this phenomenon: agglomeration is due to the savings obtained in multiple-stop shopping trips. Amazingly, these trips were only recently included into optimization models for optimal location, and their inclusion indeed changes the prescribed locations.

We study the effect of multiple-stop trips, both multipurpose shopping (MPS) and comparison shopping (CS), on location of retail stores. We analyze follower and (bi-level) leader-follower models for MPS and CS, in which customers use a binary, deterministic choice rule to decide to which store to go to make a purchase, and a MPS follower problem in which customers behave according to a random utility model.

Extensions are discussed.

Keywords: Bi-level programming · Facility location · Leader-follower model

Matching Market Design: Theory and Applications

Alexander S. Nesterov 

National Research University Higher School of Economics, St.Petersburg, Russia
nesterovu@gmail.com

Abstract. How do we match supply and demand when standard price mechanisms are not available? Examples include school choice, college admissions, organ allocation, social housing. In these cases, we use matching mechanisms that elicit the agents' preferences and then determine who gets what, so that the outcome is desirable by various standards. In this talk, I introduce the matching theory and present the results addressing the recent methodological difficulty: how to compare different matching mechanisms according to various desirable properties when the standard axiomatic approach is not applicable? I also present a practical case in point: the Russian college admissions system.

Keywords: Price mechanism · Agent's preference · Matching market design

Numerical Infinities and Infinitesimals in Optimization

Yaroslav D. Sergeyev 

University of Calabria, Rende, Italy
yaro@dimes.unical.it

Abstract. In this talk, a recent computational methodology is described. It has been introduced with the intention to allow one to work with infinities and infinitesimals numerically in a unique computational framework. It is based on the principle ‘The part is less than the whole’ applied to all quantities (finite, infinite, and infinitesimal) and to all sets and processes (finite and infinite). The methodology uses as a computational device the Infinity Computer (a new kind of supercomputer patented in several countries) working numerically with infinite and infinitesimal numbers that can be written in a positional system with an infinite radix. On a number of examples (numerical differentiation, divergent series, ordinary differential equations, fractals, set theory, etc.) it is shown that the new approach can be useful from both theoretical and computational points of view. The main attention is dedicated to applications in optimization (local, global, and multi-objective). The accuracy of the obtained results is continuously compared with results obtained by traditional tools used to work with mathematical objects involving infinity. The Infinity Calculator working with infinities and infinitesimals numerically is shown during the lecture.

Keywords: Finite · Infinite · Infinitesimal

Three Efficient Methods of Finding Near-Optimal Solution for NP-Hard Discrete Optimization Problems. Illustrated by their Application to Scheduling Problems

Sergey Sevastyanov 

Sobolev Institute of Mathematics, Novosibirsk, Russia
seva@math.nsc.ru

Abstract. Three fairly universal and efficient methods of finding near-optimal solutions for NP-hard problems will be discussed in our talk and illustrated by examples of their application to scheduling problems (although, they are surely applicable to a much wider area of Discrete Optimization problems):

1. Methods of ‘compact’ vector summation in a ball (of any norm) of minimum radius, and methods of non-strict summation of vectors in a given area of d -dimensional space. These methods are applicable to problems of finding uniform distributions of multicomponent objects.
2. Application of the maximum flow and the minimum cut algorithms to problems of finding uniform distributions of one-component objects with prescribed constraints on the distribution areas of objects.
3. The method of a gradual reduction of the feasible solutions domain.

Keywords: Compact vector summation · Method of a gradual reduction · Scheduling

Coordination in Closed-Loop Supply Chains: A Dynamic Games Perspective

Georges Zaccour 

GERAD, HEC Montreal, Canada
georges.zaccour@hec.ca

Abstract. Lack of coordination between the parties involved in a supply chain typically leads to lower outcomes for manufacturers and retailers, and to lower consumer surplus. Further, the collection of past-purchased products at the end of their useful life, for remanufacturing or recycling purposes, is a crucial activity in optimizing operations management and achieving a better environmental performance.

In this talk, I will discuss some coordination mechanisms that could be implemented to improve the efficiency of a closed-loop supply chain, in a context of long-term strategic interactions between the agents and in the presence of demand uncertainties.

Keywords: Coordination mechanism · Closed-loop supply chain · Uncertainty

Abstracts of Tutorials

Markov Decision Process and Convex Optimization

Alexander Gasnikov 

Moscow Institute of Physics and Technology, Russia
gasnikov@yandex.ru

Abstract. The problem of constrained Markov decision process is considered. An agent aims to maximize the expected accumulated discounted reward subject to multiple constraints on its costs (the number of constraints is small enough). A new dual approach is proposed with an integration of two ingredients: entropy regularized policy optimizer and Vayda's dual optimizer, all of which are critical to achieve a faster convergence. The finite-time error bound of the proposed approach is characterized. Despite the challenge of the nonconcave objective subject to nonconcave constraints, the proposed approach is shown to converge (with linear rate) to the global optimum with a complexity of in terms of the optimality gap and the constraint violation, which significantly improves the complexity of the existing primal-dual approach.

Keywords: Dual approach · Entropy regularized policy · Vayda's dual optimizer

Hyperheuristics for Automated Synthesis and Control of Evolutionary Optimization Algorithms

Evgenii Sopov 

Siberian Federal University, Krasnoyarsk, Russia
ESopov@sfu-kras.ru

Abstract. Evolutionary algorithms and other nature-inspired techniques have proved their efficiency in solving different hard optimization problems. At the same time, the number of different heuristics and meta-heuristics is still growing, and one must deal with the problem of selecting, fine-tuning, and combining simple heuristics to design an optimization algorithm for each specific task. In this tutorial, we will present a promising approach to automated synthesis and control of evolutionary algorithms using hyperheuristics. We will discuss novel evolutionary hyperheuristics and the experimental results for some classes of hard global “black-box” optimization problems, including multimodal, non-stationary and large-scale optimization.

Keywords: Metaheuristic · Evolutionary approach · Adaptive control

Contents

Mathematical Programming

On the Convergence Analysis of Aggregated Heavy-Ball Method	3
<i>Marina Danilova</i>	
Noisy Zeroth-Order Optimization for Non-smooth Saddle Point Problems	18
<i>Darina Dvinskikh, Vladislav Tominin, Iaroslav Tominin, and Alexander Gasnikov</i>	
Application of the Subdifferential Descent Method to a Classical Nonsmooth Variational Problem	34
<i>Alexander Fominyh</i>	
Primal-Dual Method for Optimization Problems with Changing Constraints	46
<i>Igor Konnov</i>	
Decentralized Convex Optimization Under Affine Constraints for Power Systems Control	62
<i>Demyan Yarmoshik, Alexander Rogozin, Oleg. O. Khamisov, Pavel Dvurechensky, and Alexander Gasnikov</i>	

Heuristics and Metaheuristics

Metaheuristic Approach to Spectral Reconstruction of Graphs	79
<i>Petar Ćirković, Predrag Đorđević, Miloš Milićević, and Tatjana Davidović</i>	
Variable Neighborhood Search for Multi-label Feature Selection	94
<i>Luka Matijević</i>	
Dispersion Problem Under Capacity and Cost Constraints: Multiple Neighborhood Tabu Search	108
<i>Nenad Mladenović, Raca Todosijević, and Dragan Urošević</i>	
Multiple Project Scheduling for a Network Roll-Out Problem: MIP Formulation and Heuristic	123
<i>Igor Vasilyev, Dmitry Rybin, Sergey Kudria, Jie Ren, and Dong Zhang</i>	

Applications

- On a Nonconvex Distance-Based Clustering Problem 139
Tatiana V. Gruzdeva and Anton V. Ushakov
- On Solving One Spectral Problem 153
Vladimir Zubov and Alla Albu

Mathematical Economy

- Optimal Arrivals to Preemptive Queueing System 169
Julia V. Chirkova and Vladimir V. Mazalov
- Multistage Inventory Model with Probabilistic and Quantile Criteria 182
Sergey V. Ivanov and Aleksandra V. Mamchur
- Pricing in Two-Sided Markets on the Plain with Different Agent Types 194
Elena Konovalchikova and Anna Ivashko
- On the Existence of a Fuzzy Core in an Exchange Economy 210
Valeriy Marakulin

Game Theory

- Value of Cooperation in a Differential Game of Pollution Control 221
*Angelina Chebotareva, Shimai Su, Elizaveta Voronina,
 and Ekaterina Gromova*
- A Cooperation Scheme in Multistage Game of Renewable Resource
 Extraction with Asymmetric Players 235
Denis Kuzyutin, Yulia Skorodumova, and Nadezhda Smirnova
- Two Level Cooperation in Dynamic Network Games with Partner Sets 250
Leon Petrosyan and Yaroslavna Pankratova
- Multicriteria Dynamic Games with Asymmetric Horizons 264
Anna Rettieva
- A Novel Payoff Distribution Procedure for Sustainable Cooperation
 in an Extensive Game with Payoffs at All Nodes 279
Denis Kuzyutin and Nadezhda Smirnova

The Core of Cooperative Differential Games on Networks 295
Anna Tur and Leon Petrosyan

Author Index 315

Mathematical Programming



On the Convergence Analysis of Aggregated Heavy-Ball Method

Marina Danilova^{1,2}(✉) 

¹ Institute of Control Sciences of RAS, Moscow, Russia
danilovamarina15@gmail.com

² Moscow Institute of Physics and Technology, Moscow, Russia

Abstract. Momentum first-order optimization methods are the workhorses in various optimization tasks, e.g., in the training of deep neural networks. Recently, Lucas et al. (2019) [7] proposed a method called Aggregated Heavy-Ball (AggHB) that uses multiple momentum vectors corresponding to different momentum parameters and averages these vectors to compute the update direction at each iteration. Lucas et al. (2019) [7] show that AggHB is more stable than the classical Heavy-Ball method even with large momentum parameters and performs well in practice. However, the method was analyzed only for quadratic objectives and for online optimization tasks under uniformly bounded gradients assumption, which is not satisfied for many practically important problems. In this work, we address this issue and propose the first analysis of AggHB for smooth objective functions in non-convex, convex, and strongly convex cases without additional restrictive assumptions. Our complexity results match the best-known ones for the Heavy-Ball method. We also illustrate the efficiency of AggHB numerically on several non-convex and convex problems.

Keywords: First-order methods · Momentum methods · Smooth optimization

1 Introduction

Momentum [14] and acceleration [10] are popular techniques for speeding up first-order optimization methods both from practical and theoretical perspectives. Historically, one of the first examples of such methods is Heavy-Ball (HB) method proposed by B. Polyak in 1964 [14]. This method received a lot of attention from various research communities due to its efficiency in different convex and, more importantly, non-convex problems [2]. In particular, during the last few years, a lot of variants of HB were proposed and analyzed by machine learning (ML) researchers, especially due to its efficiency in computer vision tasks [15].

The research was supported by Russian Foundation for Basic Research (Theorem 1, project No. 20-31-90073) and by Russian Science Foundation (Theorem 2, project No. 21-71-30005).

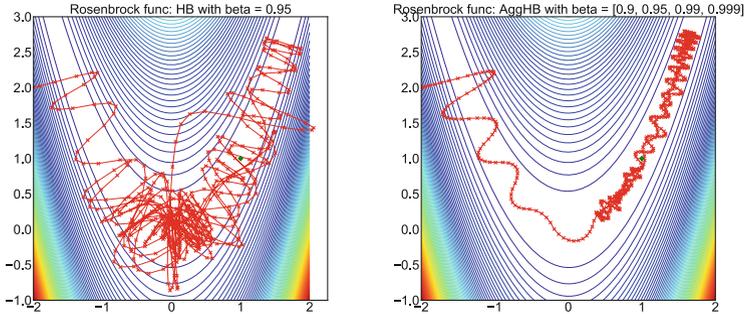


Fig. 1. Trajectories of HB (left) and AggHB (right) with different momentum parameters β applied to minimize Rosenbrock function. Stepsize γ was tuned for each method. We use the package from [12] for the visualization.

Recently, another modification of HB called Aggregated Heavy-Ball (AggHB) method was proposed in [7]. In contrast to HB, AggHB has $m \geq 1$ different momentum parameters and m corresponding momentum vectors. An average of these vectors is used as an update direction at each iteration. Such an averaging helps to make the method more stable via reducing the oscillations of the iterates, as the authors of [7] illustrated empirically. Moreover, the numerical results from [7] show the superiority of AggHB to HB at training several ML models.

1.1 Motivational Example

In this section, we consider the behavior of AggHB on Rosenbrock function, which is well-known non-convex test function. The set of momentum parameters for AggHB were chosen as $[0.9, 0.95, 0.99, 0.999]$ (see Algorithm 2) and for HB a standard momentum parameter $\beta = 0.95$ was taken (see Algorithm 1). Stepsize γ was tuned for each method. The results are presented in Fig. 1. We observe much smaller oscillations for AggHB than for HB. Moreover, the trajectory of AggHB achieves better accuracy. This example motivates the detailed study of AggHB and, in particular, the theoretical study of its convergence.

1.2 Our Contributions

However, little is known about theoretical convergence guarantees for AggHB. In particular, the authors of [7] analyzed AggHB for quadratic optimization problems, which is a very small class of problems, and for convex online optimization problems such that the gradients of the objective function are bounded on the whole domain. The former assumption is not satisfied for many practically important tasks. *In this paper, we remove this limitation and derive new convergence results for AggHB for smooth non-convex and (strongly) convex problems.*

Our main contributions can be summarized as follows.

- ◇ **First analysis of AggHB for non-convex problems.** For the problems with smooth but not necessary convex objective function f , we prove that AggHB finds an ε -stationary point (point x such that $\|\nabla f(x)\| \leq \varepsilon$) after $\mathcal{O}(1/\varepsilon^2)$ iterations neglecting the dependence on momentum parameters, smoothness constant, and initial functional suboptimality. When $m = 1$ we recover the complexity of HB and when $m > 1$ our rate is better than the corresponding rate of HB with maximal momentum parameter (see Theorem 1 and Corollary 1 for the details).
- ◇ **First analysis of AggHB without bounded gradient assumption.** In the smooth (strongly) convex case, we derive the first complexity upper bounds for AggHB without assuming that the gradients are uniformly bounded. As in the non-convex case, we recover the complexity of HB when $m = 1$ and our rate is better than the corresponding rate of HB with maximal momentum parameter when $m > 1$ (see Theorem 2 and Corollary 2).
- ◇ **Numerical experiments.** We compare the performance of AggHB and HB on the logistic regression problem with ℓ_2 -regularization and special non-convex regularization. In our experiments, AggHB converges faster than HB.

1.3 Technical Preliminaries

We consider an unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1)$$

where function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is L -smooth, i.e., for all $x, y \in \mathbb{R}^n$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2. \quad (2)$$

Next, we assume that $f(x)$ is either bounded from below $f_{\inf} = \inf_{x \in \mathbb{R}^n} f(x) > -\infty$ or μ -strongly convex

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|_2^2. \quad (3)$$

The notation we use is standard for optimization literature [11, 13], e.g., by x_* we denote the solution of (1), the distance from the starting point to the solution is denoted by $R_0 = \|x_0 - x_*\|_2$.

1.4 Related Work

Theoretical convergence guarantees for HB. The first convergence analysis of Heavy-Ball method (HB, Algorithm 1) was given in the original work by B. Polyak in 1964 [14], where *local* $\mathcal{O}(\sqrt{L/\mu} \log(1/\varepsilon))$ convergence rate was shown for twice continuously differentiable L -smooth and μ -strongly convex functions. After 50 years Ghadimi et al. (2015) [5] derived the first *global* convergence rates for HB (and its version with averaging). In particular, they shown $\mathcal{O}(L/\mu \log(1/\varepsilon))$ and $\mathcal{O}(LR_0^2/\varepsilon)$ complexity bounds for L -smooth μ -strongly convex and convex

Algorithm 1. Heavy-Ball method (HB)

Input: starting points x_0, x_1 (by default $x_0 = x_1$), number of iterations N , stepsize $\gamma > 0$, momentum parameter $\beta \in [0, 1]$

- 1: **for** $k = 0, \dots, N - 1$ **do**
- 2: $V_k = \beta V_{k-1} + \nabla f(x_k)$
- 3: $x_{k+1} = x_k - \gamma V_k$
- 4: **end for**

Output: x_N

functions respectively. In contrast to the local convergence guarantees, these rates are not accelerated [9, 10]. Although one can improve the analysis of HB for quadratic functions and get an asymptotically accelerated rate [6], it is still unclear whether this result can be generalized to the general non-quadratic functions. The non-triviality of this question is supported by the negative result from [16] showing that one cannot derive an accelerated rate of HB for the standard choice of parameters using quadratic potentials in the analysis.

HB with aggregation and averaging. As we already mentioned, Aggregated Heavy-Ball method (AggHB, Algorithm 2) was proposed in [7], where authors have empirically shown that aggregation helps to stabilize the behavior of the methods, speeds up the method in practice, and they also derive some convergence guarantees under uniformly bounded gradients assumption in the stochastic case. Recently, in [3], another approach for stabilizing HB was considered. In particular, the authors of [3] considered several averaging techniques for HB and showed that they help to reduce the maximal deviation of the method and improve the performance of the method in practice.

2 Analysis of Aggregated Heavy-Ball Method

In this section we propose a new convergence analysis for Aggregated Heavy-Ball method (AggHB, Algorithm 2). The key difference between HB and AggHB is that instead of one direction determined by parameter β the method uses to the vector of momentum parameters $\beta = [\beta_1, \dots, \beta_m]$ and takes an average over m corresponding directions. When $m = 1$ AggHB recovers HB. Moreover, we consider a slight generalization of the method proposed in [7], since we allow to use different stepsizes for different momentum parameters.

Following [8, 17] we consider *perturbed/virtual* iterates:

$$\tilde{x}_k = x_k - \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} V_{k-1}^{(i)}, \quad k \geq 0. \quad (4)$$

This representation is used for the analysis only and there is no need to compute this sequence when running the method. Virtual iterates satisfy the following useful recursion: for all $k \geq 0$

Algorithm 2. Aggregated Heavy-Ball method (AggHB)

Input: number of iterations N , stepsize $\gamma_i > 0$, momentum parameters $\{\beta_i\}_{i=1}^m \in [0, 1]$, starting points x_0, x_1 (by default $x_1 = x_0 - \alpha \nabla f(x_0)$)

1: **for** $k = 1, \dots, N - 1$ **do**

2: $V_k^{(i)} = \beta_i V_{k-1}^{(i)} + \nabla f(x_k)$ for $i = 1, \dots, m$

3: $x_{k+1} = x_k - \frac{1}{m} \sum_{i=1}^m \gamma_i V_k^{(i)}$

4: **end for**

Output: x_N

$$\begin{aligned}
 \tilde{x}_{k+1} &= x_{k+1} - \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} V_k^{(i)} = x_k - \frac{1}{m} \sum_{i=1}^m \gamma_i V_k^{(i)} - \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} V_k^{(i)} \\
 &= x_k - \frac{1}{m} \sum_{i=1}^m \frac{\gamma_i V_k^{(i)}}{1 - \beta_i} = x_k - \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} V_{k-1}^{(i)} - \frac{1}{m} \sum_{i=1}^m \frac{\gamma_i}{1 - \beta_i} \nabla f(x_k) \\
 &= \tilde{x}_k - \frac{1}{m} \sum_{i=1}^m \frac{\gamma_i}{1 - \beta_i} \nabla f(x_k). \tag{5}
 \end{aligned}$$

2.1 Non-convex Case

Below we present our main convergence result¹ for non-convex problems.

Theorem 1. *Let be f is L -smooth and possibly non-convex function with values lower bounded by f_{\inf} . Assume that*

$$-\frac{A}{2} \left(1 - \frac{CDEL^2}{2m^2} - LA \right) < 0, \tag{6}$$

where

$$A = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i}, \quad C = \sum_{i=1}^m \frac{\gamma_i}{(1 - \beta_i)^2}, \quad D = \max_{i=1, m} \frac{\gamma_i}{1 - \beta_i}, \quad E = \sum_{i=1}^m \frac{1}{1 - \beta_i}. \tag{7}$$

Then, for all $K \geq 1$ we have

$$\min_{k=1, K} \|\nabla f(x_k)\|_2^2 \leq \frac{2}{K} \frac{f(x_0) - f_{\inf}}{A \left(1 - \frac{CDEL^2}{m^2} - LA \right)}. \tag{8}$$

The above result provides a convergence guarantee in the general non-convex case and allows to use different γ_i such that (6) holds. To illustrate this result and, in particular, condition (6) we derive the following corollary² of Theorem 1.

¹ We defer all the proofs to the Appendix.

² Due to the space limitation, we omit the proofs of Corollaries 1 and 2 that can be verified via simple computations. We provide them in the arXiv version of our paper.

Corollary 1. *Let the assumptions of Theorem 1 hold. Assume that the stepsize is constant $\gamma_i \equiv \gamma$ for $i = 1, \dots, m$ and consider new constants $\tilde{\beta}$ and $\hat{\beta}$ satisfying the following conditions: $\frac{1}{m} \sum_{i=1}^m \frac{\beta_i}{(1-\beta_i)^2} = \frac{\tilde{\beta}}{(1-\tilde{\beta})^2}$, $\frac{1}{m} \sum_{i=1}^m \frac{1}{1-\beta_i} = \frac{1}{1-\hat{\beta}}$. Let*

$$\gamma = \frac{1}{L \left(\frac{2\hat{\beta}}{1-\hat{\beta}} + \sqrt{2 \left(\frac{\tilde{\beta}}{(1-\tilde{\beta})^2} + \frac{1}{1-\hat{\beta}} \right) \frac{1}{\left(1 - \max_{i=1,m} \beta_i\right) (1-\hat{\beta})}} \right)}.$$

Then, to achieve $\min_{k=1,K} \|\nabla f(x_k)\|_2^2 \leq \varepsilon^2$ for $\varepsilon > 0$ AggHB requires

$$\mathcal{O} \left(\frac{L(f(x_0) - f_{\inf})}{\varepsilon^2} + \frac{L(f(x_0) - f_{\inf}) \sqrt{\left(\frac{\tilde{\beta}(1-\hat{\beta})}{(1-\tilde{\beta})^2} + 1 \right) \frac{1}{\left(1 - \max_{i=1,m} \beta_i\right) \hat{\beta}^2}}}{\varepsilon^2} \right). \quad (9)$$

First of all, when $m = 1$, we have $\beta = \tilde{\beta} = \hat{\beta} = \max_{i=1,m} \beta_i$ and the above convergence rate can be simplified to

$$\mathcal{O} \left(\frac{L(f(x_0) - f_{\inf})}{\varepsilon^2} + \frac{L(f(x_0) - f_{\inf})}{\varepsilon^2 \beta (1 - \beta)} \right)$$

that matches the rate of HB in the non-convex case (e.g., see [4]). Next, constants $\tilde{\beta}$ and $\hat{\beta}$ can be viewed as special ‘‘averaged’’ momentum parameters. Indeed, we know that

$$\begin{aligned} \frac{\min_{i=1,m} \beta_i}{(1 - \min_{i=1,m} \beta_i)^2} &\leq \frac{1}{m} \sum_{i=1}^m \frac{\beta_i}{(1-\beta_i)^2} \leq \frac{\max_{i=1,m} \beta_i}{(1 - \max_{i=1,m} \beta_i)^2}, \\ \frac{1}{1 - \min_{i=1,m} \beta_i} &\leq \frac{1}{m} \sum_{i=1}^m \frac{1}{1-\beta_i} \leq \frac{1}{1 - \max_{i=1,m} \beta_i}, \end{aligned}$$

since $\frac{x}{(1-x)^2}$ and $\frac{1}{1-x}$ are increasing functions for $x \in (0, 1)$, i.e., $\tilde{\beta}, \hat{\beta}$ lie in $[\min_{i=1,m} \beta_i, \max_{i=1,m} \beta_i]$. This allows to use larger stepsize than maximal possible stepsize for HB with $\beta = \max_{i=1,m} \beta_i$, i.e., the rate of AggHB is better than the one of HB with $\beta = \max_{i=1,m} \beta_i$.

2.2 Convex and Strongly-Convex Cases

Lemma 1. *Let f is L -smooth and μ -strongly convex. Let γ_i and β_i satisfy $\gamma_i > 0$, $\beta_i \in [0, 1)$, and*

$$F = \frac{1}{m} \sum_{i=1}^m \frac{\gamma_i}{1 - \beta_i} \leq \frac{1}{4L}. \quad (10)$$

Then, for all $k \geq 0$

$$\frac{F}{2} (f(x_k) - f(x_*)) \leq \left(1 - \frac{F\mu}{2}\right) \|\tilde{x}_k - x_*\|_2^2 - \|\tilde{x}_{k+1} - x_*\|_2^2 + 3LF \|x_k - \tilde{x}_k\|_2^2. \quad (11)$$

Next, it is sufficient to sum up (11) for $k = 0, 1, \dots, K$ with weights $w_k = (1 - \mu F/2)^{-(k+1)}$, $W_k = \sum_{k=0}^K w_k$ to get the bound on $f(\bar{x}_K) - f(x_*)$, where $\bar{x}_K = \frac{1}{W_K} \sum_{i=1}^K w_k (f(x_k) - f(x_*))$. To get final result one needs to upper bound the sum $3LF \sum_{k=0}^K w_k \|x_k - \tilde{x}_k\|_2^2$. For this we consider the following lemma.

Lemma 2. *Assume that f is L -smooth and μ -strongly convex. Let γ_i and β_i satisfy*

$$0 < \gamma_i \leq \frac{(1 - \max_{i=1,m} \beta_i)(1 - \beta_i)}{2\mu}, \quad \beta_i \in [0, 1), \quad (12)$$

$$F = \frac{1}{m} \sum_{i=1}^m \frac{\gamma_i}{1 - \beta_i} \leq \frac{1}{4L}, \quad BF \leq \frac{1 - \max_{i=1,m} \beta_i}{48L^2}, \quad (13)$$

where $B = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i (1 - \beta_i^{K+1})}{(1 - \beta_i)^2}$. Then, for all $k \geq 0$ and $w_k = (1 - \mu F/2)^{-(k+1)}$

$$3LF \sum_{k=0}^K w_k \|x_k - \tilde{x}_k\|_2^2 \leq \frac{F}{4} \sum_{k=0}^K w_k (f(x_k) - f(x_*)) \quad (14)$$

Combining these lemmas, we get the main result in (strongly) convex case.

Theorem 2. *Assume that f is L -smooth and μ -strongly convex. Let γ_i and β_i satisfy conditions from (12) and (13). Then, after $K \geq 0$ iterations of AggHB we have*

$$f(\bar{x}_K) - f(x_*) \leq \frac{4\|x_0 - x_*\|_2^2}{FW_K}, \quad \bar{x}_K = \frac{1}{W_K} \sum_{i=1}^K w_k (f(x_k) - f(x_*)) \quad (15)$$

where $w_k = (1 - \mu F/2)^{-(k+1)}$, $W_k = \sum_{k=0}^K w_k$, i.e.,

$$f(\bar{x}_K) - f(x_*) \leq \left(1 - \frac{\mu F}{2}\right)^K \frac{4\|x_0 - x_*\|_2^2}{F}, \quad \text{if } \mu > 0, \quad (16)$$

$$f(\bar{x}_K) - f(x_*) \leq \frac{4\|x_0 - x_*\|_2^2}{FK}, \quad \text{if } \mu = 0. \quad (17)$$

As in the non-convex case, the above result gives convergence guarantees in the general convex and strongly convex cases and allows to use different γ_i such that (12) and (13) hold. To illustrate this result and, in particular, conditions (12) and (13) we derive the following corollary of Theorem 2.

Corollary 2. *Let the assumptions of Theorem 2 hold. Assume that the stepsize is constant $\gamma_i \equiv \gamma$ for $i = 1, \dots, m$ and consider constants $\tilde{\beta}$ and $\hat{\beta}$ satisfying the following conditions: $\frac{1}{m} \sum_{i=1}^m \frac{\beta_i}{(1 - \beta_i)^2} = \frac{\tilde{\beta}}{(1 - \tilde{\beta})^2}$, $\frac{1}{m} \sum_{i=1}^m \frac{1}{1 - \beta_i} = \frac{1}{1 - \hat{\beta}}$. Let*

$$\gamma = \min \left\{ \frac{\left(1 - \max_{i=1,m} \beta_i\right)^2}{2\mu}, \frac{1 - \hat{\beta}}{4L}, \frac{(1 - \tilde{\beta}) \sqrt{(1 - \hat{\beta}) \left(1 - \max_{i=1,m} \beta_i\right)}}{4\sqrt{3}L\sqrt{\tilde{\beta}}} \right\}.$$

Then, to achieve $f(\bar{x}_K) - f(x_*) \leq \varepsilon$ for $\varepsilon > 0$ AggHB requires

$$\mathcal{O}\left(\left(\frac{L}{\mu} + \frac{1 - \hat{\beta}}{\left(1 - \max_{i=1,m} \beta_i\right)^2} + \frac{L\sqrt{\tilde{\beta}(1 - \hat{\beta})}}{\mu(1 - \tilde{\beta})\sqrt{1 - \max_{i=1,m} \beta_i}}\right) \cdot \ln\left(\frac{R_0^2}{\varepsilon} \cdot \left(L + \frac{1 - \hat{\beta}}{\left(1 - \max_{i=1,m} \beta_i\right)^2} + \frac{L\sqrt{\tilde{\beta}(1 - \hat{\beta})}}{(1 - \tilde{\beta})\sqrt{1 - \max_{i=1,m} \beta_i}}\right)\right)\right) \quad (18)$$

iterations when $\mu > 0$, and

$$\mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{LR_0^2\sqrt{\tilde{\beta}(1 - \hat{\beta})}}{\varepsilon(1 - \tilde{\beta})\sqrt{1 - \max_{i=1,m} \beta_i}}\right) \quad (19)$$

iterations when $\mu = 0$, where $R_0 = \|x_0 - x_*\|_2$.

First of all, when $m = 1$, we have $\beta = \tilde{\beta} = \hat{\beta} = \max_{i=1,m} \beta_i$ and the above convergence rates can be simplified to

$$\begin{aligned} \mathcal{O}\left(\left(\frac{L}{\mu} + \frac{L\sqrt{\beta}}{\mu(1 - \beta)}\right) \log\left(\frac{R_0^2}{\varepsilon} \cdot \left(L + \frac{L\sqrt{\beta}}{1 - \beta}\right)\right)\right), & \text{ when } \mu > 0, \\ \mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{LR_0^2\sqrt{\beta}}{\varepsilon(1 - \beta)}\right), & \text{ when } \mu = 0 \end{aligned}$$

that matches the rate of HB in the strongly convex and convex cases (e.g., see [5]). Next, as we already mentioned before, constants $\tilde{\beta}$ and $\hat{\beta}$ can be viewed as special ‘‘averaged’’ momentum parameters. This allows to use larger stepsize than maximal possible stepsize for HB with $\beta = \max_{i=1,m} \beta_i$, i.e., the rate of AggHB is better than the one of HB with $\beta = \max_{i=1,m} \beta_i$.

3 Numerical Experiments

We compare the behavior of HB and AggHB on solving logistic regression problem with ℓ_2 -regularization and with special non-convex regularization:

$$\min_{x \in \mathbb{R}^n} \left\{ f(x) = \frac{1}{M} \sum_{i=1}^M \log(1 + \exp(-y_i \cdot [Ax]_i)) + \frac{l_2}{2} \|x\|_2^2 \right\}, \quad (20)$$

$$\min_{x \in \mathbb{R}^n} \left\{ f(x) = \frac{1}{M} \sum_{i=1}^M \log(1 + \exp(-y_i \cdot [Ax]_i)) + \lambda \sum_{j=1}^n \frac{x_j^2}{1 + x_j^2} \right\}, \quad (21)$$

where M denotes the number of samples in the dataset, $A \in \mathbb{R}^{M \times n}$ is a ‘‘feature matrix’’, $y_1, \dots, y_M \in \{-1, 1\}$ are labels, and $l_2, \lambda \geq 0$ are the regularization

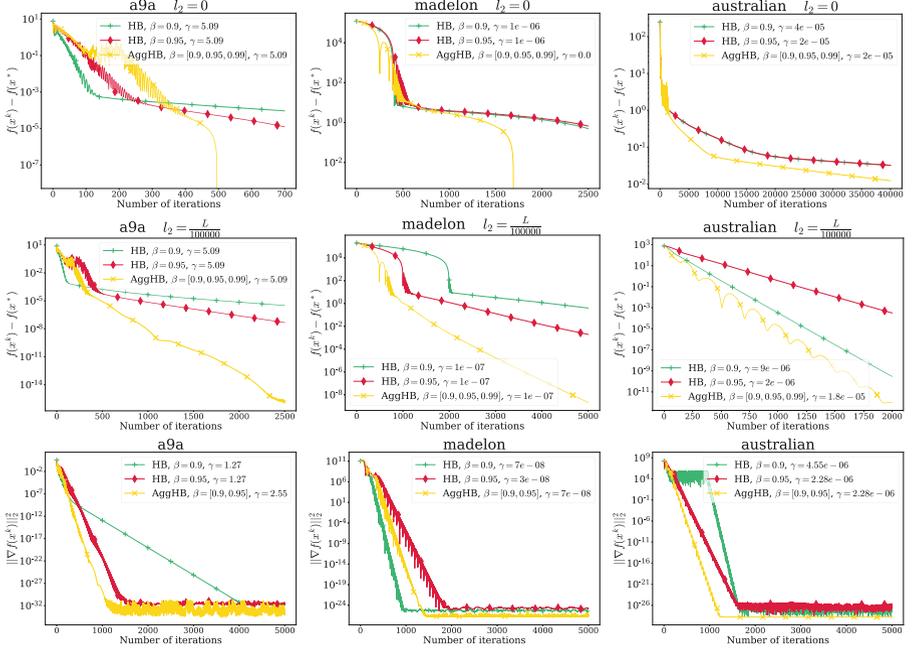


Fig. 2. Trajectories of HB and AHB with different momentum parameters β applied to solve logistic regression problem with ℓ_2 -regularization (the first two rows) and non-convex regularization (the third row) for **a9a**, **madelon**, and **australian** datasets. Stepsize γ was tuned for each method.

parameters. One can show that $f(x)$ is L -smooth and μ -strongly convex with $L = \frac{1}{4M} \lambda_{\max}(A^\top A) + l_2$ and $\mu = l_2$ in the first case, and L -smooth and non-convex with $L = \frac{1}{4M} \lambda_{\max}(A^\top A) + 2\lambda$. To construct the problems we use the following datasets from LIBSVM [1]: **a9a** ($M = 32561$, $n = 123$), **madelon** ($M = 2000$, $n = 500$), and **australian** ($M = 690$, $n = 14$). Regularization parameter l_2 is either 0 (convex problem) or $\frac{L}{100000}$ (strongly convex problem) and λ is chosen as $\lambda = \frac{L}{1000}$. We run HB with standard momentum parameters $\beta = 0.9, 0.95$ for both problems. AggHB was tested with $m = 3$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\beta_3 = 0.99$, and $\gamma_1 = \gamma_2 = \gamma_3 = \gamma$ for ℓ_2 -regularized problem and with $m = 2$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\gamma_1 = \gamma_2 = \gamma$. For each method we tune stepsize parameter γ as follows: we choose $\gamma = \frac{a}{L}$ with the best $a \in \{2^{-6}, 2^{-5}, 2^{-4}, \dots, 2^8\}$, i.e., the method achieves the best accuracy with the chosen a from the considered set.

The results are shown in the Fig. 2. We observe that AggHB outperforms HB in all cases. In particular, for ℓ_2 -regularized problem the large value of β_3 does not slow down the convergence of AggHB. In contrast, we observed that HB performs relatively bad with $\beta = 0.99$. Next, in the experiments with non-convex regularization, AggHB takes the best from two choices of momentum parameters.

4 Conclusion

In this paper, we obtain the first convergence guarantees for **AggHB** without assuming that the gradients of the objective function are uniformly bounded. In the special case when $m = 1$, our results recover the known ones for **HB** and outperform the corresponding guarantees for **HB** with $\beta = \max_{i=1,m} \beta_i$ when $m > 1$. Our numerical results show the superiority of **AggHB** to **HB**. Together with the results from [7] they indicate high practical potential of **AggHB**.

A Missing Proofs from Section 2

A.1 Proof of Theorem 1

From L -smoothness of f we have

$$f(\tilde{x}_{k+1}) \leq f(\tilde{x}_k) - A \langle \nabla f(\tilde{x}_k), \nabla f(x_k) \rangle + \frac{LA^2}{2} \|\nabla f(x_k)\|_2^2, \quad (22)$$

where $A = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i}{1-\beta_i} \gamma_i$. Next, we estimate a second term $-A \langle \nabla f(\tilde{x}_k), \nabla f(x_k) \rangle$ in the previous expression:

$$\begin{aligned} -A \langle \nabla f(\tilde{x}_k), \nabla f(x_k) \rangle &= A \frac{1}{2} (\|\nabla f(\tilde{x}_k) - \nabla f(x_k)\|_2^2 - \|\nabla f(\tilde{x}_k)\|_2^2 - \|\nabla f(x_k)\|_2^2) \\ &\stackrel{(22)}{\leq} \frac{A}{2} (L^2 \|\tilde{x}_k - x_k\|_2^2 - \|\nabla f(x_k)\|_2^2) \\ &\stackrel{(4)}{=} \frac{AL^2}{2m^2} \left\| \sum_{i=1}^m \frac{\beta_i \gamma_i}{1-\beta_i} V_{k-1}^{(i)} \right\|_2^2 - \frac{A}{2} \|\nabla f(x_k)\|_2^2. \end{aligned} \quad (23)$$

From **AggHB** update rule we know that $V_k^{(i)}$ is linear combination of gradients: $V_k^{(i)} = \sum_{l=0}^k (\beta_i)^l \nabla f(x_{k-l})$. Applying this to (23) we have

$$\frac{AL^2}{2m^2} \left\| \sum_{i=1}^m \frac{\beta_i \gamma_i}{1-\beta_i} V_{k-1}^{(i)} \right\|_2^2 \leq \frac{AL^2 B}{2m^2} \sum_{l=0}^{k-1} \|\nabla f(x_{k-1-l})\|_2^2 \sum_{i=1}^m \frac{(\beta_i)^l \gamma_i}{1-\beta_i}, \quad (24)$$

where $B = \sum_{l=0}^{k-1} \sum_{i=1}^m \frac{(\beta_i)^l \gamma_i}{1-\beta_i} \leq \sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2}$. Combining (23), (24), we continue the derivation from (22):

$$\begin{aligned} f(\tilde{x}_{k+1}) &\leq f(\tilde{x}_k) - \frac{A}{2} (1-LA) \|\nabla f(x_k)\|_2^2 + \frac{AL^2 B}{2m^2} \sum_{l=0}^{k-1} \|\nabla f(x_{k-1-l})\|_2^2 \sum_{i=1}^m \frac{(\beta_i)^l \gamma_i}{1-\beta_i} \\ &\leq f(\tilde{x}_k) - \frac{A}{2} (1-LA) \|\nabla f(x_k)\|_2^2 \\ &\quad + \frac{AL^2}{2m^2} \left(\sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2} \right) \sum_{l=0}^{k-1} \|\nabla f(x_l)\|_2^2 \sum_{i=1}^m \frac{(\beta_i)^{k-1-l} \gamma_i}{1-\beta_i} \\ &\leq f(\tilde{x}_k) - \frac{A}{2} (1-LA) \|\nabla f(x_k)\|_2^2 \\ &\quad + \frac{AL^2}{2m^2} \left(\sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2} \right) \left(\max_{i=1,m} \frac{\gamma_i}{1-\beta_i} \right) \sum_{l=0}^{k-1} \sum_{i=1}^m (\beta_i)^{k-1-l} \|\nabla f(x_l)\|_2^2. \end{aligned} \quad (25)$$

Summing up (25) for $k = 0, 1, \dots, K$ we get

$$\begin{aligned}
 f(\tilde{x}_{k+1}) &\leq f(\tilde{x}_0) + \sum_{k=1}^K \left(\frac{LA^2 - A}{2} \right) \|\nabla f(x_k)\|_2^2 \\
 &\quad + \sum_{k=1}^K \left(\frac{AL^2}{2m^2} \left(\sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2} \right) \left(\max_{i=1,m} \frac{\gamma_i}{1-\beta_i} \right) \sum_{l=k+1}^m \sum_{i=l}^{K-1} \beta_i^{l-1-k} \right) \|\nabla f(x_k)\|_2^2 \\
 &\leq f(\tilde{x}_0) + \sum_{k=1}^K \left(\frac{LA^2 - A}{2} \right) \|\nabla f(x_k)\|_2^2 \\
 &\quad + \sum_{k=1}^K \left(\frac{AL^2}{2m^2} \left(\sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2} \right) \left(\max_{i=1,m} \frac{\gamma_i}{1-\beta_i} \right) \sum_{i=1}^m \frac{1}{1-\beta_i} \right) \|\nabla f(x_k)\|_2^2 \\
 &= f(\tilde{x}_0) + \sum_{k=1}^K \left(\left(\frac{LA^2 - A}{2} \right) - \frac{ACDEL^2}{2m^2} \right) \|\nabla f(x_k)\|_2^2,
 \end{aligned}$$

where $A = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1-\beta_i}$, $C = \sum_{i=1}^m \frac{\gamma_i}{(1-\beta_i)^2}$, $D = \max_{i=1,m} \frac{\gamma_i}{1-\beta_i}$, $E = \sum_{i=1}^m \frac{1}{1-\beta_i}$. Finally, by choosing sufficiently small γ_i one can ensure that $-\frac{A}{2} \left(1 - \frac{CDEL^2}{2m^2} - LA \right) \leq 0$ and get (8)

A.2 Proof of Lemma 1

Applying the virtual iterates determined in (4), we obtain

$$\begin{aligned}
 \|\tilde{x}_{k+1} - x_*\|_2^2 &= \|\tilde{x}_k - x_*\|_2^2 - 2F \langle \tilde{x}_k - x_*, \nabla f(x_k) \rangle + F^2 \|\nabla f(x_k)\|_2^2 \\
 &= \|\tilde{x}_k - x_*\|_2^2 - 2F \langle x_k - x_*, \nabla f(x_k) \rangle - 2F \langle \tilde{x}_k - x_k, \nabla f(x_k) \rangle \\
 &\quad + F^2 \|\nabla f(x_k)\|_2^2.
 \end{aligned} \tag{26}$$

From μ -strong convexity and L -smoothness of f we have (e.g., see [11])

$$\begin{aligned}
 \langle x_k - x_*, \nabla f(x_k) \rangle &\geq f(x_k) - f(x_*) + \frac{\mu}{2} \|x_k - x_*\|^2 \\
 \|\nabla f(x_k)\|_2^2 &\leq 2L (f(x_k) - f(x_*)).
 \end{aligned} \tag{27}$$

Using these inequalities for (26) we get

$$\begin{aligned}
 \|\tilde{x}_{k+1} - x_*\|_2^2 &\leq \|\tilde{x}_k - x_*\|_2^2 - \mu F \|x_k - x_*\|_2^2 - 2F (f(x_k) - f(x_*)) \\
 &\quad - 2F \langle \tilde{x}_k - x_k, \nabla f(x_k) \rangle + F^2 \|\nabla f(x_k)\|_2^2.
 \end{aligned}$$

Firstly, we evaluate the second term $-\mu F \|x_k - x_*\|_2^2$ using that $\|a + b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$ for all $a, b \in \mathbb{R}^n$ as follows

$$-\mu F \|x_k - x_*\|_2^2 \leq -\frac{\mu F}{2} \|\tilde{x}_k - x_*\|_2^2 + \mu F \|x_k - \tilde{x}_k\|_2^2.$$

Secondly, we estimate the fourth term $-2F\langle \tilde{x}_k - x_k, \nabla f(x_k) \rangle$ using Fenchel-Young inequality³ and get

$$\begin{aligned} -2F\langle \tilde{x}_k - x_k, \nabla f(x_k) \rangle &\leq -2LF\|\tilde{x}_k - x_k\|_2^2 + \frac{F}{2L}\|\nabla f(x_k)\|_2^2 \\ &\stackrel{(27)}{\leq} -2LF\|\tilde{x}_k - x_k\|_2^2 + F\|(f(x_k) - f(x_*))\|. \end{aligned}$$

Combining the results above, we finish the proof

$$\|\tilde{x}_{k+1} - x_*\|_2^2 \stackrel{(4),(11)}{\leq} \left(1 - \frac{\mu F}{2}\right) \|\tilde{x}_k - x_*\|_2^2 - \frac{F}{2}(f(x_k) - f(x_*)) + 3LF\|x_k - \tilde{x}_k\|_2^2.$$

A.3 Proof of Lemma 2

From AggHB update rule we know that $V_k^{(i)}$ is linear combination of gradients: $V_k^{(i)} = \sum_{t=0}^k \beta_i^t \nabla f(x_{k-t})$. Next, by the definition of \tilde{x}_k we have

$$\begin{aligned} \|x_{k+1} - \tilde{x}_{k+1}\|_2^2 &= \left\| \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} V_k^{(i)} \right\|_2^2 = \left\| \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} \sum_{t=0}^k \beta_i^t \nabla f(x_{k-t}) \right\|_2^2 \\ &= \left\| \sum_{t=0}^k \left(\frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{t+1} \gamma_i}{1 - \beta_i} \right) \nabla f(x_{k-t}) \right\|_2^2 \\ &= \left\| \sum_{t=0}^k \left(\frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{k-t+1} \gamma_i}{1 - \beta_i} \right) \nabla f(x_t) \right\|_2^2. \end{aligned} \quad (28)$$

Define constant B_k as following

$$B_k = \sum_{t=0}^k \frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{k-t+1} \gamma_i}{1 - \beta_i} = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i}{1 - \beta_i} \sum_{t=0}^k \beta_i^{k-t} = \frac{1}{m} \sum_{i=1}^m \frac{\beta_i \gamma_i (1 - \beta_i^{k+1})}{(1 - \beta_i)^2}.$$

Using this, we continue the derivation from (28)

$$\begin{aligned} \|x_{k+1} - \tilde{x}_{k+1}\|_2^2 &= B_k^2 \cdot \left\| \sum_{t=0}^k \frac{1}{B} \left(\frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{k-t+1} \gamma_i}{1 - \beta_i} \right) \nabla f(x_t) \right\|_2^2 \\ &\stackrel{\text{Jensen's inequality}}{\leq} B_k^2 \cdot \sum_{t=0}^k \frac{1}{B} \left(\frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{k-t+1} \gamma_i}{1 - \beta_i} \right) \|\nabla f(x_t)\|_2^2 \\ &= B_k \cdot \sum_{t=0}^k \left(\frac{1}{m} \sum_{i=1}^m \frac{\beta_i^{k-t+1} \gamma_i}{1 - \beta_i} \right) \|\nabla f(x_t)\|_2^2 \\ &\stackrel{(13), B_k \leq B_{k+1}}{\leq} B_K F \cdot \sum_{t=0}^k \max_{i=1, m} \beta_i^{k-t+1} \|\nabla f(x_t)\|_2^2. \end{aligned} \quad (29)$$

³ $|\langle a, b \rangle| \leq \frac{\|a\|_2^2}{2\lambda} + \frac{\lambda \|b\|_2^2}{2}$ for all $a, b \in \mathbb{R}^n$ and $\lambda > 0$.

For simplicity, we denote $B_K \equiv B$. Summing up these inequalities for $k = 0, 1, \dots, K$ with weights $w_k = \left(1 - \frac{\mu F}{2}\right)^{-(k+1)}$, we get

$$\begin{aligned} 3LF \sum_{k=0}^K w_k \|x_k - \tilde{x}_k\|_2^2 &\leq 3LBF^2 \cdot \sum_{k=0}^K \sum_{t=0}^{k-1} w_k \max_{i=1,m} \beta_i^{k-t} \|\nabla f(x_t)\|_2^2 \\ &\leq 3LBF^2 \cdot \sum_{k=0}^K \sum_{t=0}^k w_k \max_{i=1,m} \beta_i^{k-t} \|\nabla f(x_t)\|_2^2. \end{aligned} \quad (30)$$

Next, we estimate w_k using that $(1 - q/2)^{-1} \leq 1 + q$ for any $q \in (0, 1]$: for all $t = 0, 1, \dots, k$

$$w_k = \left(1 - \frac{\mu F}{2}\right)^{-(k-t)} w_t \leq (1 + \mu F)^{k-t} w_t \stackrel{(12)}{\leq} \left(1 + \frac{1 - \max_{i=1,m} \beta_i}{2}\right)^{k-t} w_t.$$

Using an inequality above and $(1 + q/2)(1 - q) \leq 1 - q/2$ for $q = 1 - \max_{i=1,m} \beta_i$, we continue the previous derivation (30)

$$\begin{aligned} 3LF \sum_{k=0}^K w_k \|x_k - \tilde{x}_k\|_2^2 &\leq 3LBF^2 \sum_{k=0}^K \sum_{t=0}^k w_t \|\nabla f(x_t)\|_2^2 \left(1 + \frac{1 - \max_{i=1,m} \beta_i}{2}\right)^{k-t} \max_{i=1,m} \beta_i^{k-t} \\ &\leq 3LBF^2 \sum_{k=0}^K \sum_{t=0}^k w_t \|\nabla f(x_t)\|_2^2 \left(1 - \frac{1 - \max_{i=1,m} \beta_i}{2}\right)^{k-t} \\ &\leq 3LBF^2 \left(\sum_{k=0}^K w_k \|\nabla f(x_k)\|_2^2\right) \left(\sum_{k=0}^{\infty} \left(1 - \frac{1 - \max_{i=1,m} \beta_i}{2}\right)^k\right) \\ &= \frac{6LBF^2}{1 - \max_{i=1,m} \beta_i} \sum_{k=0}^K w_k \|\nabla f(x_k)\|_2^2 \\ &\leq \frac{12L^2BF^2}{1 - \max_{i=1,m} \beta_i} \sum_{k=0}^K w_k (f(x_k) - f(x_*)). \end{aligned} \quad (31)$$

We take parameters γ_i, β_i (12) implying (13). Combining this with the last result (31), we obtain (14).

A.4 Proof of Theorem 2

Using Lemma 1 we get

$$\frac{F}{2} (f(x_k) - f(x_*)) \leq \left(1 - \frac{\mu F}{2}\right) \|\tilde{x}_k - x_*\|_2^2 - \|\tilde{x}_{k+1} - x_*\|_2^2 + 3LF \|x_k - \tilde{x}_k\|_2^2.$$

Summing up these inequalities for $k = 0, 1, \dots, K$ with weights $w_k = \left(1 - \frac{\mu F}{2}\right)^{-(k+1)}$, we have

$$\begin{aligned}
\frac{F}{2} \sum_{k=0}^K w_k (f(x_k) - f(x_*)) &\leq \sum_{k=0}^K \left(w_k \left(1 - \frac{\mu F}{2}\right) \|\tilde{x}_k - x_*\|_2^2 - w_k \|\tilde{x}_{k+1} - x_*\|_2^2 \right) \\
&\quad + 3LF \sum_{k=0}^K w_k \|x_k - \tilde{x}_k\|_2^2 \\
&\stackrel{(14)}{\leq} \sum_{k=0}^K (w_{k-1} \|\tilde{x}_k - x_*\|_2^2 - w_k \|\tilde{x}_{k+1} - x_*\|_2^2) \\
&\quad + \frac{F}{4} \sum_{k=0}^K w_k (f(x_k) - f(x_*)) \\
&\leq \|x_0 - x_*\|_2^2 + \frac{F}{4} \sum_{k=0}^K w_k (f(x_k) - f(x_*)).
\end{aligned}$$

Rearranging and multiplying by $\frac{1}{W_K} = \frac{1}{\sum_{k=0}^K w_k}$ this inequality, we have

$$\frac{1}{W_K} \sum_{k=0}^K w_k (f(x_k) - f(x_*)) \leq \frac{4\|x_0 - x_*\|_2^2}{FW_K}.$$

Next, we obtain (15) by using Jensen's inequality:

$$f(\bar{x}_K) \leq \frac{1}{W_K} \sum_{k=0}^K w_k f(x_k).$$

In strongly convex case ($\mu > 0$), we have $W_K \geq w_{K-1} = \left(1 - \frac{\mu F}{2}\right)^{-K}$, hence (16) holds. In convex case ($\mu = 0$), $W_K = K + 1 > K$ that implies (17).

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 1–27 (2011)
2. Danilova, M., et al.: Recent theoretical advances in non-convex optimization. arXiv preprint [arXiv:2012.06188](https://arxiv.org/abs/2012.06188) (2020)
3. Danilova, M., Malinovsky, G.: Averaged heavy-ball method. arXiv preprint [arXiv:2111.05430](https://arxiv.org/abs/2111.05430) (2021)
4. Defazio, A.: Momentum via primal averaging: theoretical insights and learning rate schedules for non-convex optimization. arXiv preprint [arXiv:2010.00406](https://arxiv.org/abs/2010.00406) (2020)
5. Ghadimi, E., Feyzmahdavian, H.R., Johansson, M.: Global convergence of the heavy-ball method for convex optimization. In: 2015 European Control Conference (ECC), pp. 310–315. IEEE (2015)
6. Lessard, L., Recht, B., Packard, A.: Analysis and design of optimization algorithms via integral quadratic constraints. SIAM J. Optim. **26**(1), 57–95 (2016)

7. Lucas, J., Sun, S., Zemel, R., Grosse, R.: Aggregated momentum: stability through passive damping. In: International Conference on Learning Representations (2019)
8. Mania, H., Pan, X., Papailiopoulos, D., Recht, B., Ramchandran, K., Jordan, M.I.: Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM J. Optim.* **27**(4), 2202–2229 (2017)
9. Nemirovsky, A., Yudin, D.: *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York (1983)
10. Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In: *Doklady an USSR*, vol. 269, pp. 543–547 (1983)
11. Nesterov, Y.: *Lectures on Convex Optimization*. SOIA, vol. 137. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-91578-4>
12. Novik, M.: torch-optimizer – collection of optimization algorithms for PyTorch. GitHub repository (2020). <https://github.com/jettify/pytorch-optimizer>
13. Polyak, B.: *Introduction to Optimization*. Optimization Software, New York (1987)
14. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **4**(5), 1–17 (1964)
15. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: *International Conference on Machine Learning*, pp. 1139–1147. PMLR (2013)
16. Taylor, A., Bach, F.: Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. In: *Conference on Learning Theory*, pp. 2934–2992. PMLR (2019)
17. Yang, T., Lin, Q., Li, Z.: Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization. arXiv preprint [arXiv:1604.03257](https://arxiv.org/abs/1604.03257) (2016)



Noisy Zeroth-Order Optimization for Non-smooth Saddle Point Problems

Darina Dvinskikh^{1,2,3} , Vladislav Tominin¹ , Iaroslav Tominin¹ ,
and Alexander Gasnikov^{1,2,4} 

¹ Moscow Institute of Physics and Technology, Dolgoprudny, Russia
{[darina.dvinskikh](mailto:darina.dvinskikh@phystech.edu), [tominin.vd](mailto:tominin.vd@phystech.edu), [tominin.yad](mailto:tominin.yad@phystech.edu)}@phystech.edu,
gasnikov.av@mipt.ru

² Institute for Information Transmission Problems RAS, Moscow, Russia

³ ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

⁴ Caucasus Mathematical Center, Adyge State University, Maikop, Russia

Abstract. This paper investigates zeroth-order methods for non-smooth convex-concave saddle point problems (with r -growth condition for duality gap). We assume that a black-box gradient-free oracle returns an inexact function value corrupted by an adversarial noise. In this work we prove that the standard zeroth-order version of the mirror descent method is optimal in terms of the oracle calls complexity and the maximum admissible noise.

Keywords: stochastic optimization · non-smooth optimization · saddle point problems · gradient-free optimization

1 Introduction

In this paper, we consider a stochastic non-smooth *saddle point* problem of the form

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y), \quad (1)$$

where $f(x, y) \triangleq \mathbb{E}_\xi [f(x, y, \xi)]$ is the expectation, w.r.t. $\xi \in \Xi$, $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is convex-concave and Lipschitz continuous, and $\mathcal{X} \subseteq \mathbb{R}^{d_x}$, $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$ are convex compact sets. The standard interpretation of such min-max problems is the antagonistic game between a learner and an adversary, where the equilibria are the saddle points [20]. Now the interest in saddle point problems is renewed due to the popularity of generative adversarial networks (GANs), whose training involves solving min-max problems [6, 14].

Motivated by many applications in the field of reinforcement learning [7, 17] and statistics, where only a black-box access to the function values of the objective is available, we consider *zeroth-order* oracle (also known as gradient-free oracle). Particularly, we mention the classical problem of adversarial multi-armed bandit [1, 5, 10], where a learner receives a feedback given by the function

The research was supported by Russian Science Foundation (project No. 21-71-30005).

evaluations from an adversary. Thus, zeroth-order methods [8] are the workhorse technique when the gradient information is prohibitively expensive or even not available and optimization is performed based only on the function evaluations.

Related Work and Contribution. Zeroth-order methods in the non-smooth setup were developed in a wide range of works [4, 8, 9, 11, 12, 19, 21, 23, 24, 26]. Particularly, in [24], an optimal algorithm was provided as an improvement to the work [9] for a non-smooth case but Lipschitz continuous in stochastic convex optimization problems. However, this algorithm uses the exact function evaluations that can be infeasible in some applications. Indeed, the objective $f(z, \xi)$ can be not directly observed but instead, its noisy approximation $\varphi(z, \xi) \triangleq f(z, \xi) + \delta(z)$ can be queried, where $\delta(z)$ is some adversarial noise. This noisy-corrupted setup was considered in many works, however, such an algorithm that is optimal in terms of the number of oracle calls complexity and the maximum value of adversarial noise has not been proposed. For instance, in [2, 4], optimal algorithms in terms of oracle calls complexity were proposed, however, they are not optimal in terms of the maximum value of the noise. In papers [22, 27], algorithms are optimal in terms of the maximum value of the noise, however, they are not optimal in terms of the oracle calls complexity. In this paper, we provide an accurate analysis of a gradient-free version of the mirror descent method with an inexact oracle that shows that the method is optimal both in terms of the inexact oracle calls complexity and the maximum admissible noise. We consider two possible scenarios for the nature of the adversarial noise arising in different applications: the noise is bounded by a small value or is Lipschitz. Table 1 demonstrates our contribution by comparing our results with the existing optimal bounds. Finally, we consider the case when the objective satisfies the r -growth condition and restate the results.

Paper Organization. This paper is organized as follows. In Sect. 2, we begin with background material, notation, and assumptions. In Sect. 3, we present the main results of the paper: the algorithm and the analysis of its convergence. In Sect. 4, we consider an additional assumption of r -growth condition and restate the results. In Sect. 5, we comment on the unboundedness of the second moment of the stochastic gradient.

2 Preliminaries

Notation. We use $\langle x, y \rangle \triangleq \sum_{i=1}^d x_i y_i$ to define the inner product of $x, y \in \mathbb{R}^d$, where x_i is the i -th component of x . By norm $\|\cdot\|_p$ we mean the ℓ_p -norm. Then the dual norm of the norm $\|\cdot\|_p$ is $\|\lambda\|_q \triangleq \max\{\langle x, \lambda \rangle \mid \|x\|_p \leq 1\}$. Operator $\mathbb{E}[\cdot]$ is the full expectation and operator $\mathbb{E}_\xi[\cdot]$ is the conditional expectation, w.r.t. ξ .

Table 1. Summary of the contribution

PAPER	PROBLEM ^(a)	IS THE NOISE LIPSCHITZ?	NUMBER OF ORACLE CALLS	MAXIMUM NOISE
[2]	convex	✗	d/ϵ^2	$\epsilon^2/d^{3/2}$
[4]	saddle point	✗	d/ϵ^2	ϵ^2/d
[27]	convex	✗	$\text{Poly}(d, 1/\epsilon)$	ϵ^2/\sqrt{d}
[22]	convex	✗	$\text{Poly}(d, 1/\epsilon)$	$\max\{\epsilon^2/\sqrt{d}, \epsilon/d\}$ ^(b)
THIS WORK	saddle point	✗	d/ϵ^2	ϵ^2/\sqrt{d}
THIS WORK	saddle point	✓	d/ϵ^2	ϵ/\sqrt{d} ^(c)

^(a) The results obtained for saddle point problems are also valid for convex problems.

^(b) Notice, that this bound (up to a logarithmic factor) is also an upper bound for maximum possible value of noise. It is important to note, that $\epsilon/d \lesssim \epsilon^2/\sqrt{d}$, when $\epsilon^{-2} \lesssim d$. That is in the large-dimension regime, when subgradient method is better than center of gravity types methods [18], the upper bound on the value of admissible noise (that allows one to solve the problem with accuracy ϵ) will be ϵ^2/\sqrt{d} .

^(c) This estimate is for the Lipschitz constant of the noise and it possibly is tight. However, we prove that the upper bound (possibly not tight) for the Lipschitz-noise constant is ϵ

Setup. Let us introduce the embedding space $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$, and then $z \in \mathcal{Z}$ means $z \triangleq (x, y)$, where $x \in \mathcal{X}, y \in \mathcal{Y}$. On this embedding space, we introduce the ℓ_p -norm and a prox-function $\omega(z)$ compatible with this norm. Then we define the Bregman divergence associated with $\omega(z)$ as

$$V_z(v) \triangleq \omega(z) - \omega(v) - \langle \nabla \omega(v), z - v \rangle \geq \|z - v\|_p^2/2, \quad \text{for all } z, v \in \mathcal{Z}.$$

We also introduce a prox-operator as follows

$$\text{Prox}_z(\xi) \triangleq \arg \min_{v \in \mathcal{Z}} (V_z(v) + \langle \xi, v \rangle), \quad \text{for all } z \in \mathcal{Z}.$$

Finally, we denote ω -diameter of \mathcal{Z} by $\mathcal{D} \triangleq \max_{z, v \in \mathcal{Z}} \sqrt{2V_z(v)} = \tilde{\mathcal{O}}\left(\max_{z, v \in \mathcal{Z}} \|z - v\|_p\right)$. Here $\tilde{\mathcal{O}}(\cdot)$ is $\mathcal{O}(\cdot)$ up to a $\sqrt{\log d}$ -factor.

Assumption 1 (Lipschitz continuity of the objective). *Function $f(z, \xi)$ is M_2 -Lipschitz continuous in $z \in \mathcal{Z}$ w.r.t. the ℓ_2 -norm, i.e., for all $z_1, z_2 \in \mathcal{Z}$ and $\xi \in \Xi$,*

$$|f(z_1, \xi) - f(z_2, \xi)| \leq M_2(\xi) \|z_1 - z_2\|_2.$$

Moreover, there exists a positive constant M_2 such that $\mathbb{E}[M_2^2(\xi)] \leq M_2^2$.

Assumption 2. *For all $z \in \mathcal{Z}$, it holds $|\delta(z)| \leq \Delta$.*

Assumption 3 (Lipschitz continuity of the noise). Function $\delta(z)$ is $M_{2,\delta}$ -Lipschitz continuous in $z \in \mathcal{Z}$ w.r.t. the ℓ_2 -norm, i.e., for all $z_1, z_2 \in \mathcal{Z}$,

$$|\delta(z_1) - \delta(z_2)| \leq M_{2,\delta} \|z_1 - z_2\|_2.$$

3 Main Results

In this section, we present the main results of the paper (Theorem 1 and Corollary 1). For problem (1), we present a numerical algorithm (Algorithm 1) which is optimal in terms of the number of inexact zeroth-order oracle calls and the maximum adversarial noise. The algorithm is based on a gradient-free version of the stochastic mirror descent (SMD) [3].

Black-Box Oracle. We assume that we can query zeroth-order oracle corrupted by some adversarial noise $\delta(z)$

$$\varphi(z, \xi) \triangleq f(z, \xi) + \delta(z). \quad (2)$$

Gradient Approximation. The gradient of $\varphi(z, \xi)$ from (2), w.r.t. z , can be approximated by function evaluations in two random points closed to z . To do so, we define vector \mathbf{e} picked uniformly at random from the Euclidean unit sphere $\{\mathbf{e} : \|\mathbf{e}\|_2 = 1\}$. Let $\mathbf{e} \triangleq (\mathbf{e}_x^\top, -\mathbf{e}_y^\top)^\top$, where $\dim(\mathbf{e}_x) \triangleq d_x$, $\dim(\mathbf{e}_y) \triangleq d_y$ and $\dim(\mathbf{e}) \triangleq d = d_x + d_y$. Then the gradient of $\varphi(z, \xi)$ can be estimated by the following approximation with a small variance [24]:

$$g(z, \xi, \mathbf{e}) = \frac{d}{2\tau} (\varphi(z + \tau\mathbf{e}, \xi) - \varphi(z - \tau\mathbf{e}, \xi)) \begin{pmatrix} \mathbf{e}_x \\ -\mathbf{e}_y \end{pmatrix}, \quad (3)$$

where $\tau > 0$ is a small parameter.

Input: iteration number N
 $z^1 \leftarrow \arg \min_{z \in \mathcal{Z}} d(z)$ **for**
 $k = 1, \dots, N$ **do**
 Sample \mathbf{e}^k, ξ^k independently
 Initialize γ_k
 Calculate $g(z^k, \xi^k, \mathbf{e}^k)$ via (3)
 $z^{k+1} \leftarrow \text{Prox}_{z^k}(\gamma_k g(z^k, \xi^k, \mathbf{e}^k))$

end

Output:

$$\hat{z}^N \leftarrow \left(\sum_{k=1}^N \gamma_k \right)^{-1} \sum_{k=1}^N \gamma_k z^k$$

Algorithm 1: Zeroth-order SMD

Randomized Smoothing. For a non-smooth objective $f(z)$, we define the following function

$$f^\tau(z) \triangleq \mathbb{E}_{\tilde{\mathbf{e}}} f(z + \tau\tilde{\mathbf{e}}), \quad (4)$$

where $\tau > 0$ and $\tilde{\mathbf{e}}$ is a vector picked uniformly at random from the Euclidean unit ball: $\{\tilde{\mathbf{e}} : \|\tilde{\mathbf{e}}\|_2 \leq 1\}$. Function $f^\tau(z)$ can be referred as a smooth approximation of $f(z)$. Here $f(z) \triangleq \mathbb{E}f(z, \xi)$.

The next theorem presents the rate of convergence of Algorithm 1.

Theorem 1. Let function $f(x, y, \xi)$ satisfy the Assumption 1. Then the following holds for $\epsilon_{\text{sad}} \triangleq \max_{y \in \mathcal{Y}} f(\hat{x}^N, y) - \min_{x \in \mathcal{X}} f(x, \hat{y}^N)$, where $\hat{z}^N \triangleq (\hat{x}^N, \hat{y}^N)$ is the output of Algorithm 1:

1. under Assumption 2 and learning rate $\gamma_k = \frac{\mathcal{D}}{M_{\text{case1}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case1}}^2 \triangleq \mathcal{O}(da_q^2 M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2})$

$$\mathbb{E}[\epsilon_{\text{sad}}] \leq M_{\text{case1}} \mathcal{D} \sqrt{2/N} + \sqrt{d} \Delta \mathcal{D} \tau^{-1} + 2\tau M_2. \quad (5)$$

2. under Assumption 3 and learning rate $\gamma_k = \frac{\mathcal{D}}{M_{\text{case2}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case2}}^2 \triangleq \mathcal{O}\left(da_q^2 \left(M_2^2 + M_{2,\delta}^2\right)\right)$

$$\mathbb{E}[\epsilon_{\text{sad}}] \leq M_{\text{case2}} \mathcal{D} \sqrt{2/N} + M_{2,\delta} \sqrt{d} \mathcal{D} + 2\tau M_2, \quad (6)$$

where $\sqrt{\mathbb{E}[\|e\|_q^4]} = \mathcal{O}(\min\{q, \log d\} d^{2/q-1}) = a_q^2$ [15].

The next corollary presents our contribution.

Corollary 1. Let function $f(x, y, \xi)$ satisfy the Assumption 1, and let τ in randomized smoothing (4) be chosen as $\tau = \mathcal{O}(\epsilon/M_2)$, where ϵ is the desired accuracy to solve problem (1). If one of the two following statement is true

1. Assumption 2 holds and $\Delta = \mathcal{O}\left(\frac{\epsilon^2}{\mathcal{D} M_2 \sqrt{d}}\right)$
2. Assumption 3 holds and $M_{2,\delta} = \mathcal{O}\left(\frac{\epsilon}{\mathcal{D} \sqrt{d}}\right)$

then for the output $\hat{z}^N \triangleq (\hat{x}^N, \hat{y}^N)$ of Algorithm 1, it holds $\mathbb{E}[\epsilon_{\text{sad}}(\hat{z}^N)] = \epsilon$ after

$$N = \mathcal{O}(da_q^2 M_2^2 \mathcal{D}^2 / \epsilon^2)$$

iterations (zeroth-order oracle calls), where $\sqrt{\mathbb{E}[\|e\|_q^4]} = \mathcal{O}(\min\{q, \log d\} d^{2/q-1}) = a_q^2$ [15].

Next we clarify the Corollary 1 in the two following special setups: the ℓ_2 -norm and the ℓ_1 -norm in the two following examples.

Example 1. Let $p = 2$, then $q = 2$ and $\sqrt{\mathbb{E}_e[\|e\|_2^4]} = 1$. Thus, $a_2^2 = 1$ and $\mathcal{D}^2 = \max_{z,v \in \mathcal{Z}} \|z - v\|_2^2$. Consequently, the number of iterations in the Corollary 1 can be rewritten as follows

$$N = \mathcal{O}\left(d M_2^2 \epsilon^{-2} \max_{z,v \in \mathcal{Z}} \|z - v\|_2^2\right).$$

Example 2 [24, Lemma 4]. Let $p = 1$ then, $q = \infty$ and $\sqrt{\mathbb{E}_e[\|e\|_\infty^4]} = \mathcal{O}\left(\frac{\log d}{d}\right)$. Thus, $a_\infty^2 = \mathcal{O}\left(\frac{\log d}{d}\right)$ and $\mathcal{D}^2 = \mathcal{O}(\log d \max_{z,v \in \mathcal{Z}} \|z - v\|_1^2)$. Consequently, the number of iterations in the Corollary 1 can be rewritten as follows

$$N = \mathcal{O}\left(M_2^2 \log^2 d \epsilon^{-2} \max_{z,v \in \mathcal{Z}} \|z - v\|_1^2\right).$$

Proof of Theorem 1. By the definition $z^{k+1} = \text{Prox}_{z^k}(\gamma_k g(z^k, \mathbf{e}^k, \xi^k))$ from Algorithm 1 we get [3], for all $u \in \mathcal{Z}$

$$\gamma_k \langle g(z^k, \mathbf{e}^k, \xi^k), z^k - u \rangle \leq V_{z^k}(u) - V_{z^{k+1}}(u) + \gamma_k^2 \|g(z^k, \mathbf{e}^k, \xi^k)\|_q^2 / 2.$$

Taking the conditional expectation w.r.t. ξ, \mathbf{e} and summing for $k = 1, \dots, N$ we obtain, for all $u \in \mathcal{Z}$

$$\sum_{k=1}^N \gamma_k \mathbb{E}_{\mathbf{e}^k, \xi^k} [\langle g(z^k, \mathbf{e}^k, \xi^k), z^k - u \rangle] \leq V_{z^1}(u) + \sum_{k=1}^N \frac{\gamma_k^2}{2} \mathbb{E}_{\mathbf{e}^k, \xi^k} [\|g(z^k, \mathbf{e}^k, \xi^k)\|_q^2]. \quad (7)$$

Step 1

For the second term in the r.h.s of inequality (7) we use Lemma 6 and obtain

1. under Assumption 2:

$$\mathbb{E}_{\mathbf{e}^k, \xi^k} [\|g(z^k, \xi^k, \mathbf{e}^k)\|_q^2] \leq ca_q^2 d M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}, \quad (8)$$

2. under Assumption 3:

$$\mathbb{E}_{\mathbf{e}^k, \xi^k} [\|g(z^k, \xi^k, \mathbf{e}^k)\|_q^2] \leq ca_q^2 d (M_2^2 + M_{2,\delta}^2), \quad (9)$$

where c is some constant and $\sqrt{\mathbb{E}[\|e\|_q^4]} = \mathcal{O}(\min\{q, \log d\} d^{2/q-1}) = a_q^2$ [15].

Step 2

For the l.h.s. of inequality (7), we use Lemma 4

1. under Assumption 2

$$\begin{aligned} \sum_{k=1}^N \gamma_k \mathbb{E}_{\mathbf{e}^k, \xi^k} [\langle g(z^k, \mathbf{e}^k, \xi^k), z^k - u \rangle] &\geq \sum_{k=1}^N \gamma_k \langle \nabla f^\tau(z^k), z^k - u \rangle \\ &\quad - \sum_{k=1}^N \gamma_k \mathbb{E}_{\mathbf{e}^k} [\langle d \Delta \tau^{-1} \mathbf{e}^k, z^k - u \rangle]. \end{aligned} \quad (10)$$

2. under Assumption 3

$$\begin{aligned} \sum_{k=1}^N \gamma_k \mathbb{E}_{\mathbf{e}^k, \xi^k} [\langle g(z^k, \mathbf{e}^k, \xi^k), z^k - u \rangle] &\geq \sum_{k=1}^N \gamma_k \langle \nabla f^\tau(z^k), z^k - u \rangle \\ &\quad - \sum_{k=1}^N \gamma_k \mathbb{E}_{\mathbf{e}^k} [\langle d M_{2,\delta} \mathbf{e}^k, z^k - u \rangle]. \end{aligned} \quad (11)$$

For the first term in the r.h.s. of inequalities (10) and (11), we have

$$\begin{aligned}
\sum_{k=1}^N \gamma_k \langle \nabla f^\tau(z^k), z^k - u \rangle &= \sum_{k=1}^N \gamma_k \left\langle \begin{pmatrix} \nabla_x f^\tau(x^k, y^k) \\ -\nabla_y f^\tau(x^k, y^k) \end{pmatrix}, \begin{pmatrix} x^k - x \\ y^k - y \end{pmatrix} \right\rangle \\
&= \sum_{k=1}^N \gamma_k (\langle \nabla_x f^\tau(x^k, y^k), x^k - x \rangle - \langle \nabla_y f^\tau(x^k, y^k), y^k - y \rangle) \\
&\geq \sum_{k=1}^N \gamma_k (f^\tau(x^k, y^k) - f^\tau(x, y^k)) - (f^\tau(x^k, y^k) - f^\tau(x^k, y)) \\
&= \sum_{k=1}^N \gamma_k (f^\tau(x^k, y) - f^\tau(x, y^k)), \tag{12}
\end{aligned}$$

where $u \triangleq (x^\top, y^\top)^\top$. Then we use the fact function $f^\tau(x, y)$ is convex in x and concave in y and obtain

$$\left(\sum_{i=1}^N \gamma_k \right)^{-1} \sum_{k=1}^N \gamma_k (f^\tau(x^k, y) - f^\tau(x, y^k)) \leq f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N), \tag{13}$$

where (\hat{x}^N, \hat{y}^N) is the output of Algorithm 1. Using (13) for (12) we get

$$\sum_{k=1}^N \gamma_k \langle \nabla f^\tau(z^k), z^k - u \rangle \geq \sum_{k=1}^N \gamma_k f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N). \tag{14}$$

Next we estimate the term $\mathbb{E}_{e^k} [|\langle e^k, z^k - u \rangle|]$ in (10) and (11), by Lemma 1

$$\mathbb{E}_{e^k} [|\langle e^k, z^k - u \rangle|] \leq \|z^k - u\|_2 / \sqrt{d}. \tag{15}$$

Now we substitute (14) and (15) to (10) and (11), and get

1. under Assumption 2

$$\begin{aligned}
\sum_{k=1}^N \gamma_k \mathbb{E}_{e^k, \xi^k} [\langle g(z^k, e^k, \xi^k), z^k - u \rangle] &\geq \sum_{k=1}^N \gamma_k f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) \\
&\quad - \sum_{k=1}^N \gamma_k \sqrt{d} \Delta \|z^k - u\|_2 \tau^{-1}. \tag{16}
\end{aligned}$$

2. under Assumption 3

$$\begin{aligned}
\sum_{k=1}^N \gamma_k \mathbb{E}_{e^k, \xi^k} [\langle g(z^k, e^k, \xi^k), z^k - u \rangle] &\geq \sum_{k=1}^N \gamma_k f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) \\
&\quad - \sum_{k=1}^N \gamma_k \sqrt{d} M_{2, \delta} \|z^k - u\|_2. \tag{17}
\end{aligned}$$

Step 3 (under Assumption 2)

Now we combine Eq. (16) with Eq. (8) for Eq. (7) and obtain under Assumption 2 the following

$$\begin{aligned} \sum_{k=1}^N \gamma_k f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) - \sum_{k=1}^N \gamma_k \sqrt{d} \Delta \|z^k - u\|_2 \tau^{-1} &\leq V_{z^1}(u) \\ &+ \sum_{k=1}^N \frac{\gamma_k^2}{2} (ca_q^2 d M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}). \end{aligned} \quad (18)$$

Using Lemma 2 we obtain

$$f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) \geq f(\hat{x}^N, y) - f(x, \hat{y}^N) - 2\tau M_2.$$

Using this we can rewrite (18) as follows

$$\begin{aligned} f(\hat{x}^N, y) - f(x, \hat{y}^N) &\leq \frac{V_{z^1}(u)}{\sum_{k=1}^N \gamma_k} + \frac{ca_q^2 d M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}}{\sum_{k=1}^N \gamma_k} \sum_{k=1}^N \frac{\gamma_k^2}{2} \\ &+ \sqrt{d} \Delta \max_k \|z^k - u\|_2 \tau^{-1} + 2\tau M_2. \end{aligned} \quad (19)$$

For the r.h.s. of (19) we use the definition of the ω -diameter of \mathcal{Z} :

$\mathcal{D} \triangleq \max_{z, v \in \mathcal{Z}} \sqrt{2V_z(v)}$ and estimate $\|z^k - u\|_2 \leq \mathcal{D}$ for all z^1, \dots, z^k and all $u \in \mathcal{Z}$. Using this for (19) and taking the maximum in $(x, y) \in (\mathcal{X}, \mathcal{Y})$, we obtain

$$\begin{aligned} \max_{y \in \mathcal{Y}} f(\hat{x}^N, y) - \min_{x \in \mathcal{X}} f(x, \hat{y}^N) &\leq \frac{\mathcal{D}^2 + (ca_q^2 d M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}) \sum_{k=1}^N \gamma_k^2 / 2}{\sum_{k=1}^N \gamma_k} \\ &+ \sqrt{d} \Delta \mathcal{D} \tau^{-1} + 2\tau M_2. \end{aligned} \quad (20)$$

Then we use the definition of the ω -diameter of \mathcal{Z} : $\mathcal{D} \triangleq \max_{z, v \in \mathcal{Z}} \sqrt{2V_z(v)}$ and estimate $\|z^k - u\|_2 \leq \mathcal{D}$ for all z^1, \dots, z^k and all $u \in \mathcal{Z}$. Thus, taking the expectation of (20) and choosing learning rate $\gamma_k = \frac{\mathcal{D}}{M_{\text{case1}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case1}}^2 \triangleq cda_q^2 M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}$ in Eq. (20) we get

$$\mathbb{E} \left[\max_{y \in \mathcal{Y}} f(\hat{x}^N, y) - \min_{x \in \mathcal{X}} f(x, \hat{y}^N) \right] \leq M_{\text{case1}} \mathcal{D} \sqrt{2/N} + \frac{\Delta \mathcal{D} \sqrt{d}}{\tau} + 2\tau M_2.$$

Step 4 (under Assumption 3)

Now we combine Eq. (17) with Eq. (9) for Eq. (7) and obtain under Assumption 3

$$\begin{aligned} \sum_{k=1}^N \gamma_k f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) - \sum_{k=1}^N \gamma_k \sqrt{d} M_{2,\delta} \|z^k - u\|_2 &\leq V_{z^1}(u) \\ &+ \sum_{k=1}^N \frac{\gamma_k^2}{2} ca_q^2 d (M_2^2 + M_{2,\delta}^2). \end{aligned} \quad (21)$$

Using Lemma 2 we obtain

$$f^\tau(\hat{x}^N, y) - f^\tau(x, \hat{y}^N) \geq f(\hat{x}^N, y) - f(x, \hat{y}^N) - 2\tau M_2.$$

Using this we can rewrite (21) as follows

$$\begin{aligned} f(\hat{x}^N, y) - f(x, \hat{y}^N) &\leq \frac{V_{z^1}(u)}{\sum_{k=1}^N \gamma_k} + \frac{ca_q^2 d(M_2^2 + M_{2,\delta}^2)}{\sum_{k=1}^N \gamma_k} \sum_{k=1}^N \frac{\gamma_k^2}{2} \\ &\quad + \sqrt{d} M_{2,\delta} \max_k \|z^k - u\|_2 + 2\tau M_2. \end{aligned} \quad (22)$$

For the r.h.s. of (22) we use the definition of the ω -diameter of \mathcal{Z} :

$\mathcal{D} \triangleq \max_{z,v \in \mathcal{Z}} \sqrt{2V_z(v)}$ and estimate $\|z^k - u\|_2 \leq \mathcal{D}$ for all z^1, \dots, Z^k and all $u \in \mathcal{Z}$. Using this for (22) and taking the maximum in $(x, y) \in (\mathcal{X}, \mathcal{Y})$, we obtain

$$\begin{aligned} \max_{y \in \mathcal{Y}} f(\hat{x}^N, y) - \min_{x \in \mathcal{X}} f(x, \hat{y}^N) &\leq \frac{\mathcal{D}^2 + ca_q^2 d(M_2^2 + M_{2,\delta}^2) \sum_{k=1}^N \gamma_k^2 / 2}{\sum_{k=1}^N \gamma_k} \\ &\quad + M_{2,\delta} \sqrt{d} \mathcal{D} + 2\tau M_2. \end{aligned} \quad (23)$$

Then we use the definition of the ω -diameter of \mathcal{Z} : $\mathcal{D} \triangleq \max_{z,v \in \mathcal{Z}} \sqrt{2V_z(v)}$ and estimate $\|z^k - u\|_2 \leq \mathcal{D}$ for all z^1, \dots, Z^k and all $u \in \mathcal{Z}$. Thus, taking the expectation of (20) and choosing learning rate $\gamma_k = \frac{\mathcal{D}}{M_{\text{case2}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case2}}^2 \triangleq cda_q^2(M_2^2 + M_{2,\delta}^2)$ in Eq. (23) we get

$$\mathbb{E} \left[\max_{y \in \mathcal{Y}} f(\hat{x}^N, y) - \min_{x \in \mathcal{X}} f(x, \hat{y}^N) \right] \leq M_{\text{case2}} \mathcal{D} \sqrt{2/N} + M_{2,\delta} \mathcal{D} \sqrt{d} + 2\tau M_2.$$

□

4 Restarts

In this section, we assume that we additionally have the r -growth condition for duality gap (see, [25] for convex optimization problems). For such a case, we apply the restart technique [16] to Algorithm 1

Assumption 4 (r -growth condition). *There is $r \geq 2$ and $\mu_r > 0$ such that for all $z = (x, y) \in \mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$*

$$\frac{\mu_r}{2} \|z - z^*\|_p^r \leq f(x, y^*) - f(x^*, y),$$

where (x^*, y^*) is a solution of problem (1).

Next we restate the Theorem 1 under this Assumption 4 together with restarts technique.

Theorem 2. *Let $f(x, y, \xi)$ satisfy Assumption 1. Then the following holds for $\epsilon_{\text{sad}} \triangleq f(\hat{x}^N, y^*) - f(x^*, \hat{y}^N)$, where $\hat{z}^N \triangleq (\hat{x}^N, \hat{y}^N)$ is the output of Algorithm 1*

1. under Assumption 2 and learning rate $\gamma_k = \frac{\sqrt{\mathbb{E}[V_{z^1}(z^*)]}}{M_{\text{case1}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case1}}^2 \triangleq \mathcal{O}(da_q^2 M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2})$

$$\mathbb{E}[\epsilon_{\text{sad}}] \leq \sqrt{\frac{2}{N}} M_{\text{case1}} \sqrt{\mathbb{E}[V_{z^1}(z^*)]} + \Delta \mathcal{D} \sqrt{d} \tau^{-1} + 2\tau M_2. \quad (24)$$

2. under Assumption 3 and learning rate $\gamma_k = \frac{\sqrt{\mathbb{E}[V_{z^1}(z^*)]}}{M_{\text{case2}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case2}}^2 \triangleq \mathcal{O}(da_q^2 (M_2^2 + M_{2,\delta}^2))$

$$\mathbb{E}[\epsilon_{\text{sad}}] \leq \sqrt{\frac{2}{N}} M_{\text{case2}} \sqrt{\mathbb{E}[V_{z^1}(z^*)]} + M_{2,\delta} \mathcal{D} \sqrt{d} + 2\tau M_2, \quad (25)$$

where $\sqrt{\mathbb{E}_e[\|\mathbf{e}\|_q^4]} \leq a_q^2$.

Moreover, let τ be chosen as $\tau = \mathcal{O}(\epsilon/M_2)$ in randomized smoothing (4), where ϵ is the desired accuracy to solve problem (1). If the Assumption 4 and one of the two following statement are satisfied

1. Assumption 2 holds and $\Delta = \mathcal{O}\left(\frac{\epsilon^2}{\mathcal{D} M_2 \sqrt{d}}\right)$
2. Assumption 3 holds and $M_{2,\delta} = \mathcal{O}\left(\frac{\epsilon}{\mathcal{D} \sqrt{d}}\right)$

then for the output $\hat{z}^N \triangleq (\hat{x}^N, \hat{y}^N)$ of Algorithm 1, we can apply the restart technique to achieve $\mathbb{E}[\epsilon_{\text{sad}}] \leq \epsilon$ in N_{acc} iterations, where N_{acc} is given by

$$N_{\text{acc}} = \tilde{\mathcal{O}}\left(\frac{a_q^2 M_2^2 d}{\mu_r^{2/r} \epsilon^{2(r-1)/r}}\right). \quad (26)$$

Proof of Theorem 2. We repeat the proof of Theorem 1, except that now z^1 can be chosen in a stochastic way. Moreover, now we use a rougher inequality instead of (15)

$$\mathbb{E}_{e^k} [|\langle e^k, z^k - u \rangle|] \leq \mathcal{D} / \sqrt{d}. \quad (27)$$

Step 1 (under Assumption 2)

Taking the expectation in (19), choosing $(x, y) = (x^*, y^*)$, and learning rate $\gamma_k = \frac{\sqrt{\mathbb{E}[V_{z^1}(z^*)]}}{M_{\text{case1}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case1}}^2 \triangleq cda_q^2 M_2^2 + d^2 a_q^2 \Delta^2 \tau^{-2}$ we get

$$\mathbb{E}[f(\hat{x}^N, y^*) - f(x^*, \hat{y}^N)] \leq \sqrt{\frac{2}{N}} M_{\text{case1}} \sqrt{\mathbb{E}[V_{z^1}(z^*)]} + \frac{\sqrt{d} \Delta \mathcal{D}}{\tau} + 2\tau M_2. \quad (28)$$

Step 2 (under Assumption 3)

Taking the expectation in (22), choosing $(x, y) = (x^*, y^*)$, and learning rate $\gamma_k = \frac{\sqrt{\mathbb{E}[V_{z^1}(z^*)]}}{M_{\text{case2}}} \sqrt{\frac{2}{N}}$ with $M_{\text{case2}}^2 \triangleq cda_q^2 (M_2^2 + M_{2,\delta}^2)$ we obtain

$$\mathbb{E}[f(\hat{x}^N, y^*) - f(x^*, \hat{y}^N)] \leq \sqrt{\frac{2}{N}} M_{\text{case2}} \sqrt{\mathbb{E}[V_{z^1}(z^*)]} + M_{2,\delta} \sqrt{d} \mathcal{D} + 2\tau M_2. \quad (29)$$

Step 3 (Restarts)

Now let τ be chosen as $\tau = \mathcal{O}(\epsilon/M_2)$, where ϵ is the desired accuracy to solve problem (1). If one of the two following statement holds

1. Assumption 2 holds and $\Delta = \mathcal{O}\left(\frac{\epsilon^2}{\mathcal{D}M_2\sqrt{d}}\right)$
2. Assumption 3 holds and $M_{2,\delta} = \mathcal{O}\left(\frac{\epsilon}{\mathcal{D}\sqrt{d}}\right)$

then using (28) we obtain the convergence rate of the following form

$$\mathbb{E}[f(\hat{x}^{N_1}, y^*) - f(x^*, \hat{y}^{N_1})] = \tilde{\mathcal{O}}\left(\frac{a_q M_2 \sqrt{d}}{\sqrt{N_1}} \sqrt{\mathbb{E}[V_{z^1}(z^*)]}\right). \quad (30)$$

In this step we will employ the restart technique that is a generalization of the technique proposed in [16].

For the l.h.s. of Eq. (30) we use the Assumption 4. For the r.h.s. of Eq. (30) we use $V_{z^1}(z^*) = \tilde{\mathcal{O}}(\|z^1 - z^*\|_p^2)$ from [13, Remark 3]

$$\begin{aligned} \frac{\mu_r}{2} \mathbb{E}[\|z^{N_1} - z^*\|_p^r] &\leq \mathbb{E}[f(\hat{x}^{N_1}, y^*) - f(x^*, \hat{y}^{N_1})] \\ &= \tilde{\mathcal{O}}\left(\frac{a_q M_2 \sqrt{d}}{\sqrt{N_1}} \sqrt{\mathbb{E}[\|z^1 - z^*\|_p^2]}\right). \end{aligned} \quad (31)$$

Then the l.h.s of Eq. (31) we use the Jensen inequality and get the following

$$\begin{aligned} \frac{\mu_r}{2} (\mathbb{E}[\|z^{N_1} - z^*\|_p^2])^{r/2} &\leq \frac{\mu_r}{2} \mathbb{E}[\|z^{N_1} - z^*\|_p^r] \leq \mathbb{E}[f(\hat{x}^{N_1}, y^*) - f(x^*, \hat{y}^{N_1})] \\ &= \tilde{\mathcal{O}}\left(\frac{a_q M_2 \sqrt{d}}{\sqrt{N_1}} \sqrt{\mathbb{E}[\|z^1 - z^*\|_p^2]}\right). \end{aligned} \quad (32)$$

Finally, let us introduce $R_k \triangleq \sqrt{\mathbb{E}[\|z^{N_k} - z^*\|_p^2]}$ and $R_0 \triangleq \sqrt{\mathbb{E}[\|z^1 - z^*\|_p^2]}$. Then we take N_1 so as to halve the distance to the solution and get

$$N_1 = \tilde{\mathcal{O}}\left(\frac{a_q^2 M_2^2 d}{\mu_r^2 R_0^{2(r-1)}}\right).$$

Next, after N_1 iterations, we restart the original method and set $z^1 = z^{N_1}$. We determine N_2 similarly: we halve the distance R_1 to the solution, and so on. Thus, after k restarts, the total number of iterations will be

$$N_{acc} = N_1 + \dots + N_k = \tilde{\mathcal{O}}\left(\frac{2^{2(r-1)} a_q^2 M_2^2 d}{\mu_r^2 R_0^{2(r-1)}} \left(1 + 2^{2(r-1)} + \dots + 2^{2(k-1)(r-1)}\right)\right). \quad (33)$$

Now we need to determine the number of restarts. To do this, we fix the desired accuracy and using the inequality (31) we obtain

$$\mathbb{E}[\epsilon_{sad}] = \tilde{\mathcal{O}}\left(\frac{\mu_r R_k^r}{2}\right) = \tilde{\mathcal{O}}\left(\frac{a_q M_2 \sqrt{d}}{\sqrt{N_k}} R_{k-1}\right) = \tilde{\mathcal{O}}\left(\frac{\mu_r R_0^r}{2^{kr}}\right) \leq \epsilon. \quad (34)$$

Then to fulfill this condition, one can choose $k = \log_2(\tilde{\mathcal{O}}(\mu_r R_0^r/\epsilon))/r$ and using Eq. (33) we get the total number of iterations

$$N_{acc} = \tilde{\mathcal{O}}\left(\frac{2^{2k(r-1)} a_q^2 M_2^2 d}{\mu_r^2 R_0^{2(r-1)}}\right) = \tilde{\mathcal{O}}\left(\frac{a_q^2 M_2^2 d}{\mu_r^{2/r} \epsilon^{2(r-1)/r}}\right).$$

□

Remark 1. If in Theorem 2, we use a tighter inequality (15) instead of (27) (as in Theorem 1), then the estimations on the Δ and $M_{2,\delta}$ can be improved. Choosing $u = (x^*, y^*)$ we can provide exponentially decreasing sequence of $\mathcal{D}^k = \mathbb{E}\|z^k - u\|_2$ in Eq. (30) and get

1. under Assumption 2 $\Delta \lesssim \frac{\mu_r^{1/r} \epsilon^{2-1/r}}{M_2 \sqrt{d}}$
2. under Assumption 3 $M_{2,\delta} \lesssim \frac{\mu_r^{1/r} \epsilon^{1-1/r}}{\sqrt{d}}$.

5 Infinite Noise Variance

When the second moment of the stochastic gradient $\nabla f(z, \xi)$ is unbounded the rate of convergence may changes dramatically, see the next section. For such a case, we consider a more general inequality for Assumptions 1. We suppose that there exists a positive constant \tilde{M}_2 such that for $M_2(\xi)$ Assumptions 1 the following holds

$$\mathbb{E}[M_2(\xi)^{1+\kappa}] \leq \tilde{M}_2^{1+\kappa},$$

where $\kappa \in (0, 1]$. From [24, Lemmas 9–11] the following can be obtained

1. under Assumption 2:

$$\mathbb{E}[\|g(z, \xi, \mathbf{e})\|_q^{1+\kappa}] \leq \tilde{c} a_q^2 d^{(1+\kappa)/2} \tilde{M}_2^{1+\kappa} + 2^{1+\kappa} d^{1+\kappa} a_q^2 \Delta^2 \tau^{-2} = \tilde{M}_{case1}^{1+\kappa},$$

2. under Assumption 3:

$$\mathbb{E}[\|g(z, \xi, \mathbf{e})\|_q^{1+\kappa}] \leq \tilde{c} a_q^2 d^{(1+\kappa)/2} (\tilde{M}_2^{1+\kappa} + M_{2,\delta}^{1+\kappa}) = \tilde{M}_{case2}^{1+\kappa},$$

where \tilde{c} is some numerical constant and $\sqrt{\mathbb{E}_e[\|e\|_q^{2+2\kappa}]} \leq \tilde{a}_q^2$. As a particular case: $\tilde{a}_2^2 = 1$, $\tilde{a}_\infty^2 = \mathcal{O}\left(\frac{\log^{(1+\kappa)/2} d}{d^{(1+\kappa)/2}}\right)$.

Let us assume that $q \in [1 + \kappa, \infty)$, $1/p + 1/q = 1$ and is prox-function determined by

$$\omega(x) = K_q^{1/\kappa} \frac{\kappa}{1 + \kappa} \|x\|_p^{\frac{1+\kappa}{\kappa}} \text{ with } K_q = 10 \max\left\{1, (q-1)^{(1+\kappa)/2}\right\}.$$

Based on [28] one can prove the first part of Theorem 2 with M_2 replaced by \tilde{M}_2 , and the following stepsize

$$\gamma_k = \frac{((1 + \kappa)V_{z^1}(z^*)/\kappa)^{\frac{1}{1+\kappa}}}{\tilde{M}_{case}} N^{-\frac{1}{1+\kappa}}.$$

The first terms in the r.h.s. of (24), (25) will be determined as follows

$$\tilde{M}_{case} \left(\frac{1 + \kappa}{\kappa} V_{z^1}(z^*) \right)^{\frac{\kappa}{1+\kappa}} N^{-\frac{\kappa}{1+\kappa}}.$$

These results can be further generalized to the r -growth condition ($r \geq 2$) for duality gap.

6 Conclusion

In this paper, we demonstrate how to solve non-smooth stochastic convex-concave saddle point problems with two-point gradient-free oracle. In the Euclidean proximal setup, we obtain oracle complexity bound proportional to d/ϵ^2 that is optimal. We also generalize this result for an arbitrary proximal setup and obtain a tight upper bound on maximal level of additive adversary noise in oracle calls proportional to ϵ^2/\sqrt{d} . We generalize this result for the class of saddle point problems satisfying the r -growth condition for duality gap and get a bound which is proportional to $d/\epsilon^{2(r-1)/r}$ for the oracle complexity in the Euclidean proximal setup with $\sim \epsilon^2/\sqrt{d}$ maximal level of additive adversarial noise in oracle calls. For more details, see the arXiv version <https://arxiv.org/pdf/2202.06114.pdf> with complete proofs of all statements. The obtained results can be probably generalized to infinite noise variance [28]. We plan to develop this idea in a separate paper.

A Auxiliary Results

This appendix presents auxiliary results to prove Theorem 1 from Sect. 3.

Lemma 1. *Let vector \mathbf{e} be a random unit vector from the Euclidean unit sphere $\{\mathbf{e} : \|\mathbf{e}\|_2 = 1\}$. Then it holds for all $r \in \mathbb{R}^d$*

$$\mathbb{E}_{\mathbf{e}} [|\langle \mathbf{e}, r \rangle|] \leq \|r\|_2 / \sqrt{d}.$$

Lemma 2. *Let $f(z)$ be M_2 -Lipschitz continuous. Then for $f^\tau(z)$ from (4), it holds*

$$\sup_{z \in \mathcal{Z}} |f^\tau(z) - f(z)| \leq \tau M_2.$$

Lemma 3. *Function $f^\tau(z)$ is differentiable with the following gradient*

$$\nabla f^\tau(z) = \mathbb{E}_{\mathbf{e}} \left[\frac{d}{\tau} f(z + \tau \mathbf{e}) \mathbf{e} \right].$$

Lemma 4. *For $g(z, \xi, \mathbf{e})$ from (3) and $f^\tau(z)$ from (4), the following holds*

1. *under Assumption 2*

$$\mathbb{E}_{\xi, \mathbf{e}} [\langle g(z, \xi, \mathbf{e}), r \rangle] \geq \langle \nabla f^\tau(z), r \rangle - d \Delta \tau^{-1} \mathbb{E}_{\mathbf{e}} [|\langle \mathbf{e}, r \rangle|],$$

2. under Assumption 3

$$\mathbb{E}_{\xi, \mathbf{e}} [\langle g(z, \xi, \mathbf{e}), r \rangle] \geq \langle \nabla f^\tau(z), r \rangle - dM_{2, \delta} \mathbb{E}_{\mathbf{e}} [\|\mathbf{e}, r\|],$$

Lemma 5 [24, Lemma 9]. For any function $f(\mathbf{e})$ which is M -Lipschitz w.r.t. the ℓ_2 -norm, it holds that if \mathbf{e} is uniformly distributed on the Euclidean unit sphere, then

$$\sqrt{\mathbb{E} [(f(\mathbf{e}) - \mathbb{E}f(\mathbf{e}))^4]} \leq cM_2^2/d$$

for some numerical constant c .

Lemma 6. For $g(z, \xi, \mathbf{e})$ from (3), the following holds under Assumption 1

1. and Assumption 2

$$\mathbb{E}_{\xi, \mathbf{e}} [\|g(z, \xi, \mathbf{e})\|_q^2] \leq ca_q^2 dM_2^2 + d^2 a_q^2 \Delta^2 / \tau^2,$$

2. and Assumption 3

$$\mathbb{E}_{\xi, \mathbf{e}} [\|g(z, \xi, \mathbf{e})\|_q^2] \leq ca_q^2 d(M_2^2 + M_{2, \delta}^2),$$

where c is some numerical constant and $\sqrt{\mathbb{E} [\|\mathbf{e}\|_q^4]} \leq a_q^2$.

References

1. Bartlett, P., Dani, V., Hayes, T., Kakade, S., Rakhlin, A., Tewari, A.: High-probability regret bounds for bandit online linear optimization. In: Proceedings of the 21st Annual Conference on Learning Theory, COLT 2008, pp. 335–342. Omnipress (2008)
2. Bayandina, A.S., Gasnikov, A.V., Lagunovskaya, A.A.: Gradient-free two-point methods for solving stochastic nonsmooth convex optimization problems with small non-random noises. *Autom. Remote. Control.* **79**(8), 1399–1408 (2018)
3. Ben-Tal, A., Nemirovski, A.: *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM (2013)
4. Beznosikov, A., Sadiev, A., Gasnikov, A.: Gradient-free methods with inexact oracle for convex-concave stochastic saddle-point problem. In: Kochetov, Y., Bykadorov, I., Gruzdeva, T. (eds.) *MOTOR 2020*. CCIS, vol. 1275, pp. 105–119. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58657-7_11
5. Bubeck, S.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends® Mach. Learn.* **5**(1), 1–122 (2012). <https://doi.org/10.1561/22000000024>
6. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26 (2017)
7. Choromanski, K., Rowland, M., Sindhvani, V., Turner, R., Weller, A.: Structured evolution with compact architectures for scalable policy optimization. In: *International Conference on Machine Learning*, pp. 970–978. PMLR (2018)

8. Conn, A., Scheinberg, K., Vicente, L.: Introduction to Derivative-Free Optimization. Society for Industrial and Applied Mathematics (2009). <https://doi.org/10.1137/1.9780898718768>. <http://epubs.siam.org/doi/abs/10.1137/1.9780898718768>
9. Duchi, J.C., Jordan, M.I., Wainwright, M.J., Wibisono, A.: Optimal rates for zero-order convex optimization: the power of two function evaluations. *IEEE Trans. Inf. Theor.* **61**(5), 2788–2806 (2015). <https://doi.org/10.1109/TIT.2015.2409256>. [arXiv:1312.2139](https://arxiv.org/abs/1312.2139)
10. Flaxman, A.D., Kalai, A.T., McMahan, H.B.: Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint arXiv:cs/0408007* (2004)
11. Gasnikov, A.V., Krymova, E.A., Lagunovskaya, A.A., Usmanova, I.N., Fedorenko, F.A.: Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case. *Autom. Remote Control* **78**(2), 224–234 (2017). <https://doi.org/10.1134/S0005117917020035>. [arXiv:1509.01679](https://arxiv.org/abs/1509.01679)
12. Gasnikov, A., et al.: The power of first-order smooth optimization for black-box non-smooth problems. *arXiv preprint arXiv:2201.12289* (2022)
13. Gasnikov, A.V., Nesterov, Y.E.: Universal method for stochastic composite optimization problems. *Comput. Math. Math. Phys.* **58**(1), 48–64 (2018)
14. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems 27* (2014)
15. Gorbunov, É.A., Vorontsova, E.A., Gasnikov, A.V.: On the upper bound for the expectation of the norm of a vector uniformly distributed on the sphere and the phenomenon of concentration of uniform measure on the sphere. *Math. Notes* **106**, 11–19 (2019)
16. Juditsky, A., Nesterov, Y.: Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stoch. Syst.* **4**(1), 44–80 (2014). <https://doi.org/10.1287/10-SSY010>
17. Mania, H., Guy, A., Recht, B.: Simple random search of static linear policies is competitive for reinforcement learning. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 1805–1814 (2018)
18. Nemirovskij, A.S., Yudin, D.B.: *Problem Complexity and Method Efficiency in Optimization* (1983)
19. Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. *Found. Comput. Math.* **17**(2), 527–566 (2017). <https://doi.org/10.1007/s10208-015-9296-2>. First appeared in 2011 as CORE discussion paper 2011/16
20. Neumann, J.: Zur theorie der gesellschaftsspiele. *Mathematische annalen* **100**(1), 295–320 (1928)
21. Polyak, B.: *Introduction to Optimization*. Optimization Software, New York (1987)
22. Risteski, A., Li, Y.: Algorithms and matching lower bounds for approximately-convex optimization. *Adv. Neural. Inf. Process. Syst.* **29**, 4745–4753 (2016)
23. Sergeyev, Y.D., Candelieri, A., Kvasov, D.E., Perego, R.: Safe global optimization of expensive noisy black-box functions in the δ -lipschitz framework. *Soft. Comput.* **24**(23), 17715–17735 (2020)
24. Shamir, O.: An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *J. Mach. Learn. Res.* **18**, 52:1–52:11 (2017). <http://jmlr.org/papers/v18/papers/v18/16-632.html>. First appeared in [arXiv:1507.08752](https://arxiv.org/abs/1507.08752)
25. Shapiro, A., Dentcheva, D., Ruszczyński, A.: *Lectures on Stochastic Programming: Modeling and Theory*. SIAM (2021)
26. Spall, J.C.: *Introduction to Stochastic Search and Optimization*, 1st edn. Wiley, New York (2003)

27. Vasin, A., Gasnikov, A., Spokoiny, V.: Stopping rules for accelerated gradient methods with additive noise in gradient (2021)
28. Vural, N.M., Yu, L., Balasubramanian, K., Volgushev, S., Erdogdu, M.A.: Mirror descent strikes again: optimal stochastic convex optimization under infinite noise variance (2022)



Application of the Subdifferential Descent Method to a Classical Nonsmooth Variational Problem

Alexander Fominyh^(✉) 

St. Petersburg State University, 7/9 Universitetskaya nab.,
St. Petersburg 199034, Russia
alexfomster@mail.ru

Abstract. The paper considers a classical problem of calculus of variations with a nonsmooth integrand of the minimized functional. The integrand is assumed to be only subdifferentiable. Under some natural conditions the subdifferentiability of the functional considered is proved. The steepest (subdifferential) descent is found. Then the subdifferential descent method is applied to solve the initial problem. Some numerical examples demonstrate the algorithm implementation.

Keywords: Nonsmooth variational problem · Subdifferential · Subdifferential descent method

1 Introduction

Despite the fact that there are many deep theoretical results regarding nonsmooth problems of calculus of variations (see, e.g., [1–4]), the practical side of solving nondifferentiable variational problems remains rather underdeveloped. There are some numerical methods constructed for such problems (see, e.g., [5, 6]), but they usually consider only some particular cases of the problem or use some kind of smoothing technique (and the disadvantages of smoothing nondifferentiable functions are known).

This paper aims at solving the classical variational problem with a nondifferentiable (but only subdifferentiable) integrand. The method is based on reduction the initial problem to minimization of a functional of a special form. This work developed the apparatus of V. F. Demyanov scientific school (see, e.g., [7–12]).

2 Statement of the Problem

Let us give some notations of the paper. $C_n[0, T]$ is a space of n -dimensional continuous on $[0, T]$ vector-functions, which are supposed to be piecewise continuously differentiable with bounded on its domain derivative; $P_n[0, T]$ denotes

Supported by Russian Science Foundation (project 21-71-00021).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 34–45, 2022.
https://doi.org/10.1007/978-3-031-09607-5_3

a space of piecewise continuous and bounded on $[0, T]$ n -dimensional vector-functions. Denote $L_p^n(M)$, $1 \leq p < \infty$, the space of measurable on M n -dimensional vector-functions which are p -summable and $L_\infty^n(M)$ —the space of measurable on M and almost everywhere bounded n -dimensional vector-functions, where M is a measurable subset of the interval $[0, T]$. Use $\text{co}P$ to denote the convex hull of the set P . Let $B_r(c)$ ($D_r(c)$) denote a closed (open) ball in corresponding space with the radius r and the center c ; for some set C in this space $B_r(C)$ ($D_r(C)$) denotes the union of all closed (open) balls with the radius r and the centers from the set C . Denote $\langle a, b \rangle$ the scalar product of the vectors $a, b \in R^d$. Let X be some normed space, then $\|\cdot\|_X$ denotes norm of the introduced space and X^* —the space conjugate to the space X . For some positive number $\alpha \in R$ let $o(\alpha)$ denote such a value that we have $o(\alpha)/\alpha \rightarrow 0$ if $\alpha \rightarrow 0$.

Consider a classical variational problem: one has to minimize the following functional

$$\bar{J}(x) = \int_0^T f(x(t), \dot{x}(t), t) dt \tag{1}$$

with the boundary constraints

$$x(0) = x_0, \quad x(T) = x_T. \tag{2}$$

In expression (1) $f(x, \dot{x}, t)$, $t \in [0, T]$, is a known function, $T \in R$ is a positive known moment of time, $x(t)$ is an n -dimensional continuous vector-function, which is supposed to be continuously differentiable at each $t \in [0, T]$ with the exception of only the finite number of points, and we assume that its derivative is bounded on its domain. The function $f(x, \dot{x}, t)$ is continuous in (x, \dot{x}, t) and locally Lipschitz continuous in (x, \dot{x}) at each fixed time moment $t \in [0, T]$. In formula (2) $x_0, x_T \in R^n$ are known vectors.

In the paper we use definitions of both subdifferentials of functions in a finite-dimensional space and subdifferentials of functionals in a functional space. For convenience of presentation we separately introduce these definitions below.

Consider the space $R^n \times R^n$ with its standard norm. Let $g = [g_1, g_2]$ be an arbitrary vector from the space $R^n \times R^n$. Assume that at each moment $t \in [0, T]$ of time at the point $(x, \dot{x}) \in R^n \times R^n$ there exists such a convex compact $\partial f(x, \dot{x}, t) \subset R^n \times R^n$ that

$$\frac{\partial f(x, \dot{x}, t)}{\partial g} = \lim_{\alpha \downarrow 0} \frac{1}{\alpha} (f(x + \alpha g_1, \dot{x} + \alpha g_2, t) - f(x, \dot{x}, t)) = \max_{v \in \partial f(x, \dot{x}, t)} \langle v, g \rangle. \tag{3}$$

In this case the function $f(x, \dot{x}, t)$ is called subdifferentiable at the point (x, \dot{x}) and the set $\partial f(x, \dot{x}, t)$ is called the subdifferential of the function $f(x, \dot{x}, t)$ at the point (x, \dot{x}) .

From expression (3) it is easy to see that at each $t \in [0, T]$ we have the formula

$$f(x + \alpha g_1, \dot{x} + \alpha g_2, t) = f(x, \dot{x}, t) + \alpha \frac{\partial f(x, \dot{x}, t)}{\partial g} + o(\alpha, x, \dot{x}, g, t), \tag{4}$$

$$\frac{o(\alpha, x, \dot{x}, g, t)}{\alpha} \rightarrow 0, \alpha \downarrow 0.$$

Let for each positive number ε there exist such positive numbers δ and α_0 that at $\bar{g} \in B_\delta(g)$ and $\alpha \in (0, \alpha_0)$ we have $|o(\alpha, x, \dot{x}, \bar{g}, t)| < \alpha\varepsilon$, then the function $f(x, \dot{x}, t)$ is called uniformly subdifferentiable at the point (x, \dot{x}) . Recall [13] that if at each $t \in [0, T]$ the function $f(x, \dot{x}, t)$ is subdifferentiable at the point (x, \dot{x}) and locally Lipschitz continuous in some vicinity of the point (x, \dot{x}) , then it will be uniformly subdifferentiable at the point (x, \dot{x}) . Let for the uniformly subdifferentiable function $f(x, \dot{x}, t)$ in formula (4) one has $\frac{o(\alpha, x, \dot{x}, g, t)}{\alpha} \rightarrow 0$, $\alpha \downarrow 0$, uniformly in $t \in [0, T]$, then such a function is called absolutely uniformly subdifferentiable.

Consider the space $C_n[0, T] \times P_n[0, T]$ with the norm $L_2^n[0, T] \times L_2^n[0, T]$. Let $g = [g_1, g_2]$ be an arbitrary vector-function from the space $C_n[0, T] \times P_n[0, T]$. Assume that at the point $(x, z) \in C_n[0, T] \times P_n[0, T]$ there exists such a convex weakly* compact set $\underline{\partial}I(x, z)$ from the space $(C_n[0, T] \times P_n[0, T], \|\cdot\|_{L_2^n[0, T] \times L_2^n[0, T]})^*$ that

$$\frac{\partial I(x, z)}{\partial g} = \lim_{\alpha \downarrow 0} \frac{1}{\alpha} (I(x + \alpha g_1, z + \alpha g_2) - I(x, z)) = \max_{v \in \underline{\partial}I(x, z)} v(g). \quad (5)$$

Then the functional $I(x, z)$ is called subdifferentiable at the point (x, z) , and the set $\underline{\partial}I(x, z)$ is called the subdifferential of the functional $I(x, z)$ at this point (x, z) .

From formula (5) it is easy to check that we have the following expression

$$I(x + \alpha g_1, z + \alpha g_2) = I(x, z) + \alpha \frac{\partial I(x, z)}{\partial g} + o(\alpha, x, z, g), \quad (6)$$

$$\frac{o(\alpha, x, z, g)}{\alpha} \rightarrow 0, \alpha \downarrow 0.$$

Thus, one has to obtain such a vector-function $x^* \in C_n[0, T]$, which minimizes functional (1) and fullfills boundary restrictions (2). Suppose that there exists a required solution. Note that in classical problems of variational calculus the integrand is smooth and in the problem under consideration it is supposed to be only subdifferentiable.

3 Reduction to an Unconstrained Minimization Problem

Consider the following functional which takes into account all the restrictions in the formulation of the original problem. Let $z(t) = \dot{x}(t)$ (as we have noted, $z \in P_n[0, T]$), then by the first equality in formula (2) we obtain

$$x(t) = x_0 + \int_0^t z(\tau) d\tau. \quad (7)$$

The idea of this paper is to “forcibly” consider the points z and x as the “independent” variables. Of course, in fact, there is obvious relationship (7) between these variables, so let us take this into account by using the last in the following functional (on the space $C_n[0, T] \times P_n[0, T]$)

$$\begin{aligned}
 I(x, z) &= J(x, z) + \lambda\psi(z) + \lambda\varphi(x, z) \tag{8} \\
 &= \int_0^T f(x(t), z(t), t)dt \\
 &+ \lambda \frac{1}{2} \left(x_0 + \int_0^T z(t)dt - x_T \right)^2 + \lambda \frac{1}{2} \int_0^T \left(x(t) - x_0 - \int_0^t z(\tau)d\tau \right)^2 dt.
 \end{aligned}$$

We see that the dimension of this functional arguments is n more the dimension of the initial problem, but the structure of its subdifferential (in the space $C_n[0, T] \times P_n[0, T]$ as a normed space with the norm $L_2^n[0, T] \times L_2^n[0, T]$), as will be seen below, is rather simple. This fact will allow us to develop a numerical method for solving the original problem (and there will be effective well-known methods for solving the arising subproblems).

As is well-known [14], if the value λ is sufficiently large, then the solution of problem (1), (2) is arbitrarily close (with regard to the used metric $L_2^n[0, T]$) to the trajectory $\bar{x}(t)$ where (\bar{x}, \bar{z}) is a point of the global minimum of functional (8) with some fixed value $\bar{\lambda}$. So, we have reduced the original problem to minimizing functional (8) on the space $C_n[0, T] \times P_n[0, T]$. In practice, first, one has to solve this problem for some fixed number $\bar{\lambda}$. If the solution obtained (at $\lambda = \bar{\lambda}$) satisfies the restrictions in the form of differential relation (7) and right endpoint constraint from (2) with the required accuracy (i.e. the value of the functional $\psi + \varphi$ on the solution obtained is rather small), then we finish the process; otherwise, one should increase the value λ and restart the process with this new number.

So, now our aim is to solve the unconditional minimization problem for the functional $I(x, z)$ (for some rather large value $\bar{\lambda}$) on the space

$$X = (C_n[0, T] \times P_n[0, T], \| \cdot \|_{L_2^n[0, T] \times L_2^n[0, T]}). \tag{9}$$

Remark 1. Recall that the space $(C_n[0, T], \| \cdot \|_{L_2^n[0, T]})$ is everywhere dense in the space $L_2^n[0, T]$ and also the space $(P_n[0, T], \| \cdot \|_{L_2^n[0, T]})$ is everywhere dense in the space $L_2^n[0, T]$, so as is known [15] the space X^* conjugate to the space X (see (9)) is isometrically isomorphic to the space $L_2^n[0, T] \times L_2^n[0, T]$; hence, below in the paper these spaces (X^* and $L_2^n[0, T] \times L_2^n[0, T]$) are identified.

4 Minimum Conditions for the Functional $I(x, z)$

In order to obtain the minimum condition, useful for constructing numerical methods for the original, at first, let us investigate the differential properties of the functional $I(x, z)$.

With the help of classical variation one can easily check Gateaux differentiability of the functional $\psi(z)$:

$$\nabla\psi(z) = x_0 + \int_0^T z(t)dt - x_T.$$

With the help of classical variation and integration by parts one can also easily check Gateaux differentiability of the functional $\varphi(x, z)$:

$$\nabla\varphi(x, z, t) = \left(\begin{array}{c} x(t) - x_0 - \int_0^t z(\tau)d\tau \\ - \int_t^T \left(x(\tau) - x_0 - \int_0^\tau z(s)ds \right) d\tau \end{array} \right).$$

Turn to the differential properties of the functional $\int_0^T f(x(t), z(t), t)dt$. Recall that the variables x and z are considered in this functional as independent ones, so put $\xi(t) = (x(t), z(t))$ for brevity and prove the following theorem (we retain the previous notation for the functional $J(x, z)$ for convenience).

Theorem 2. There is a functional

$$J(\xi) = \int_0^T f(\xi(t), t)dt,$$

where $\xi \in C_n[0, T] \times P_n[0, T]$, the function $f(\xi, t)$ is continuous in (ξ, t) and is absolutely uniformly subdifferentiable and its subdifferential is $\underline{\partial}f(\xi, t)$. Assume also the mapping $t \rightarrow \underline{\partial}f(\xi(t), t)$ to be upper semicontinuous.

Then the functional $J(\xi)$ is subdifferentiable, i.e.

$$\frac{\partial J(\xi)}{\partial g} = \lim_{\alpha \downarrow 0} \frac{1}{\alpha} (J(\xi + \alpha g) - J(\xi)) = \max_{v \in \underline{\partial}J(\xi)} \int_0^T \langle v(t), g(t) \rangle dt, \tag{10}$$

where $g \in C_n[0, T] \times P_n[0, T]$ and the set $\underline{\partial}J(\xi)$ is defined as follows:

$$\underline{\partial}J(\xi) = \left\{ v(t) \in L_\infty^{2n}[0, T] \mid v(t) \in \underline{\partial}f(\xi(t), t) \ \forall t \in [0, T] \right\}. \tag{11}$$

Proof. Let us give just the scheme of proof for brevity:

- 1) With the use of the expansion (6), the absolutely uniformly quasidifferentiability definition and Filippov lemma we first show that the direction derivative of the functional $J(\xi)$ is of form (10).
- 2) Now our aim is to show that the set $\underline{\partial}J(\xi)$ in formula (11) is convex and weakly* compact in the space X^* :

- a) this set is convex due to the convexity of the function $f(\xi, t)$ quasidifferential;
- b) we show that the set $\underline{\partial}J(\xi)$ is bounded in $L_2^{2n}[0, T]$ -norm via quasidifferentiability of the function $f(\xi, t)$ and upper-semicontinuity of the mapping $t \rightarrow \underline{\partial}f(\xi(t), t)$;
- c) recall the following known fact from functional analysis: if v_n is the sequence of functions from the set $\underline{\partial}J(\xi)$ converging to the function v^* in the strong topology of the space $L_2^{2n}[0, T]$, then this sequence has the subsequence v_{n_k} converging pointwise to v^* almost everywhere on $[0, T]$; using this fact and the function $f(\xi, t)$ quasidifferential definition we show that the set $\underline{\partial}J(\xi)$ is weakly closed in the space $L_2^{2n}[0, T]$;
- d) finally, we use Remark 1, the fact that weak compactness is equivalent to weak* compactness in the space $L_2^{2n}[0, T]$ and the fact that weak compactness is equivalent to boundedness in norm and weak closedness in the space $L_2^{2n}[0, T]$ in order to finish the proof.

As is seen from Theorem 2, the subdifferential of the functional $J(\xi)$ is defined by the subdifferential of its integrand (at each time moment $t \in [0, T]$). Hence, in order to obtain the subdifferential of the functional $J(x, z)$, one has to calculate the set $\underline{\partial}f(x, \dot{x}, t)$ for each time moment $t \in [0, T]$ with the use of subdifferential calculus [13]. In book [13] one can find the required rules for calculating the subdifferential for a wide range of functions.

Using formula (11) and the noted subdifferential calculus rules, we finally obtain the expression for calculating the functional $I(x, z)$ subdifferential (at the point (x, z))

$$\underline{\partial}I(x, z) = \sum_{k=1}^3 \underline{\partial}I_k(x, z), \tag{12}$$

(we have formally put $I_1(x, z) = J(x, z)$, $I_2(x, z) = \lambda\psi(z)$, $I_3(x, z) = \lambda\varphi(x, z)$).

Recall the well-known necessary minimum condition of the subdifferentiable functional $I(x, z)$ at the point (\bar{x}, \bar{z}) [4]

$$0_{2n} \in \underline{\partial}I(\bar{x}, \bar{z}),$$

where 0_{2n} is a zero element of the space $L_2^{2n}[0, T]$. Finally we obtain the theorem.

Theorem 3. For the point (\bar{x}, \bar{z}) to minimize functional (8), it is necessary for the inclusion

$$\mathbf{0}_{2n} \in \underline{\partial}I(\bar{x}(t), \bar{z}(t)) \tag{13}$$

to be satisfied for almost every $t \in [0, T]$, where $\mathbf{0}_{2n}$ is a zero element of the space R^{2n} , and the subdifferential $\underline{\partial}I(x, z)$ is calculated by formula (12).

Remark 2. Inclusion (13) is a minimum condition in constructive form, since it is possible to develop a numerical method on its basis (the simple version of possible numerical methods is described in the next section). The idea of this paper is to apply the well-known algorithm to the specially constructed functional $I(x, z)$ in formula (8) (recall that its variables are considered as “independent” ones). As it will be seen in the next section, in this case it is possible to solve the arising subproblems of the method via known effective algorithms.

5 The Subdifferential Descent Method

Recall the known subdifferential descent algorithm for minimization the functional $I(x, z)$.

Fix some arbitrary initial point $(x_{(1)}, z_{(1)}) \in C_n[0, T] \times P_n[0, T]$. Let the point $(x_{(k)}, z_{(k)}) \in C_n[0, T] \times P_n[0, T]$ be already constructed. If minimum condition (13) is satisfied (in practice with some fixed accuracy $\bar{\varepsilon}$), then the point $(x_{(k)}, z_{(k)})$ is a stationary point of the functional $I(x, z)$ and the process ends. Otherwise, put

$$(x_{(k+1)}, z_{(k+1)}) = (x_{(k)}, z_{(k)}) + \gamma_{(k)} G(x_{(k)}, z_{(k)}),$$

where the vector-function $G(x_{(k)}, z_{(k)})$ is the steepest (subdifferential) descent direction of the functional $I(x, z)$ at the point $(x_{(k)}, z_{(k)})$, and the value $\gamma_{(k)}$ is a solution of the following one-dimensional minimization problem

$$\min_{\gamma \geq 0} I\left((x_{(k)}, z_{(k)}) + \gamma G(x_{(k)}, z_{(k)})\right) = I\left((x_{(k)}, z_{(k)}) + \gamma_{(k)} G(x_{(k)}, z_{(k)})\right). \quad (14)$$

Then, as one can easily check, we obtain

$$I(x_{(k+1)}, z_{(k+1)}) < I(x_{(k)}, z_{(k)})$$

(and the vector-function $G(x_{(k)}, z_{(k)})$ is indeed the steepest descent direction).

As is seen from the algorithm description, one has to solve three subproblems on each iteration. The first one is obtaining the subdifferential of the functional $I(x, z)$ at the point $(x_{(k)}, z_{(k)})$. The solution of this problem is given by formula (12). The second problem is finding the steepest descent direction $G(x_{(k)}, z_{(k)})$; two next paragraphs are devoted to solving this subproblem. The third problem is one-dimensional minimization (14); and there exist many known methods [14] which solve this problem effectively.

In order to find the vector-function $G(x_{(k)}, z_{(k)})$, consider the problem

$$\min_{v \in \partial I(x_{(k)}, z_{(k)})} \|v\|_{L_2^n[0, T] \times L_2^n[0, T]}^2 = \min_{v \in \partial I(x_{(k)}, z_{(k)})} \int_0^T v^2(t) dt. \quad (15)$$

Denote $\bar{v}_{(k)}$ its solution. The vector-function $\bar{v}_{(k)}(t)$ depends on the point $(x_{(k)}, z_{(k)})$, but we omit this dependence in the paper for convenience. The vector-function

$$G(x_{(k)}(t), z_{(k)}(t), t) = -\bar{v}_{(k)}(x_{(k)}(t), z_{(k)}(t), t)$$

is called a subdifferential descent direction of the functional $I(x, z)$ at the point $(x_{(k)}, z_{(k)})$.

One can easily check that the solution of this problem is such selector of the multivalued mapping $t \rightarrow \partial I(x_{(k)}(t), z_{(k)}(t), t)$ that minimizes the distance from zero point to the set $\partial I(x_{(k)}(t), z_{(k)}(t), t)$ at each time moment $t \in [0, T]$. So, in order to solve problem (15) one has to solve the following problem

$$\min_{v(t) \in \partial I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) \quad (16)$$

for each $t \in [0, T]$. Actually, for every $t \in [0, T]$ we have the obvious inequality

$$\min_{v \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) \leq v^2(t),$$

where $v(t)$ is a measurable selector of the mapping $t \rightarrow \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)$ (by virtue of the noted in the Theorem 2 proof scheme property of the set $\underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)$ boundedness uniformly in $t \in [0, T]$ we have $v \in L_\infty^{2n}[0, T]$), then we obtain the inequality

$$\int_0^T \min_{v \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) dt \leq \min_{v \in \underline{\partial}I(x_{(k)}, z_{(k)})} \int_0^T v^2(t) dt.$$

Insofar as for every $t \in [0, T]$ we have

$$\min_{v \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) \in \left\{ v^2(t) \mid v(t) \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t) \right\}$$

and the set $\underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)$ is closed and bounded at every fixed t by definition of the subdifferential and the mapping $t \rightarrow \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)$ is upper semicontinuous by assumption and besides, the norm is continuous in its argument, then due to Filippov lemma [16] there exists such a measurable selector $\bar{v}_k(t)$ of the mapping $t \rightarrow \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)$ that for every $t \in [0, T]$ one obtains

$$\min_{v \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) = \bar{v}_k^2(t),$$

so we have found the element \bar{v}_k of the set $\underline{\partial}I(x_{(k)}, z_{(k)})$ which brings the equality to the previous inequality. Hence, finally we obtain

$$\int_0^T \min_{v \in \underline{\partial}I(x_{(k)}(t), z_{(k)}(t), t)} v^2(t) dt = \min_{v \in \underline{\partial}I(x_{(k)}, z_{(k)})} \int_0^T v^2(t) dt.$$

Problem (16) at each fixed time moment $t \in [0, T]$ is the finite-dimensional problem of finding the distance from zero point to a convex compact (the subdifferential) $\underline{\partial}I(x(t), z(t))$. In order to solve this problem in practice one makes a (uniform) partition of the interval $[0, T]$, and solves this problem for every point of the partition, i.e. one has to calculate $G(x_{(k)}(t_i), z_{(k)}(t_i), t_i)$ where $t_i \in [0, T]$, $i = \bar{1}, \bar{N}$, are the points of discretization (see notation in Lemma 1 below). Under some natural assumption this lemma guarantees that the vector-function obtained with the help of piecewise linear interpolation of the subdifferential descent directions evaluated at every point of such partition of the interval $[0, T]$, converges to the sought vector-function $G(x_{(k)}(t), z_{(k)}(t), t)$ in the space $L_2^{2n}[0, T]$ when the discretization rank tends to infinity. Note that in most practical cases the subdifferential $\underline{\partial}I(x(t), z(t))$ at each moment of time $t \in [0, T]$ has a rather simple structure [13]. If, for example, the integrand is a maximum of the finite number of continuously differentiable functions (as in Example 1 of the next section), then the subdifferential $\underline{\partial}I(x(t), z(t))$ is a convex polyhedron

at each $t \in [0, T]$. This problem of finding the Euclidean distance from a point to a convex polyhedron can be effectively solved by various methods (see, e.g., [17, 18]). In a more general case the subdifferential at each moment $t \in [0, T]$ of time may be a convex compact set (for example, if the integrand depends on the norm of some coordinates of the vector-functions $x(t)$, $z(t)$ (as in Example 2 of the next section), then the subdifferential at some points $t \in [0, T]$ may be an ellipsoid (with its interior points), lying in some subspace of the space R^{2n}). In this case it is required to solve the problem of finding the Euclidean distance from a point to a convex compact set, and if (for example) ellipsoids are considered, then some methods for solving this problem can be found in [19].

Prove the following lemma with a simple natural condition which holds true for applications and at the same time guarantees that the function $L(t)$ obtained with the help of piecewise linear interpolation of the sought function $p \in L^1_\infty[0, T]$, converges to this function in the space $L^1_2[0, T]$ while the rank of a (uniform) partition of the interval $[0, T]$ tends to infinity.

Lemma 1. Let the function $p \in L^1_\infty[0, T]$ satisfy the following condition: for every $\bar{\delta} > 0$ the function $p(t)$ is piecewise continuous on the set $[0, T]$ with the exception of only the finite number of the intervals $(\bar{t}_1(\bar{\delta}), \bar{t}_2(\bar{\delta}))$, \dots , $(\bar{t}_r(\bar{\delta}), \bar{t}_{r+1}(\bar{\delta}))$ whose union length does not exceed the number $\bar{\delta}$.

Choose the (uniform) finite splitting $t_1 = 0, t_2, \dots, t_{N-1}, t_N = T$ of the interval $[0, T]$ and calculate the values $p(t_i)$, $i = \overline{1, N}$, at these points. Let $L(t)$ be the function obtained with the help of piecewise linear interpolation with the nodes $(t_i, p(t_i))$, $i = \overline{1, N}$. Then for every $\varepsilon > 0$ there exists such number $\bar{N}(\varepsilon)$ that for every $N > \bar{N}(\varepsilon)$ one has $\|L - p\|_{L^1_2[0, T]} \leq \varepsilon$.

Proof. Let us give just the scheme of proof for brevity:

- 1) We denote $M(\bar{\delta}) := \bigcup_{k=1}^r (\bar{t}_k(\bar{\delta}), \bar{t}_{k+1}(\bar{\delta}))$ and divide the interval $[0, T]$ into two parts: $M(\bar{\delta})$ and $[0, T] \setminus M(\bar{\delta})$.
- 2) We use the property of the set $M(\bar{\delta})$ and boundedness of the functions $p(t)$ and $L(t)$ for all (uniform) finite partitions of the interval $[0, T]$ in order to show that the value $\|L - p\|_{L^1_2(M(\bar{\delta}))}^2$ can be made arbitrarily small.
- 3) We use the property of the set $M(\bar{\delta})$ and the fact that the piecewise continuous function $p(t)$ can be arbitrarily closely approximated by the function $L(t)$ on the set $[0, T] \setminus M(\bar{\delta})$ in order to show that the value $\|L - p\|_{L^1_2([0, T] \setminus M(\bar{\delta}))}^2$ can be made arbitrarily small.

6 Numerical Examples

Let us give some examples of the subdifferential descent method implementation. The stopping criteria of the method is the inequality $\|\bar{v}^{(k)}\|_{L^2_2[0, T] \times L^2_2[0, T]} \leq \bar{\varepsilon}$ (see problem (15)). The value $\bar{\varepsilon}$ was taken equal to $5 \times 10^{-2} - 9 \times 10^{-2}$. Such a

choice of accuracy is explained by the compromise between the permissible for practical needs accuracy and a not very large number of iterations to realize. As it will be seen, the error of the minimized functional and the restrictions on the right endpoint in the examples of this section did not exceed the value 5×10^{-3} .

Example 1. Consider a minimization of the functional

$$\begin{aligned} \bar{J}(x) &= \int_0^1 |x(t) - \max\{t - 0.5, 0\}| dt \\ &= \int_0^1 \max\{x(t) - \max\{t - 0.5, 0\}, -x(t) + \max\{t - 0.5, 0\}\} dt, \\ x(0) &= 0, \end{aligned}$$

with the only obvious solution $x^*(t) = \max\{t - 0.5, 0\} \forall t \in [0, 1]$ and $\bar{J}(x^*) = 0$. As the considered functional is independent of the derivative $\dot{x}(t)$ and the right endpoint is free, the functionals $\psi(z)$ and $\varphi(x, z)$ are absent in functional (8), so we can put $I(x) := I(x, z)$. (Note that the solution $x^*(t)$ satisfies the initial condition.) Take $x_{(1)} = 2t - 1$ as the initial approximation, then $I(x_{(1)}) = 0.375$. As the iteration number increased, the discretization rank also increased during the solution of the subproblem of finding the direction of the steepest descent described in the algorithm developed, and in the end the discretization step was equal to 10^{-1} . At the 28-th iteration the point

$$\begin{aligned} x_{(17)} &= 23.594853t^6 - 70.788855t^5 + 77.166539t^4 - 36.346867t^3 \\ &\quad + 7.447309t^2 - 0.573011t + 0.009928 \end{aligned}$$

was obtained, herewith $\bar{J}(x_{(17)}) = I(x_{(17)}) \approx 0.00448$, so the error does not exceed the value 5×10^{-3} . For the convenience, the Lagrange interpolation polynomial has been presented which sufficiently accurately approximates the resulting trajectory. This means that the interpolation error does not affect the value of the functional given with a set accuracy but (insignificantly) affects the presented value of the smallest in norm subgradient. Herewith, $\|\bar{v}_{(17)}\|_{L_2^1[0, T]} \approx 0.083$.

Example 2. Consider the minimization problem of the functional

$$\begin{aligned} \bar{J}(x) &= \int_0^5 \sqrt{(\dot{x}_1(t) - 1)^2 + x_2^2(t)} + (x_1(t) - x_3(t) - \sin(t))^2 dt, \\ x_1(0) &= 0, \quad x_2(0) = 0, \quad x_3(0) = 0, \end{aligned}$$

with the only obvious solution $x_1^*(t) = t$, $x_2^*(t) = 0$, $x_3^*(t) = t - \sin(t)$, $t \in [0, 5]$, and $J(x^*) = 0$. As the endpoint is free here, the functional $\psi(z)$ is absent. So, it is required to minimize the functional

$$I(x, z) = \int_0^5 \sqrt{(\dot{x}_1(t) - 1)^2 + x_2^2(t)} + (x_1(t) - x_3(t) - \sin(t))^2 dt$$

$$+ \int_0^5 \left(x(t) - \int_0^t z(\tau) d\tau \right)^2 dt,$$

where the value $\lambda = 2$ is taken. It is obvious that $z_1^*(t) = 1$, $z_2^*(t) = 0$, $z_3^*(t) = 1 - \cos(t)$, $t \in [0, 5]$, $I(x^*, z^*) = 0$.

Take $(x_{(1)}, z_{(1)}) = (0, 0, 0, 1, 0, 0)'$ as the first approximation, then $I(x_{(1)}, z_{(1)}) = 44.30267$. As the iteration number increased, the discretization rank also increased during the solution of the subproblem of finding the direction of the steepest descent described in the algorithm developed and in the end of the process the discretization step was equal to 5×10^{-2} . At the 121-st iteration the point

$$\begin{aligned} x_{(121)} &= (-0.000709t^5 + 0.005769t^4 - 0.009043t^3 - 0.015847t^2 + 1.019026t, \\ &0, 0.003552t^5 - 0.067694t^4 + 0.335731t^3 - 0.214808t^2 + 0.098457t)', \\ z_{(121)} &= (-0.000028t^4 + 0.000284t^3 - 0.001009t^2 + 0.001565t + 0.999079, \\ &0, 0.017760t^4 - 0.270776t^3 + 1.007193t^2 - 0.429616t + 0.098457)' \end{aligned}$$

was constructed and the functional value obtained is $I(x_{(121)}, z_{(121)}) = 0.00353$. For the convenience, the Lagrange interpolation polynomial has been presented which sufficiently accurately approximates the resulting trajectory. This means that the interpolation error does not affect the value of the functional given with a set accuracy but (insignificantly) affects the presented value of the smallest in norm subgradient. Herewith, we have $\|\bar{v}_{(121)}\|_{L_2^3[0,T] \times L_2^3[0,T]} \approx 0.0474$.

Since in fact we know that $z(t) = \dot{x}(t)$, $t \in [0, 5]$, then one may put

$$\begin{aligned} x_{(121)} &= (-0.000006t^5 + 0.000071t^4 - 0.000336t^3 + 0.000783t^2 + 0.999079t, \\ &0, 0.003552t^5 - 0.067694t^4 + 0.335731t^3 - 0.214808t^2 + 0.098457t)', \\ z_{(121)} &= (-0.000028t^4 + 0.000284t^3 - 0.001009t^2 + 0.001565t + 0.999079, \\ &0, 0.017760t^4 - 0.270776t^3 + 1.007193t^2 - 0.429616t + 0.098457)', \end{aligned}$$

and then $\bar{J}(x_{(121)}) = I(x_{(121)}, z_{(121)}) \approx 0.00553$, i.e. the error of the functional value does not exceed the value 6×10^{-3} .

Remark 3. In practice rather large values of λ lead to additional computational difficulties. So in the future research it is interesting to consider the functional

$$\bar{\varphi}(x, z) = \int_0^T \left| x(t) - x_0 - \int_0^t z(\tau) d\tau \right| dt$$

instead of the functional $\varphi(x, z)$ (see formula (8)). Unlike the functional $\varphi(x, z)$ the functional $\bar{\varphi}(x, z)$ is nonsmooth. However, with the use of Hölder's inequality one can make sure that its structure allows us to improve the accuracy of fulfillment of the corresponding constraint and significantly decrease the value λ . So it makes sense to compare the use of these functionals in practice.

References

1. Ioffe, A.D.: An existence theorem for problems of the calculus of variations. *Dokl. Akad. Nauk SSSR* **205**(2), 277–280 (1972)
2. Clarke, F.H.: The Erdmann condition and Hamiltonian inclusions in optimal control and the calculus of variations. *Can. J. Math.* **32**(2), 494–509 (1980)
3. Ioffe, A.D.: On generalized Bolza problem and its application to dynamic optimization. *J. Optim. Theor. Appl.* **182**, 285–309 (2019)
4. Dolgopolik, M.: Nonsmooth problems of calculus of variations via codifferentiation. *ESAIM Control Optim. Calc. Var.* **20**(4), 1153–1180 (2014)
5. Teo, K.L., Goh, C.J.: On constrained optimization problems with nonsmooth cost functionals. *Appl. Math. Optim.* **18**(1), 181–190 (1988)
6. Wu, C.Z., Teo, K.L., Zhao, Y.: Numerical method for a class of optimal control problems subject to nonsmooth functional constraints. *J. Comput. Appl. Math.* **217**(2), 311–325 (2008)
7. Tamasyan, G.S.: Numerical methods in problems of calculus of variations for functionals depending on higher order derivatives. *J. Math. Sci.* **188**(3), 299–321 (2013)
8. Demyanov, F.V., Tamasyan, G.S.: On direct methods for solving variational problems. *Proc. Steklov Instit. Math.* **16**(5), 36–47 (2010)
9. Fominyh, A.V., Karelin, V.V., Polyakova, L.N.: Application of the hypodifferential descent method to the problem of constructing an optimal control. *Optim. Lett.* **12**(8), 1825–1839 (2018)
10. Fominyh, A.V.: The quasidifferential descent method in a control problem with nonsmooth objective functional. *Optim. Lett.* **15**(2), 2773–2792 (2021)
11. Fominyh, A.V.: Methods of subdifferential and hypodifferential descent in the problem of constructing an integrally constrained program control. *Autom. Remote. Control.* **78**, 608–617 (2017)
12. Fominyh, A.V.: Open-loop control of a plant described by a system with nonsmooth right-hand side. *Comput. Math. Math. Phys.* **59**(10), 1639–1648 (2019)
13. Demyanov, V.F., Vasil'ev, L.V.: *Nondifferentiable Optimization*. Springer, New York (1986)
14. Vasil'ev, F.P.: *Optimization Methods*. Factorial Press, Moscow (2002). (in Russian)
15. Kolmogorov, A.N., Fomin, S.V.: *Elements of the Theory of Functions and Functional Analysis*. Dover Publications Inc., New York (1999)
16. Filippov, A.F.: On certain questions in the theory of optimal control. *J. Soc. Ind. Appl. Math. Ser. A Control* **1**, 76–84 (1959)
17. Demyanov, F.F., Malozemov, V.N.: *Introduction to Minimax*. Dover Publications Inc., New York (1990)
18. Wolfe, P.: The simplex method for quadratic programming. *Econometrica* **27**, 382–398 (1959)
19. Dolgopolik, M.V.: The alternating direction method of multipliers for finding the distance between ellipsoids. *Appl. Math. Comput.* **409**, 1–19 (2021)



Primal-Dual Method for Optimization Problems with Changing Constraints

Igor Konnov(✉)

Institute of Computational Mathematics and Information Technologies,
Kazan Federal University, Kremlevskaya st. 18, 420008 Kazan, Russia
konn-igor@ya.ru

<https://kpfu.ru/Igor.Konnov>

Abstract. We propose a modified primal-dual method for general convex optimization problems with changing constraints. We obtain properties of Lagrangian saddle points for these problems which enable us to establish convergence of the proposed method. We describe specializations of the proposed approach to multi-agent optimization problems under changing communication topology and to feasibility problems.

Keywords: Convex optimization · changing constraints · primal-dual method · constrained multi-agent optimization · feasibility problem

1 Introduction

It is well known that the general optimization problem consists in finding the minimal value of some goal function \tilde{f} on a feasible set \tilde{D} . For brevity, we write this problem as

$$\min_{v \in \tilde{D}} \rightarrow \tilde{f}(v).$$

In many cases, only some approximations are known instead of the exact values of the goal function and the feasible set. This situation is caused by various circumstances. On the one hand, this is due to inevitable calculation errors of values of cost and constraint functions. On the other hand, this is due to incompleteness of information about these functions since their parameters may be specialized during the computational process. Such problems are called non stationary; see e.g. [1] and [2, Chapter VI, §3]. Besides, some perturbations can be inserted for attaining better properties in comparison with the initial one as in various regularization methods; see e.g. [3]. In these problems, only some sequences of approximations $\{\tilde{D}_k\}$ and $\{\tilde{f}_k\}$ are known, which however must converge in some sense to the exact values of \tilde{D} and \tilde{f} . The case where the convergence is not obligatory seems more difficult, but it also appears in many applied problems. For instance, large-scale models may contain superfluous constraints and variables together with the necessary ones, but only some of them can be utilized at a given iterate. Various decentralized multi-agent optimization

problems can serve as examples of such systems; see e.g. [4–6] and the references therein.

In this paper we investigate just general convex optimization problems with changing constraints. This means that the set of constraints may vary from iteration to iteration. First we obtain properties of Lagrangian saddle points for these problems. They enable us to propose a modification of the primal-dual method from [7] for finding their solutions. We establish different convergence properties of the proposed method under rather weak assumptions. We describe specializations of the proposed approach to multi-agent optimization problems under changing communication topology and to feasibility problems.

2 The General Problem with Changing Constraints and Its Properties

Let us consider first a general optimization problem of the form

$$\min_{x \in D} \rightarrow f(x) \quad (1)$$

for some function $f : \mathbb{E} \rightarrow \mathbb{R}$ and set $D \subseteq \mathbb{E}$ in a finite-dimensional space \mathbb{E} . The set of its solutions is denoted by D^* , and the optimal function value by f^* , i.e.

$$f^* = \inf_{x \in D} f(x).$$

It will be suitable for us to specialize this problem as follows. For each $x \in \mathbb{E}$, let $x = (x_i)_{i=1, \dots, m}$, i.e. $x^\top = (x_1^\top, \dots, x_m^\top)$, where $x_i = (x_{i1}, \dots, x_{in})^\top$ for $i = 1, \dots, m$, hence $\mathbb{E} = \mathbb{R}^{mn}$. This means that each vector x is divided into m subvectors $x_i \in \mathbb{R}^n$. In case $n = 1$ we obtain the custom coordinates of x . Next, we suppose that

$$D = \{x \in X \mid Ax = b\}, \quad (2)$$

where X is a subset of \mathbb{R}^{mn} , the matrix A has ln rows and mn columns, so that $b = (b_i)_{i=1, \dots, l}$, $b_i \in \mathbb{R}^n$ for $i = 1, \dots, l$, and $b \in \mathbb{R}^{ln}$.

In what follows, we will use the following basic assumptions.

(A1) The set D^* is nonempty, X is a convex and closed set in \mathbb{R}^{mn} .

(A2) $f : \mathbb{R}^{mn} \rightarrow \mathbb{R}$ is a convex function.

For brevity, we set $M = \{1, \dots, m\}$ and $L = \{1, \dots, l\}$. It is clear that the matrix A is represented as follows:

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \dots \\ A_l \end{pmatrix},$$

where A_i is the corresponding $n \times mn$ sub-matrix of A for $i \in L$. We will write this briefly

$$A = (\{A_i^\top\}_{i \in L})^\top.$$

Similarly, we can determine some other submatrices

$$A_I = (\{A_i^\top\}_{i \in I})^\top$$

for any $I \subseteq L$, hence $A = A_L$. Setting

$$F_I = \{x \in \mathbb{R}^{mn} \mid A_I x = b_I\} \text{ and } D_I = \{x \in X \mid A_I x = b_I\} = X \cap F_I, \quad (3)$$

where $b_I = (b_i)_{i \in I}$, we obtain a family of optimization problems

$$\min_{x \in D_I} \rightarrow f(x). \quad (4)$$

As above, we denote the solution set of problem (3)–(4) by D_I^* , and the optimal function value by f_I^* , so that $D_L^* = D^*$ and $f_L^* = f^*$. Clearly, if $I \subset J$, then $f_I^* \leq f_J^*$. We intend to establish some properties related to superfluous constraints. We will denote by F^* the solution set of the optimization problem

$$\min_{x \in X} \rightarrow f(x),$$

and its optimal function value by f^{**} .

Lemma 1. *Suppose the set $F^* \cap F_I$ is nonempty for some $I \subseteq L$. Then $f^{**} = f_I^*$ and $F^* \cap F_I = D_I^*$.*

Proof. If $x^* \in F^* \cap F_I$, then clearly $x^* \in D_I^*$, hence $f^{**} = f_I^*$. It follows that $F^* \cap F_I = D_I^*$. \square

Definition 1. *We say that $I \subseteq J$ is a basic index set with respect to J if*

$$A_I x = b_I \implies A_J x = b_J.$$

We say that $I \subseteq L$ is a basic index set if it is a basic index set with respect to L .

From the definitions we obtain immediately the simple but useful properties.

Lemma 2.

- (i) *If $I \subset J$ is a basic index set with respect to J , then $f_I^* = f_J^*$, $D_I = D_J$, and $D_I^* = D_J^*$.*
- (ii) *If I is a basic index set, then $f_I^* = f^*$, $D_I = D$, and $D_I^* = D^*$.*

For each problem (3)–(4) associated with an index set $I \subseteq L$ we can define its Lagrange function

$$\mathcal{L}_I(x, y) = f(x) + \langle y_I, A_I x - b_I \rangle$$

and the corresponding saddle point problem. It appears more suitable to utilize the general Lagrange function

$$\mathcal{L}(x, y) = f(x) + \langle y, Ax - b \rangle,$$

with the modified dual feasible set. Namely, we say that $w^* = (x^*, y^*) \in X \times Y_I$ is a saddle point for problem (3)–(4) if

$$\forall y \in Y_I, \quad \mathcal{L}(x^*, y) \leq \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*) \quad \forall x \in X, \quad (5)$$

where

$$Y_I = \{y = (y_i)_{i \in L} \in \mathbb{R}^{ln} \mid y_i = \mathbf{0} \in \mathbb{R}^n \text{ for } i \notin I\}.$$

We denote by $W_I^* = D_I^* \times Y_I^*$ the set of saddle points in (5) since D_I^* is precisely the solution set of problem (3)–(4), whereas Y_I^* is the set of its Lagrange multipliers. Since $D_L^* = D^*$, we also set $Y^* = Y_L^*$, i.e. $W^* = D^* \times Y^*$ is the set of saddle points for the initial problem (1)–(2). Observe that (5) is rewritten equivalently as follows:

$$A_I x^* = b_I, \quad \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*) \quad \forall x \in X. \quad (6)$$

Besides, if we take $I = \emptyset$, then $Y_I = \{\mathbf{0}\}$, hence we can write $D_I^* = F^*$ and $Y_I^* = \{\mathbf{0}\}$.

Proposition 1. *Suppose that assumptions (A1)–(A2) are fulfilled. If $I \subset J$ is a basic index set with respect to J , then $D_I^* = D_J^*$ and $Y_I^* \subseteq Y_J^*$.*

Proof. The first equality follows from Lemma 2 (i). If $(x^*, y^*) \in D_I^* \times Y_I^*$, then (6) holds, which now implies (6) with $I = J$. Hence $y^* \in Y_J^*$. \square

Corollary 1. *Suppose that assumptions (A1)–(A2) are fulfilled. If I is a basic index set, then $D_I^* = D^*$ and $Y_I^* \subseteq Y^*$.*

We can establish similar relations for dual variables in case $F^* \cap F_I \neq \emptyset$.

Proposition 2. *Suppose that assumptions (A1)–(A2) are fulfilled, the set $F^* \cap F_I$ is nonempty for some $I \subseteq L$. Then $F^* \cap F_I = D_I^*$ and $\mathbf{0} \in Y_I^*$.*

Proof. The first equality follows from Lemma 1. Take any $x^* \in F^* \cap F_I$, then $x^* \in D_I^*$ and (6) holds with $y^* = \mathbf{0}$. Therefore, $\mathbf{0} \in Y_I^*$. \square

3 Primal-Dual Method for the Family of Saddle Point Problems

We intend to find saddle points in (5) by a modification of the primal-dual method that was proposed in [7]. First we note that the set of saddle points for the initial problem (1)–(2) is nonempty under the assumptions in (A1)–(A2); see e.g. [8, Corollary 28.2.2]. Therefore, this is the case for each saddle point problem in (5) associated with a basic index set I . Denote by $\pi_U(u)$ the projection of u onto U . Also, for simplicity we will write $Y_{(k)} = Y_{I_k}$, $Y_{(k)}^* = Y_{I_k}^*$, etc. Then the method is described as follows.

Method (PDM). *Step 0:* Choose an index set $I_0 \subseteq L$, a point $w^0 = (x^0, y^0) \in X \times Y_{(0)}$. Set $k = 1$.

Step 1: Choose an index set $I_k \subseteq L$ and a number $\lambda_k > 0$.

Step 2: Take $p^k = \pi_{Y_{(k)}}[y^{k-1} + \lambda_k(Ax^{k-1} - b)]$.

Step 3: Take $x^k = \operatorname{argmin}\{f(x) + \langle p^k, Ax - b \rangle + 0.5\lambda_k^{-1}\|x - x^{k-1}\|^2 \mid x \in X\}$.

Step 4: Take $y^k = \pi_{Y_{(k)}}[y^{k-1} + \lambda_k(Ax^k - b)]$. Set $k = k + 1$ and go to Step 1.

First we observe that

$$p^k = \operatorname{argmin}\{-\mathcal{L}(x^{k-1}, p) + 0.5\lambda_k^{-1}\|p - y^{k-1}\|^2 \mid p \in Y_{(k)}\}$$

and

$$y^k = \operatorname{argmin}\{-\mathcal{L}(x^k, y) + 0.5\lambda_k^{-1}\|y - y^{k-1}\|^2 \mid y \in Y_{(k)}\}.$$

Therefore, each iteration of (PDM) involves two projection (proximal) steps in the dual variable y and one proximal step in the primal variable x . The point $w^k = (x^k, y^k)$ belongs to $X \times Y_{(k)}$. The next two properties follow the usual substantiation schemes for this method; see [7] and also [9].

Lemma 3. *Suppose U is a closed convex set in a finite-dimensional space \mathbb{E} , $\varphi : \mathbb{E} \rightarrow \mathbb{R}$ is a convex function, u is a point in \mathbb{E} . If*

$$\mu(z) = \varphi(z) + 0.5\lambda^{-1}\|z - u\|^2, \quad \lambda > 0,$$

and

$$v = \operatorname{argmin}\{\mu(z) \mid z \in U\},$$

then

$$2\lambda\{\varphi(v) - \varphi(z)\} \leq \|z - u\|^2 - \|z - v\|^2 - \|v - u\|^2 \quad \forall z \in U. \quad (7)$$

Proof. Since the function μ is strongly convex with constant λ^{-1} , we have

$$\mu(z) - \mu(v) \geq 0.5\lambda^{-1}\|z - v\|^2 \quad \forall z \in U.$$

This inequality gives (7). \square

Proposition 3. *Suppose that assumptions (A1)–(A2) are fulfilled. For any pair $w^* = (x^*, y^*) \in D_{(k)}^* \times Y_{(k)}^*$ we have*

$$\begin{aligned} \|w^k - w^*\|^2 &\leq \|w^{k-1} - w^*\|^2 - \|p^k - y^k\|^2 - \|p^k - y^{k-1}\|^2 - \|x^k - x^{k-1}\|^2 \\ &\quad + 2\lambda_k \langle y^k - p^k, A(x^k - x^{k-1}) \rangle \\ &= \|w^{k-1} - w^*\|^2 - \|p^k - y^k\|^2 - \|p^k - y^{k-1}\|^2 - \|x^k - x^{k-1}\|^2 \\ &\quad + 2\lambda_k^2 \|A_{(k)}(x^k - x^{k-1})\|^2. \end{aligned} \quad (8)$$

Proof. Choose any $w^* = (x^*, y^*) \in D_{(k)}^* \times Y_{(k)}^*$. Setting $\varphi(z) = \mathcal{L}(z, p^k)$, $\lambda = \lambda_k$, $U = X$, $u = x^{k-1}$, $v = x^k$, and $z = x^*$ in (7) gives

$$2\lambda_k \{\mathcal{L}(x^k, p^k) - \mathcal{L}(x^*, p^k)\} \leq \|x^* - x^{k-1}\|^2 - \|x^* - x^k\|^2 - \|x^k - x^{k-1}\|^2.$$

Also, using (5) with $I = I_k$, $x = x^k$, and $y = p^k$ gives

$$2\lambda_k \{\mathcal{L}(x^*, p^k) - \mathcal{L}(x^k, y^*)\} \leq 0.$$

Adding these inequalities, we obtain

$$\|x^k - x^*\|^2 \leq \|x^{k-1} - x^*\|^2 - \|x^k - x^{k-1}\|^2 + 2\lambda_k \langle p^k - y^*, Ax^k - b \rangle. \quad (9)$$

On the other hand, setting $\varphi(z) = -\mathcal{L}(x^{k-1}, z)$, $\lambda = \lambda_k$, $U = Y_{(k)}$, $u = y^{k-1}$, $v = p^k$, and $z = y^k$ in (7) gives

$$2\lambda_k \{\mathcal{L}(x^{k-1}, y^k) - \mathcal{L}(x^{k-1}, p^k)\} \leq \|y^k - y^{k-1}\|^2 - \|p^k - y^k\|^2 - \|p^k - y^{k-1}\|^2.$$

Next, setting $\varphi(z) = -\mathcal{L}(x^k, z)$, $\lambda = \lambda_k$, $U = Y_{(k)}$, $u = y^{k-1}$, $v = y^k$, and $z = y^*$ in (7) gives

$$2\lambda_k \{\mathcal{L}(x^{k-1}, y^*) - \mathcal{L}(x^k, y^k)\} \leq \|y^* - y^{k-1}\|^2 - \|y^* - y^k\|^2 - \|y^k - y^{k-1}\|^2.$$

Adding these inequalities, we obtain

$$\begin{aligned} \|y^k - y^*\|^2 &\leq \|y^{k-1} - y^*\|^2 - \|p^k - y^k\|^2 - \|p^k - y^{k-1}\|^2 \\ &\quad - 2\lambda_k \{\langle y^* - y^k, Ax^k - b \rangle + \langle y^k - p^k, Ax^{k-1} - b \rangle\}. \end{aligned} \quad (10)$$

Now adding (9) and (10) gives the first inequality in (8). Since

$$\langle y^k - p^k, A(x^k - x^{k-1}) \rangle = \lambda_k \|A_{(k)}(x^k - x^{k-1})\|^2,$$

we conclude also that the second relation in (8) holds true. \square

Now we can indicate conditions that provide basic convergence properties.

Theorem 1. *Suppose that assumptions (A1)–(A2) are fulfilled,*

$$\bigcap_{k=j}^{\infty} W_{(k)}^* \neq \emptyset \text{ for some } j \geq 1, \quad (11)$$

the sequence $\{\lambda_k\}$ satisfies the condition

$$\lambda_k \in [\tau, \sqrt{(1-\tau)/(\sqrt{2}\|A_{(k)}\|)}] \quad (12)$$

for some $\tau \in (0, 1)$. Then:

- (i) *the sequence $\{w^k\}$ has limit points,*
- (ii) *each of these limit points is a solution of problem (5) for some $I \subseteq L$,*
- (iii) *for any limit point \bar{w} of $\{w^k\}$ such that*

$$\bar{w} \in \bigcap_{k=j}^{\infty} W_{(k)}^* \text{ for some } j \geq 1,$$

it holds that

$$\lim_{k \rightarrow \infty} w^k = \bar{w}. \quad (13)$$

Proof. Take any point

$$w^* \in \bigcap_{k=j}^{\infty} W_{(k)}^*.$$

Then from (8) and (12) we have

$$\|w^k - w^*\|^2 \leq \|w^{k-1} - w^*\|^2 - \|p^k - y^k\|^2 - \|p^k - y^{k-1}\|^2 - \tau \|x^k - x^{k-1}\|^2 \quad (14)$$

for $k = j, j+1, \dots$. Hence, the sequence $\{w^k\}$ is bounded and has limit points, i.e. part (i) is true. Besides, (14) gives

$$\lim_{k \rightarrow \infty} \|w^k - w^*\| = \sigma \geq 0 \quad (15)$$

and

$$\lim_{k \rightarrow \infty} \|p^k - y^k\| = \lim_{k \rightarrow \infty} \|p^k - y^{k-1}\| = \lim_{k \rightarrow \infty} \|x^k - x^{k-1}\| = 0, \quad (16)$$

hence

$$\lim_{k \rightarrow \infty} \|y^k - y^{k-1}\| = 0. \quad (17)$$

Let $\bar{w} = (\bar{x}, \bar{y})$ be an arbitrary limit point of $\{w^k\}$, i.e.

$$\bar{w} = \lim_{s \rightarrow \infty} w^{k_s}.$$

Then there exists $J \subseteq L$ such that $J = I_{k_s}$ for infinitely many times. Without loss of generality we can suppose that $J = I_{k_s}$ for any s . Then $w^{k_s} = (x^{k_s}, y^{k_s}) \in X \times Y_J$ for any s , hence $\bar{w} = (\bar{x}, \bar{y}) \in X \times Y_J$. Setting $\varphi(z) = \mathcal{L}(z, p^k)$, $\lambda = \lambda_k$, $U = X$, $u = x^{k-1}$, $v = x^k$, and $z = x \in X$ in (7) gives

$$2\lambda_k \{\mathcal{L}(x^k, p^k) - \mathcal{L}(x, p^k)\} \leq \|x - x^{k-1}\|^2 - \|x - x^k\|^2 - \|x^k - x^{k-1}\|^2.$$

Taking the limit $k = k_s \rightarrow \infty$ due to (16)–(17) gives

$$\mathcal{L}(\bar{x}, \bar{y}) - \mathcal{L}(x, \bar{y}) \leq 0. \quad (18)$$

Also, setting $\varphi(z) = -\mathcal{L}(x^k, z)$, $\lambda = \lambda_k$, $U = Y_J$, $u = y^{k-1}$, $v = y^k$, and $z = y \in Y_J$ in (7) gives

$$2\lambda_k \{\mathcal{L}(x^k, y) - \mathcal{L}(x^k, p^k)\} \leq \|y^{k-1} - y\|^2 - \|y^k - y\|^2 - \|y^k - y^{k-1}\|^2.$$

Taking the limit $k = k_s \rightarrow \infty$ due to (16)–(17) gives

$$\mathcal{L}(\bar{x}, y) - \mathcal{L}(\bar{x}, \bar{y}) \leq 0. \quad (19)$$

It follows from (18) and (19) that $\bar{w} = (\bar{x}, \bar{y}) \in W_J^* = D_J^* \times Y_J^*$. Hence, part (ii) is also true.

Next, if

$$\bar{w} \in \bigcap_{k=j}^{\infty} W_{(k)}^* \text{ for some } j \geq 1,$$

we can set $w^* = \bar{w}$ in (15). However, now $\sigma = 0$, which gives (13) and part (iii) is true. \square

These properties enable us to establish convergence to a solution under suitable conditions.

Definition 2. We say that $I \subseteq L$ is a support index set with respect to the sequence $\{w^k\}$ if $I = I_k$ for infinitely many k . We say that $I \subseteq L$ is a strongly support index set with respect to the sequence $\{w^k\}$ if it is a support index set and

$$\inf_{I=I_l, k < l} \sup_{I=I_k} (l - k) \leq d < \infty.$$

We denote by \mathcal{P} (respectively, by \mathcal{P}^*) the collection of all support (respectively, strongly support) index sets with respect to the sequence $\{w^k\}$. Also, we set

$$J^s = \bigcap_{I \in \mathcal{P}} I \text{ and } J^* = \bigcap_{I \in \mathcal{P}^*} I,$$

then clearly $J^s \subseteq J^*$ if $\mathcal{P}^* \neq \emptyset$.

Theorem 2. Suppose that assumptions (A1)–(A2) are fulfilled, the sequence $\{\lambda_k\}$ satisfies condition (12) for some $\tau \in (0, 1)$.

- (i) If J^s is a basic index set, then the sequence $\{w^k\}$ has limit points and each of these limit points belongs to W^* .
- (ii) If J^s is a basic index set and $J^s \in \mathcal{P}$ or $J^s = J^*$, then

$$\lim_{k \rightarrow \infty} w^k = w^* \in W^*. \tag{20}$$

Proof. Let $J = J^s$ be a basic index set. By assumption, the sets W_J^* and W^* are now nonempty. Due to Proposition 1, $W_J^* \subseteq W_{(k)}^*$ for k large enough, hence condition (11) holds. Then the sequence $\{w^k\}$ has limit points due to Theorem 1 (i). Also, there exists $I \subseteq L$ such that $J \subseteq I = I_{k_s}$ for infinitely many times. But now I is a nonempty basic index set, hence $W_I^* \subseteq W^*$. Following the lines of part (ii) of Theorem 1, we obtain that any limit point of $\{w^{k_s}\}$ will belong to $W_I^* \subseteq W^*$. Therefore, part (i) is true.

In case (ii) we first take the case where $J \in \mathcal{P}$. It follows that $J = I_{k_s}$ for infinitely many times. Then we have similarly that any limit point w^* of $\{w^{k_s}\}$ will belong to W_J^* , but

$$w^* \in \bigcap_{k=j}^{\infty} W_{(k)}^* \text{ for some } j \geq 1, \tag{21}$$

and (20) follows from Theorem 1 (iii).

Now we take the case where $J = J^s = J^*$. Let $w^* = (x^*, y^*)$ be a limit point of $\{w^k\}$, i.e.

$$w^* = \lim_{s \rightarrow \infty} w^{k_s}$$

and let $I \in \mathcal{P}^*$. By definition, for each k_s there exists a number l_s such that $I = I_{l_s}$ and $k_s \leq l_s \leq k_s + d$. Due to (16)–(17) we have

$$w^* = \lim_{s \rightarrow \infty} w^{l_s},$$

but $y_i^{l_s} = \mathbf{0}$ for any $i \notin I$, hence $y_i^* = \mathbf{0}$ for any $i \notin I$. It follows that $w^* = (x^*, y^*) \in D_J^* \times Y_J^*$ and (21) holds. Then (20) also follows from Theorem 1 (iii). \square

Theorem 3. *Suppose that assumptions (A1)–(A2) are fulfilled, the sequence $\{\lambda_k\}$ satisfies condition (12) for some $\tau \in (0, 1)$.*

- (i) *If $F^* \cap F_L \neq \emptyset$, then the sequence $\{w^k\}$ has limit points.*
- (ii) *If $F^* \cap F_L \neq \emptyset$ and each $I \in \mathcal{P}$ is a basic index set, then all the limit points of $\{w^k\}$ belong to W^* .*
- (iii) *If $F^* \cap F_L \neq \emptyset$, J^s is a basic index set and $J^s \in \mathcal{P}$ or $J^s = J^*$, then the sequence $\{w^k\}$ converges to a point of W^* .*

Proof. Due to Proposition 2, we now have $F^* \cap F_I = D_I^*$, $D_I^* \neq \emptyset$, and $\mathbf{0} \in Y_I^*$ for any $I \subseteq L$. It follows that

$$\left\{ F^* \cap F_L \right\} \times \{\mathbf{0}\} \subseteq \bigcap_{k=1}^{\infty} W_{(k)}^*.$$

Therefore, (11) holds and assertion (i) follows from Theorem 1 (i). Following the lines of part (ii) of Theorem 1, we obtain that any limit point of $\{w^{k_s}\}$ will belong to $W_I^* \subseteq W^*$ where I is a nonempty basic index set. Therefore, assertion (ii) is also true. Assertion (iii) clearly follows from Theorem 2. \square

The conditions of part (ii) of Theorem 2 are satisfied if for instance we take the rule $I_k \subseteq I_{k+1}$ or $I_{k+1} \subseteq I_k$ for index sets. These rules can be also applied in part (iii) of Theorem 3.

4 Primal-Dual Method for Multi-agent Optimization Problems

We now describe a specialization of the proposed approach to the multi-agent optimization problem

$$\min \rightarrow \left\{ \sum_{i=1}^m f_i(v) \mid \bigcap_{i=1}^m X_i \right\}, \quad (22)$$

where m is the number of agents (units) in the system. That is, the information about the function f_i and set X_i is known only to the i -th agent and may be unknown even to its neighbours. Besides, it is usually supposed that the agents are joined by some transmission links for information exchange so that the system is usually a connected network, whose topology may vary from time to time. This decentralized system has to find a concordant solution defined by (22).

For this reason, we replace (22) with the family of optimization problems of the form

$$\min_{x \in D_I} \rightarrow f(x) = \sum_{i=1}^m f_i(x_i), \quad (23)$$

where $x = (x_i)_{i=1,\dots,m} \in \mathbb{R}^{mn}$, i.e. $x^\top = (x_1^\top, \dots, x_m^\top)$, $x_i = (x_{i1}, \dots, x_{in})^\top$ for $i = 1, \dots, m$,

$$D_I = X \bigcap F_I, \quad X = X_1 \times \dots \times X_m = \prod_{i=1}^m X_i, \quad X_i \subseteq \mathbb{R}^n, \quad i = 1, \dots, m; \quad (24)$$

the set F_I describes the information exchange scheme within the current topology of the communication network, and I is the index set of arcs of the corresponding oriented graph. More precisely, the maximal (full) communication network with non-oriented edges denoted by \mathcal{F} corresponds to the set

$$\tilde{F} = \{x \in \mathbb{R}^{mn} \mid x_s = x_t, \quad s, t = 1, \dots, m, \quad s \neq t\},$$

i.e. each edge is associated with two directions or equations ($x_s = x_t$ and $x_t = x_s$). However, this definition of topology is superfluous. It seems more suitable to introduce some other graph topology for writing the multi-agent optimization problem in addition to the graph \mathcal{F} . For this reason, we associate each pair of vertices (agents) (s, t) to one oriented arc i , so that $L = \{1, \dots, l\}$ is the index set of all these arcs, hence $l = m(m-1)/2$. That is, each arc (s, t) is in fact used in both the directions in the communication network \mathcal{F} , but we fix only one direction for definition of the multi-agent optimization problem and obtain the graph \mathcal{G} . Taking subsets $I \subseteq L$, we obtain various constraint sets

$$F_I = \{x \in \mathbb{R}^{mn} \mid x_s - x_t = \mathbf{0}, \quad i = (s, t) \in I\}, \quad (25)$$

corresponding to the oriented graphs \mathcal{G}_I in the multi-agent optimization problem formulation. Replacing the arcs in \mathcal{G}_I with non-oriented edges, we obtain the corresponding communication network \mathcal{F}_I of the system. It follows that $\mathcal{F} = \mathcal{F}_L$, $\mathcal{G} = \mathcal{G}_L$, and $F = F_L$. Next, for each arc $i = (s, t)$ we can define the $n \times mn$ sub-matrix

$$A_i = (A_{i1} \dots A_{im}),$$

where

$$A_{ij} = \begin{cases} E, & \text{if } j = s, \\ -E, & \text{if } j = t, \\ \Theta, & \text{otherwise,} \end{cases}$$

E is the $n \times n$ unit matrix, Θ is the $n \times n$ zero matrix. Then clearly

$$F_I = \{x \in \mathbb{R}^{mn} \mid A_I x = \mathbf{0}\},$$

where

$$A_I = (\{A_i^\top\}_{i \in I})^\top,$$

which corresponds to the definition in (3) for $b_I = \mathbf{0}$ and any $I \subseteq L$, hence we can set $A = A_L$. Therefore, our problem (23)–(25) corresponds to (3)–(4).

In what follows, we will use the following basic assumptions.

- (B1) For each $i = 1, \dots, m$, X_i is a convex and closed set in \mathbb{R}^n , $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function.
 (B2) The set $D^* = D_L^*$ is nonempty.

These assumptions imply (A1)–(A2). If the graph \mathcal{F}_I for some $I \subseteq L$ is connected, then I a basic index set. Now we present an implementation of Method (PDM) for the multi-agent optimization problem (23)–(25), where each agent (or unit) receives information only from its neighbours. Given an oriented graph \mathcal{G}_I and an agent j , we denote by $\mathcal{N}_I^+(j)$ and $\mathcal{N}_I^-(j)$ the sets of incoming and outgoing arcs at j . Since many oriented graphs \mathcal{G}_I are associated with the same graph \mathcal{F}_I , we suppose that agent j is responsible for calculation of the current values of the primal variable x_j and all the dual variables y_i and p_i such that $i \in \mathcal{N}_I^-(j)$. That is, we will fix the oriented graph \mathcal{G} and its subgraphs \mathcal{G}_I such that agent j is associated with all the outgoing arcs for vertex j . The general Lagrange function for problems (23)–(25) is written as follows:

$$\begin{aligned} \mathcal{L}(x, y) &= f(x) + \langle y, Ax \rangle = \sum_{j \in M} f_j(x_j) + \sum_{i \in L} \langle y_i, A_i x \rangle \\ &= \sum_{j \in M} \left\{ f_j(x_j) + \sum_{i \in \mathcal{N}_L^-(j)} \langle y_i, x_j \rangle - \sum_{i \in \mathcal{N}_L^+(j)} \langle y_i, x_j \rangle \right\}. \end{aligned} \quad (26)$$

The saddle point problems are defined in (5). As in Sect. 3, for simplicity we will write $Y_{(k)} = Y_{I_k}$, $Y_{(k)}^* = Y_{I_k}^*$, etc.

Method (PDMI). At the beginning, the agents choose the communication topology by choosing the active arc index set $I_0 \subseteq L$. Next, each s -th agent chooses x_s^0 and y_i^0 for $i \in \mathcal{N}_{(0)}^-(s)$ and reports these values to its neighbours. This means that $y_i^0 = \mathbf{0}$ for $i \notin I_0$.

At the k -th iteration, $k = 1, 2, \dots$, each s -th agent has the values x_s^{k-1} and y_i^{k-1} , $i \in \mathcal{N}_{(k-1)}^-(s)$, and the same values of its neighbours. The agents choose the current communication topology by choosing the active arc index set $I_k \subseteq L$ and determine the stepsize λ_k . This means that they set $y_i^k = \mathbf{0}$ for $i \notin I_k$.

Step 1: Each s -th agent sets

$$p_i^k = y_i^{k-1} + \lambda_k(x_s^{k-1} - x_t^{k-1}) \quad \forall i = (s, t), \quad i \in \mathcal{N}_{(k)}^-(s). \quad (27)$$

Then each s -th agent reports these values to its neighbours.

Step 2: Each s -th agent calculates

$$v_s^k = \sum_{i \in \mathcal{N}_{(k)}^-(s)} p_i^k - \sum_{i \in \mathcal{N}_{(k)}^+(s)} p_i^k$$

and

$$x_s^k = \arg \min_{x_s \in X_s} \{ f_s(x_s) + \langle v_s^k, x_s \rangle + 0.5\lambda_k^{-1} \|x_s - x_s^{k-1}\|^2 \} \quad (28)$$

and reports this value to its neighbours.

Step 3: Each s -th agent sets

$$y_i^k = y_i^{k-1} + \lambda_k(x_s^k - x_t^k) \quad \forall i = (s, t), i \in \mathcal{N}_{(k)}^-(s). \quad (29)$$

Then each s -th agent reports these values to its neighbours. The k -th iteration is complete.

We observe that the agents do not store the dual variables related to the inactive arcs, i.e. $y_i^k = \mathbf{0}$ for $i \notin I_k$. If some arc $i = (s, t) \notin I_{k-1}$ becomes active at the k -th iteration, i.e. $i \in I_k$, then agent s simply sets $y_i^{k-1} = \mathbf{0}$.

Due to (26), relations (27)–(29) correspond to Steps 2–4 of (PDM), respectively. Hence, the convergence properties of (PDMI) will follow directly from Theorems 2 and 3.

Corollary 2. *Suppose that assumptions (B1)–(B2) are fulfilled, the sequence $\{\lambda_k\}$ satisfies condition (12) for some $\tau \in (0, 1)$.*

- (i) *If J^s is a basic index set, then the sequence $\{w^k\}$, $w^k = (x^k, y^k)$, generated by (PDMI) has limit points and each of these limit points belongs to W^* .*
- (ii) *If J^s is a basic index set and $J^s \in \mathcal{P}$ or $J^s = J^*$, then (20) holds.*

Corollary 3. *Suppose that assumptions (B1)–(B2) are fulfilled, the sequence $\{\lambda_k\}$ satisfies condition (12) for some $\tau \in (0, 1)$.*

- (i) *If $F^* \cap F_L \neq \emptyset$, then the sequence $\{w^k\}$, $w^k = (x^k, y^k)$, generated by (PDMI) has limit points.*
- (ii) *If $F^* \cap F_L \neq \emptyset$ and each $I \in \mathcal{P}$ is a basic index set, then all the limit points of $\{w^k\}$ belong to W^* .*
- (iii) *If $F^* \cap F_L \neq \emptyset$, J^s is a basic index set and $J^s \in \mathcal{P}$ or $J^s = J^*$, then the sequence $\{w^k\}$ converges to a point of W^* .*

Convergence of (PDMI) requires for all the agents to choose the stepsize λ_k in accordance with (12), hence they have to evaluate the norm $\|A_{(k)}\|$ at the k -th iteration. Fix some $I \subseteq L$, then

$$A_I^\top A_I = H_I \otimes E,$$

where H_I is the Kirchhoff matrix of the graph \mathcal{F}_I , \otimes denotes the Kronecker product of matrices. Application of the Gershgorin theorem (see Theorem 5 in [10, Chapter XIV]) gives

$$\|A_I\| = \sqrt{\|H_I\|} \leq \sqrt{2d(\mathcal{F}_I)},$$

where $d(\mathcal{F}_I)$ is the maximal vertex degree of the graph \mathcal{F}_I . There exist more precise estimates for some special classes of graphs; see e.g. [11, 12]. Together with (12) we obtain the bound

$$\lambda_k \in \left[\tau, 0.5 \sqrt{(1 - \tau)/d(\mathcal{F}_{(k)})} \right] \quad (30)$$

for some $\tau \in (0, 1)$. In case of varying topology the separate agents may meet difficulties in evaluation of $d(\mathcal{F}_{(k)})$ since the graph then may be non-regular. The concordant value of $\lambda = \lambda_k$ satisfying (30) can be obtained by determining some upper bound for $d(\mathcal{F}_{(k)})$. It seems suitable to apply the following strategy. First we choose the fixed topology that corresponds to an arc index set $J \subset L$ so that it gives the connected graph \mathcal{F}_J and $J \subseteq I_k$ for any k . This means that all the arcs in J remain always active. The status of the other arcs may vary, but the maximal vertex degree of the graph $\mathcal{F}_{(k)}$ can not exceed some fixed number v . Then each agent can take $\lambda = 0.5\sqrt{(1-\tau)/v}$ and the assumptions of Corollary 2 (i) and Corollary 3 (i)–(ii) on the choice of parameters hold.

We now give a natural example of problem (23)–(25) such that $F^* \cap F_L \neq \emptyset$. Namely, set $X_i = \mathbb{R}^n$, $f_i(v) = (1/p)(\max\{h_i(v), 0\})^p$, $p \geq 1$ for $i = 1, \dots, m$. Then (23)–(25) corresponds to a penalized problem for finding a point of the set

$$\tilde{V} = \{u \in \mathbb{R}^n \mid h_i(u) \leq 0, i = 1, \dots, m\}.$$

If $\tilde{V} \neq \emptyset$, then clearly $F^* \cap F_L \neq \emptyset$, which gives stronger convergence properties.

It should be noticed that primal-dual methods are usually applied to large-scale convex optimization problems with binding constraints in order to keep the decomposability properties. However, the streamlined primal-dual gradient projection method requires strengthened assumptions. Utilization of extrapolation steps enables one to attain convergence under custom convex-concavity; see [13]. These methods admit a fixed positive stepsize that yields a linear rate of convergence; see e.g. [14, Chapter VI] and the references therein. However, replacing projections with proximal steps also will enhance convergence, besides the method becomes applicable to non-smooth problems. This primal-dual method with proximal steps was proposed in [7]. Similar methods were described in [9, 15]. It should be also noticed that known iterative methods for multi-agent optimization problems with changing communication topology are based on different conditions; see e.g. [16, 17].

5 Computational Experiments

In order to check the performance of the proposed method we carried out preliminary series of computational experiments. We evaluated the total number of iterations of (PDMI) for obtaining some desired accuracy. We implemented the method in Delphi with double precision arithmetic.

We took the well-known Fermat-Weber problem:

$$\min_{v \in \mathbb{R}^n} \rightarrow \tilde{\varphi}(v) = \sum_{i=1}^m \|v - \tilde{a}_i\|,$$

where \tilde{a}_i , $i = 1, \dots, m$ are some given points (anchors). Clearly, this is a particular case of problem (22) with the coercive, convex, and non-smooth cost function over the whole space \mathbb{R}^n . It is rewritten in the format (23) as follows:

$$\min_{x \in F_I} \rightarrow \sum_{i=1}^m \|x_i - \tilde{a}_i\|,$$

where the set F_I describes the current topology of the communication network in accordance with (25), i.e. $X_i = \mathbb{R}^n$, $f_i(x_i) = \|x_i - \tilde{a}_i\|$ for $i = 1, \dots, m$. Then its solution set D_I^* is nonempty and bounded. In the multi-agent setting, the i -th unit of the network knows only the vector \tilde{a}_i . We created the communication network by using the following three simple cycles:

$$\begin{aligned} C_1 &= \{(1, 2), (2, 3), \dots, (m-1, m), (m, 1)\}, \\ C_2 &= \{(1, 3), (3, 5), \dots, (m-3, m-1), (m-1, 1)\}, \\ C_3 &= \{(2, 4), (4, 6), \dots, (m-2, m), (m, 2)\}, \end{aligned}$$

m was chosen to be even. More precisely, the fixed topology of the communication network was defined by C_1 , whereas the changing topology was given by the iteration cycle scheme

$$C_1 \implies C_1 \cup C_2 \implies C_1 \cup C_3 \implies C_1 \cup C_2 \cup C_3 \implies C_1.$$

We calculated the total number of the iterations for attaining the accuracy δ with respect to the distance between two points: $\Delta_k = \|w^k - w^{k-1}\|$. Besides, given a primal point $x = (x_i)_{i=1, \dots, m} \in \mathbb{R}^{mn}$, we can calculate the average point $z = (1/m) \sum_{i=1}^m x_i$ and the value of the cost function $\tilde{\varphi}(z)$. This average point z^k was calculated at x^k in a separate block and this value was not used in the method itself. For brevity, we write $\tilde{\varphi}_k = \tilde{\varphi}(z^k)$. The elements of the vectors \tilde{a}_i were defined by

$$\tilde{a}_{ij} = 5 \sin(i/j) \cos(ij), \quad j = 1, \dots, n, \quad i = 1, \dots, m.$$

We took the same starting point $x^0 = (5, \dots, 5)^\top$ and the fixed stepsize $\lambda = 0.25$ for all the cases. Table 1 describes the results of application of (PDMI), where (kt) denotes the number of its iterations for attaining the accuracy $\delta = 0.001$. Besides, it gives the value of the function $\tilde{\varphi}(z^k)$ at the average point for the same number of iterations (k) close to (kt). We can conclude that (PDMI) demonstrated more rapid convergence with respect to Δ_k in the case of changing topology, but gave somewhat greater cost function values after the same number of iterations. However, this difference appeared not so essential, and in general the convergence of (PDMI) was rather stable.

Table 1. Computations by (PDMI)

		Fixed topology			Changing topology		
m	n	kt	$\tilde{\varphi}_k$	k	kt	$\tilde{\varphi}_k$	k
20	10	509	152.3378	390	397	152.3383	390
50	10	557	382.2441	480	485	382.2443	480
100	10	568	759.3882	480	496	759.3883	480
100	20	700	1094.8977	580	585	1094.8979	580
100	50	1090	1760.8916	960	963	1760.8918	960

Acknowledgement. This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program (“PRIORITY-2030”).

References

1. Eremin, I.I., Mazurov, V.D.: Non-stationary Processes of Mathematical Programming. Nauka, Moscow (1979). (in Russian)
2. Polyak, B.T.: Introduction to Optimization. Nauka, Moscow (1983). (Engl. transl. in Optimization Software, New York (1987))
3. Vasil'yev, F.P.: Methods for Solving Extremal Problems. Nauka, Moscow (1981). (in Russian)
4. Khan, M., Pandurangan, G., Anil Kumar, V.S.: Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **20**(1), 124–139 (2009). <https://doi.org/10.1109/TPDS.2008.57>
5. Lobel, I., Ozdaglar, A., Feijer, D.: Distributed multi-agent optimization with state-dependent communication. *Math. Program.* **129**, 255–284 (2011)
6. Peng, Z., Yan, M., Yin, W.: Parallel and distributed sparse optimization. In: The 47th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, pp. 646–659. IEEE (2013)
7. Antipin, A.S.: On non-gradient methods for optimization of saddle functions. In: Karmanov, V.G. (ed.) *Problems of Cybernetics. Methods and Algorithms for the Analysis of Large Systems*, pp. 4–13. Nauchn. Sovet po Probleme “Kibernetika”, Moscow (1988). (in Russian)
8. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton (1970)
9. Chen, G., Teboulle, M.: A proximal-based decomposition method for convex minimization problems. *Math. Program.* **64**, 81–101 (1994)
10. Gantmacher, F.R.: *The Theory of Matrices*. Nauka, Moscow (1966). [In Russian]
11. Li, J.-S., Zhang, X.-D.: A new upper bound for eigenvalues of the Laplacian matrix of a graph. *Linear Algebra Appl.* **265**, 93–100 (1997)
12. Pan, Y.-L.: Sharp upper bounds for the Laplacian graph eigenvalues. *Linear Algebra Appl.* **355**, 287–295 (2002)
13. Arrow, K.J., Solow, R.M.: Gradient methods for constrained maxima, with weakened assumptions. In: Arrow, K.J., Hurwicz, L., Uzawa, H. (eds.) *Studies in Linear and Nonlinear Programming*, pp. 166–176. Stanford University Press, Stanford (1958)

14. Gol'shtein, E.G., Tret'yakov, N.V.: Modified Lagrange Functions. Nauka, Moscow (1989). (Engl. transl. in John Wiley and Sons, New York (1996))
15. Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.* **3**, 1015–1046 (2010)
16. Nedić, A., Olshevsky, A.: Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control* **60**, 601–615 (2015)
17. Aybat, N.S., Hamedani, E.Y.: A primal-dual method for conic constrained distributed optimization problems. In: *Advances in Neural Information Processing Systems*, pp. 5049–5057. Neural Information Processing Systems Foundation, Barcelona (2016)



Decentralized Convex Optimization Under Affine Constraints for Power Systems Control

Demyan Yarmoshik¹ , Alexander Rogozin¹ , Oleg. O. Khamisov², Pavel Dvurechensky³ , and Alexander Gasnikov^{1,4,5} 

¹ Moscow Institute of Physics and Technology, Institutskii ave., 9, Dolgoprudny, Russia

{yarmoshik.dv,aleksandr.rogozin,gasnikov.av}@phystech.edu

² Skolkovo Institute of Science and Technology, Moscow, Russia

Oleg.Khamisov@skolkovotech.ru

³ Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany

pavel.dvurechensky@wias-berlin.de

⁴ Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, Russia

⁵ Caucasus Mathematical Center, Adyge State University, Maikop, Russia

Abstract. Modern power systems are now in continuous process of massive changes. Increased penetration of distributed generation, usage of energy storage and controllable demand require introduction of a new control paradigm that does not rely on massive information exchange required by centralized approaches. Distributed algorithms can rely only on limited information from neighbours to obtain an optimal solution for various optimization problems, such as optimal power flow, unit commitment etc.

As a generalization of these problems we consider the problem of decentralized minimization of the smooth and convex partially separable function $f = \sum_{k=1}^l f^k(x^k, \tilde{x})$ under the coupled $\sum_{k=1}^l (A^k x^k - b^k) \leq 0$ and the shared $\tilde{A}\tilde{x} - \tilde{b} \leq 0$ affine constraints, where the information about A^k and b^k is only available for the k -th node of the computational network.

One way to handle the coupled constraints in a distributed manner is to rewrite them in a distributed-friendly form using the Laplace matrix of the communication graph and auxiliary variables (Khamisov, CDC, 2017). Instead of using this method we reformulate the constrained optimization problem as a saddle point problem (SPP) and utilize the consensus constraint technique to make it distributed-friendly. Then we provide a complexity analysis for state-of-the-art SPP solving algorithms applied to this SPP.

Keywords: Constrained convex optimization · Distributed optimization · Energy system · Distributed control · Saddle point problem

The work of D. Yarmoshik was supported by the program “Leading Scientific Schools” (grant no. NSh-775.2022.1.1). The work of A. Rogozin and A. Gasnikov was supported by Russian Science Foundation (project No. 21-71- 30005).

1 Introduction

Optimal operation of power systems relies heavily on the ability of system operator to solve efficiently a number of optimization problems such as optimal power flow, unit commitment, as well as a number of online problems such as frequency and voltage control. Traditionally such problems were solved by System Operators in a centralized way. However, recent developments in implementation of distributed energy sources, storage systems and possibility of demand response can be effectively controlled by distributed algorithms. Such approach has a number of potential benefits, namely reduction of necessary communications between agents, increased robustness with respect to malfunction of any agent and possibility to increase cybersecurity and privacy of each agent.

The detailed surveys on the application of distributed algorithms in power systems is given in [10, 14]. These applications often lead to the necessity of solving an optimization problem, which can be formulated as distributed optimization problem with coupled constraints. Distributed approaches for optimization problems with coupled constraints can be separated into two main groups: (i) primal, dual or primal-dual consensus algorithms [2, 8, 9, 11–13, 19]; (ii) ADMM-based algorithms [1, 3, 16, 18].

In this paper we propose a novel optimization approach for convex optimization problems with coupled linear equality and inequality constraints. Here introduction of specially placed Laplace matrices is used to model communications between neighboring agents in a computational network described as a connected graph. In the core of our approach lies: 1) the reduction of the decentralized optimization problem with constraints to decentralized saddle point problem; 2) applying decentralized Mirror Prox algorithm from [15] to solve the obtained saddle point problem. We obtain the same rate of convergence $\sim 1/N$ (N – number of communication steps/oracle calls) as the best known competitors, like ADMM [7]. The main benefit of our approach is that the local optimization problem at each node is much simpler than in the ADMM-based approaches since we use only gradient oracle instead of complicated proximal mapping which may require a matrix inversion. Compared to the dual algorithms of [11–13], we consider a more general setting in which the objective may be non-separable and there are local linear constraints at each node of the computational network.

2 Problem Statement

Let us consider the following optimization problem:

$$\min_{x \in \mathbb{R}^n} f(\mathbf{x}), \quad (1a)$$

$$A'\mathbf{x} - b' = 0, \quad A' \in \mathbb{R}^{m \times n}, \quad b' \in \mathbb{R}^m, \quad (1b)$$

$$C'\mathbf{x} - d' \leq 0, \quad C' \in \mathbb{R}^{h \times n}, \quad d' \in \mathbb{R}^h, \quad (1c)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable strictly convex function. It is assumed that constraints (1b) and (1c) are consistent and there exists a unique solution

x^* . Thus, Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient optimality conditions.

Let us now consider the case, when problem (1) must be solved by a multi-agent network with l agents connected by a graph defined by a Laplacian matrix W . For this case, we assume, that each agent seeks to find its own subvector $x^k \in \mathbb{R}^{n_k}$, $k \in \{1, \dots, l\}$ ($\sum_{k=1}^l n_k = n$) and the shared vector $\tilde{x} \in \mathbb{R}^{\tilde{n}}$. We denote vector of private variables by $\mathbf{x} = (x^{1\top}, \dots, x^{l\top})^\top$. Additionally, function f is partially separable:

$$f(\mathbf{x}, \tilde{x}) = \sum_{k=1}^l f^k(x^k, \tilde{x})$$

and each f^k is known only to agent k . Each agent has partial information $A^k \in \mathbb{R}^{m \times n_k}$, $b^k \in \mathbb{R}^m$, $C^k \in \mathbb{R}^{h \times n_k}$ and $d^k \in \mathbb{R}^h$ about constraints' parts corresponding only to variables x^k : $A := [A^1, \dots, A^l]$, $b := \sum_{k=1}^l b^k$, $C := [C^1, \dots, C^l]$ and $n := \sum_{k=1}^l n_k$. Additionally we assume that there are shared constraints with matrices $\tilde{A} \in \mathbb{R}^{\tilde{m} \times \tilde{n}}$, $\tilde{C} \in \mathbb{R}^{\tilde{p} \times \tilde{n}}$ and vectors $\tilde{b} \in \mathbb{R}^{\tilde{m}}$, $\tilde{d} \in \mathbb{R}^{\tilde{p}}$ which are known to all agents.

As a result, each agent k has only its own part of the objective function $f^k(x^k, \tilde{x})$ and parts of the coupled equality and inequality constraints respectively: $A^k x^k - b^k$ and $C^k x^k - d^k$.

Therefore, we have an optimization problem of the following form:

$$\min_{x \in \mathbb{R}^{n+\tilde{n}}} \sum_{k=1}^l f^k(x^k, \tilde{x}), \quad (2a)$$

$$\text{s.t.} \sum_{k=1}^l (A^k x^k - b^k) = 0, \quad (2b)$$

$$\sum_{k=1}^l (C^k x^k - d^k) \leq 0, \quad (2c)$$

$$\tilde{A}\tilde{x} - \tilde{b} = 0, \quad (2d)$$

$$\tilde{C}\tilde{x} - \tilde{d} \leq 0. \quad (2e)$$

Here $\tilde{x} \in \mathbb{R}^{\tilde{n}}$ is a subvector of x that contains global variables used by all agents.

3 Mathematical Setting

Assumption 1. For every $k = 1, \dots, l$

1. $f^k(x^k, \tilde{x})$ is differentiable.
2. (Convexity) $\forall x^k, x'^k \in \mathcal{X}^k, \forall \tilde{x}, \tilde{x}' \in \tilde{\mathcal{X}}$

$$f^k(x'^k, \tilde{x}') \geq f^k(x^k, \tilde{x}) + \left\langle \nabla f^k(x^k, \tilde{x}), \begin{pmatrix} x'^k - x^k \\ \tilde{x}' - \tilde{x} \end{pmatrix} \right\rangle.$$

3. (Lipschitz smoothness)

$$\|\nabla f^k(x'^k, \tilde{x}') - \nabla f^k(x^k, \tilde{x})\| \leq L_k \left\| \begin{pmatrix} x'^k - x^k \\ \tilde{x}' - \tilde{x} \end{pmatrix} \right\|.$$

Assumption 2. Variable x is subject to block constraints: $x^k \in \prod_{i=1}^{n_k} [\xi^{k,i}, \eta^{k,i}] = \mathcal{X}^k$, $\xi^{k,i}, \eta^{k,i} \in \mathbb{R}$ and $\tilde{x} \in \prod_{i=1}^{\tilde{n}} [\tilde{\xi}^i, \tilde{\eta}^i] = \tilde{\mathcal{X}}$, $\tilde{\xi}^i, \tilde{\eta}^i \in \mathbb{R}$.

This is a natural assumption since in a real-world system maximal and minimal values of every control and auxiliary variable are limited. Let us also denote

- $\lambda_{max}(A), \lambda_{min}^+(A)$ — the largest and the smallest positive eigenvalues of a matrix A .
- $\sigma_{max}(A) = \sqrt{\lambda_{max}(A^\top A)}$ and $\sigma_{min}^+(A) = \sqrt{\lambda_{min}^+(A^\top A)}$ — the largest and the smallest positive singular values of a matrix A .
- $\chi(A) = \frac{\sigma_{max}(A)}{\sigma_{min}^+(A)}$ — condition number of a matrix A on $(\text{Ker}A)^\top$.
- $\text{Proj}_S(x)$ — projection of x onto a set S .

The key instrument in separating shared variables and coupled constraints is introducing the consensus constraint with the help of matrix W defined as follows:

1. W is symmetric positive semi-definite matrix.
2. (Network compatibility) For all $i, j = 1, \dots, l$ the entry of W : $[W]_{ij} = 0$ if $i \neq j$ and there is no edge in the communication graph between nodes i and j . This property allows to perform multiplications by W in a distributed manner (only using information from neighbours in the communication graph).
3. (Kernel property) For any $v = [v_1, \dots, v_m]^\top \in \mathbb{R}^m$, $Wv = 0$ if and only if $v_1 = \dots = v_m$, i.e. $\text{Ker}W = \text{span}\{\mathbf{1}\}$. This property allows to rewrite pairwise equality constraint in a distributed way.

An example of matrix satisfying this assumption is the graph Laplacian $W \in \mathbb{R}^{m \times m}$:

$$[W]_{ij} = \begin{cases} -1, & \text{if } (i, j) \in E, \\ \text{deg}(i), & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$$

where $\text{deg}(i)$ is the degree of the node i , i.e., the number of neighbors of the node.

Matrix W can be used to rewrite pairwise equality of scalars. To rewrite pairwise equality of vector variables with equal dimension we will use the following extension of matrix W , called *communication matrix*:

$$\mathbf{W} = W \otimes I_d, \tag{3}$$

where \otimes denotes the Kronecker product and d is the dimension of the vector variables.

4 Distributed Saddle Point Problem Formulation

4.1 Saddle Point Problem and Consensus Constraints

We reformulate problem (1) as saddle point problem:

$$\min_{\mathbf{x}, \tilde{\mathbf{x}}} \max_{\substack{\lambda, \tilde{\lambda} \\ \mu, \tilde{\mu} \geq 0}} \sum_{k=1}^l [f^k(x^k, \tilde{x}) + \lambda^\top (A^k x^k - b^k) + \mu^\top (C^k x^k - d^k)] + \tilde{\lambda}^\top (\tilde{A}\tilde{x} - \tilde{b}) + \tilde{\mu}^\top (\tilde{C}\tilde{x} - \tilde{d}). \quad (4)$$

Let us unify the analysis of equality and inequality constraints by stacking Lagrange multipliers λ and μ in a single dual variable

$$y = \begin{pmatrix} \lambda \\ \mu \end{pmatrix}, \quad y \in \mathcal{Y} = \mathbb{R}^m \times \mathbb{R}_+^h.$$

And similarly we introduce the joined dual variable for the coupled constraints:

$$\tilde{y} = \begin{pmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{pmatrix}, \quad \tilde{y} \in \tilde{\mathcal{Y}} = \mathbb{R}^{\tilde{m}} \times \mathbb{R}_+^{\tilde{p}}.$$

To solve this saddle point problem in a distributed manner we have to separate dual variables y by making their copies at each node and introducing consensus constraint into the saddle point problem, as described in [15]. That brings us to the following formulation:

$$\min_{\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{z}} \max_{\mathbf{y}, \tilde{\mathbf{y}}} \sum_{k=1}^l \left[f^k(x^k, \tilde{x}) + y^{k\top} \begin{pmatrix} A^k x^k - b^k \\ C^k x^k - d^k \end{pmatrix} \right] + \mathbf{z}^\top \mathbf{W} \mathbf{y} + \tilde{\mathbf{y}}^\top \begin{pmatrix} \tilde{A}\tilde{x} - \tilde{b} \\ \tilde{C}\tilde{x} - \tilde{d} \end{pmatrix} \quad (5)$$

To separate the terms corresponding to the shared constraints (2d), (2e) we should go back to the optimization problem (2) and do the same trick with them: make a copy of \tilde{x} at each node and introduce consensus constraint. So we transform (2d), (2e) into equivalent system

$$\tilde{A}\tilde{x}^k + \tilde{b} = 0, \quad k \in \{1, \dots, l\}, \quad (6a)$$

$$\tilde{C}\tilde{x}^k + \tilde{d} \leq 0, \quad k \in \{1, \dots, l\}, \quad (6b)$$

$$\tilde{\mathbf{W}}\tilde{\mathbf{x}} = 0, \quad (6c)$$

where $\tilde{\mathbf{x}} = (\tilde{x}^{1\top}, \dots, \tilde{x}^{l\top})^\top$, $\tilde{\mathbf{W}} = W \otimes I_{\tilde{n}}$.

Note, that each node can handle constraints (6a) and (6b) independently, so we don't have to introduce additional consensus constraints over corresponding dual variables in the final saddle point problem:

$$\begin{aligned}
 \min_{\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{z}} \max_{\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}} \sum_{k=1}^l & \left[f^k(x^k, \tilde{x}^k) + y^{k\top} \begin{pmatrix} A^k x^k - b^k \\ C^k x^k - d^k \end{pmatrix} + \tilde{y}^{k\top} \begin{pmatrix} \tilde{A} \tilde{x}^k - \tilde{b} \\ \tilde{C} \tilde{x}^k - \tilde{d} \end{pmatrix} \right] + \mathbf{z}^\top \mathbf{W} \mathbf{y} + \tilde{\mathbf{z}}^\top \tilde{\mathbf{W}} \tilde{\mathbf{x}} \\
 & = \min_{\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{z}} \max_{\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}} \sum_{k=1}^l g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k) + \mathbf{z}^\top \mathbf{W} \mathbf{y} + \tilde{\mathbf{z}}^\top \tilde{\mathbf{W}} \tilde{\mathbf{x}}, \quad (7)
 \end{aligned}$$

where $\mathbf{y} = (y^{1\top}, \dots, y^{l\top})^\top$, $\mathbf{W} = W \otimes I_{m+h}$.

We will also use the following notation:

$$G(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}}) = \sum_{k=1}^l g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k), \quad (8)$$

and

$$G_w(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}}) = G(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}}) + \mathbf{z}^\top \mathbf{W} \mathbf{y} + \tilde{\mathbf{z}}^\top \tilde{\mathbf{W}} \tilde{\mathbf{x}}. \quad (9)$$

4.2 Comparison with [4, 5]

In this subsection we show the equivalence of our approach and approach from [4, 5] from the perspective of saddle point problems. Since the shared variables \tilde{x} are handled in the same way in both approaches (by introducing the constraint $\tilde{\mathbf{W}} \tilde{\mathbf{x}} = 0$ into the optimization problem), we consider the case without shared variables and only with equality-type constraints to simplify the derivations.

Let us introduce a set of new matrices and vectors:

$$\mathbf{A} = \text{diag}(A^1, \dots, A^l), \mathbf{b} = (b^{1\top}, \dots, b^{l\top})^\top, \quad (10)$$

$$W^{mk} = \text{diag}(W_{k\bullet}, \dots, W_{k\bullet}) \in \mathbb{R}^{m \times ml}, \mathbf{W}^m = \begin{bmatrix} W^{m1} \\ \vdots \\ W^{ml} \end{bmatrix} \in \mathbb{R}^{ml \times ml}, \quad (11)$$

In [4, 5] the following distributed-friendly reformulation of problem (2) is proposed, and its equivalence to the original problem is shown:

$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^{ml}} \left\{ f(\mathbf{x}) = \sum_{k=1}^l f^k(x^k) \right\}, \quad (12a)$$

$$\mathbf{A} \mathbf{x} - \mathbf{b} + \mathbf{W}^m \mathbf{y} = 0. \quad (12b)$$

Here a sort of consensus constraint is integrated directly into the minimization problem, which differs from our technique of adding consensus constraint into the corresponding saddle point problem. Note also that \mathbf{W}^m and \mathbf{W} differ in

their structure (the way of constructing communication matrix for using it with multi-dimensional variables).

The saddle point problem corresponding to the minimization problem (12) is

$$\min_{\mathbf{x}, \mathbf{y}} \max_{\mathbf{z}} L(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \min_{\mathbf{x}, \mathbf{y}} \max_{\mathbf{z}} f(\mathbf{x}) + \mathbf{z}^\top (\mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{W}^m \mathbf{y}). \quad (13)$$

Let us now compare this problem with the saddle point problem (7). By rewriting sum in (7) and using the symmetry of \mathbf{W} we have

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \max_{\mathbf{y}} \sum_{k=1}^l [f^k(x^k) + y^{k\top} (A^k x^k - b^k)] + \mathbf{z}^\top \mathbf{W} \mathbf{y} \\ = \min_{\mathbf{x}, \mathbf{z}} \max_{\mathbf{y}} f(\mathbf{x}) + \mathbf{y}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) + \mathbf{z}^\top \mathbf{W} \mathbf{y} \\ = \min_{\mathbf{x}, \mathbf{z}} \max_{\mathbf{y}} f(\mathbf{x}) + \mathbf{y}^\top (\mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{W}\mathbf{z}). \end{aligned} \quad (14)$$

Since \mathbf{W} and \mathbf{W}^m differ only in the arrangement of columns, problems (13) and (14) differ only in the arrangement of components of maximized variables. Therefore, both approaches leads to the same saddle point problem.

5 Algorithm

We use classical Extragradient algorithm from [6]. Being applied to the problem (2) it converges to the solutions of the primal and the dual problems as will be shown in the next sections. Here we describe it in an explicit form, so it is ready to be applied to the problem (2), see Algorithm 1.

Note, that the projection in our case is a simple clipping and can be performed independently for each component of the variable.

6 Smoothness and Domain Size Analysis

In this section we will perform some technical analysis to obtain the relations between parameters of the input data to the problem (object functions and constrains) and parameters of Extragradient's convergence rate.

6.1 Bounds on $\|\mathbf{y}^*\|$, $\|\tilde{\mathbf{y}}^*\|$

To calculate Lipschitz smoothness constants of the problem we have to localize \mathbf{y}^* (dual part of solution of the initial saddle problem (4), which is also a solution to the dual problem under our assumptions), i.e. find R_y such that \mathcal{Y} lies in a ball in \mathbb{R}^m with center in 0 and radius R_y , and $\mathbf{y}^* \in \mathcal{Y}$. From optimality conditions for dual problem of (2)

$$\nabla_{\mathbf{x}} L = \nabla_{\mathbf{x}} f + (A^\top, C^\top) \mathbf{y}^* = 0, \quad (15)$$

Algorithm 1. Decentralized Extragradient for problem (2)

1: Initialize $\mathbf{x}_0 \in \mathcal{X}, \mathbf{y}_0 = \mathbf{z}_0 = \mathbf{0}_{l(m+h)}, \tilde{\mathbf{x}}_0 \in \tilde{\mathcal{X}}^l, \tilde{\mathbf{y}} = \mathbf{0}_{l(\tilde{m}+\tilde{p})}, \tilde{\mathbf{z}}_0 = \mathbf{0}_{l\tilde{m}}$

2: **for** $i = 0, \dots, N - 1$ **do**

3: Compute $\mathbf{z}'_i = \mathbf{W}\mathbf{z}_i, \mathbf{y}'_i = \mathbf{W}\mathbf{y}_i, \tilde{\mathbf{z}}'_i = \tilde{\mathbf{W}}\tilde{\mathbf{z}}_i, \tilde{\mathbf{x}}'_i = \tilde{\mathbf{W}}\tilde{\mathbf{x}}_i$.

4: Make intermediate gradient step

$$x_{i+\frac{1}{2}}^k = \text{Proj}_{\mathcal{X}} \left(x_i^k - h \nabla_{\mathbf{x}^k} f^k(x_i^k, \tilde{x}_i^k) - h(A^{k\top}, C^{k\top})y_i^k \right)$$

$$\tilde{x}_{i+\frac{1}{2}}^k = \text{Proj}_{\tilde{\mathcal{X}}} \left(\tilde{x}_i^k - h \nabla_{\tilde{\mathbf{x}}^k} f^k(x_i^k, \tilde{x}_i^k) - h\tilde{z}_i^{k\top} \right)$$

$$y_{i+\frac{1}{2}}^k = \text{Proj}_{\mathcal{Y}} \left(y_i^k + h \begin{pmatrix} A^k x_{i+\frac{1}{2}}^k - b^k \\ C^k x_{i+\frac{1}{2}}^k - d^k \end{pmatrix} + h z_i^{k\top} \right)$$

$$\tilde{y}_{i+\frac{1}{2}}^k = \text{Proj}_{\tilde{\mathcal{Y}}} \left(\tilde{y}_i^k + h \begin{pmatrix} \tilde{A}^k \tilde{x}_{i+\frac{1}{2}}^k - \tilde{b}^k \\ \tilde{C}^k \tilde{x}_{i+\frac{1}{2}}^k - \tilde{d}^k \end{pmatrix} \right)$$

$$z_{i+\frac{1}{2}}^k = z_i^k - h y_i^{k\top}$$

$$\tilde{z}_{i+\frac{1}{2}}^k = \tilde{z}_i^k + h \tilde{x}_i^{k\top}$$

5: Compute $\mathbf{z}'_{i+\frac{1}{2}} = \mathbf{W}\mathbf{z}_{i+\frac{1}{2}}, \mathbf{y}'_{i+\frac{1}{2}} = \mathbf{W}\mathbf{y}_{i+\frac{1}{2}}, \tilde{\mathbf{z}}'_{i+\frac{1}{2}} = \tilde{\mathbf{W}}\tilde{\mathbf{z}}_{i+\frac{1}{2}}, \tilde{\mathbf{x}}'_{i+\frac{1}{2}} = \tilde{\mathbf{W}}\tilde{\mathbf{x}}_{i+\frac{1}{2}}$.

6: Make gradient step

$$x_{i+1}^k = \text{Proj}_{\mathcal{X}} \left(x_{i+\frac{1}{2}}^k - h \nabla_{\mathbf{x}^k} f^k(x_{i+\frac{1}{2}}^k, \tilde{x}_{i+\frac{1}{2}}^k) - h(A^{k\top}, C^{k\top})y_{i+\frac{1}{2}}^k \right)$$

$$\tilde{x}_{i+1}^k = \text{Proj}_{\tilde{\mathcal{X}}} \left(\tilde{x}_{i+\frac{1}{2}}^k - h \nabla_{\tilde{\mathbf{x}}^k} f^k(x_{i+\frac{1}{2}}^k, \tilde{x}_{i+\frac{1}{2}}^k) - h\tilde{z}_{i+\frac{1}{2}}^{k\top} \right)$$

$$y_{i+1}^k = \text{Proj}_{\mathcal{Y}} \left(y_{i+\frac{1}{2}}^k + h \begin{pmatrix} A^k x_{i+1}^k - b^k \\ C^k x_{i+1}^k - d^k \end{pmatrix} + h z_{i+\frac{1}{2}}^{k\top} \right)$$

$$\tilde{y}_{i+1}^k = \text{Proj}_{\tilde{\mathcal{Y}}} \left(\tilde{y}_{i+\frac{1}{2}}^k + h \begin{pmatrix} \tilde{A}^k \tilde{x}_{i+1}^k - \tilde{b}^k \\ \tilde{C}^k \tilde{x}_{i+1}^k - \tilde{d}^k \end{pmatrix} \right)$$

$$z_{i+1}^k = z_{i+\frac{1}{2}}^k - h y_{i+\frac{1}{2}}^{k\top}$$

$$\tilde{z}_{i+1}^k = \tilde{z}_{i+\frac{1}{2}}^k + h \tilde{x}_{i+\frac{1}{2}}^{k\top}$$

7: **end for**

Ensure: For $\mathbf{t} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}\}$ compute $\hat{\mathbf{t}}^N = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{t}^{k+\frac{1}{2}}$.

$$\nabla_{\tilde{\mathbf{x}}} L = \nabla_{\tilde{\mathbf{x}}} f + (\tilde{A}^\top, \tilde{C}^\top) \tilde{\mathbf{y}}^* = 0. \quad (16)$$

Since for any $y \in \ker A^T$ vector $y^* + y$ is also a solution, we consider only solution with the smallest norm (it's enough for saddle point problem solution's quality criteria and convergence analysis), i. e. $y^* \in (\ker A^T)^\perp$.

Therefore

$$\|y^*\|^2 \leq \frac{\|\nabla_{\mathbf{x}} f(\mathbf{x}^*, \tilde{x}^*)\|^2}{(\sigma_{\min}^+((A^\top, C^\top)))^2},$$

$$\|\tilde{y}^*\|^2 \leq \frac{\|\nabla_{\tilde{x}} f(\mathbf{x}^*, \tilde{x}^*)\|^2}{(\sigma_{\min}^+((\tilde{A}^\top, \tilde{C}^\top)))^2},$$

where $\sigma_{\min}^+(A) = \sqrt{\min\{\lambda > 0 : \exists x \neq 0 : AA^\top x = \lambda x\}}$. Hence we get

Lemma 1. *Saddle point problem (7), which is unconstrained on variables $\mathbf{y}, \tilde{\mathbf{y}}$, is equivalent to the same problem with constraints $\|\mathbf{y}\| \leq R_{\mathbf{y}}$ and $\|\tilde{\mathbf{y}}\| \leq R_{\tilde{\mathbf{y}}}$, where*

$$R_{\mathbf{y}} = \sqrt{l} \frac{\max_{\mathbf{x} \in \mathcal{X}, \tilde{x} \in \tilde{\mathcal{X}}} \|\nabla_{\mathbf{x}} f(\mathbf{x}, \tilde{x})\|}{\sigma_{\min}^+((A^\top, C^\top))}, R_{\tilde{\mathbf{y}}} = \sqrt{l} \frac{\max_{\mathbf{x} \in \mathcal{X}, \tilde{x} \in \tilde{\mathcal{X}}} \|\nabla_{\tilde{x}} f(\mathbf{x}, \tilde{x})\|}{\sigma_{\min}^+((\tilde{A}^\top, \tilde{C}^\top))}. \quad (17)$$

6.2 Bounds on $\|z^*\|$, $\|\tilde{z}^*\|$

Next we want to find constants for Euclidean-case bounds for Theorem 3.5 [15]. To specify, how the convergence rate depends on problem's parameters, we need to find scalars $M_y, M_{\tilde{x}}, L_{xx}, L_{yx}, L_{xy}, L_{yy}$, determined by inequalities

$$\|\nabla_y g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k)\| \leq M_y \quad \forall k, x_k \in \mathcal{X}_k, y_k \in \mathcal{Y}, \quad (18a)$$

$$\|\nabla_{\tilde{x}} g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k)\| \leq M_{\tilde{x}} \quad \forall k, x_k \in \mathcal{X}_k, y_k \in \mathcal{Y}, \quad (18b)$$

$$\|\nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_x G(\bar{\mathbf{x}}', \bar{\mathbf{y}})\| \leq L_{xx} \|\bar{\mathbf{x}} - \bar{\mathbf{x}}'\| \quad \forall \bar{\mathbf{x}}, \bar{\mathbf{x}}' \in \bar{\mathcal{X}}, \bar{\mathbf{y}} \in \bar{\mathcal{Y}}, \quad (18c)$$

$$\|\nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}')\| \leq L_{xy} \|\bar{\mathbf{y}} - \bar{\mathbf{y}}'\| \quad \forall \bar{\mathbf{x}} \in \bar{\mathcal{X}}, \bar{\mathbf{y}}, \bar{\mathbf{y}}' \in \bar{\mathcal{Y}}, \quad (18d)$$

$$\|\nabla_y G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_y G(\bar{\mathbf{x}}', \bar{\mathbf{y}})\| \leq L_{yx} \|\bar{\mathbf{x}} - \bar{\mathbf{x}}'\| \quad \forall \bar{\mathbf{x}}, \bar{\mathbf{x}}' \in \bar{\mathcal{X}}, \forall \bar{\mathbf{y}} \in \bar{\mathcal{Y}}, \quad (18e)$$

$$\|\nabla_y G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_y G(\bar{\mathbf{x}}, \bar{\mathbf{y}}')\| \leq L_{yy} \|\bar{\mathbf{y}} - \bar{\mathbf{y}}'\| \quad \forall \bar{\mathbf{x}} \in \bar{\mathcal{X}}, \forall \bar{\mathbf{y}}, \bar{\mathbf{y}}' \in \bar{\mathcal{Y}}, \quad (18f)$$

where $\bar{\mathbf{x}} = (\mathbf{x}^\top, \tilde{\mathbf{x}}^\top)^\top$ and $\bar{\mathbf{y}} = (\mathbf{y}^\top, \tilde{\mathbf{y}}^\top)^\top$.

By using the triangle inequality

$$\|\nabla_{y^k} g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k)\| = \|\nabla_{y^k} y^\top (A^k x^k - b^k)\| = \left\| \begin{pmatrix} A^k x^k - b^k \\ C^k x^k - d^k \end{pmatrix} \right\| \leq \max_{k \in \{1, \dots, l\}} \left\{ \sigma_{\max} \left((A^k{}^\top, C^k{}^\top) \right) R_{x^k} + \left\| (b^k{}^\top, d^k{}^\top) \right\| \right\} = M_y,$$

and

$$\|\nabla_{\tilde{x}^k} g^k(x^k, \tilde{x}^k, y^k, \tilde{y}^k)\| = \|(\tilde{A}^\top, \tilde{C}^\top) \tilde{y}^k\| \leq \sigma_{\max}((\tilde{A}^\top, \tilde{C}^\top)) R_{\tilde{\mathbf{y}}}$$

$$= \chi((\tilde{A}^\top, \tilde{C}^\top)) \max_{\mathbf{x} \in \mathcal{X}, \tilde{x} \in \tilde{\mathcal{X}}} \|\nabla_{\tilde{x}} f(\mathbf{x}, \tilde{x})\| = M_{\tilde{x}}.$$

Then by directly applying Lemma 4.2 in [15] we have

Lemma 2. *Saddle point problem (7), which is unconstrained on variables $\mathbf{z}, \tilde{\mathbf{z}}$, is equivalent to the same problem with constraints $\|\mathbf{z}\| \leq R_{\mathbf{z}}$ and $\|\tilde{\mathbf{z}}\| \leq R_{\tilde{\mathbf{z}}}$, where*

$$R_{\mathbf{z}} = \frac{\sqrt{2l}M_y}{\lambda_{\min}^+(\mathbf{W})}, R_{\tilde{\mathbf{z}}} = \frac{\sqrt{2l}M_{\tilde{\mathbf{x}}}}{\lambda_{\min}^+(\tilde{\mathbf{W}})}. \quad (19)$$

6.3 Smoothness Constants

Let us find smoothness constants of function G . From (7) we have

$$\nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_x G(\bar{\mathbf{x}}', \bar{\mathbf{y}}) = \begin{pmatrix} \nabla f^1(x^1, \tilde{x}^1) - \nabla f^1(x^{1'}, \tilde{x}^{1'}) \\ \vdots \\ \nabla f^l(x^l, \tilde{x}^l) - \nabla f^l(x^{l'}, \tilde{x}^{l'}) \end{pmatrix}.$$

By Assumption 1

$$\begin{aligned} \|\nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_x G(\bar{\mathbf{x}}', \bar{\mathbf{y}})\|^2 &= \sum_{k=1}^l \|\nabla f^k(x^k, \tilde{x}^k) - \nabla f^k(x^{k'}, \tilde{x}^{k'})\|^2 \\ &\leq \sum_{k=1}^l L_k^2 \left\| \begin{pmatrix} x'^k - x^k \\ \tilde{x}'^k - \tilde{x}^k \end{pmatrix} \right\|^2 \leq \max_k L_k^2 \|\bar{\mathbf{x}} - \bar{\mathbf{x}}'\|^2. \end{aligned}$$

Taking square root from both parts of the inequality we get

$$L_{xx} = \max_{k \in \{1, \dots, l\}} L_k.$$

Similarly, for other variables

$$\begin{aligned} \nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_x G(\bar{\mathbf{x}}, \bar{\mathbf{y}}') &= \begin{pmatrix} (A^\top, C^\top) (y - y') \\ (\tilde{A}^\top, \tilde{C}^\top) (\tilde{y}^l - \tilde{y}^{l'}) \end{pmatrix}, \\ \nabla_y G(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \nabla_y G(\bar{\mathbf{x}}', \bar{\mathbf{y}}) &= \begin{pmatrix} (A^{1\top}, C^{1\top})^\top (x^1 - x^{1'}) \\ \vdots \\ (A^{l\top}, C^{l\top})^\top (x^l - x^{l'}) \\ (\tilde{A}^\top, \tilde{C}^\top)^\top (\tilde{x}^1 - \tilde{x}^{1'}) \\ \vdots \\ (\tilde{A}^\top, \tilde{C}^\top)^\top (\tilde{x}^l - \tilde{x}^{l'}) \end{pmatrix}. \end{aligned}$$

and

$$L_{xy} = \max \left\{ \max_{k \in \{1, \dots, l\}} \sigma_{\max} \left((A^k{}^\top, C^k{}^\top) \right), \sigma_{\max} \left((\tilde{A}^\top, \tilde{C}^\top) \right) \right\} = L_{yx},$$

$$L_{yy} = 0.$$

7 Main Result

Let us denote

$$L_\zeta = 2 \cdot \max\{R_{\tilde{\mathbf{x}}}^2 L_{x\tilde{x},x\tilde{x}}, R_{\tilde{\mathbf{y}}}^2 L_{y\tilde{y},y\tilde{y}}, \\ \sqrt{2}R_{\mathbf{x}\tilde{\mathbf{x}}}R_{\mathbf{y}\tilde{\mathbf{y}}}L_{x\tilde{x},y\tilde{y}} + 2M_{\mathbf{x}\tilde{\mathbf{x}}}R_{\mathbf{x}\tilde{\mathbf{x}}}\frac{\lambda_{\max}(\tilde{\mathbf{W}})}{\lambda_{\min}^+(\tilde{\mathbf{W}})} + 2M_{\mathbf{y}\tilde{\mathbf{y}}}R_{\mathbf{y}\tilde{\mathbf{y}}}\frac{\lambda_{\max}(\mathbf{W})}{\lambda_{\min}^+(\mathbf{W})}\}.$$

Then, following the arguments presented in Theorem 4.5 from [15], we introduce $\zeta = (\mathbf{x}^\top, \tilde{\mathbf{x}}^\top, \mathbf{y}^\top, \tilde{\mathbf{y}}^\top, \mathbf{z}^\top, \tilde{\mathbf{z}}^\top)^\top$. We also define a norm for ζ as follows:

$$\|\zeta\|^2 = \frac{\|\mathbf{x}\|^2}{R_{\mathbf{x}}^2} + \frac{\|\tilde{\mathbf{x}}\|^2}{R_{\tilde{\mathbf{x}}}^2} + \frac{\|\mathbf{y}\|^2}{R_{\mathbf{y}}^2} + \frac{\|\tilde{\mathbf{y}}\|^2}{R_{\tilde{\mathbf{y}}}^2} + \frac{\|\mathbf{z}\|^2}{R_{\mathbf{z}}^2} + \frac{\|\tilde{\mathbf{z}}\|^2}{R_{\tilde{\mathbf{z}}}^2}.$$

According to the standard analysis of Mirror-Prox algorithm, the duality gap is bounded as follows:

$$G_w(\mathbf{x}_N, \tilde{\mathbf{x}}_N, \mathbf{z}_N, \mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) - G_w(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{z}, \mathbf{y}_N, \tilde{\mathbf{y}}_N, \tilde{\mathbf{z}}_N) \leq \frac{L_\zeta}{2N} \|\zeta - \zeta_0\|^2, \quad (20)$$

Substituting $\mathbf{y} = 0, \tilde{\mathbf{y}} = 0, \tilde{\mathbf{z}} = 0, \mathbf{x} = \mathbf{x}_*, \tilde{\mathbf{x}} = \tilde{\mathbf{x}}_*, \mathbf{z} = 0$ we get complexity estimate by function residual:

$$\sum_{k=1}^{\ell} f(x_N^k, \tilde{x}_N^k) - \sum_{k=1}^{\ell} f(x_*^k, \tilde{x}_*^k) \leq \frac{3L_\zeta}{N}. \quad (21)$$

Analogously, we obtain bounds for affine constraints and consensus constraints

$$\begin{aligned} \|A\mathbf{x}_N - \mathbf{b}\| + \|C\mathbf{x}_N - \mathbf{d}\| &\leq \frac{17\sqrt{2}L_\zeta}{N} \min_{k=1,\dots,\ell} \sigma_{\min}^+(A^{k\top}, C^{k\top}), \\ \|\tilde{\mathbf{A}}\tilde{\mathbf{x}} - \mathbf{b}\| + \|\tilde{\mathbf{C}}\tilde{\mathbf{x}} - \mathbf{d}\| &\leq \frac{17\sqrt{2}L_\zeta}{N} \min_{k=1,\dots,\ell} \sigma_{\min}^+(\tilde{\mathbf{A}}^\top, \tilde{\mathbf{C}}^\top), \\ \|\mathbf{W}\mathbf{y}_N\| &\leq \frac{17\sqrt{2}L_\zeta}{2N} \lambda_{\min}^+(\mathbf{W}), \\ \|\tilde{\mathbf{W}}\tilde{\mathbf{x}}_N\| &\leq \frac{17\sqrt{2}L_\zeta}{2N} \lambda_{\min}^+(\tilde{\mathbf{W}}). \end{aligned}$$

Remark 1. In the problem formulation (2) we can additionally assume that $x^k \in Q^k \subseteq \mathbb{R}^{n_k}, \tilde{x} \in \tilde{Q} \subseteq \mathbb{R}^{\tilde{n}}$, where Q^k and \tilde{Q} – simple convex sets, i.e. simplex, ball, half plane e.t.c. In this case instead of decentralized Extragradient method for saddle point problem (Mirror Prox with euclidian prox-function) one should use general decentralized Mirror Prox algorithm [15].

8 Numerical Experiment

For the purpose of numerical experiment data is taken from [17]. Here 6 bus system contains 2 generators. DC optimal power flow problem of the following form is considered:

$$\min_{\substack{p_i^G \in \mathcal{P} \\ \theta \in \Theta}} \sum_{i \in G} c_i(p_i^G) \quad (22a)$$

$$p_i^G - p_i^D = B_{ij}(\theta_i - \theta_j), \quad (22b)$$

$$|(\theta_i - \theta_j)/X_{ij}| \leq F_{ij}^{max}. \quad (22c)$$

Optimization Variables:

- p_i^G , $i \in \{1, \dots, l\}$ —generator power output;
- θ_i , $i \in \{1, \dots, l\}$ —phase angle of the bus i .

Parameters:

- $\mathcal{P} = \prod_{i=1}^l [p_i^{G,\min}, p_i^{G,\max}]$ — minimal and maximal generation. For nodes without generation $p_i^{G,\min} = p_i^{G,\max} = 0$;
- $\Theta = \prod_{i=1}^l [-\theta_i^{max}, \theta_i^{max}]$ — maximal phase angle;
- p_i^D , $i \in \{1, \dots, l\}$ —demand;
- $B_{ij} = B_{ji}$, $i, j \in \{1, \dots, l\}$ —line susceptances. If no power line between nodes i and j then $B_{ij} = 0$ else $B_{ij} > 0$. $X_{ij} = -B_{ij}$, $i, j \in \{1, \dots, l\}$ are line reactances;
- F_{ij}^{max} , $i, j \in \{1, \dots, l\}$ —maximal power flow on the line (i, j) .

Cost Functions:

$c_i(\cdot)$, $i \in \{1, \dots, l\}$ —convex sufficiently smooth functions, representing the cost of operating a generator at given power.

The obtained results are consistent with the results in [17]: generation is equal to 110 MW and 200 MW for the 1-st and 2-nd generators respectively. The results of numerical experiment are given in Fig. 1. Here the plots of function value and constraint residual convergence.

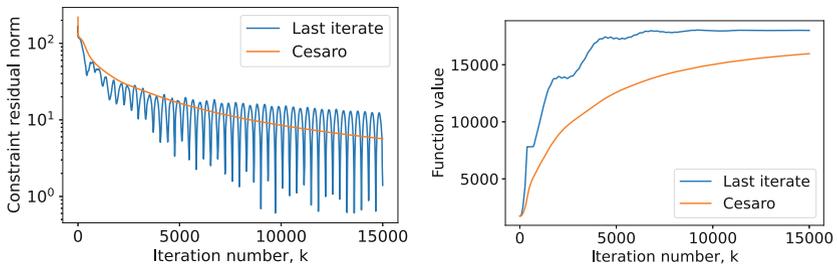


Fig. 1. Results of the numerical experiment for DC optimal power flow problem on 6-bus system [17]

References

1. Erseghe, T.: Distributed optimal power flow using ADMM. *IEEE Trans. Power Syst.* **29**(5), 2370–2380 (2014). <https://doi.org/10.1109/TPWRS.2014.2306495>
2. Falsone, A., Margellos, K., Garatti, S., Prandini, M.: Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica* **84**, 149–158 (2017)
3. Falsone, A., Notarnicola, I., Notarstefano, G., Prandini, M.: Tracking-ADMM for distributed constraint-coupled optimization. *Automatica* **117**, 1–13 (202)
4. Khamisov, O.O.: Direct disturbance based decentralized frequency control for power systems. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 3271–3276, December 2017. <https://doi.org/10.1109/CDC.2017.8264139>
5. Khamisov, O.O., Chernova, T., Bialek, J.W.: Comparison of two schemes for closed-loop decentralized frequency control and overload alleviation. In: 2019 IEEE Milan PowerTech, pp. 1–6, June 2019. <https://doi.org/10.1109/PTC.2019.8810926>
6. Korpelevich, G.M.: The extragradient method for finding saddle points and other problems. *Matecon* **12**, 747–756 (1976)
7. Lan, G.: *First-Order and Stochastic Optimization Methods for Machine Learning*. SSSS, Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-39568-1>
8. Liang, S., Wang, L.Y., Yin, G.: Distributed smooth convex optimization with coupled constraints. *IEEE Trans. Autom. Control* **65**, 347–353 (2020)
9. Liang, S., Zheng, X., Hong, Y.: Distributed nonsmooth optimization with coupled inequality constraints via modified Lagrangian function. *IEEE Trans. Autom. Control* **63**, 1753–1759 (2018)
10. Molzahn, D.K., et al.: A survey of distributed optimization and control algorithms for electric power systems. *IEEE Trans. Smart Grid* **8**(6), 2941–2962 (2017). <https://doi.org/10.1109/TSG.2017.2720471>
11. Necoara, I., Nedelcu, V.: Distributed dual gradient methods and error bound conditions. [arXiv:1401.4398](https://arxiv.org/abs/1401.4398) (2014)
12. Necoara, I., Nedelcu, V.: On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica* **55**, 209–216 (2015)
13. Necoara, I., Nedelcu, V., Dumitrache, I.: Parallel and distributed optimization methods for estimation and control in networks. *J. Process Control* **21**(5), 756–766 (2011). <https://www.sciencedirect.com/science/article/pii/S095915241000257X>, Special Issue on Hierarchical and Distributed Model Predictive Control <https://doi.org/10.1016/j.jprocont.2010.12.010>
14. Patari, N., Venkataramanan, V., Srivastava, A., Molzahn, D.K., Li, N., Annaswamy, A.: Distributed optimization in distribution systems: use cases, limitations, and research needs. *IEEE Trans. Power Syst.*, 1–1 (2021). <https://doi.org/10.1109/TPWRS.2021.3132348>. Early access
15. Rogozin, A., Beznosikov, A., Dvinskikh, D., Kovalev, D., Dvurechensky, P., Gasnikov, A.: Decentralized distributed optimization for saddle point problems. [arXiv preprint arXiv:2102.07758](https://arxiv.org/abs/2102.07758) (2021)
16. Rostampour, V., Haar, O.T., Keviczky, T.: Distributed stochastic reserve scheduling in AC power systems with uncertain generation. *IEEE Trans. Power Syst.* **34**(2), 1005–1020 (2019). <https://doi.org/10.1109/TPWRS.2018.2878888>
17. Wang, Y., Wu, L., Wang, S.: A fully-decentralized consensus-based ADMM approach for DC-OPF with demand response. *IEEE Trans. Smart Grid* **8**(6), 2637–2647 (2017). <https://doi.org/10.1109/TSG.2016.2532467>

18. Wnag, Z., Ong, C.J.: Distributed model predictive control of linear discrete-times systems with local and global constraints. *Automatica* **81**, 184–195 (2017)
19. Yuan, D., Ho, D.W.C., Jiang, G.P.: An adaptive primal-dual subgradient algorithm for online distributed constrained optimization. *IEEE Trans. Cybern.* **48**, 3045–3055 (2018)

Heuristics and Metaheuristics



Metaheuristic Approach to Spectral Reconstruction of Graphs

Petar Ćirković¹, Predrag Đorđević¹, Miloš Milićević²,
and Tatjana Davidović³(✉)

¹ Faculty of Science and Mathematics, University of Niš, niš, Serbia
{petar.cirkovic,predrag.djordjevic}@pmf.edu.rs

² Faculty of Mathematics, University of Belgrade, Belgrade, Serbia
mi19019@alas.matf.bg.ac.rs

³ Mathematical Institute of the Serbian Academy of Sciences and Arts,
Belgrade, Serbia
tanjad@mi.sanu.ac.rs

Abstract. Characterization of a graph by its spectrum is a very attractive research problem that has numerous applications. It is shown that the graph is not necessarily uniquely determined by its spectrum in the most general case, i.e., there could be several non-isomorphic graphs corresponding to the same spectrum. All such graphs are called cospectral. However, in most of the cases, it is important to find at least one graph whose spectrum is equal to a given constant vector. This process is called Spectral Reconstruction of Graph (SRG) and it is known as one of the most difficult optimization problems. We address the SRG problem by the metaheuristic methods, more precisely, by Basic Variable Neighborhood Search (BVNS) and improvement-based Bee Colony Optimization (BCOi) methods. The resulting heuristics are called SRG-BVNS and SRG-BCOi, respectively. Both methods are implemented in such a way to take into account the graph properties defined by its spectrum. We compare the performance of the proposed methods with each other and with the results obtained by other approaches from the relevant literature on the reconstruction of some well-known graphs.

Keywords: Spectral graph theory · Spectral distance · Cospectral graphs · Metaheuristics

1 Introduction

Graphs are mathematical objects defined as 2-tuples $G = (V, E)$ [8], where $V = \{v_1, v_2, \dots, v_n\}$, represents the set of *vertices* v_i , while $E \subseteq V \times V$ denotes the connections (relations) between the pairs of vertices and is called the set of *edges*. If there is a connection (edge) between vertices v_i and v_j , we say that $\{v_i, v_j\} \in E$ and that vertices v_i and v_j are *adjacent*. Graphs are used to model numerous problems in science, engineering, industry, etc. Usually, V is finite set, however, the infinite cases are also studied in the literature starting with [22]. In this paper, we consider only finite and undirect graphs.

The simplest graph representation is by the *Adjacency matrix* A with elements 0 or 1 defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_j\} \in E; \\ 0, & \text{otherwise.} \end{cases}$$

If graph is undirected, A is symmetric, i.e., $a_{ij} = a_{ji}$. The *degree* of vertex v_i (denoted by d_i) in graph G represents the number of vertices adjacent to v_i , i.e., the number of edges having v_i as an end-vertex and it is calculated as $d_i = \sum_{j=1}^n a_{ij}$. *Eigenvalues* λ_i , $i = 1, 2, \dots, n$ for the graph G are actually the eigenvalues of matrix A , i.e., the roots of its characteristic polynomial $P_G(x) = \det(xI - A)$. As the adjacency matrix A is symmetric its eigenvalues are real numbers. The set of all eigenvalues of graph G is called *spectrum*. It can contain negative, positive values and zeros, with some repeated values. It is usual to represent the spectrum as a non-increasing array of values $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Then, the largest eigenvalue λ_1 of graph G is called *index*. An array x such that $Ax = \lambda x$ is known as *eigenvector* (corresponding to the eigenvalue λ) of graph G , and it actually represents the eigenvector of matrix A .

Spectral graph theory (SGT) [11,23] studies graphs based on their adjacency matrix, more precisely, based on the *eigenvalues* and *eigenvectors* of this matrix. In recent literature, some other matrices associated with graphs are defined and analyzed, such as Laplacian matrix and signless Laplacian matrix ([11], Sect. 1.3). However, they will not be considered in this paper. SGT has important applications in various fields of computer science [13], some of them including graph recognition problems [5,7,13,19] as graphs represent natural models for various types of objects. It has been shown in the literature that some special classes of graphs, e.g., complete graphs, paths, cycles, are determined (to the isomorphism) by the spectrum with respect to the adjacency matrix A . However, it has been proved that it does not hold in the general case, i.e., an arbitrary graph cannot be fully characterized by its spectrum, and there may exist non-isomorphic graphs having the same spectrum. In particular, it has been shown that trees cannot be characterized by a spectrum, nor can molecules in chemistry. Non-isomorphic graphs that have identical spectra are called *cospectral*. In [17,24], the number of non-isomorphic cospectral graphs is analyzed in relation to the three mentioned matrices for all graphs with $n \leq 11$ vertices. Graphs with $n = 12$ vertices are considered in [3]. In these papers, it was noticed that the number of graphs with non-isomorphic co-spectral mates decreases from $n = 10$ with respect to the total number of graphs with the same number of vertices. Based on that observation, a hypothesis has been introduced stating that graphs with a large number of vertices (for $n \rightarrow \infty$) may be determined by their spectrum. The hypothesis is still an open problem in SGT.

Our work is inspired by the results published in [5] and presents their generalization and expansion. In the first part of [5], the authors discussed the problem of Spectral Reconstruction of Graphs (SRG) [9] with the help of AutoGraphiX (AGX) software package developed at GERAD Institute in Montreal [1,6]. AGX uses the Variable Neighborhood Search (VNS) metaheuristic method [18,21] to find graphs that have extreme values of selected invariants or their combinations.

We have also applied AGX, optimized its execution by adding new constraints that enable to reduce the search space, and consequently, to decrease the time required to obtain the results.

As AGX is a general purpose software, it obviously contains many auxiliary functions that are not necessary for the considered problem. Therefore, we do not expect its good performance and we propose the application of metaheuristic methods to efficiently find a graph with the given spectrum. We have implemented a basic version of VNS (BVNS) and an improvement-based Bee Colony Optimization (BCOi) [14, 15] to tackle the SRG problem. The methods are called SRG-BVNS and SRG-BCOi, respectively. The stochastic nature of metaheuristic methods allows us to perform restarts from different random initial graphs, and to generate mutually non-isomorphic cospectral graphs (if any). However, finding all cospectral graphs still remains a challenging task because it is actually a NP-hard optimization problem: it is necessary to examine all graphs with n vertices and m edges and the number of such graphs is $\binom{n(n-1)/2}{m}$, i.e., the number of ways m edges can be distributed in $n(n-1)/2$ places.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the relevant literature. The SRG problem is described in detail in Sect. 3, specifying its complexity and some special cases in which there are efficient algorithms for finding all non-isomorphic cospectral graphs with a given spectrum. In Sects. 4 and 5, the implementations of SRG-BVNS and SRG-BCOi are described. The results obtained by applying the implemented methods to some known graphs from the literature, are presented in Sect. 6. Concluding remarks and guidelines for future work are given in Sect. 7.

2 Literature Review

The study of graphs based on their spectra has become very popular in the past two decades because the spectrum can be determined relatively quickly (the computational complexity is, in the general case, $O(n^3)$, and for special classes of graphs this complexity may be significantly reduced). Based on the spectrum, various information can be determined on the structure of the corresponding graph [9, 11, 19], especially on some parameters of the graph that require exponential time for calculation. As already mentioned in the introduction, graphs are not uniquely determined by the spectrum, i.e., for some graphs there exist non-isomorphic cospectral graphs. However, due to its great importance, spectral recognition of graphs is intensively studied in the literature [9, 17, 24].

The review paper [9] defines 4 basic problems that are considered in connection with spectral recognition of graphs: characterization of graphs with a given spectrum; construction (exact or approximate) of a graph with a given spectrum; spectral similarity of graphs; and spectral perturbations of graphs. Let us note once again that all these problems are related to the spectrum of the graph and use *spectral distance*. This distance is defined for graphs with the same number of vertices as the distance between their spectra. Various types of distances may

be used, such as Euclidean, Manhattan or some other distance between ordered sequences of eigenvalues, i.e., vectors in the n -dimensional space.

The first problem (characterization of a graph with the given spectrum) [9] involves describing as many of its properties as possible based on the spectrum. For the second problem, it is necessary to find a graph whose spectrum is represented by a given vector (which actually represents the SRG problem that is considered in this paper). The characterization of a graph by the spectrum is achieved by solving the optimization problem representing minimization of the spectral distance between the given vector and the spectrum of the constructed graph. The solution to this problem does not have to be unique, several mutually non-isomorphic cospectral graphs can be obtained. It is obvious that the cospectral graphs are at a distance equal to zero, which is the minimal value of any spectral distance. The spectral distance can be considered as a measure of graphs' similarity, i.e., we say that the two graphs are similar if their spectral distance is small. Similar graphs are obtained from each other by small perturbations (changes in the structure or spectrum of graphs). Examples of perturbations are removing or adding edges, moving edges from one position to some other, and so on.

One of the first algorithms for spectral reconstruction of graphs based on the Laplacian matrix was developed in [7]. It is based on the Tabu Search (TS) metaheuristic method, starts from a random graph with n vertices and tries to minimize the spectral distance. The algorithm was tested on several classes of networks (random, regular, cluster graphs, etc.).

The VNS method is used in [5] indirectly, through the AGX program package. The authors have performed the reconstruction of some classes of graphs based on Euclidean and Manhattan spectral distances defined with respect to the various matrices associated with the graph. Graphs of up to 20 vertices have been analyzed and the number of successful reconstructions in 100 restarts was reported. The stopping criterion in each execution was 100,000 evaluations of the objective function (i.e., calculations of the spectral distance), and the initial solution was always a randomly generated graph. The paper [5] served as the inspiration for our work. We aim to maximally exploit the information that can be obtained about the target graph from its spectrum and to develop efficient implementations of our methods and to generate the desired graph in the shortest possible time. By repeated restarts, it is possible to obtain several non-isomorphic cospectral graphs.

3 Finding Graph with a Given Spectrum

As it is already mentioned, SRG implies finding (one or more) graphs whose spectrum is equal to a given vector. In this paper we use the Euclidean distance, that is (among others) used in [5] as well. Let $C = (c_1, c_2, \dots, c_n)$ be a given vector, let $G = (V, E)$ be a graph having n vertices, and let $S = (\lambda_1, \lambda_2, \dots, \lambda_n)$ represent its spectrum. It is necessary to perform transformations of the graph G with an aim to minimize (nullify) the spectral distance defined by Eq. (1).

$$d = \sqrt{(c_1 - \lambda_1)^2 + (c_2 - \lambda_2)^2 + \dots + (c_n - \lambda_n)^2}. \quad (1)$$

First, we applied AGX software [1, 6]. It is an interactive software package designed to find extreme graphs, i.e., graphs that minimize or maximize certain graph invariant or a function of graph invariants. A graph invariant is a parameter of a graph that is independent of the vertex and edge labeling. Graph invariants are, for example, the index of a graph (i.e., the largest eigenvalue λ_1), minimum (δ) and maximum (Δ) vertex degree, etc. [8, 11]. Searching for extremal graphs, AGX uses an optimization module based on the VNS meta-heuristics and generates the corresponding graph examples for some special cases (for some given values of n). The researchers use these experimentally obtained graphs to set hypotheses for the general case and then try to prove them theoretically, “by hand” or applying some automatic theorem prover [2, 5, 6]. The latest version of the AGX software package (AGX 3.3.9) as well as the accompanying documentation can be downloaded from the Internet address <https://www.gerad.ca/Gilles.Caporossi/agx/AGX/AutoGraphiX.html>.

To solve SRG problem, we need to ask AGX to minimize the Euclidean distance between the given constant vector C and the ordered vector of eigenvalues of the required graph. To make things easier for AGX, we exploit the fact ([11], p. 85) that the number of edges in a graph can be calculated by the following equation $m = \frac{1}{2} \sum_{i=1}^n \lambda_i^2$. As the input vector C is actually the spectrum of the desired graph, we can calculate the number of its edges by Eq. (2), i.e., using the scalar product of the vector C with itself:

$$m = \frac{1}{2} \sum_{i=1}^n c_i^2. \quad (2)$$

On the other hand, the number of edges in the graph equals one half the sum of all elements of the adjacency matrix A , and we can reduce the search space for AGX by equalizing this sum with the scalar product of the vector C with itself. The block-diagram of the corresponding optimization task performed by AGX software is presented in Fig. 1. Constant vector C is an input parameter, while the adjacency matrix A is provided by AGX in the process of graph generation/transformation. The initial graph is generated randomly with the number of edges depending on the input vector C according to Eq. (2), while all other graphs are obtained by performing transformations (that preserve the number of edges) of the currently best found graph. The goal is to minimize the spectral distance d (given by Eq. (1)) between input vector C and graph defined by the adjacency matrix A , and therefore, the loop is executed until d becomes zero. However, it may take too much time and it is necessary to define some stopping criterion that will interrupt the execution of AGX even if the solution is not found. This is required also for fair comparison of AGX with other approaches.

It is important to note that AGX is a stochastic search engine, and therefore, each of its executions can give a different result, with respect to either the solution itself (when a non-isomorphic cospectral graph is obtained) or the

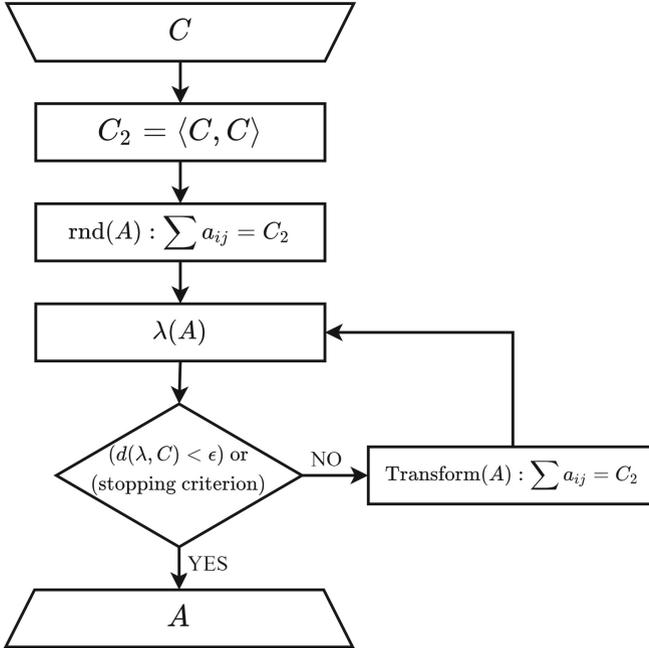


Fig. 1. Minimization of spectral distance according to AGX software

time required to find the same graph. Consequently, for the analysis of AGX performance, it is necessary to repeat executions and determine some average (mean) results. Although AGX cannot guarantee a complete search of the solution space (graphs with a given number of vertices and edges), repeating the execution allows to find some non-isomorphic cospectral graphs (if any). Of course, the fact that AGX failed to generate a cospectral graph, i.e., it obtained the same solution in all executions, does not imply that there are no non-isomorphic cospectral graphs for the given graph. As it is already mentioned, in order to find all non-isomorphic cospectral graphs for a given graph, it is necessary to perform a complete search of graphs with the same number of vertices n and edges m . For example, for a graph with $n = 8$ vertices and $m = 9$ edges, it is necessary to examine 6,906,900 graphs. It is clear that (in the general case) as the number of vertices increases, the complexity of complete search is increasing nonlinearly. On the other hand, there are algorithms developed for some special classes of graphs that employ *a priori* knowledge about these graphs in order to reduce the number of analyzed graphs. An example of such an algorithm is described in [12]. The authors considered Smith graphs (whose spectrum is limited to the interval $[-2, 2]$). They identified transformations that translate one Smith graph into another, mutually non-isomorphic cospectral with the starting one, and developed an algorithm for generating all such graphs. Our goal is to develop algorithm that can be applied to any graph, and therefore, we cannot

compare against the methodology proposed in [12]. We develop two metaheuristic methods (VNS and BCOi) that are compatible with the block-diagram from Fig. 1, however, transformations of solutions are performed in more systematic ways.

4 Variable Neighborhood Search for SRG

In this section we briefly recall some information about the VNS method and then describe its implementation for the considered SRG problem.

4.1 Variable Neighborhood Search

Variable Neighborhood Search (VNS) is a trajectory-based metaheuristic method proposed in [21]. It uses distances between solutions and employs one or more neighborhood structures to efficiently search the solution space of a considered optimization problem. VNS uses some problem-specific local search procedure(s) in the exploitation phase and changing distances between solutions to ensure the exploration of solution space. The role of exploration (diversification, perturbation) phase is to ensure escaping from local optima traps. VNS is widely used optimization tool with many variants and successful applications [18] and we used its basic variant (BVNS) for the SRG problem.

Algorithm 1. Pseudo-code for BVNS method

```

procedure BVNS(Problem input data,  $k_{max}$ , STOP)
   $x_{best} \leftarrow \text{InitSolution}()$ 
  repeat
     $k \leftarrow 1$ 
    repeat
       $x' \leftarrow \text{RandomSolution}(x_{best}, \mathcal{N}_k)$  ▷ Shaking
       $x'' \leftarrow \text{LS}(x')$  ▷ Local Search
      if ( $f(x'') < f(x_{best})$ ) then ▷ Neighborhood Change
         $x_{best} \leftarrow x''$ 
         $k \leftarrow 1$ 
      else
         $k \leftarrow k + 1$ 
      end if
       $\text{Terminate} \leftarrow \text{StoppingCriterion}(\text{STOP})$ 
    until ( $k > k_{max} \vee \text{Terminate}$ )
  until (Terminate)
  return ( $x_{best}, f(x_{best})$ )
end procedure

```

BVNS employs a single type of neighborhood and consists of three main steps: Shaking, Local Search, and Neighborhood Change (see Algorithm 1). The role of Shaking step is to ensure the diversification of the search. It performs a random perturbation of x_{best} in the given neighborhood and provides a starting solution x' to the next step. Local Search tries to improve x' by visiting its neighbors with respect to the selected neighborhood. After the Local Search, BVNS performs Neighborhood Change step in which it examines the quality of the obtained local optimum x'' . If it is better than x_{best} , the search is concentrated around it (the global best solution x_{best} and the neighborhood index k are updated properly). Otherwise, only k is changed. The three main steps are repeated until a pre-specified stopping criterion is satisfied [18].

The main parameter of BVNS is k_{max} , the maximum number of neighborhoods for Shaking. Actually, the current value of k represents the distance between x_{best} and x' obtained within the Shaking phase. BVNS is known as the First Improvement (FI) search strategy because the search is always concentrated around x_{best} : as soon as this solution is improved, k is reset to 1.

4.2 Implementation Details

Let us remind that the number of edges in the graph, which we want to generate based on the given spectrum C , is known, i.e., it can be calculated by Eq. (2). Therefore, we have implemented BVNS because it is enough to consider only one type of neighborhood: moving an edge from one place to another. This neighborhood preserves the number of edges in the graph. The resulting graphs do not have to be connected because this condition is not set for the starting graphs either (although all analyzed examples are connected graphs, they may have non-connected cospectral mates).

The solution of the considered problem is a graph denoted here by g , which should have a spectral distance given by Eq. (1) from the given constant vector C less than some predetermined constant ε . The initial solution is chosen randomly from all graphs with a given number of vertices n and edges m calculated by Eq. (2). Then, the transformations of the initial graph are performed following the steps of BVNS. The solutions in BVNS are represented by three data structures in order to reduce the computational complexity required to find neighbors of the considered graph in Local Search, as well as a random graph at a given distance (with respect to the number of transformations). Obviously, memory usage is sacrificed to increase the efficiency.

The first structure is the adjacency matrix $A = [a_{ij}]_{n \times n}$. It is needed for calculating the spectrum. The second structure contains the lists of adjacent $g[i].ls$ and non-adjacent $g[i].ln$ vertices for each vertex i in graph g . In addition, for each vertex i , it is necessary to always know the number of neighbors/non-neighbors, and this information is stored in the arrays $g[i].ns$ and $g[i].nn$. All these data structures allow to perform any transformation of the graph in a constant number steps $O(1)$. Each deleted vertex is replaced by the last one in the list and the corresponding number of elements is reduced by one ($--ns[i]$ and $--nn[i_1]$). A new vertex is always added to the end of the list, while

the number of list elements is increased by one ($nn[i]++$ and $ns[i_1]++$). Of course, it must be checked that some of the used lists are not empty. The mentioned operations are performed on randomly selected pairs of vertices (i, j) and (i_1, j_1) in the Shaking, while they are applied to all pairs of vertices from the neighborhood of the current solution in the Local Search. Of course, there are still some steps that cannot be performed in less than polynomial (or at least $\log n$) number of operations.

5 Bee Colony Optimization for SRG

This section contains the brief description of Bee Colony Optimization (BCO) metaheuristic, more precisely its improvement-based variant BCOi, as well as the implementation of BCOi for finding graphs with given spectrum.

5.1 Bee Colony Optimization

Bee Colony Optimization (BCO) is a population-based metaheuristic that mimics the foraging process of honeybees in nature [15]. The population consists of artificial bees, each responsible for one solution of the considered problem. During the execution of BCO, artificial bees build (in the constructive BCO variant, BCOc) or transform (in the improvement-based BCOi) their solutions in order to find the best possible with respect to the given objective. The BCO algorithm runs in iterations until a stopping condition is met and the best found solution (the so called global best) is reported as the final one.

Algorithm 2. Pseudo-code of the BCO algorithm

```

procedure BCO(Problem input data,  $B, NC, STOP$ )
  repeat ▷ Main BCO loop
    for  $b \leftarrow 1, B$  do ▷ Initializing population
       $Sol(b) \leftarrow SelectSolution()$ 
    end for
    for  $u \leftarrow 1, NC$  do
      for  $b \leftarrow 1, B$  do ▷ Forward pass
         $EvaluateMove(Sol(b))$ 
         $SelectMove(Sol(b))$ 
      end for
       $EvaluateSolutions()$  ▷ Backward pass
       $Loyalty()$ 
       $Recruitment()$ 
    end for
     $Update(x_{best}, f(x_{best}))$ 
     $Terminate \leftarrow StoppingCriterion(STOP)$ 
  until ( $Terminate$ )
  return ( $x_{best}, f(x_{best})$ )
end procedure

```

Each BCO iteration contains several execution steps divided into two alternating phases: *forward pass* and *backward pass* (see Algorithm 2). Within forward passes, all bees explore the search space by applying a predefined number of moves and obtain new population of solutions. Moves are related to building or transforming solutions, depending on the used BCO variant and they explore *a priori* knowledge about the considered problem. When a new population is obtained, the second phase (backward pass) is executed, where the information about the quality of solutions is exchanged between bees. The solution's quality is defined by the corresponding value of the objective function. The next step in backward pass is to select a subset of promising solutions to be further explored by applying *loyalty decision* and *recruitment* steps. Depending on the relative quality of its current solution with respect to the best solution in the current population, each bee decides with a certain probability should it stay *loyal* to that solution and become a *recruiter* that advertises its solution by simulating waggle dance of honeybees [15]. Obviously, bees with better solutions should have more chances to keep their solutions. A non-loyal bees are referred to as *uncommitted followers*, they abandon current solutions, and have to select one of the solutions held by recruiters. This selection is taken with a probability, such that better advertised solutions have greater opportunities to be chosen for further exploration. In the basic variant of BCO there are only two parameters:

- B – the number of bees involved in the search and
- NC – the number of forward/backward passes in a single BCO iteration.

5.2 Implementation of SRG-BCOi

As in BVNS, we used multiple data structures to represent solutions. The first is adjacency matrix, represented by 2-D arrays in the C(C++) programming language. For each bee b we introduced variable $A[b]$ as array of arrays containing $n * n$ elements. Therefore, our data structure A is actually an 3-D array. If in the solution handled by bee b , vertices i and j of the corresponding graph are connected, then $A[b][(i - 1) * n + j - 1] = A[b][(j - 1) * n + i - 1] = 1$ (as A is symmetric matrix), otherwise, the corresponding elements are equal to 0. We chose 1-D array for storing matrix because it is used in a version of Jacobi algorithm for calculating eigenvalues, which we found on the internet [4]. Our algorithm heavily relies on powerful data structures *vector* and *unordered_set* from C++. Unordered sets take (approximately) constant time to perform insert, delete and find operations, which is very important for obtaining efficient implementation of iterative algorithms. These data structures are used to model lists of adjacent and non-adjacent vertices for a given vertex v_i . It is important to note that here the dimension of used vectors and unordered sets increases by one, for counting bees in our population-based BCOi algorithm. To increase efficiency even more, we store in a separate vector (for each bee) non-isolated vertices, i.e., the ones that have at least one neighbour. We use this vector to select the first end-vertex of an edge to be removed. In addition, we list vertices that have less than $n - 1$ adjacent vertices, to efficiently select the end-vertices of an edge to be added.

An initial population of each BCOi iteration is constructed randomly, by adding edges starting from an empty graph (containing only vertices). For each initial solution, we calculate spectrum by Jacobi algorithm and its spectral distance from the input vector C to evaluate the obtained solutions and to check if we already found the desired graph.

Forward pass involves the required transformations. Each transformation consists of moving a (randomly selected) number (o) of edges from one position to another one. As the first step, we need to select a random value for variable o from the interval $[1, 2 * m]$. The range for o is determined experimentally, having in mind that we should enable performing significant changes of the current solution. Although the total number of edges to be moved is only m , we allow o to take larger values, i.e., to move some edges more than once and, possibly, increase the diversity of the obtained solution. The value for o is determined for each bee separately, ensuring various treatment of the same solutions assigned to different bees (after recruitment). The second step in solution transformation assumes substituting o times an existing edge with a non-existing one. To determine the edge to be removed, we randomly select an element i from the set of non-isolated vertices (as the first end-vertex of the corresponding edge) and then pick randomly one of the vertices (j) adjacent to i from the corresponding unordered set. In a similar way, we select an edge (i_1, j_1) to be included in the transformed graph. Vertex i_1 is selected randomly from the set of vertices that have less than $n - 1$ adjacent vertices, while j_1 is determined as a random element from the set of non-adjacent vertices of i_1 . Random selection from a set usually takes linear time but we found smart trick to avoid it, on the internet [20]. When o transformations are completed, the spectrum of the resulting graph is calculated by Jacobi algorithm and used to determine spectral distance from the input vector C . Among all B solutions, the one with the smallest spectral distance is identified and used to check if we already found the desired graph or if the current best solution is improved.

The backward pass is performed in standard way described in [15]. The probability that bee is loyal to the current solution equals the normalized value of the corresponding objective function, while the recruitment is performed using the roulette wheel composed of solutions advertised by recruiters. Redundant solution representation helps to reduce the complexity of these steps also.

6 Experimental Evaluation

Here we present the results of applying AGX software and the proposed SRG-BVNS and SRG-BCOi methods to the reconstruction of some graph examples with a small number of vertices.

6.1 Testing Environment

SRG-BVNS method is implemented in the R language [16] within RStudio Ver. 2022.02.0 for Windows and executed on Intel Core i7-11800H 2.30 GHz (24 MB

Cache, up to 4.6 GHz) 16 GB DDR4, 512 GB SSD, NVidia GeForce RTX 3050 Ti, GDDR6 4 GB VRAM. AGX software is run on the same computer. SRG-BCOi is coded in C++ and executed on Intel(R) Core(TM) i3-7020U CPU 2.30 GHz, 8 GB DDR4, 512 GB NVMe SSD, Nvidia GeForce MX130 with 2 GB VRAM. In order to be able to ensure fair comparison of the tested methods, we set the stopping criterion for all of them to be the maximum number of objective function evaluation. As in [5], this number is set to 100000.

SRG is specific optimization problem because we know the optimal value of the objective function (when the optimal solution found, the spectral distance between the corresponding graph and a given input vector equals zero). Therefore, we “just” need to find a graph with n vertices and m (calculated by Eq. (2)) edges satisfying the condition $d = 0$, where d is calculated by Eq. (1). This also means that we can stop the execution of the algorithms after the optimal solution is found. As we already noted, the solution space (depending on n and m) can be quite large, making our task very hard.

All of the compared methods are stochastic search algorithms, and therefore, we need to execute them repeatedly (for different values of random generator’s seed) in order to evaluate their stability and performance. We set the number of repetitions to 100 as it ensures statistical significance of the obtained results. As the performance measure, we report the number of successful runs, i.e., the number of graph reconstructions in 100 repetitions, as well as the average number of required objective function evaluations. In the cases when solution was not found in each of 100 executions, we report the average value of the objective function. Regarding the parameters of the compared methods, we used default settings for AGX and performed some preliminary experiments to determine the values of SRG-BVNS and SRG-BCOi parameters. For SRG-BVNS the parameters are specified as follows: $k_{max} = m$ and we apply a FI strategy in LS in order to reduce the time spent in the intensification phase. Parameters of SRG-BCOi are set to the following values: $B = 6$ and $NC = 30$.

6.2 Results of Spectral Reconstruction of Some Graph Examples

The graphs that we selected as the test examples for comparison are presented in Fig. 2 and Fig. 3. These are graphs with 8, 9, and 10 vertices that have been identified in [10] as suitable models for multiprocessor systems. To be able to control the experiment and to replicate the results, we used a fixed set of values for seed in SRG-BVNS, and SRG-BCOi. For the sake of simplicity, seed value in the i -th execution equals i . To the best of our knowledge, it is not possible to control seed value in AGX and its results may be slightly different in some new executions. We hope that 100 repetitions is enough to have a general judgement about AGX performance. The comparison results of these three algorithms are presented in Table 1.

Table 1 is organized as follows: the first column contains the name of the graph example used to define the input vector C ; the remaining columns are grouped by three and they contain the results for each of the compared methods. The first group of three columns show the number of successful reconstructions,

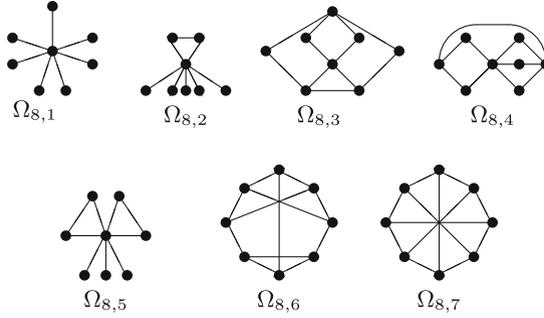


Fig. 2. Test examples with 8 vertices

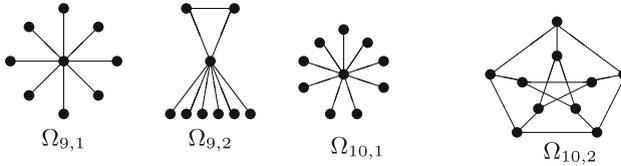


Fig. 3. Test examples with 9 and 10 vertices

Table 1. Comparison of AGX, SRG-BVNS, SRG-BCOi

Graph	AGX			SRG-BVNS			SRG-BCOi		
	#graphs	av. eval.	av. obj.	#graphs	av. eval.	av. obj.	#graphs	av. eval.	av. obj.
$\Omega_{8,1}$	42	65676.66	0.56	100	445.76	0.00	100	595.37	0.00
$\Omega_{8,2}$	16	88186.25	0.23	100	735.64	0.00	100	339.73	0.00
$\Omega_{8,3}$	100	1390.70	0.00	100	324.88	0.00	100	1017.03	0.00
$\Omega_{8,4}$	45	58939.98	0.20	100	489.14	0.00	100	1557.70	0.00
$\Omega_{8,5}$	100	1698.89	0.00	100	336.16	0.00	100	123.73	0.00
$\Omega_{8,6}$	100	2454.73	0.00	100	409.97	0.00	100	*10288.97	0.00
$\Omega_{8,7}$	100	4563.33	0.00	100	415.99	0.00	100	11818.75	0.00
$\Omega_{9,1}$	98	23256.09	0.05	100	369.55	0.00	100	446.10	0.00
$\Omega_{9,2}$	0	100000.00	0.27	100	1182.50	0.00	100	1611.08	0.00
$\Omega_{10,1}$	0	100000.00	0.97	100	1143.97	0.00	100	1866.42	0.00
$\Omega_{10,2}$	95	20094.89	0.06	100	1513.41	0.00	46	*75939.58	1.01

* - the results are obtained when 2 (out of 6) initial solutions are set to the current best solution.

the average number of function evaluations, and the average value of the objective function for AGX, respectively. The corresponding results for SRG-BVNS and SRG-BCOi are presented in the columns 5–7 and 8–10.

Comparing the results from Table 1 we can conclude that both problem-oriented metaheuristic implementations outperformed AGX (except for one example where AGX performed better than SRG-BCOi). This result was expected having in mind that AGX is a general-purpose graph optimization

software. SRG-BVNS was able to find a graph with given spectra in all executions, while SRG-BCOi had troubles with the last tested graph, the Petersen graph $\Omega_{10,2}$. With respect to the average number of objective function evaluations, the superiority of SRG-BVNS is evident in all but two examples, where SRG-BCOi managed to reconstruct a graph faster. Our main conclusion is that single-solution metaheuristic performs better for these examples and we believe that it is a consequence of more systematic search with less randomness, that may lead to the situations where some solutions are visited more than once. We tried to resolve this problem by recording visited solutions in a hash table, however, it turned out that searching this table is also time consuming.

7 Conclusion

We considered the problem of Spectral Reconstruction of a Graph (SRG) and developed the Basic Variable Neighborhood Search (BVNS) and the improvement-based Bee Colony Optimization (BCOi). The SRG problem consists of finding at least one graph whose spectrum coincides with a given vector. The implemented metaheuristic methods take into account the well-known relationship between the number of edges in the graph and its spectrum. The results of applying SRG-BVNS and SRG-BCOi to the reconstruction of some known graphs are compared with each other and with the results obtained using the AutoGraphiX (AGX) package. They clearly show the superiority of the proposed SRG-BVNS implementation with respect to both solution quality and search speed measured by the number of objective function evaluations needed for reconstruction. Potential topics for future research include experiments with graphs of larger dimensions, comparison with similar methods from the literature, and generation of more (as much as possible) non-isomorphic cospectral graphs (if any). In addition, we plan to incorporate other known connections between the parameters of the graph and its spectrum in order to reduce the search space and speedup the execution of our SRG-BVNS and SRG-BCOi methods. Other matrices associated with graphs and other types of distances can be used as well.

Acknowledgements. This work has been supported by the Serbian Ministry of Education, Science and Technological Development, Agreement No. 451-03-9/2021-14/200029, by the Serbian Academy of Sciences and Arts under the project F-159, and by the Science Fund of Republic of Serbia, under the project AI4TrustBC. The authors are grateful to Academician Dragoš Cvetković for numerous suggestions and comments that contributed the quality of this paper.

References

1. Aouchiche, M., et al.: Variable neighborhood search for extremal graphs 14: the AutoGraphiX 2 system. In: Liberti, L., Maculan, N. (eds.) *Global Optimization: From Theory to Implementation*, pp. 281–310. Springer, Boston, MA (2006). https://doi.org/10.1007/0-387-30528-9_10

2. Aouchiche, M., Hansen, P.: A survey of automated conjectures in spectral graph theory. *Linear Algebra Appl.* **432**(9), 2293–2322 (2010)
3. Brouwer, A.E., Spence, E.: Cospectral graphs on 12 vertices. *Electron. J. Comb.* **16**(20), 1–3 (2009)
4. Burkardt, J.: Eigenvalues and Eigenvectors of a Symmetric Matrix (Jacobi Algorithm) (2019). https://people.sc.fsu.edu/~jburkardt/c_src/jacobi_eigenvalue/jacobi_eigenvalue.html
5. Caporossi, G., Cvetković, D., Rowlinson, P.: Spectral reconstruction and isomorphism of graphs using variable neighbourhood search. *Bull. Acad. Serbe Sci. Arts Cl. Sci. Math. Natur. Sci. Math.* **146**(39), 23–38 (2014)
6. Caporossi, G., Hansen, P.: Variable neighborhood search for extremal graphs: 1 the AutoGraphiX system. *Discret. Math.* **212**(1–2), 29–44 (2000)
7. Comellas, F., Diaz-Lopez, J.: Spectral reconstruction of complex networks. *Phys. A* **387**(25), 6436–6442 (2008)
8. Cvetković, D.: *Graph Theory and Applications*, 3rd edn. Naučna knjiga, Beograd (1990)
9. Cvetković, D.: Spectral recognition of graphs. *YUJOR* **22**(2), 145–161 (2012)
10. Cvetković, D., Davidović, T.: *Multiprocessor interconnection networks*, 2nd edn. In: *Zbornik radova, special issue Selected Topics on Applications of Graph Spectra*, vol. 14, no. 22, pp. 35–62. Mathematical Institute SANU (2011)
11. Cvetković, D., Doob, M., Sachs, H.: *Spectra of Graphs: Theory and Application*, 3rd edn. Johann Ambrosius Barth Verlag, Heidelberg-Leipzig (1995)
12. Cvetković, D., Jerotijević, M.: Compositions of cospectrality graphs of smith graphs. *Kragujevac J. Math.* **47**(2), 271–279 (2023)
13. Cvetković, D., Simić, S.: Graph spectra in computer science. *Linear Algebra Appl.* **434**(6), 1545–1562 (2011)
14. Davidović, T., Ramljak, D., Šelmić, M., Teodorović, D.: Bee colony optimization for the p-center problem. *Comput. Oper. Res.* **38**(10), 1367–1376 (2011)
15. Davidović, T., Teodorović, D., Šelmić, M.: Bee colony optimization part i: the algorithm overview. *YUJOR* **25**(1), 33–56 (2015)
16. Dessau, R.B., Pipper, C.B.: “R”-project for statistical computing. *Ugeskrift for laeger* **170**(5), 328–330 (2008)
17. Haemers, W.H., Spence, E.: Enumeration of cospectral graphs. *Eur. J. Comb.* **25**(2), 199–211 (2004)
18. Hansen, P., Mladenović, N., Brimberg, J., Pérez, J.A.M.: Variable neighborhood search. In: Gendreau, M., Potvin, J.-Y. (eds.) *Handbook of Metaheuristics*. ISORMS, vol. 272, pp. 57–97. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91086-4_3
19. Jovanović, I.M.: *Spectral Recognition of Graphs and Networks*. Ph.D. thesis (in Serbian), University of Belgrade, Faculty of Mathematics (2014)
20. Matovitch: Random Element from unordered_set in $O(1)$ (2015). <https://stackoverflow.com/questions/12761315/random-element-from-unordered-set-in-o1/31522686#31522686>
21. Mladenović, N., Hansen, P.: Variable neighborhood search. *Comput. Oper. Res.* **24**(11), 1097–1100 (1997)
22. Nash-Williams, C.: Infinite graphs - a survey. *J. Comb. Theor.* **3**(3), 286–301 (1967)
23. Spielman, D.: Spectral graph theory. In: Naumann, U., Schenk, O. (eds.) *Combinatorial Scientific Computing*, vol. 18, pp. 18:1–18:30. CRC Press (2012)
24. Van Dam, E.R., Haemers, W.H.: Which graphs are determined by their spectrum? *Linear Algebra Appl.* **373**, 241–272 (2003)



Variable Neighborhood Search for Multi-label Feature Selection

Luka Matijević (✉) 

Mathematical Institute of the Serbian Academy of Sciences and Arts,
Kneza Mihaila 36, 11001 Belgrade, Serbia
luka@mi.sanu.ac.rs

Abstract. With the growing dimensionality of the data in many real-world applications, feature selection is becoming an increasingly important preprocessing step in multi-label classification. Finding a smaller subset of the most relevant features can significantly reduce resource consumption of model training, and in some cases, it can even result in a model with higher accuracy. Traditionally, feature selection has been done by employing some statistical measure to determine the most influential features, but in recent years, more and more metaheuristics have been proposed to tackle this problem more effectively. In this paper, we propose using the *Basic Variable Neighborhood Search* (BVNS) algorithm to search for the optimal subset of features, combined with a local search method based on mutual information. The algorithm can be considered a hybrid between the wrapper and filter methods, as it uses statistical knowledge about features to reduce the number of examined solutions during the local search. We compared our approach against *Ant Colony Optimization* (ACO) and *Memetic Algorithm* (MA), using the *K-nearest neighbors* classifier to evaluate solutions. The experiments conducted using three different metrics on a total of four benchmark datasets suggest that our approach outperforms ACO and MA.

Keywords: K-nearest neighbors · Metaheuristics · Optimization · Mutual information

1 Introduction

During the last decade we have witnessed the growing popularity of machine learning and data mining algorithms. These algorithms for predicting the outcome based on the given data have found numerous applications in the business and day-to-day lives of billions of people. This was made possible by advances in both hardware and software, but also by the ever-growing amount of accumulated data used for training these algorithms. This data is often high-dimensional and may sometimes range in tens of thousands of attributes (*features*). Taking into account all of these features can be computationally expensive and can result in models that are essentially useless in practice. However, not all of the

features contribute equally to the accuracy of the particular model, hence finding an appropriate subset of features is an essential preprocessing step of every machine learning model.

The traditional feature selection problem most often considers the situation where each row in the data is associated with exactly one label. This might not always be the case as many datasets have several labels associated with the same row. In other words, labels in these datasets are not mutually exclusive. Applying several labels to the same data can be useful in many fields, such as biology [7, 26], text processing [11, 18], image and video analysis [20, 29], chemistry [1], etc. The version of the problem which considers multiple labels is referred to as *Multi-label feature selection (MLFS)*.

There are two main approaches when dealing with multi-label feature selection. The first approach transforms multi-label data into single-label data and then applies some traditional feature selection techniques to the transformed dataset. The second approach works directly on a multi-label dataset, utilizing specialized metrics and local search methods to optimize the solution. In both approaches, three main types of algorithms can be distinguished: *filter*, *wrapper*, and *embedded* [17]. Filter methods calculate the statistical relevance of each attribute, which are then used to determine a good subset of features without actually calling the learning algorithm. Wrapper methods use a specific learning algorithm to determine the quality of each considered feature subset. Most commonly, these methods are based on some heuristic or metaheuristic method that calls a learning algorithm as a black box, thus exploring the search space consisting of all possible feature subsets. Finally, embedded methods select features during the learning process.

In this paper, we use a direct hybrid method for finding a satisfactory solution to the multi-label feature selection problem, combining wrapper and filter techniques. More precisely, we utilize the Variable Neighborhood Search (VNS) algorithm, originally proposed by Mladenović and Hansen (1997) [14]. VNS has since been applied to numerous optimization problems with a great deal of success.

The rest of the paper is organized as follows. In Sect. 2 we present a more concise definition of the problem, along with a suitable evaluation metrics. Section 3 is a brief overview of the existing literature. In Sect. 4 we present our method, and in Sect. 5 we discuss the experimental results obtained on several benchmark datasets. Finally, the conclusion is presented in Sect. 6.

2 Problem Description

Feature selection (FS) problem can be formulated as follows:

Definition 1. *Let us assume that we have a dataset \mathcal{D} where each row is described with a set of features $S = \{s_1, s_2, \dots, s_n\}$ and a specific machine learning model \mathcal{M} . The goal is to find a feature subset of size k ($k < n$) such that*

$$\max_{S_k \subset S} \text{accuracy}(\mathcal{M}(\mathcal{D}[S_k])) \quad (1)$$

where $\mathcal{D}[S_k]$ is the original data where only the features from S_k are present.

The function *accuracy* can be substituted with any appropriate metric. For the purpose of our study, we adopted three metrics suitable for multi-label feature selection, but a lot more metrics can be found in the paper by Tsoumakas et al. (2009) [25].

Let us assume that we have a set of test instances (x_i, Y_i) , $i = 1..n$, where Y_i is a subset of labels associated with instance x_i . Also, let us denote a predicted set of labels for each test instance as Z_i . We can then define the following metrics.

– **Classification Accuracy**

$$\text{Classification Accuracy} = \frac{1}{n} \sum_{i=1}^n I(Z_i = Y_i) \quad (2)$$

where $I(\text{true}) = 1$ and $I(\text{false}) = 0$.

– **Hamming loss**

$$\text{Hamming Loss} = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \Delta Z_i|}{L} \quad (3)$$

where Δ is the symmetric difference between two sets, and L is the number of all available labels.

– **Precision**

$$\text{Precision} = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (4)$$

3 Related Work

In this section, we give a brief overview of some multi-label feature selection methods, as well as existing single-label feature selection methods based on VNS.

There have been several studies in which the authors used VNS for single-label feature selection. Mucherino and Liberti (2013) [15] formulated the feature selection problem as a bilevel program and used VNS to optimize it. Marinaki and Marinakis (2015) [13] combined *clonal selection algorithm* with *iterated local search* and VNS for finding a good subset of features. García-Torres et al. (2015) [8] applied VNS to high-dimensional datasets, having previously used *Markov blankets* to group subsets of features. Boughaci and Alkhawaldeh (2018) [2] explored three methods: VNS, hill-climbing local search, and stochastic local search. Chen et al. (2020) [5] combined VNS with the *estimation of distribution technique* to find a solution to the feature selection problem. To the best of our knowledge, VNS has not yet been used for multi-label feature selection.

Several surveys were conducted on the topic of multi-label feature selection [10, 17, 22]. As previously stated, the transformation-based methods transform a multi-label problem into a single-label problem and then apply some of the existing feature selection methods, usually the filter methods. An example of transformation-based method is a study by Spolaôr et al. (2013) [21].

In a study by Zhang et al. (2009) [28], the principal component analysis is used to remove irrelevant and redundant features before applying a genetic algorithm (**GA**) to the remaining features to determine the best possible feature subset.

Shao et al. (2013) [19] combined mutation-based simulated annealing, genetic algorithm, and the greedy hill-climbing algorithm. The simulated annealing is used to find promising parts of the search space. The best-found solutions detected by simulated annealing are then used as the initial population for the genetic algorithm. Finally, the hill-climbing algorithm is used to further refine and improve solutions found by GA, as well as to determine the best possible subset of features.

Yu et al. (2014) [27] used a forward search strategy to obtain a relevant feature subset for each label by employing dependence maximization as a metric. In the next step, GA is used to find the globally optimal feature subset.

In the study by Lee and Kim (2015) [12], the authors used a *memetic algorithm*, combined with a local search based on the approximated mutual information.

Jungjit and Freitas (2015) [9] applied a standard genetic algorithm to the MLFS, but presented a novel fitness function based on the correlation between pairs of features, and between features and labels.

An algorithm based on *particle swarm optimization* was presented in a study by Zhang et al. (2017) [30] for MLFS. The features are encoded as a vector of real numbers, each representing the probability that a corresponding feature will be selected. Furthermore, a *local learning strategy* is utilized to improve the overall performance of the algorithm.

Dowlatshahi et al. (2017) [6] proposed a novel algorithm called *Epsilon-Greedy Swarm Optimizer*. In each iteration of the algorithm, a random particle is chosen. Next, the nearest better particle to the selected particle in the swarm is determined, after which the *epsilon-greedy method* is applied to those two particles in order to obtain the new one. If the new particle is better than the randomly selected one, it replaces the latter in the swarm. This algorithm is further hybridized with *filter-based rankers* to improve its performance.

In a study by Paniri et al. (2019) [16], the ant colony optimization algorithm is utilized for MLFS. Pheromone values are initialized by using the cosine similarity between features and labels. Furthermore, correlations between pairs of features and between features and labels are incorporated in the state transition rule.

4 Proposed Method

In this section, we will present our method based on the VNS metaheuristic.

The solution is represented as a set of selected features, where cardinality is limited by the input parameter k , which represents the number of features to be selected.

The solution quality is determined by invoking a *multi-label k -nearest neighbors* algorithm [29] on the given sets of training and test instances and predictions

are then evaluated by using one of the metrics described in Sect. 2. Multi-label KNN classifier is used as it does not require any model training other than storing the training dataset into memory. Nonetheless, any multi-label classifier can be used to evaluate subsets of features. The number of neighbors is preset to 10. The *standard scaler* is applied to all the data, which normalizes each feature individually so that its mean (μ) is equal to zero, and its standard deviation (σ) is equal to one, using the formula 5.

$$z = \frac{x - \mu}{\sigma} \quad (5)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (6)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (7)$$

Basic VNS consists of three main steps: shaking procedure, local search, and move-or-not procedure. In Algorithm 1, we present our shaking procedure. In essence, we substitute k features in the incumbent set with k other features, selected at random.

Algorithm 1. Shaking procedure

```

1: procedure SHAKE( $k$ ,  $features$ )
2:    $selected \leftarrow features$ 
3:    $available \leftarrow F \setminus features$             $\triangleright F$  is the set of all the possible features
4:   for  $i = 1$  to  $k$  do
5:      $f^- \leftarrow$  randomly choose a feature from  $selected$ 
6:      $f^+ \leftarrow$  randomly choose a feature from  $available$ 
7:      $features \leftarrow features \cup f^+$ 
8:      $features \leftarrow features \setminus f^-$ 
9:      $selected \leftarrow selected \setminus f^-$ 
10:     $available \leftarrow available \setminus f^+$ 
11:  end for
12:  return  $features$ 
13: end procedure

```

Our local search method is inspired by the procedure presented in [12]. Let us first introduce a helper function $Q(f)$, where f is a specific feature.

$$Q(f) = \sum_{y_j \in Y} I(f, y_j) - \sum_{f_j \in S_k} I(f, f_j) \quad (8)$$

where Y is the set of all possible labels, and S_k is a set of currently selected features. The function $I(f, h)$ represents mutual information, and can be calculated

as:

$$I(f, h) = H(f) + H(h) - H(f, h) \quad (9)$$

where $H(X)$ is the entropy of the variable X (Eq. 10), and $H(X, Y)$ is the joint entropy between variables X and Y (Eq. 11). In Eq. 10, $p(x)$ denotes the probability that the outcome x is observed in variable X . Likewise, $p(x, y)$ in Eq. 11 designates the probability that outcomes x and y are jointly observed in variables X and Y . We can notice that these values can be calculated just once, during the preprocessing step.

$$H(X) = - \sum_x p(x) \log p(x) \quad (10)$$

$$H(X, Y) = - \sum_x \sum_y p(x, y) \log p(x, y) \quad (11)$$

We can now define two functions: *ADD_FEATURE* (Algorithm 2) and *DEL_FEATURE* (Algorithm 3). The function *ADD_FEATURE* finds a feature f from the set of unselected features that maximizes function $Q(f)$ and adds it to the set of selected features. On the contrary, function *DEL_FEATURE* finds a feature f from the set of selected features that minimizes function $Q(f)$ and removes it from the set of selected features.

Algorithm 2. Procedure that adds a feature into the set of selected features

```

1: procedure ADD_FEATURE(features)
2:    $A \leftarrow F \setminus \text{features}$  ▷  $F$  is the set of all the possible features
3:    $f \leftarrow \arg \max_{f \in A} Q(f)$ 
4:    $\text{features} \leftarrow \text{features} \cup f$ 
5:   return features
6: end procedure

```

Algorithm 3. Procedure that removes a feature from the set of selected features

```

1: procedure DEL_FEATURE(features)
2:    $f \leftarrow \arg \min_{f \in \text{features}} Q(f)$ 
3:    $\text{features} \leftarrow \text{features} \setminus f$ 
4:   return features
5: end procedure

```

Finally, our local search method is presented in Algorithm 4. In each step of the algorithm, we first add i features to the incumbent solution (Lines 4–6) and then remove i different features from the solution (Lines 7–9). If the newly

created solution is better than the incumbent solution, it is accepted as the new incumbent solution and the search resumes from the smallest neighborhood (Line 12). Determining the quality of the solution is done by using the *EVALUATE* function, which consists of invoking the multi-label K-nearest neighbors algorithm on the testing dataset and applying one of the aforementioned metrics to the result.

Algorithm 4. Local search

```

1: procedure LOCAL_SEARCH(features, h)
2:   for  $i = 1$  to  $h$  do
3:      $features' \leftarrow features$ 
4:     for  $j = 1$  to  $i$  do
5:       ADD_FEATURE( $features'$ )
6:     end for
7:     for  $j = 1$  to  $i$  do
8:       DEL_FEATURE( $features'$ )
9:     end for
10:    if EVALUATE(features) < EVALUATE( $features'$ ) then
11:       $feature \leftarrow features'$ 
12:       $i \leftarrow 1$ 
13:    end if
14:  end for
15:  return features
16: end procedure

```

Using the same idea as in the local search procedure, we can construct a procedure that generates the initial solution. The pseudocode of that procedure is given in Algorithm 5, and it consists of k consecutive calls to the *ADD_FEATURE* procedure.

Algorithm 5. Procedure for generating the initial solution

```

1: procedure INITIAL_SOLUTION( $k$ )
2:    $features \leftarrow \emptyset$ 
3:   for  $j = 1$  to  $k$  do
4:     ADD_FEATURE(features)
5:   end for
6:   return features
7: end procedure

```

The whole proposed algorithm is presented in Algorithm 6.

Algorithm 6. Basic Variable Neighborhood Search

```

1: procedure BVNS(training_data, test_data, k, lmin, lmax, h)
2:   features  $\leftarrow$  initial_solution(k)
3:   while stopping criterion is not met do
4:     l  $\leftarrow$  lmin
5:     while l < lmax do
6:       features'  $\leftarrow$  SHAKE(features, l)
7:       features''  $\leftarrow$  LOCAL_SEARCH(features', h)
8:       if EVALUATE(features) < EVALUATE(features'') then
9:         features  $\leftarrow$  features''
10:        l  $\leftarrow$  lmin
11:      else
12:        l  $\leftarrow$  l + 1
13:      end if
14:    end while
15:  end while
16:  return features
17: end procedure

```

5 Experimental Evaluation

5.1 Experimental Settings

For the purpose of evaluating our approach, we implemented three algorithms: the multi-label ant colony optimization (**MLACO**), as presented in [16], memetic algorithm (**MA**) with the local refinement procedure based on mutual information from [12], and our own basic variable neighborhood search (**BVNS**).

Each of these algorithms has a set of parameters which have to be fine-tuned in order to achieve the best performance. For this purpose, we used the *iRace*¹ package for the R programming language with a budget of 200 tests for each algorithm. The results are presented in Table 1. The meaning of each parameter for MLACO and MA can be found in original publications. The size of the feature subset that needs to be selected is set to 10 for the purpose of testing. In general, there is no optimal number of features that would work for every dataset, as it is highly dependent on the properties of the dataset under consideration.

The experiments were conducted on a personal laptop with Intel i7-10750H CPU and 32 GB of RAM, under Ubuntu 20.04 operating system. The algorithms were implemented in Python programming language, using packages such as scikit-learn, scikit-multilearn [23], and numpy. The tests were performed on four benchmark datasets for multi-label classification, presented in Table 2. Features with continuous values in the datasets were discretized by linearly dividing the interval into 10 bins. The tests were performed with 30 repetitions to ensure the stability of the obtained results. Three different metrics were tested as the fitness function, presented in Sect. 2.

¹ <https://cran.r-project.org/web/packages/irace/index.html>.

Table 1. The optimal parameter values

MLACO	MA	BVNS
number_of_ants = 25	population_size = 15	$l_{min} = 2$
$\beta = 0.8$	v = 500	$l_{max} = 8$
$\rho = 0.1$	h = 15	h = 5
	crossover_probability = 0.5	
	mutation_probability = 0.1	

Table 2. Multi-label datasets used for experimental evaluation

Dataset	Domain	Instances	Features	Labels	Source
Birds	Audio	645	260	19	[4]
Emotions	Music	593	72	6	[24]
Scene	Image	2407	294	6	[3]
Yeast	Biology	2417	103	14	[7]

The stopping criterion for all three algorithms was the number of times the fitness function was called. This way we wanted to show which algorithm was the most successful in navigating the search space with the limited resources. The total number of fitness function calls for each algorithm execution was set to 500. In practice, this limit would be set much higher allowing algorithms to obtain much better solutions.

5.2 Obtained Results

In Table 3, we present the average accuracy over 30 independent runs for all three algorithms and all four datasets, with standard deviation presented in the parenthesis. The algorithm with the best performance for each dataset is bolded. Furthermore, we performed the Wilcoxon pair-wise statistical test between BVNS and other proposed methods, taking into account all 30 independent runs for each dataset and each pair of methods. The p-values of these tests are given in Table 6, using accuracy as the metric. Nonetheless, the results are almost identical when using the other two metrics. With a significance level of $p=0.05$ we can observe that there is a statistically significant difference between the results obtained by BVNS and MLACO for all four datasets, and the results obtained by MA statistically differ from BVNS results in two out of four cases. In cases of *Yeast* and *Scene* datasets, BVNS and MA performed rather similarly.

Similarly, the results obtained by using the Hamming loss as the fitness function are presented in Table 4. It is important to mention that in the case of Hamming loss metric the lower value corresponds to a better solution. Likewise, in Table 5 we gave the average precision for implemented method.

From these tables one can conclude that BVNS outperformed other methods for each dataset. On the other hand, we have not observed any significant difference between different fitness functions.

Table 3. Average Accuracy over 30 independent runs

Dataset	MLACO	MA	BVNS
Emotions	0.2574 (0.0107)	0.3054 (0.0066)	0.3178 (0.0109)
Birds	0.5108 (0.0039)	0.5012 (0.0045)	0.5170 (0.0044)
Yeast	0.1586 (0.0037)	0.1747 (0.0073)	0.1783 (0.0063)
Scene	0.4153 (0.0151)	0.5099 (0.0052)	0.5115 (0.0060)

Table 4. Average Hamming loss over 30 independent runs

Dataset	MLACO	MA	BVNS
Emotions	0.2524 (0.0097)	0.2195 (0.0075)	0.2138 (0.0079)
Birds	0.0490 (0.0019)	0.0484 (0.0008)	0.0472 (0.0022)
Yeast	0.2192 (0.0031)	0.2168 (0.0045)	0.2150 (0.0043)
Scene	0.1445 (0.0054)	0.1250 (0.0033)	0.1235 (0.0029)

Table 5. Average Precision over 30 independent runs

Dataset	MLACO	MA	BVNS
Emotions	0.4897 (0.0161)	0.5451 (0.0121)	0.2138 (0.0079)
Birds	0.0969 (0.0085)	0.0899 (0.0074)	0.1016 (0.0122)
Yeast	0.3516 (0.0024)	0.3523 (0.0049)	0.3523 (0.0034)
Scene	0.4104 (0.0158)	0.4843 (0.0071)	0.4891 (0.0069)

Table 6. p-values of Wilcoxon pair-wise statistical tests comparing BVNS to other proposed methods

Dataset	MLACO	MA
Emotions	0.00195	0.01141
Birds	0.0373	0.00195
Yeast	0.00195	0.09289
Scene	0.00097	0.0625

In Fig. 1, we present a boxplot for each dataset, where we compared the obtained accuracy for all three methods. Again, it is clear that BVNS outperformed other methods, but it is noteworthy that in the case of the *Scene* dataset, BVNS and MA performed quite similarly.

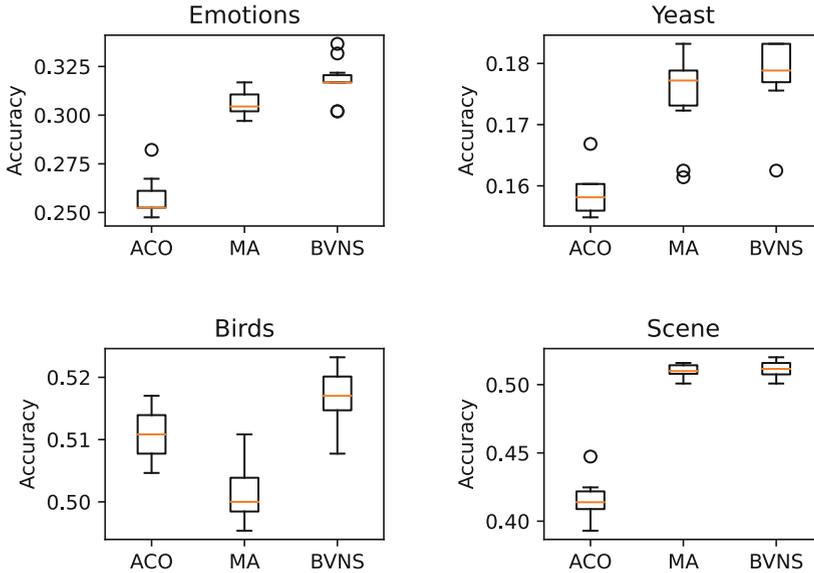


Fig. 1. Boxplots showcasing the difference in the accuracy of tested algorithms on four datasets

In Fig. 2, we present the accuracy BVNS was able to achieve with the limited number of fitness function calls. The accuracy was measured after every 50 calls. The blue line represents the average accuracy after n calls, the orange and green lines symbolize the best and worst accuracy respectively, and the shaded area denotes the interval that 80% of the results belong to. It is easy to see that the quality of the 80% of the results closely follows the average accuracy, with very little deviation, suggesting the stability of the method. While in the worst-case BVNS reported solutions that are below average, it is compensated by a relatively steady climb of the average accuracy. Given enough time, it is safe to assume that the results from all executions would converge at some point.

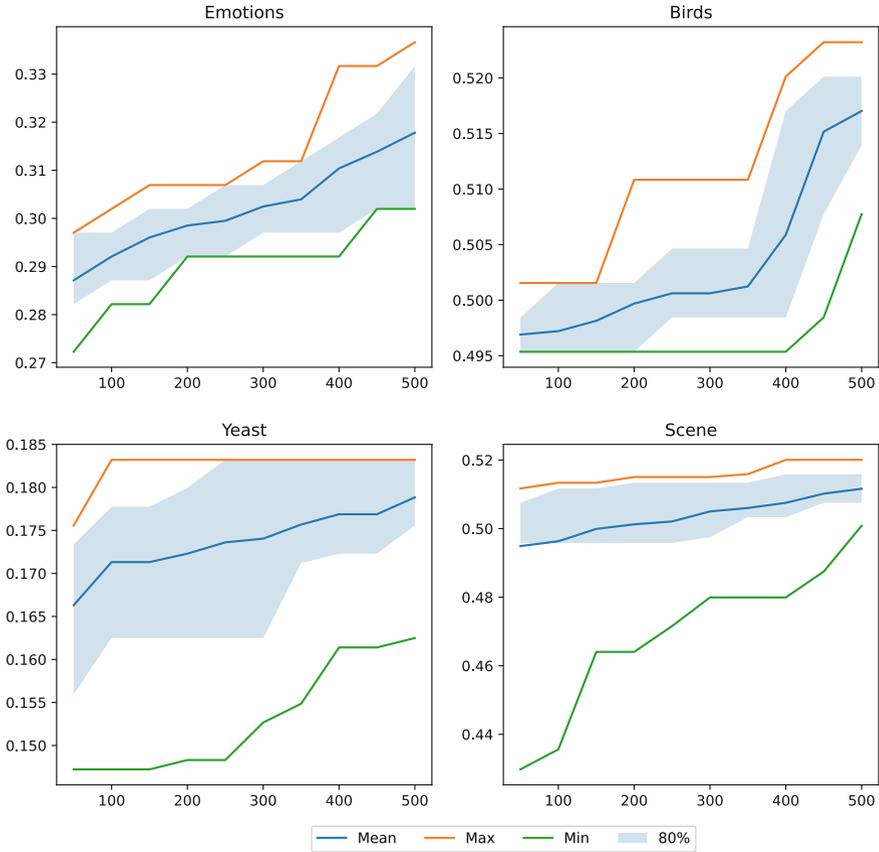


Fig. 2. Diagram displaying the obtained accuracy after every 50 calls to the fitness function (30 repetitions)

6 Conclusion

In this paper, we have proposed a Basic Variable Neighborhood Search (BVNS) algorithm for finding a good subset of features for multi-label classification. To increase the performance of this algorithm, we incorporated the local search procedure based on mutual information, as well as a procedure for generating the initial solution which uses the same idea. A multi-label k-nearest neighbors algorithm was used to evaluate the quality of proposed solutions.

We have tested our approach in comparison with two methods present in the literature: Multi-Label Ant Colony Optimization (MLACO) and Memetic Algorithm (MA). Four different benchmark datasets for this problem were used. Based on conducted experiments, BVNS was able to outperform MLACO and MA, with the limited numbers of calls to the fitness function. Three different fitness functions were tested: classification accuracy, Hamming loss, and precision,

but we concluded that the choice of these measures did not affect the overall performance of the algorithm.

In future work, we will approach this problem as a multi-objective optimization problem, in which we need to maximize the accuracy of the model, while simultaneously minimizing the number of features.

Acknowledgements. This work was supported by the Serbian Ministry of Education, Science and Technological Development, Agreement No. 451-03-9/2021-14/200029 and by the Science Fund of the Republic of Serbia, Grant AI4TrustBC: Advanced Artificial Intelligence Techniques for Analysis and Design of System Components Based on Trustworthy Blockchain Technology.

References

1. Blockeel, H., Džeroski, S., Grbović, J.: Simultaneous prediction of multiple chemical parameters of river water quality with TILDE. In: Zytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 32–40. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48247-5_4
2. Boughaci, D., Alkhalaf, A.A.S.: Three local search-based methods for feature selection in credit scoring. *Vietnam J. Comput. Sci.* **5**(2), 107–121 (2018)
3. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recogn.* **37**(9), 1757–1771 (2004)
4. Briggs, F., et al.: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In: *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–8 (2013)
5. Chen, W., Li, Z., Guo, J.: A VNS-EDA algorithm-based feature selection for credit risk classification. In: *Mathematical Problems in Engineering 2020* (2020)
6. Dowlatshahi, M.B., Derhami, V., Nezamabadi-pour, H.: Ensemble of filter-based rankers to guide an epsilon-greedy swarm optimizer for high-dimensional feature subset selection. *Information* **8**(4), 152 (2017)
7. Elisseeff, A., Weston, J.: A Kernel method for multi-labelled classification. In: *Advances in Neural Information Processing Systems*, vol. 14 (2001)
8. García-Torres, M., Gómez-Vela, F., Melián-Batista, B., Moreno-Vega, J.M.: High-dimensional feature selection via feature grouping: a variable neighborhood search approach. *Inf. Sci.* **326**, 102–118 (2016)
9. Jungjit, S., Freitas, A.A.: A new genetic algorithm for multi-label correlation-based feature selection. In: *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 285–290 (2015)
10. Kashif, S., Nezamabadi-pour, H., Nikpour, B.: Multilabel feature selection: a comprehensive review and guiding experiments. *Wiley Interdisc. Rev. Data Mining Knowl. Discov.* **8**(2), e1240 (2018)
11. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: *Proceedings of the ECML/PKDD*, vol. 18, p. 5. Citeseer (2008)
12. Lee, J., Kim, D.W.: Memetic feature selection algorithm for multi-label classification. *Inf. Sci.* **293**, 80–96 (2015)
13. Marinaki, M., Marinakis, Y.: A hybridization of clonal selection algorithm with iterated local search and variable neighborhood search for the feature selection problem. *Memetic Comput.* **7**(3), 181–201 (2015). <https://doi.org/10.1007/s12293-015-0161-2>

14. Mladenović, N., Hansen, P.: Variable neighborhood search. *Comput. Oper. Res.* **24**(11), 1097–1100 (1997)
15. Mucherino, A., Liberti, L.: A VNS-based heuristic for feature selection in data mining. In: Talbi, E.G. (eds.) *Hybrid Metaheuristics. Studies in Computational Intelligence*, vol. 434, pp. 353–368. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30671-6_13
16. Paniri, M., Dowlatshahi, M.B., Nezamabadi-Pour, H.: MLACO: a multi-label feature selection algorithm based on ant colony optimization. *Knowl.-Based Syst.* **192**, 105285 (2020)
17. Pereira, R.B., Plastino, A., Zadrozny, B., Merschmann, L.H.C.: Categorizing feature selection methods for multi-label classification. *Artif. Intell. Rev.* **49**(1), 57–78 (2016). <https://doi.org/10.1007/s10462-016-9516-4>
18. Sajnani, H., Saini, V., Kumar, K., Gabrielova, E., Choudary, P., Lopes, C.: Classifying yelp reviews into relevant categories. University of California Press, Berkeley, CA USA, Technical report, Mondego Group (2012)
19. Shao, H., Li, G., Liu, G., Wang, Y.: Symptom selection for multi-label data of inquiry diagnosis in traditional Chinese medicine. *Sci. China Inf. Sci.* **56**(5), 1–13 (2013)
20. Snoek, C.G., Worring, M., Van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: *Proceedings of the 14th ACM international Conference on Multimedia*, pp. 421–430 (2006)
21. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: A comparison of multi-label feature selection methods using the problem transformation approach. *Electron. Notes Theor. Comput. Sci.* **292**, 135–151 (2013)
22. Spolaôr, N., Monard, M.C., Tsoumakas, G., Lee, H.D.: A systematic review of multi-label feature selection and a new method based on label construction. *Neurocomputing* **180**, 3–15 (2016)
23. Szymański, P., Kajdanowicz, T.: A Scikit-based Python environment for performing multi-label classification. *ArXiv e-prints*, February 2017
24. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD 2008)*, vol. 21, pp. 53–59 (2008)
25. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09823-4_34
26. Xu, J., Liu, J., Yin, J., Sun, C.: A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. *Knowl.-Based Syst.* **98**, 172–184 (2016)
27. Yu, Y., Wang, Y.: Feature selection for multi-label learning using mutual information and GA. In: Miao, D., Pedrycz, W., Ślęzak, D., Peters, G., Hu, Q., Wang, R. (eds.) *RSKT 2014. LNCS (LNAI)*, vol. 8818, pp. 454–463. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11740-9_42
28. Zhang, M.L., Peña, J.M., Robles, V.: Feature selection for multi-label Naive Bayes classification. *Inf. Sci.* **179**(19), 3218–3229 (2009)
29. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn.* **40**(7), 2038–2048 (2007)
30. Zhang, Y., Gong, D.W., Sun, X.Y., Guo, Y.N.: A PSO-based multi-objective multi-label feature selection method in classification. *Sci. Rep.* **7**(1), 1–12 (2017)



Dispersion Problem Under Capacity and Cost Constraints: Multiple Neighborhood Tabu Search

Nenad Mladenović¹ , Raca Todosijević² , and Dragan Urošević³  

¹ Department of Industrial Engineering and Research Center on Digital Supply Chain and Operations Management, Khalifa University, PO Box 127788, Abu Dhabi, United Arab Emirates

`nenad.mladenovic@ku.ac.ae`

² Polytechnic University of Hauts-de-France, Cedex 9, Valenciennes, France

³ Mathematical Institute, University of Belgrade, Knez Mihailova 36, Belgrade, Serbia

`draganu@mi.sanu.ac.rs`

Abstract. Diversity and dispersion problems consists of selecting a subset of elements from a given set so that their diversity is maximized. The one of most recently proposed variant is the MaxMin dispersion problem with capacity and cost constraints. This variant usually called the generalized dispersion problem. In this paper we propose variant of tabu search based on multiple neighborhoods to solve large-size instances. Extensive numerical computational experiments are performed to compare our tabu search metaheuristic with the state-of-art heuristic. Results on public benchmark instances show the superiority of our proposal with respect to the previous algorithms.

Keywords: Metaheuristics · diversity maximization · dispersion · tabu search

1 Introduction

In the last thirty years, the study of diversity has very popular in Operations Research and Computer Science. For large period, it was mainly devoted to continuous models. Discrete diversity maximization was introduced by Kuby [5] in a paper that was the origin of a today very large area of location problems (Martí et al., [8]).

In its simplest form, the problem of maximizing diversity or dispersion consists of selecting a subset of elements from a given set in such a way that the

This work is also partially supported by the Serbian Ministry of Education, Science and Technological Development, Agreement No. 451-03-9/2021-14/200029 and by the Science Fund of the Republic of Serbia, Grant AI4TrustBC: Advanced Artificial Intelligence Techniques for Analysis and Design of System Components Based on Trustworthy Blockchain Technology.

distance among the selected elements is maximized. We may think on the standard distance definition based on the Euclidean formula. But many applications may require non-Euclidean geometries, such as those induced by affinities relationships expressing a relative degree of attraction between the elements. Typical examples are architectural space planning and analysis of social networks (Glover et al., [4]).

Over the last few years, different mathematical expressions have been proposed as measure of diversity, dispersion, or even equity. The sum of the distances among all pairs of the selected elements is probably the most well-known model (the MaxSum model). But, we can also use the minimum among these distances (the MaxMin model) as an efficient way to model it. Parreño et al. [11] perform an empirical comparison of the different diversity models, and conclude that maximizing the minimum of the distances among the selected elements is the best way for measuring the representativeness, which many location applications require when maximizing diversity.

Figure 1 shows in the left part the optimal solution of the MaxSum model on an Euclidean instance with 50 elements from which we select 10 of them. In the right part of this figure, we can see the optimal solution of the MaxMin model for the same instance (see [11]). We can conclude that the MaxMin model does not avoid to select points in the central region of the plane, as the MaxSum model does. Also, the MaxMin model selects almost equidistant points deployed all over the plane. If we consider the points in the plane as potential locations to set facilities over a given territory, the MaxMin solution in the right, presents a better distribution of the selected elements than the one in the left, since it covers the territory (with disperse points) in a better way.

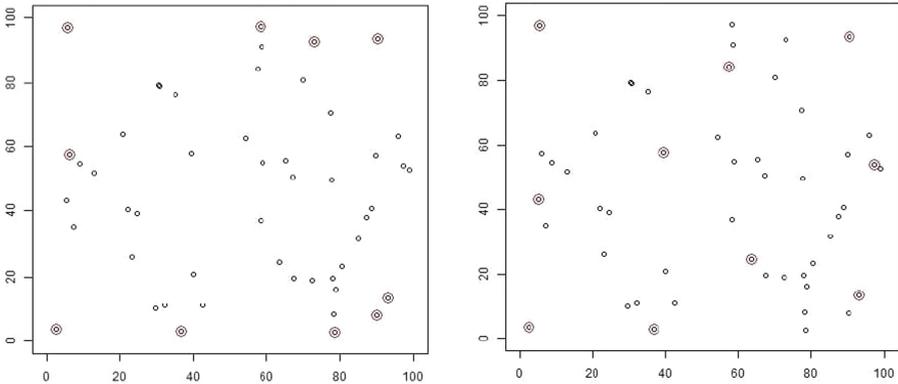


Fig. 1. Comparison of optimal solutions for MaxSum and MaxMin Dispersion Problem on instance with $n = 50$ elements (see also [11]).

In this paper we consider the generalized dispersion problem (GDP), which is based on the MaxMin diversity model, but also incorporates capacity and cost

constraints. This model was introduced for the first by Rosenkrantz et al. [13]. In short, the authors considered the allocation of the same-type of facilities, such as shops, but in such way to avoid their closeness. The capacity constraints introduce the maximal number of customers (patients, students) that corresponding facility may serve and ensures that total capacity of all allocated facilities is not fewer than given lower limit. The cost constraint allows to limit total cost (which is real situation) of setting up all facilities.

Despite of their existence, the side constraints, such as capacity and cost constraints, have been mostly ignored in the discrete diversity literature. There are only two papers considering these two types of constraints. On the other hand Martí et al. [8] found more than 50 papers considering the unconstrained discrete diversity (but including the standard constraint regarding the number of facilities to be selected). In this paper we study this more realistic \mathcal{NP} -hard problem involving capacity constraint as well as cost constraint.

The paper is organized as follows. In the Sect. 2 Mathematical model of the studied problem is presented. In the Sect. 3 overview of existing heuristics is given. The developed Multiple neighborhood tabu search for generalized dispersion problem is described in Sect. 4. Computational results are presented in Sect. 5.

2 Mathematical Model

Given a set of n potential facilities V ($V = \{1, 2, \dots, n\}$) connected by edges (links) each of which has positive length (let us denote by d_{ij} the length of edge connecting facilities i and j). Each potential facility has capacity (let c_i is capacity of the facility i) and cost of establishing (let us denote by a_i cost of establishing facility at location i). Let B be the minimum total capacity (so called service level) and let K be maximum allowed budget (maximal total cost of establishing all facilities). It is necessary select subset $P \subset V$ satisfying capacity and cost constraints such that minimum distance among elements of P is maximized.

The mathematical model for generalized diversity problem (GDP for short) is based on binary decision variables x_i ($i = 1, 2, \dots, n$) which take value 1 if facility i is selected and value 0 if facility i is not selected into subset P . Under these assumptions the GDP can be defined in the following way:

$$\max_x \min_{i,j \in V, i \neq j} d_{ij} x_i x_j + (2 - x_i - x_j)M$$

such that

$$\sum_{i=1}^n c_i x_i \geq B \tag{1}$$

$$\sum_{i=1}^n a_i x_i \leq K. \tag{2}$$

In this model, M is arbitrary big number (for example $M = \max_{i,j \in V} d_{ij}$).

In the above model the constraint 1 ensures that total capacity is not lower than minimum required capacity B . The constraint 2 ensures that total cost of establishing all facilities is not greater than allowed budget K . Above mathematical model can be rewritten as follows:

$$\max_x z$$

such that

$$\sum_{i=1}^n c_i x_i \geq B \quad (3)$$

$$\sum_{i=1}^n a_i x_i \leq K, \quad (4)$$

$$d_{ij} x_i x_j + (2 - x_i - x_j) M \geq z \quad 1 \leq i < j \leq n. \quad (5)$$

3 Previous Heuristics

Rosenkrantz et al. [13] propose a binary search based method to solve a variant of the GDP also known as the capacitated dispersion problem (CDP for short). This problem involves distances and capacity in the same way than the GDP, but does not consider the cost constraint. In short, the greedy algorithm proposed by Rosenkrantz et al. [13] performs a binary search over the non-zero distances in order to find a set of elements satisfying the capacity constraint with minimum distance as large as possible. Martínez-Gavara et al. [9] modify this algorithm to algorithm solving the GDP, and named it **TI_Ad**. Specifically, Martínez-Gavara et al. [9] consider the ratio between the capacity and cost of each element and sort elements from V according to the ratio. **TI_Ad** then selects the elements from this list in decreasing order, and checks both, capacity and cost constraint, to validate the feasibility of the selected set of elements, and stops when the capacity level is reached.

Martínez-Gavara et al. [9] applied two metaheuristics to obtain high quality solutions of the GDP: a greedy randomized adaptive search procedure, **GRASP**, and a long-term tabu search, **TS**. As it is known, **GRASP** is a multi-start method that in each iteration creates a new solution, and improves it by applying a local search to obtain a local-optimum. We will describe in a short the main features of the **GRASP** algorithm. The construction phase starts by creating the so-called candidate list (CL), which consists of all unselected elements (locations) that can be inserted in the solution P without violation the upper bound on the budget K . Then, the next element to be added in the solution is selected at random from the restricted candidate list (RCL) containing the good candidates. To obtain RCL, the value of greedy function (denoted by \tilde{g}) calculates for each element of CL, and the elements having greater values of greedy function inserts into RCL. The greedy function collects in a single expression all the three criteria involved in this problem: distance, cost, and capacity (the exact formula can be seen in [9]).

Once a feasible solution P is constructed, the algorithm explores the neighborhood of P in order to obtain a local optimum. This neighborhood is based on swap moves. Let d^* be the objective function value of solution P , and **pivotalist** is the set of all elements belonging to the solution P with minimum distance equal to d^* . Then, at each iteration, the algorithm evaluates the objective function value of each solution obtained by exchange between a randomly selected site i from the **pivotalist** with a element $j \in V \setminus P$, with distance to the selected elements (except the element i), larger than d^* . The swap move is applied if it is feasible and it improves the current solution. The local search based on this neighborhood performs while there is improving move.

The authors also propose a long-term tabu search for GDP. This is a memory-based methodology that explores efficiently the solution space [3]. The proposed algorithm, **TS**, starts by constructing an initial solution in the same way as the **GRASP** constructive phase, but without selecting the next element at random way. After that, the algorithm explores the same neighborhood of the current solution as the previously described **GRASP**. But in this method, the algorithm always performs a move even if it does not improve the solution. After executing an exchange between a site $i \in \text{pivotalist}$ and a site $j \in V \setminus P$, the tabu structure records as *tabu-active* the site i , i.e., the element that leaves the solution. The tabu status of a element i remains active for a specific number of iteration, and during these iterations, it cannot be selected for inclusion in the solution. Also, the long term phase in **TS** is diversifies the search and forces exploring unvisited areas of the solution space, by ignoring frequently used elements and forcing the non-frequently used elements which provide high quality solutions.

The computational experiments performed in [9] showed that the **TI_Ad** heuristic algorithm determines relatively good solutions for small size instances with very short running times (less than 1s). But the **TI_Ad** cannot compete with **GRASP** and **TS** in terms of the quality of the solutions, especially for large size instances. Moreover, authors state that **GRASP** has best performances, with respect to the running times.

4 Tabu Search for Generalized Dispersion Problem

4.1 Solution Representation

To store a solution and allow an efficient moving from one solution to another the following solution representation is proposed (see also [10]). The potential facility locations are stored in the vector (or array) $x = \{x_1, x_2, x_3, \dots, x_{m-1}, x_m, x_{m+1}, \dots, x_n\}$ so that the first m elements represent solution P_x , i.e., the subset of selected locations, while the remaining $n - m$ elements are the non-selected locations. This implies that pair (x, m) determines the solution P_x ($P_x = \{x_1, x_2, \dots, x_m\}$) and we will use this two notations interchangeably.

Beside the vector x (and the number of selected locations m) we all time update four auxiliary data structures:

- **array** $dm1$ where each element $dm1_i$ contains the minimum distance of the element $i \in V$ to the elements in P_x , i.e., $dm1_i = \min\{d_{i,x_j} | x_j \in P_x, x_j \neq i\}$;
- **array** $c1$ where each element $c1_i$ corresponds to the element in P_x , that is the closest to element $i \in V$. More precisely, $c1_i = \operatorname{argmin}\{d_{i,x_j} | x_j \in P_x, x_j \neq i\}$, or $dm1_i = d_{i,c1_i}$;
- **array** $dm2$ where each element $dm2_i$ contains the second minimum distance of the element $i \in V$ to the elements in P_x , i.e., $dm2_i = \min\{d_{i,x_j} | x_j \in P_x, x_j \neq i, x_j \neq c1_i\}$;
- **array** $c2$ such that each element $c2_i$ corresponds to the element in P_x , that is the second closest to element $i \in V$ and which may be determined as $c2_i = \operatorname{argmin}\{d_{i,x_j} | x_j \in P_x, x_j \neq i, x_j \neq c1_i\}$.

The array $dm1$ enables us to quickly calculate the objective function value of a solution P_x as

$$f(P_x) = \min\{dm1_{x_j} | j = 1, 2, 3, \dots, m\}.$$

On the other hand, arrays $c1$ and $c2$ will be used to speed up the exploring neighborhoods and the local search (i.e. tabu search). Beside the above data structures, we also calculate the total number of elements in the current solution that yield the objective function value, i.e. the number of elements $x_j \in P_x$ satisfying condition $dm1_{x_j} = f(P_x)$. Such elements are called *critical elements* and constitute set C_x defined as $C_x = \{x_i \in P_x | dm1_{x_i} = f(P_x)\}$. The cardinality of set C_x is denoted with $nmin_x$ (i.e., $nmin_x = |C_x|$). In case that there are a more locations at the same smallest distances, values for $c1$ and $c2$ are fewest location labels among all closest locations.

According introduced notation we propose new criteria for comparing solutions. So the solution $P_{x'}$ will be considered as *better* than the solution $P_{x''}$ if one of the following conditions is satisfied:

- $f(P_{x'}) > f(P_{x''})$ or
- $f(P_{x'}) = f(P_{x''})$ and $nmin_{x'} = |C_{x'}| < |C_{x''}| = nmin_{x''}$.

In other words, the solution x' is better than the solution x'' either if its objective function value (i.e. minimal distance between elements belonging to the solution $P_{x'}$) is strictly greater or in the case of the same minimal distance between elements if solution x' has fewer number of critical elements.

4.2 Neighborhoods

In this section, we describe neighborhoods that are explored within our heuristic. In total, three neighborhood structures have been considered (swap, 2-out-1-in, and 1-out-2-in neighborhoods). Comparing to the previous papers, this paper proposes two new neighborhood structures (2-out-1-in and 1-out-2-in) that have not been considered before for solving the GDP. In addition, the previous papers use relatively inefficient procedure for evaluating a neighboring solution (did not use any additional data structures). In this paper we explain how a neighboring solution may be efficiently evaluated by using the auxiliary data structures.

Swap Neighborhood. The swap neighborhood of a solution P_x (stored as (x, m)) is defined as a set of all solutions that may be obtained by a feasible swap move that replaces one element in P_x by one element not belonging P_x . A swap move is considered feasible if its execution yields a solution that satisfies the minimum capacity requirement and maximum cost requirement.

In order to efficiently calculate the objective function value of the solution obtained by swapping elements $out \in P_x$ and $in \notin P_x$, we will use arrays $c1$ and $c2$. Let us denote by $dm1'_k$ the minimal distance for element $k \in V$, after executing the swap move involving elements in and out , then we have:

$$dm1'_k = \begin{cases} d_{k,in}, & \text{if } d_{k,in} \leq d_{k,c1_k} \\ dm1_k, & \text{if } out \neq c1_k \text{ and } d_{k,in} > d_{k,c1_k} \\ d_{k,in}, & \text{if } out = c1_k \text{ and } d_{k,in} < dm2_k \\ dm2_k, & \text{otherwise.} \end{cases}$$

or equivalently

$$dm1'_k = \begin{cases} \min\{dm1_k, d_{k,in}\}, & \text{if } out \neq c1_k, \\ \min\{dm2_k, d_{k,in}\}, & \text{if } out = c1_k. \end{cases}$$

These formulas imply that resulting $dm1$ value for any element (after performing a swap move) can be calculated in $O(1)$ time complexity. On the other hand, to evaluate the objective function value of a resulting solution after a swap move it is necessary to calculate new $dm1$ values only for elements participating in the solution and for the element that is inserted by the swap move. Because of that, the objective function value (as well as the number of critical vertices) of the solution obtained by performing a swap move may be calculated in the time complexity $O(m) = O(n)$. On the other hand, the number of different solutions belonging to the swap neighborhood is $m \cdot (n - m) = O(m \times n)$. So the worst case time complexity of exploring swap neighborhood is $O(m^2 \times n)$.

2-out-1-in Neighborhood. 2-out-1-in neighborhood of a solution P_x contains all solutions that may be obtained by applying a feasible 2-out-1-in move on a given solution P_x . A feasible 2-out-1-in move removes two elements out_1 and out_2 from solution P_x and inserts one element in not belonging P_x , while respecting the minimum capacity requirement and the maximum cost constraint. To efficiently calculate the objective function value after performing a 2-out-1-in move, the resulting array $dm1'$ must be calculated using the following formulas:

- $\min\{d_{k,in}, dm1_k\}$, if $c1_k \notin \{out_1, out_2\}$
- $\min\{d_{k,in}, dm2_k\}$, if $c1_k \in \{out_1, out_2\}$ and $c2_k \notin \{out_1, out_2\}$
- $\min\{d_{k,in}\} \cup \{d_{k,x_i} | i = 1, 2, \dots, m, x_i \neq out_1, x_i \neq out_2\}$, otherwise.

It is easy to conclude that calculating of $dm1_k$ (for arbitrary element k) has worst time complexity $O(m)$ (it is complexity in third case of formula for calculating $dm1'$). In order to calculate objective value of the new solution (obtained

after performing a 2-out-1-in move) we must calculate value of $dm1'$ for all elements belonging to the new solution. Because of that the worst time complexity of evaluating the objective value of the neighbouring solution is $O(m^2)$. Taking into account that cardinality of the 2-out-1-in neighborhood is $O(m^2(n-m))$ we can conclude that worst case complexity of exploring the 2-out-1-in neighborhood is $O(m^4(n-m)) = O(m^4n)$. But it is very pessimistic scenario, because the time complexity of evaluating the most of neighbouring solution is not $O(m^2)$ but $O(m)$ (since the third case of the formula for calculating $dm1'$ is very rarely applied).

1-out-2-in Neighborhood. The last considered neighborhood is based on feasible 1-out-2-in moves. A feasible 1-out-2-in move removes one element *out* from a given solution and inserts two other elements in_1 and in_2 , not belonging to the solution, while respecting the minimum capacity requirement and maximum cost requirement. Each 1-out-2-in move can be considered as composition of two moves: one swap move (swaps element *out* going out and one of elements (for example in_1) going in) and one insert move (inserting the second element in_2 going in). Considering 1-out-2-in move as composition of two moves allows us to efficiently calculate values $dm1$ (for all elements belonging to the solution) after performing both moves. We already seen how it is possible calculate values $dm1$ after performing swap move (let us denote this values with $dm1'$). If we denote by $dm1''$ values of $dm1$ after inserting element in_2 into the solution, then we calculate $dm1''$ by using the following rules:

$$dm1''_k = \begin{cases} dm1'_k, & \text{if } dm1'_k \leq d_{k,in_2} \\ d_{k,in_2}, & \text{if } d_{k,in_2} < dm1'_k, \end{cases}$$

or $dm1''_k = \min\{dm1'_k, d_{k,in_2}\}$.

Because of that, the time complexity of evaluating value $dm1$ for one element is $O(1)$, and time complexity of evaluating objective value for the neighbouring solution is $O(m)$. On the other side, the cardinality of complete neighborhood is $O(m(n-m)^2) = O(mn^2)$, and the time complexity of exploring the 1-out-2-in neighborhood is $O(m^2n^2)$. But the number of infeasible moves in 1-out-2-in neighborhood can be very large and in this case time complexity of exploring 1-out-2-in neighborhood reduces (because it is not necessary evaluate objective function values for such solutions). However, it is showed that exploring the complete 1-out-2-in neighborhood is very time consuming. Because of that we decided to reduce this neighborhood by considering only the moves removing some of the critical elements. We called such defined neighborhood *Restricted 1-out-2-in neighborhood*.

4.3 Multiple Neighborhood Tabu Search for Generalized Dispersion Problem

Taking into account that three possible neighborhoods are defined we decided to implement multiple neighborhood tabu search in the following way. For beginning we define two tabu lists:

- tabu list consisting of the elements belonging to solution which are included into the solution in any of the last nt_{in} (setting the value of nt_{in} will be described later) iterations;
- tabu list consisting of the elements not belonging to solution which are excluded from the solution in any of the last nt_{out} (setting the value of nt_{out} will be described later) iterations.

One iteration of the tabu search consists of the following steps:

- At first we explore complete Swap neighborhood in order to determine the best non-tabu swap move m_1 (move involving two elements not belonging to the previously described tabu lists) as well as the best swap move m_2 (including also all the tabu swap moves). If the solution obtained by applying move m_2 is better than the best found after the last restart, then we apply such move and continue the tabu search from the obtained solution (go to the next iteration). Otherwise, if the solution obtained by applying move m_1 is better than the current solution, then we apply this move and continue the tabu search from the obtained solution (go to the next iteration).
- If the better solution in the Swap neighborhood is not found, then we explore complete 2-out-1-in neighborhood in order to determine the best non-tabu 2-out-1-in move m_3 (move involving three elements not belonging to the previously described tabu lists) as well as the best 2-out-1-in move m_4 . If the solution obtained by applying move m_4 is better than the best found after the last restart, then we apply such move and continue the tabu search from the obtained solution (go to the next iteration). Otherwise, if the solution obtained by applying move m_3 is better than the current solution, then we apply this move and continue the tabu search from the obtained solution (go to the next iteration).
- If the better solution in the 2-out-1-in neighborhood is not found, then we explore the restricted 2-out-1-in neighborhood in order to determine the best non-tabu 1-out-2-in move m_5 (move involving three elements not belonging to the previously described tabu lists) as well as the best 1-out-2-in move m_6 . If the solution obtained by applying move m_6 is better than the best found after the last restart, then we apply such move and continue the tabu search from the obtained solution (go to the next iteration). Otherwise, if the solution obtained by applying move m_5 is better than the current solution, then we apply this move and continue the tabu search from the obtained solution (go to the next iteration).
- Finally, if there is not improving move in all three neighborhoods, then we apply best of the three non-tabu moves m_1 , m_3 and m_5 (the move giving the best solution).

Tabu Lists Length. Lengths of tabu lists are not fixed. Length of each tabu list changes according to procedure previously proposed by Galinier *et al.* [2]. Complete Tabu search algorithm consists of sequence of iterations. Iterations are numbered by positive integer numbers starting from 1 (after each restart of tabu search, the numeration of iterations starts with number 1). Tabu lists lengths are change in iterations b_i ($i = 0, 1, 2, \dots$), in the following way: length of tabu list in iterations from interval $[b_i, b_{i+1}]$ is equal $a_i + rand(2)$, where

$$a_i = \left\lfloor \frac{c_i \bmod 15}{8} T_{\max} \right\rfloor.$$

In above formula:

- c is array with 15 elements as follows: 1, 2, 1, 4, 1, 2, 8, 1, 2, 1, 4, 1, 2, 1;
- T_{\max} is maximal allowed length of corresponding tabu list; and
- $rand(n)$ represents random number between 0 and n .

Sequence b_i is defined as follows:

$$b_0 = 1 \quad b_{i+1} = b_i + 5 \times a_i.$$

In other words, length of all tabu lists change periodically (shortest period is 15) and can take one of the following four values (with “small noise” not greater than 2):

$$\left\lfloor 1 \times \frac{T_{\max}}{8} \right\rfloor \quad \left\lfloor 2 \times \frac{T_{\max}}{8} \right\rfloor \quad \left\lfloor 4 \times \frac{T_{\max}}{8} \right\rfloor \quad \left\lfloor 8 \times \frac{T_{\max}}{8} \right\rfloor.$$

Number 5 in formula for calculating b_{i+1} is also proposed in [2]. Value T_{\max} is selected after detailed experimentation and set by the following formulas

- $T_{\max} = m'_{best}/2$, for the list of elements currently belonging to the solution and
- $T_{\max} = (n - m'_{best})/2$, for the list of elements currently not belonging to the solution.

In above formulas m'_{best} is number of elements in subset P'_{best} representing best solution obtained during executing of the tabu search.

Proposed algorithm restarts by calculating new initial solution and performing the tabu (local) search from the new solution. The restarts executes after performing $4 \cdot n$ iterations without improving best solution found since the last restart. The value (limit) $4 \cdot n$ was selected after long time experiments.

5 Computational Results

5.1 Instances

The benchmark set of instances was created (extracted) from the MDPLIB set. The MDPLIB set (Martí and Duarte [6]) was created from several data sets previously employed in different studies on diversity problems [1, 12, 14]. The set consists of three subsets:

- GKD: this set was originally proposed by Glover and Laguna [4] for small-size instances, and it was extended for medium-size and large-size instances in [1] and [7], respectively. Martínez–Gavara et al. [9] select 10 instances of size 50, 10 instances of size 150, and 10 instances of size 500.
- MDG: this data set was proposed in [1] and it consists of 100 matrices with real numbers randomly selected from a uniform distribution. Martínez-Gavara et al. [9] select 10 of this set of size 500.
- SOM: this data set was created by Martí et al. [7] for the maximum diversity problem, where the objective function is the sum of the distances. The matrices of this set are generated with random numbers of an integer uniform distribution between 0 and 9. Martínez-Gavara et al. [9] select 10 of them of size 50.

For each of these 50 instances, the capacities and costs were generated as real numbers with a Uniform distribution. Specifically, the capacity c_i of a site $i \in V$ was generated by a $U(1, 1000)$, the fix cost a_i was generated from its capacity c_i by a $U(c_i/2, 2c_i)$. The minimum capacity B was computed as the sum of all capacities multiplied by a factor φ_b of 0.2 or 0.3, and the maximum budget was computed as sum of all costs multiplied by a factor φ_k of 0.2 or 0.3. In this way a set of 200 instances was generated.

5.2 Detailed Results

Our Tabu search is implemented in C++ programming language. All experiments are performed on a machine equipped with Intel(R) Core(TM) i5-3470 CPU 3.20 GHz with 16 GB memory and Linux Operating System.

Due to lack of space we present detailed results on GKD instances with $n = 150$ elements (Table 1), GKD instances with $n = 500$ elements (Table 2) and MDG instances (Table 3).

Format of all tables is same. The first column contains name of corresponding instance. Columns 2 and 3 contains the objective value for solution obtained by using the Tabu Search proposed by Martínez–Gavara et al. [9], as well as running time until reaching the corresponding solution. Columns 4 and 5 contains the objective value for the solution obtained by using GRASP proposed by Martínez–Gavara et al. [9], as well as running time until reaching the corresponding solution. Columns 6–9 contains cumulative results obtained by 30 executions of our Tabu search: the objective value for the best solution, average of objective values, the objective value of the worst solution and average execution time. All results obtained by GRASP and TS (Martínez et al., [9]) are kindly provided by Ana Martínez-Gavara.

Table 1. Detailed results on GKD instances with $n = 150$ elements.

Instance	TS-M		GRASP		TS-N			
	Obj.	Time	Obj.	Time	Best	Avg.	Worst	Time
GKD-b.41_n150_b02_m15_k02.txt	156.50	300.00	163.40	300.12	166.50	166.50	166.50	0.08
GKD-b.41_n150_b02_m15_k03.txt	161.80	300.01	163.40	300.01	166.50	166.50	166.50	0.08
GKD-b.41_n150_b03_m15_k02.txt	132.60	300.00	132.60	300.00	135.80	135.80	135.80	0.35
GKD-b.41_n150_b03_m15_k03.txt	151.30	300.00	152.10	300.03	154.90	154.90	154.90	0.46
GKD-b.42_n150_b02_m15_k02.txt	77.80	300.00	81.50	300.02	82.70	82.70	82.70	0.20
GKD-b.42_n150_b02_m15_k03.txt	78.80	300.01	82.20	300.00	83.30	83.30	83.30	0.07
GKD-b.42_n150_b03_m15_k02.txt	0.00	300.00	0.00	300.00	52.00	51.35	50.10	19.97
GKD-b.42_n150_b03_m15_k03.txt	65.30	300.00	70.00	300.22	70.70	70.70	70.70	8.77
GKD-b.43_n150_b02_m15_k02.txt	59.20	300.00	62.60	300.05	65.80	65.80	65.80	8.83
GKD-b.43_n150_b02_m15_k03.txt	56.60	300.01	63.80	300.07	66.00	66.00	66.00	0.51
GKD-b.43_n150_b03_m15_k02.txt	37.50	300.00	37.50	300.00	40.70	40.70	40.70	0.21
GKD-b.43_n150_b03_m15_k03.txt	47.70	300.00	51.60	300.11	54.30	54.30	54.30	6.71
GKD-b.44_n150_b02_m15_k02.txt	90.40	300.00	100.90	300.01	102.70	102.70	102.70	2.19
GKD-b.44_n150_b02_m15_k03.txt	93.90	300.01	102.70	300.06	102.90	102.90	102.90	0.08
GKD-b.44_n150_b03_m15_k02.txt	0.00	300.00	0.00	300.00	72.80	72.45	72.30	15.41
GKD-b.44_n150_b03_m15_k03.txt	84.20	300.00	85.70	300.04	89.30	89.30	89.30	1.06
GKD-b.45_n150_b02_m15_k02.txt	103.10	300.00	107.70	300.08	110.90	110.90	110.90	0.99
GKD-b.45_n150_b02_m15_k03.txt	104.10	300.00	107.60	300.11	110.90	110.90	110.90	0.41
GKD-b.45_n150_b03_m15_k02.txt	0.00	300.00	0.00	300.00	79.20	79.20	79.20	0.11
GKD-b.45_n150_b03_m15_k03.txt	80.70	300.00	93.30	300.05	97.40	97.40	97.40	7.49
GKD-b.46_n150_b02_m45_k02.txt	118.00	300.00	121.00	300.09	124.80	124.80	124.80	0.84
GKD-b.46_n150_b02_m45_k03.txt	118.90	300.01	121.10	300.10	124.80	124.80	124.80	0.51
GKD-b.46_n150_b03_m45_k02.txt	97.70	300.00	99.20	300.01	99.20	99.20	99.20	0.02
GKD-b.46_n150_b03_m45_k03.txt	107.50	300.00	109.30	300.09	111.90	111.90	111.90	15.99
GKD-b.47_n150_b02_m45_k02.txt	158.50	300.00	163.10	300.06	164.90	164.90	164.90	0.44
GKD-b.47_n150_b02_m45_k03.txt	158.70	300.00	162.70	300.10	165.10	165.10	165.10	0.47
GKD-b.47_n150_b03_m45_k02.txt	133.90	300.00	133.90	300.00	135.00	134.91	134.90	6.51
GKD-b.47_n150_b03_m45_k03.txt	144.30	300.00	148.60	300.07	154.80	154.80	154.80	3.51
GKD-b.48_n150_b02_m45_k02.txt	94.70	300.00	97.60	300.10	99.10	99.10	99.10	0.28
GKD-b.48_n150_b02_m45_k03.txt	94.80	300.01	98.10	300.01	99.10	99.10	99.10	0.40
GKD-b.48_n150_b03_m45_k02.txt	70.30	300.00	71.20	300.00	75.00	75.00	75.00	0.01
GKD-b.48_n150_b03_m45_k03.txt	79.70	300.00	84.80	300.12	86.60	86.58	86.30	12.59
GKD-b.49_n150_b02_m45_k02.txt	161.90	300.00	166.10	300.06	166.90	166.90	166.90	0.54
GKD-b.49_n150_b02_m45_k03.txt	160.60	300.00	166.90	300.09	166.90	166.90	166.90	1.03
GKD-b.49_n150_b03_m45_k02.txt	0.00	300.00	0.00	300.00	139.40	139.40	139.40	0.06
GKD-b.49_n150_b03_m45_k03.txt	141.80	300.00	153.90	300.01	158.50	158.50	158.50	0.13
GKD-b.50_n150_b02_m45_k02.txt	97.80	300.00	106.30	300.21	111.60	111.60	111.60	0.21
GKD-b.50_n150_b02_m45_k03.txt	104.90	300.00	106.50	300.11	112.80	112.80	112.80	4.05
GKD-b.50_n150_b03_m45_k02.txt	0.00	300.00	0.00	300.00	82.00	82.00	82.00	3.69
GKD-b.50_n150_b03_m45_k03.txt	89.90	300.00	94.30	300.01	100.00	100.00	100.00	0.34
Average	92.89	300.00	96.58	300.05	109.59	109.56	109.52	3.14

Table 2. Detailed results on GKD instances with $n = 500$ elements.

Instance	TS-M		GRASP		TS-N			
	Obj.	Time	Obj.	Time	Best	Avg.	Worst	Time
GKD-c.01_n500_b02_m50_k02.txt	6.40	300.00	7.40	300.18	9.30	9.30	9.30	104.97
GKD-c.01_n500_b02_m50_k03.txt	6.50	300.45	7.40	301.32	9.30	9.30	9.30	114.47
GKD-c.01_n500_b03_m50_k02.txt	5.70	300.00	6.20	300.16	8.30	8.30	8.30	162.74
GKD-c.01_n500_b03_m50_k03.txt	5.70	300.00	6.30	300.83	8.30	8.30	8.30	162.83
GKD-c.02_n500_b02_m50_k02.txt	6.70	300.00	7.50	300.26	9.20	9.15	9.10	151.21
GKD-c.02_n500_b02_m50_k03.txt	6.70	300.24	7.50	300.20	9.20	9.16	9.10	176.82
GKD-c.02_n500_b03_m50_k02.txt	4.80	300.00	6.20	300.11	8.30	8.27	8.20	127.03
GKD-c.02_n500_b03_m50_k03.txt	5.20	300.00	6.20	300.03	8.30	8.27	8.20	127.07
GKD-c.03_n500_b02_m50_k02.txt	6.10	300.00	7.20	301.09	9.20	9.14	9.10	101.50
GKD-c.03_n500_b02_m50_k03.txt	6.10	300.39	7.20	301.56	9.20	9.15	9.10	115.29
GKD-c.03_n500_b03_m50_k02.txt	5.10	300.00	6.10	300.04	8.20	8.20	8.20	128.26
GKD-c.03_n500_b03_m50_k03.txt	5.90	300.00	6.10	300.67	8.20	8.20	8.20	128.27
GKD-c.04_n500_b02_m50_k02.txt	7.10	300.00	7.20	300.72	9.20	9.20	9.20	75.34
GKD-c.04_n500_b02_m50_k03.txt	7.10	300.03	7.20	300.97	9.20	9.20	9.20	66.21
GKD-c.04_n500_b03_m50_k02.txt	5.70	300.00	5.70	300.00	8.30	8.21	8.10	141.64
GKD-c.04_n500_b03_m50_k03.txt	5.70	300.00	6.20	301.49	8.30	8.21	8.10	141.88
GKD-c.05_n500_b02_m50_k02.txt	7.40	300.00	7.50	301.90	9.20	9.20	9.20	36.91
GKD-c.05_n500_b02_m50_k03.txt	7.40	300.53	7.50	300.25	9.20	9.20	9.20	31.27
GKD-c.05_n500_b03_m50_k02.txt	0.00	300.00	0.00	300.00	8.30	8.29	8.20	126.32
GKD-c.05_n500_b03_m50_k03.txt	5.10	300.00	7.40	302.04	8.30	8.29	8.20	126.42
GKD-c.06_n500_b02_m50_k02.txt	6.70	300.29	7.10	300.32	9.20	9.10	9.00	134.63
GKD-c.06_n500_b02_m50_k03.txt	6.70	300.21	7.00	300.68	9.20	9.11	9.10	129.47
GKD-c.06_n500_b03_m50_k02.txt	5.60	300.00	5.80	300.33	8.20	8.18	8.10	158.77
GKD-c.06_n500_b03_m50_k03.txt	5.60	300.00	5.90	301.49	8.20	8.18	8.10	158.68
GKD-c.07_n500_b02_m50_k02.txt	5.10	300.00	6.70	300.72	9.40	9.28	9.20	176.03
GKD-c.07_n500_b02_m50_k03.txt	5.10	300.37	6.70	301.88	9.40	9.29	9.20	187.42
GKD-c.07_n500_b03_m50_k02.txt	5.00	300.00	5.10	300.83	8.30	8.21	8.20	119.12
GKD-c.07_n500_b03_m50_k03.txt	5.10	300.00	5.60	300.99	8.30	8.21	8.20	119.46
GKD-c.08_n500_b02_m50_k02.txt	6.40	300.20	7.30	300.60	9.40	9.34	9.30	139.29
GKD-c.08_n500_b02_m50_k03.txt	6.80	300.40	7.30	301.75	9.40	9.34	9.30	152.35
GKD-c.08_n500_b03_m50_k02.txt	6.00	300.00	6.40	301.41	8.40	8.33	8.30	136.75
GKD-c.08_n500_b03_m50_k03.txt	6.10	300.00	6.60	301.97	8.40	8.33	8.30	136.61
GKD-c.09_n500_b02_m50_k02.txt	5.90	300.00	7.50	300.14	9.30	9.28	9.20	175.38
GKD-c.09_n500_b02_m50_k03.txt	5.90	300.08	7.10	300.44	9.30	9.28	9.20	203.75
GKD-c.09_n500_b03_m50_k02.txt	5.70	300.00	5.90	300.00	8.30	8.22	8.20	118.54
GKD-c.09_n500_b03_m50_k03.txt	6.00	301.65	6.00	300.63	8.30	8.22	8.20	118.30
GKD-c.10_n500_b02_m50_k02.txt	6.10	300.00	7.80	300.18	9.40	9.40	9.40	73.00
GKD-c.10_n500_b02_m50_k03.txt	6.10	300.42	7.40	301.39	9.40	9.40	9.40	81.27
GKD-c.10_n500_b03_m50_k02.txt	0.00	300.00	0.00	300.00	8.50	8.38	8.20	164.73
GKD-c.10_n500_b03_m50_k03.txt	6.10	300.00	6.10	300.47	8.50	8.38	8.20	165.06
Average	5.71	300.13	6.38	300.75	8.80	8.75	8.70	129.88

Table 3. Detailed results on MDG instances with $n = 500$ elements.

Instance	TS-M		GRASP		TS-N			
	Obj.	Time	Obj.	Time	Best	Avg.	Worst	Time
MDG-b_01_n500_b02_m50_k02.txt	5.30	300.00	15.90	304.39	53.60	50.34	47.70	266.52
MDG-b_01_n500_b02_m50_k03.txt	8.40	300.32	17.60	300.59	53.60	50.34	47.70	267.39
MDG-b_01_n500_b03_m50_k02.txt	0.10	300.00	0.10	300.08	26.00	24.57	23.60	243.64
MDG-b_01_n500_b03_m50_k03.txt	0.10	300.00	0.10	300.91	26.00	24.57	23.60	243.83
MDG-b_02_n500_b02_m50_k02.txt	1.00	300.00	19.40	300.30	55.10	52.54	48.70	236.09
MDG-b_02_n500_b02_m50_k03.txt	1.00	300.02	19.20	300.15	55.10	52.54	48.70	235.98
MDG-b_02_n500_b03_m50_k02.txt	0.90	300.00	1.00	300.17	26.90	25.54	24.80	198.67
MDG-b_02_n500_b03_m50_k03.txt	0.90	300.00	1.00	300.27	26.90	25.54	24.80	199.06
MDG-b_03_n500_b02_m50_k02.txt	5.40	300.00	20.30	301.91	55.50	53.93	52.10	263.84
MDG-b_03_n500_b02_m50_k03.txt	5.40	300.24	17.40	302.39	55.50	53.93	52.10	263.56
MDG-b_03_n500_b03_m50_k02.txt	1.80	300.00	2.80	300.77	27.20	26.36	25.30	217.90
MDG-b_03_n500_b03_m50_k03.txt	1.80	300.00	2.80	300.85	27.20	26.36	25.30	217.94
MDG-b_04_n500_b02_m50_k02.txt	11.60	300.00	21.10	307.89	57.20	54.20	51.50	286.36
MDG-b_04_n500_b02_m50_k03.txt	21.10	300.38	26.10	300.86	57.20	54.20	51.50	286.37
MDG-b_04_n500_b03_m50_k02.txt	0.00	300.00	0.00	300.00	28.00	26.35	25.90	189.62
MDG-b_04_n500_b03_m50_k03.txt	0.60	300.00	5.10	302.05	28.00	26.35	25.90	189.73
MDG-b_05_n500_b02_m50_k02.txt	23.60	300.00	23.60	303.56	56.70	55.21	54.30	203.99
MDG-b_05_n500_b02_m50_k03.txt	21.20	300.80	23.60	301.05	56.70	55.21	54.30	204.30
MDG-b_05_n500_b03_m50_k02.txt	0.00	300.00	0.00	300.00	27.40	26.68	25.50	239.22
MDG-b_05_n500_b03_m50_k03.txt	0.50	300.00	6.60	301.95	27.40	26.68	25.50	239.49
MDG-b_06_n500_b02_m50_k02.txt	2.50	300.00	22.40	301.90	54.50	51.53	47.90	137.56
MDG-b_06_n500_b02_m50_k03.txt	2.50	300.15	12.50	300.47	54.50	51.86	50.10	177.28
MDG-b_06_n500_b03_m50_k02.txt	0.60	300.00	1.20	300.00	27.20	26.20	25.60	170.04
MDG-b_06_n500_b03_m50_k03.txt	0.50	300.00	2.50	301.06	27.20	26.20	25.60	169.77
MDG-b_07_n500_b02_m50_k02.txt	7.90	300.00	20.40	300.50	57.40	52.53	50.30	246.64
MDG-b_07_n500_b02_m50_k03.txt	8.70	300.02	18.80	300.28	57.40	52.53	50.30	246.45
MDG-b_07_n500_b03_m50_k02.txt	0.60	300.00	0.60	300.20	27.10	26.55	25.90	243.05
MDG-b_07_n500_b03_m50_k03.txt	0.60	300.00	4.80	300.35	27.10	26.55	25.90	242.60
MDG-b_08_n500_b02_m50_k02.txt	14.80	300.00	23.90	300.27	56.80	55.14	54.30	213.97
MDG-b_08_n500_b02_m50_k03.txt	5.40	300.23	20.50	300.85	56.80	55.14	54.30	213.94
MDG-b_08_n500_b03_m50_k02.txt	0.80	300.00	2.50	300.91	30.10	29.31	28.40	202.84
MDG-b_08_n500_b03_m50_k03.txt	0.80	300.00	6.10	301.66	30.10	29.31	28.40	202.86
MDG-b_09_n500_b02_m50_k02.txt	6.20	300.00	33.50	300.60	57.50	55.31	53.20	254.52
MDG-b_09_n500_b02_m50_k03.txt	5.40	300.56	22.20	303.81	57.50	55.31	53.20	254.60
MDG-b_09_n500_b03_m50_k02.txt	0.00	300.00	0.00	300.00	26.70	25.75	24.60	259.06
MDG-b_09_n500_b03_m50_k03.txt	1.70	300.00	2.50	301.78	26.70	25.75	24.60	258.89
MDG-b_10_n500_b02_m50_k02.txt	2.70	300.00	23.90	307.83	58.10	55.95	52.80	189.98
MDG-b_10_n500_b02_m50_k03.txt	2.70	300.26	26.00	308.65	58.10	55.95	52.80	189.10
MDG-b_10_n500_b03_m50_k02.txt	0.90	300.00	0.90	300.17	29.30	27.11	26.20	236.19
MDG-b_10_n500_b03_m50_k03.txt	0.90	300.00	5.20	300.25	29.30	27.11	26.20	236.34
Average	4.42	300.07	11.85	301.54	41.92	40.06	38.49	225.98

5.3 Conclusions

From these tables we can conclude that our Tabu search significantly outperforms existing methods in both criteria: the objective function value of obtained solutions as well as time need to reach the final solution. We think that main part in our algorithm is using three neighborhoods. Note that using only one neighborhood disables (not allows) changing number of selected elements. Because of that, number of selected elements during the improving phase (in the GRASP algorithm) or during the local search (in the Tabu search proposed by Martínez-Gavara et al. [9]) stay same as in the initial solution.

Our future research will be devoted to introducing new neighborhoods into the solution space.

References

1. Duarte, A., Martí, R.: Tabu search and GRASP for the maximum diversity problem. *Eur. J. Oper. Res.* **178**, 71–84 (2007)
2. Galinier, P., Boujbel, Z., Fernandes, M.C.: An efficient memetic algorithm for the graph partitioning problem. *Ann. Oper. Res.* **19**(1), 1–22 (2011)
3. Glover, F.: Tabu search—Part I. *ORSA J. Comput.* **1**, 190–206 (1989)
4. Glover, F., Kuo, C.C., Dhir, K.S.: Heuristic algorithms for the maximum diversity problem. *J. Inf. Optim. Sci.* **19**, 109–132 (1998)
5. Kuby, M.J.: Programming models for facility dispersion: the p-Dispersion and maximum dispersion problems. *Math. Comput. Model.* **10**, 792 (1988)
6. Martí, R., Duarte, A.: MDPLIB - maximum diversity problem library (2010). <https://www.uv.es/rmarti/paper/mdp.html>
7. Martí, R., Gallego, M., Duarte, A.: A branch and bound algorithm for the maximum diversity problem. *Eur. J. Oper. Res.* **200**, 36–44 (2010)
8. Martí, R., Martínez-Gavara, A., Pérez-Peló, S., Sánchez-Oro, J.: A review on discrete diversity and dispersion maximization from an OR perspective. *Eur. J. Oper. Res.* **299**, 795–813 (2022)
9. Martínez-Gavara, A., Corberán, T., Martí, R.: GRASP and Tabu search for the generalized dispersion problem. *Expert Syst. Appl.* **173**, 114703 (2021)
10. Mladenović, N., Todosijević, R., Urošević, D., Ratli, M.: Solving the Capacitated Dispersion Problem with variable neighborhood search approaches: from basic to skewed VNS. *Comput. Oper. Res.* **139**, 105622 (2022)
11. Parreño, F., Alvarez-Valdés, R., Martí, R.: Measuring diversity. A review and an empirical analysis. *Eur. J. Oper. Res.* **289**, 515–532 (2021)
12. Resende, M.G., Martí, R., Gallego, M., Duarte, A.: Grasp and path relinking for the max-min diversity problem. *Comput. Oper. Res.* **37**, 498–508 (2010)
13. Rosenkrantz, D.J., Tayi, G.K., Ravi, S.S.: Facility dispersion problems under capacity and cost constraints. *J. Comb. Optim.* **4**, 7–33 (2000)
14. Sandoya, F., Martínez-Gavara, A., Aceves, R., Duarte, A., Martí, R.: Diversity and equity models. In: Martí, R., Pardalos, P.M., Resende, M.G.C. (eds.) *Handbook of Heuristics*, pp. 979–998. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-07124-4_61



Multiple Project Scheduling for a Network Roll-Out Problem: MIP Formulation and Heuristic

Igor Vasilyev^{1,2} , Dmitry Rybin^{3,4}  , Sergey Kudria^{3,4} , Jie Ren⁴,
and Dong Zhang⁵

- ¹ Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia
vil@icc.ru
- ² Huawei Novosibirsk Research Center, Novosibirsk, Russia
igor.vasilyev@huawei.com
- ³ The Chinese University of Hong Kong, Shenzhen, China
dmitryrybin@link.cuhk.edu.cn, sergeykudrya@bk.ru
- ⁴ Huawei Moscow Research Center, Moscow, Russia
renjie21@huawei.com
- ⁵ Algorithm and Technology Development Department, Global Technical Service Department, Huawei Technologies, Co., Ltd., Dongguan, China
zhangdong48@huawei.com

Abstract. The paper addresses the Network Roll-Out (NRO) problem aimed at scheduling the construction of mobile stations. In this case, NRO problem can be considered as a generalization of Resource-Constrained Project Scheduling Problem, where we need to find a schedule for the activities related to the construction of a set of base stations taking into account the precedence constraints and the availability of resources while minimizing some measure of performance. The generalization involves the multiple projects, multi-modes, discrete time-cost tradeoff, together with particular business requirements like precedence redundancy and workload stability constraints.

To solve this problem, we propose a MIP formulation that is based on a generalization of Disaggregated Discrete-Time formulation with pulse start variables. Using this formulation, we propose a three-stage heuristic based on a relax and fix strategy. The effectiveness and efficiency of the proposed approach are illustrated in a series of computational experiments on real-life problem instances.

Keywords: Network roll-out problem · Multiple project scheduling · MIP heuristic

1 Introduction

The modern world demands more and more internet connectivity and higher data rates provided by wireless communications. It is the most important component in the development of smart economy and world digitization. Currently,

cellular networks, mainly based on LTE (4G) and 5G, have a wider range of tasks than ever before. Wireless communications must satisfy requirements of multiple device connectivity and high data rate, much greater bandwidth, low-latency quality of service, and low interference. It requires providing high-quality service and accommodating a wide range of prospective technologies, like the Internet of Things (IoT), Internet of Vehicles (IoV), Device to Device (D2D) communications, e-healthcare, Machine to Machine (M2M) communications, self-driving and Financial Technology (FinTech) etc. [1,6].

One of the key problems is rolling-out mobile networks. A Network Roll-Out (NRO) problem involves multiple strategic decisions, including finding the sites for locating base stations, selecting types of the stations, traffic routing. The NRO is usually a large-scale telecommunication project, and its different stages faced by carriers and vendors involve the decisions that can be modelled as optimization problems. The Roll-out process consists of many stages, from regulation, budgeting, and site selection to implementation of a detailed construction plan. Several recent papers were focused on a site selection problem with respect to existing government and user requirements [2,7].

In this paper we address the last stage—construction scheduling. At this stage, the NRO problem can be considered as a generalization of Resource-Constrained Project Scheduling Problem (RCPSP). It is to find a schedule for all the activities related to the construction of stations, taking into account the precedence constraints and the availability of resources while minimizing some measure of performance. Thus, it leads us to the Multiple Project Scheduling Problem (MPSP), where construction of a station is a single project, activities of which are specified by contractors having various skills. To process a certain activity, a predefined skill is needed. This case can be tackled in terms of Multi-mode Resource-Constrained Project Scheduling Problem (MRCPSP), where the multiple activity modes are presented, and each activity can be executed under different conditions depending on skills. Moreover, the duration and cost involved in completing the activity depend on the mode of an activity. Such problem is also known as the Discrete Time-Cost Tradeoff Problem (DTCTP).

Due to their practical importance, the RCPSP and its generalizations remain a very active field of research. We address the reader to handbooks [4,5,11,14,15] for the comprehensive survey of theory and algorithms on this topic. Some more details of MRCPSP can also be found in [12,13,17], DTCTP in [5,16] and MPSP in [9,10].

In this paper, we propose a MIP formulation of NRO based on Disaggregated Discrete-Time formulation with pulse start variables. The latter can be found in [3,8] with the overview of different other formulations. Taking into account the particularity of our problem, we generalize this formulation, provided that we have many identical projects with a few activities and simple precedence constraints. Our formulation also allows for many extra requirements coming from technical and business considerations. We test our formulation in a series of computational experiments on real life instances and demonstrate that a general MIP solver cannot find good or even feasible solutions in a reasonable time.

Thus, we propose a three-stage heuristic based on a relax and fix strategy. The effectiveness and efficiency of the proposed approach are illustrated in the computational experiments on real life instances.

The rest of the paper is structured as follows. The NRO problem statement is introduced in Sect. 2 and its MIP formulation in Sect. 3. The three-stage heuristic approach is outlined in Sect. 4. Finally, Sect. 5 gives preliminary computation results and concluding remarks.

2 Problem Statement

In our particular case, the multiple projects consist of constructing mobile base stations on some sites. To construct a base station, a certain sequence of activities must be fulfilled in a prescribed order, which defines the precedence relations between activities. We suppose to have a finite number of different station types, and the sequence of activities is the same for the same station type. Therefore, we can consider each activity not separately but group them by type of station. Let us assume that

- $T = \{1, \dots, |T|\}$ is the time horizon (the time unit is a day).
- The time horizon consists of a set of equal intervals (weeks or months) $W = \{1, \dots, |W|\}$, where $\tau_w \in T$ is a beginning of interval $w \in W$. For the sake of simplicity, we suppose that $\tau_{|W|+1} = |T|$
- J is the set of activity types.
- n_j is the number of activities of type $j \in J$ that have to be fulfilled.
- $\bar{J} \subset J$ is a subset of activity types, which have a predecessor activity, i.e. $J \setminus \bar{J}$ is a set of activity types that are the first in construction sequence, and they do not have a predecessor.
- $j_j^* \in J$ is a type of activity, which is a predecessor of activity type $j \in \bar{J}$.
- f_j is the minimal time lag between the completion time of activity of type j_j^* and the start time of successor activity of type $j \in \bar{J}$.
- C is the fixed cost per week until all the activities are not completed.

The capacity constraints are defined by contractors whose teams complete activities:

- K is a set of contractors.
- B is a set of team types.
- $B^k \subset B$ is a subset of team types of contractor $k \in K$, which partition B , i.e.

$$\bigcup_{k \in K} B^k = B, \quad B^l \cap B^k = \emptyset \quad \forall l, k \in K : k \neq l.$$

- u_t^b is the number of teams of type $b \in B$ available at time $t \in T$.
- $J^b \subset J$ is a subset of activity types that a team of type $b \in B$ can do (has the skill to complete the activity).
- $B^j \subset B$ is a subset of team types with the skill to do activity type $j \in J$ (related to J^b).

- p_j is a processing time of activity type $j \in J$.
- c_{jb} is a cost of processing activity of type $j \in J$ by a team of type $b \in B$.

In the general case, the problem consists of scheduling all activities with minimal time lags with respect to the precedence constraints and assigning the contractor's teams to complete these activities with respect to their skills and capacities. We consider a more general objective. Instead of minimizing only the makespan, our objective is to minimize the processing costs. It is acceptable to have the makespan longer than minimal, but not too much. Moreover, we have the fixed cost during the makespan, which has to be taken into account if the minimal makespan is extended.

We have additional constraints that have never been considered in the literature. The first one is the precedence redundancy constraints, which should reduce the risk of non-fulfillment of the project schedule. Within a week, for each task type $j \in \bar{J}$, the cumulative completion of predecessors of type j_j^* is not less than the cumulative completion of activities of type j multiplied by the given risk factor $r_j \geq 1 \forall j \in \bar{J}$, unless the cumulative completion of activities of type j_j^* is greater or equal to the total.

The second particular condition is a “stable” contractor workload. Contractors require that:

- the daily number of involved teams should be similar in a week (close to the average);
- the maximal number of involved teams per week should be non-decreasing at the beginning and, then, non-increasing. This condition is not hard, i.e. the mild violation is acceptable.

The stability constraint is soft, not strictly defined, and cannot be analytically evaluated. Its evaluation only can be done in practice by contractors; therefore, a parametric approach is required to control this constraint.

3 MIP Formulation

Let us consider the variables x_{jbt} which are equal to the number of activities of type $j \in J$ started by teams of type $b \in B^j$ at time $t \in T$, variable H equals the number of weeks in the makespan. Let us also introduce the variables

$$h_w = \begin{cases} 1, & \text{if there is any activities processed in week } w \in W, \\ 0, & \text{otherwise.} \end{cases}$$

With these notations, the problem can be written as

$$\min \sum_{j \in J} \sum_{b \in B^j} \sum_{t \in T} c_{jb} x_{jbt} + C \cdot H \quad (1)$$

$$\sum_{j \in J} \sum_{b \in B^j} \sum_{t=\tau_w}^{\tau_{w+1}-1} u_t^b h_w \geq \sum_{j \in J} \sum_{b \in B^j} \sum_{t=\tau_w-p_j+1}^{\tau_{w+1}-1} x_{jbt} \quad \forall w \in W \quad (2)$$

$$H \geq w h_w \quad \forall w \in W \quad (3)$$

$$\sum_{b \in B^j} \sum_{t \in T} x_{jbt} \geq n_j \quad \forall j \in J \quad (4)$$

$$\sum_{b \in B^{j_j^*}} \sum_{l=0}^{t-p_{j_j^*}-f_j} x_{j_j^*kl} \geq \sum_{b \in B^j} \sum_{l=0}^t x_{jbl} \quad j \in \bar{J}, t \in T \quad (5)$$

$$\sum_{j \in J^b} \sum_{l=t}^{t-p_j+1} x_{jbl} \leq u_t^b \quad \forall b \in B, t \in T \quad (6)$$

$$x_{jbt} \in \mathbb{B} \quad \forall j \in J, b \in B^j, t \in T \quad (7)$$

$$h_w \in \mathbb{B} \quad \forall w \in W \quad (8)$$

$$w \geq 0 \quad (9)$$

The objective (1) is to minimize the total processing and makespan costs. Constraints (2) ensure that variable h_w equals one if there is any processed activity during the corresponding week, and the inequalities (3) define the bound on the number of weeks. All the activities must be completed due to the constraints (4). The precedence constraints are guaranteed by inequalities (5). Furthermore, the constraints (6) are capacity constraints on the available teams. In constraints (5) and (6), if the lower bound of summation is greater than the upper bound, then the sums are assumed to be zero.

For the precedence redundancy condition, let us consider new variables: y_{jw} equals 1 if the completion of activities of type j_j^* is less than the total in week $w \in W$, and 0 otherwise, i.e.

$$n_{j_j^*} y_{jw} \geq n_{j_j^*} - \sum_{b \in B^{j_j^*}} \sum_{t=0}^{\tau_{w+1}-p_{j_j^*}} x_{j_j^*bt} \quad j \in \bar{J}, w \in W \quad (10)$$

The redundancy is guaranteed by

$$\begin{aligned} & \sum_{b \in B^{j_j^*}} \sum_{t=0}^{\tau_{w+1}-p_{j_j^*}} x_{j_j^*bt} + n_j(1 - y_{jw}) \geq \\ & \geq r_j \sum_{b \in B^j} \sum_{t=0}^{\tau_{w+1}-1} x_{jbt} \quad \forall j \in \bar{J}, w \in W \quad (11) \end{aligned}$$

In addition, we have the following valid inequalities:

$$y_{jw} \geq y_{jw+1} \quad \forall j \in \bar{J}, w = 1, |W| - 1 \quad (12)$$

The stability condition is much harder to express, especially in soft case. Let us consider a day contractor workload (number of used teams)

$$\xi_{kt} = \sum_{b \in B^k} \sum_{j \in J^b} \sum_{l=t-p_j+1}^t x_{jbl} \quad \forall k \in K, t \in T,$$

i.e. the number of teams of contractor b which are used in this day. Let

$$\pi_{kw} = \max\{\xi_{kt} : t = \tau_w, \dots, \tau_{w+1} - 1\} \quad \forall k \in K, w \in W$$

be weekly peaks (maximum weekly workload). For the strict stability we want that

$$\pi_{kw} = \xi_{kt} \quad \forall k \in K, w \in W, t = \tau_w, \dots, \tau_{w+1} - 1$$

and the sequences π_{kw} are weakly unimodal, i.e. there exists $w^k \in W$ such that

$$\pi_{k1} \leq \dots \leq \pi_{kw^k} \geq \dots \geq \pi_{k|W|} \quad \forall k \in K.$$

These conditions are very restrictive and usually cannot be satisfied in practice. So, we need to find a way to soften them, express them in a MIP, and define a parameter that allows us to control their “softness”.

The best empirical results, compromising between quality schedules and difficulty of MIP model, were obtained with the following model. Let us consider variables ψ_{kw} such that

$$\psi_{kw} \geq \pi_{kw} \quad \forall k \in K, w \in W$$

or the same

$$\psi_{kw} \geq \sum_{b \in B^k} \sum_{j \in J^b} \sum_{l=t-p_j+1}^t x_{jbl} \quad \forall k \in K, w \in W, t = \tau_w, \dots, \tau_{w+1} - 1. \quad (13)$$

The corresponding sequences must be unimodal, i.e.

$$\begin{aligned} \psi_{kw} &\leq \psi_{kw+1} + Mz_{kw} && \forall k \in K, w = 1, d, \dots, |W| - 1 \\ \psi_{kw+1} &\leq \psi_{kw} + M(1 - z_{kw}) && \forall k \in K, w = 1, \dots, |W| - 1 \\ z_{kw} &\leq z_{kw+1} && \forall k \in K, w = 1, \dots, |W| - 1 \end{aligned} \quad (14)$$

where variables

$$z_{kw} \in \mathbb{B} \quad \forall k \in K, w = 1, \dots, |W| - 1. \quad (15)$$

It follows that

$$z_{kw} = \begin{cases} 0 & \text{if } \psi_{kw} \leq \psi_{kw+1} \\ 1 & \text{if } \psi_{kw+1} \geq \psi_{kw}. \end{cases}$$

Of course, constraints (13)–(15) cannot guaranty strict or soft stability by themselves. Let us consider a new objective function with penalized sum of all ψ_{kw} variables, i.e.

$$\min \sum_{j \in J} \sum_{b \in B^j} \sum_{t \in T} c_{jb} x_{jbt} + C \cdot H + S \sum_{k \in K} \sum_{w \in W} \psi_{kw} \tag{16}$$

where S is a given parameter. Our experiments showed that this approach gives us a good results and changing S provides us with trade-off between the processing cost and stability quality.

4 Three Stage Heuristic Approach

In the formulation (2)–(16), we have some fixed time horizon. As mentioned in the problem statement, we are interested in finding solutions with minimal processing costs, the value of which is not too longer than the minimal makespan. Therefore, a minimal makespan has to be found.

In our approach, we suppose that the makespan is fixed and our objective is to find the schedule with the minimal cost. With the fixed makespan the variables $H, h_w \forall w \in W$ and constrains (2), (3) can be omitted. The fixed cost is a constant, hence the objective is

$$\min \sum_{j \in J} \sum_{b \in B^j} \sum_{t \in T} c_{jb} x_{jbt} + S \sum_{k \in K} \sum_{w \in W} \psi_{kw}. \tag{17}$$

As we observed in our experiments, the resulting formulation (4)–(15), (17) was hard to be solved by a MIP solver for relatively large instances, and heuristic approaches are needed to find high-quality solutions quickly.

The LP relaxation of (4)–(15), (17) does not give a good direction of finding integer solution. In our case study, the optimal objective values of LP relaxation and the integer formulation are close to each other, but solutions are far away. The main reason is that the formulation contain several so-called BigM inequalities which come from the redundancy constraints (10), (11) and stability constraints (14). It makes the LP solution very fractional and useless in recovering an integer solution, saying nothing about the optimal one. Thus, our idea is to consider other relaxations, which allows us to find feasible and quite good solutions.

Testing different possibilities, we found out that a three-stage heuristic based on a relax and fix strategy gives us quite promising results.

In our heuristic, in the first stage, we relax the integrality constraints on variables x as well as the stability constraints. Obtained variables y are fixed in the second stage with stability constraints. Finally, in the third stage, we find the integer solution x with fixed binary variables y and z from the previous stages. The stages of the heuristic are summarized in detail in Table 1.

Table 1. Heuristic details

Stage	Constraints	Variables
I	(4)–(12)	x – relaxed y – binary z – skipped
II	(4)–(15)	x – relaxed y – fixed z – binary
III	(4)–(15)	x – integer y – fixed z – fixed

5 Computational Results and Concluding Remarks

Our computational experiments were carried out on a workstation with Intel(R) Core(TM) i7-8550U CPU and 16 Gb of RAM. SCIP version 7.0.2 was used as a general MIP solver.

We consider 12 test instances which are based on real data. The details of test instances are given in Table 2, where

- $Name$ is the instance name;
- $|K|$ is the number of contractors;
- $|B|$ is the number of team types;
- $|J|$ is the number of activity types;
- n is the total number of activities;
- \bar{H} is the minimal makespan in weeks.

Table 2. Instance details

$Name$	$ K $	$ B $	$ J $	$ n $	\bar{H}
Task01	6	16	12	6008	21
Task02	6	16	12	6006	42
Task03	6	16	18	3006	15
Task04	6	16	18	3010	21
Task05	6	16	21	3010	20
Task06	8	15	21	2010	15
Task07	8	15	28	2008	17
Task08	8	16	28	2012	24
Task09	7	19	28	1234	16
Task10	7	19	28	1812	8
Task11	4	9	28	1812	18
Task12	4	9	28	630	6

In our case study, it turns out that the feasibility of LP relaxation always relates to the feasibility of the original problem. Solving LP relaxation takes just a few seconds, hence the minimal makespan can be easily and fast found by the bisection of \bar{H} and solving the corresponding LP relaxations of (4)–(15), (17).

The last instance Task12 is the smallest one and can be easily solved to optimality by the MIP solver. It is used only to answer a question of how the stability penalty S can be chosen. We tried different variants based on average construction cost denoted by \bar{c} . The results on Task12 are illustrated in Figs. 1, 2, 3 and 4 and Table 3. The alternating red and blue colors represent weeks. If we understand workload stability as small oscillation within each week and unimodality of peak load, then the best tradeoff between the stability quality and construction cost is given by $S = 0.1 \cdot \bar{c}$, as mild contractor cost increase results in a near stable workload. One can observe similar behavior in other instances; hence it was chosen for all experiments.

Table 3. Contractor cost for different values of penalty S

S	0	$0.01 \cdot \bar{c}$	$0.1 \cdot \bar{c}$	\bar{c}
Cost	1 008 565	1 008 769	1 010 138	1 031 422

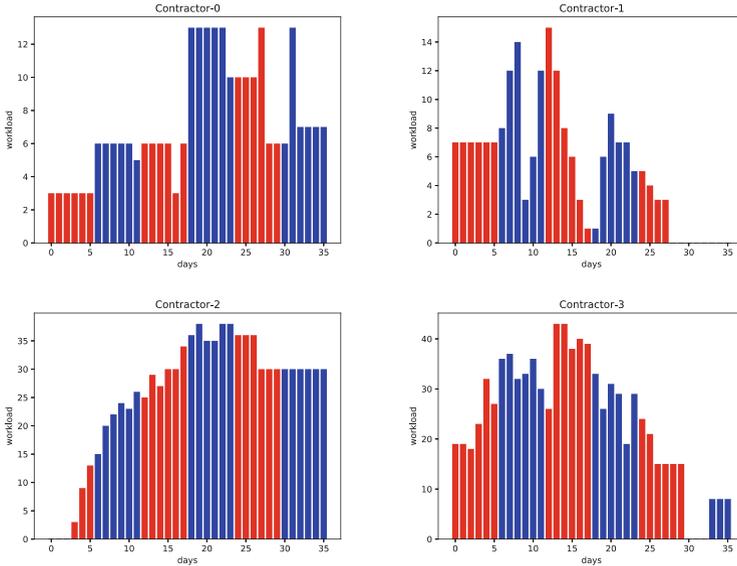


Fig. 1. Workload with $S = 0$

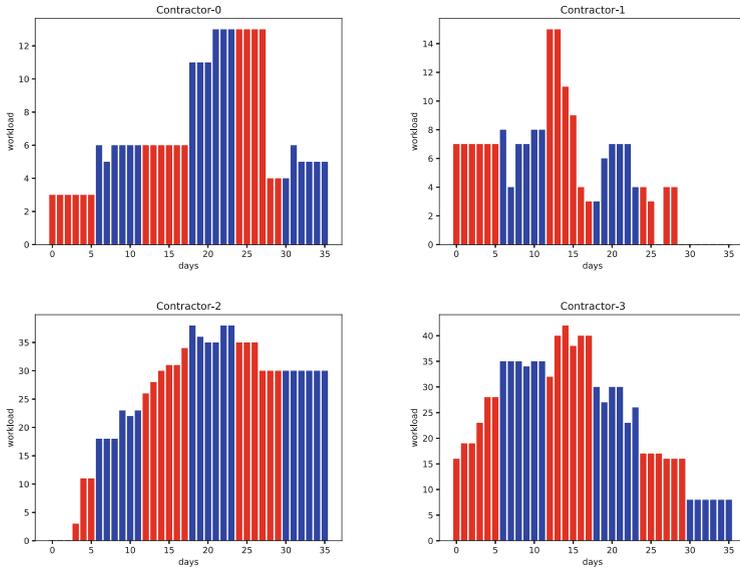


Fig. 2. Workload with $S = 0.01 \cdot \bar{c}$

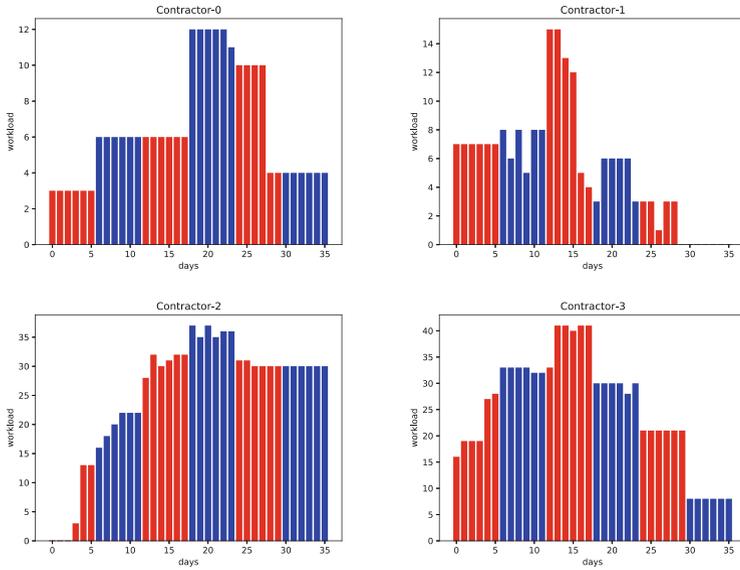


Fig. 3. Workload with $S = 0.1 \cdot \bar{c}$

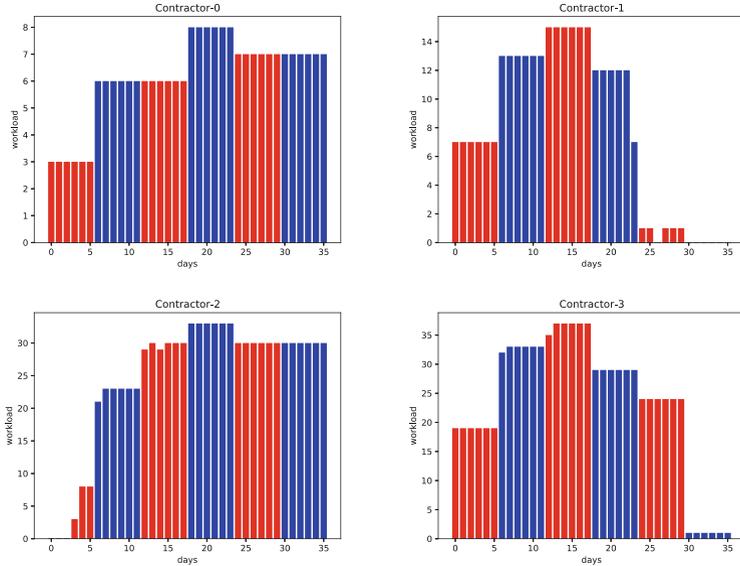


Fig. 4. Workload with $S = \bar{c}$

In practice, we are interested in fast finding a good solution in order to allow practitioners to have different solutions with different problem parameters and data. Our three-stage approach allows one to control the computation time and solution quality by setting different parameters for the MIP solver at different stages. In our preliminary experiments, we set time limit to 600 s and GAP limit to 0.1% on all stages. In Tables 4 and 5, the results are given and compared with solutions obtained by solving the formulation with SCIP within two hours of time limit. The results are given with the following notations:

- Makespan is the makespan in weeks. We consider five consequent values starting from the minimal makespan.
- GAP scip is the gap in percent between the best known upper and lower bound obtained by the MIP solver.
- GAP heur is the gap in percent between our three-stage heuristic upper bound and the MIP solver lower bound.
- Time heur is the total running time of all three stages of the heuristic.

In three cases, the heuristic yields slightly worse solutions. But the most important thing is that the heuristic can find quite good feasible solutions in all cases, while the MIP solver could not even find any feasible ones in 17 cases, that is, for 17 different pairs of data instance and makespan there was no feasible solution found by the MIP solver, hence the gap is equal to 100.0.

Our preliminary results showed that our three-stage heuristic can fast provide us with good solutions to instances that are based on real-life data. But it is worth mentioning that the MIP solver does not always give solutions with a small gap,

especially in the third stage. Further research will be focused on improving our approach by developing more problem-specific algorithms than a general MIP solver, which involves better studying of problem structure.

Table 4. Computational results

Task01					
Makespan	21	22	23	24	25
GAP scip	100.0	100.0	100.0	100.0	13.4
GAP heur	0.0	4.1	4.1	3.4	2.7
Time heur	711	196	211	385	837
Task02					
Makespan	42	43	44	45	46
GAP scip	13.3	9.5	4.8	5.9	1.5
GAP heur	0.4	0.4	0.3	0.2	0.3
Time heur	1348	1100	523	535	614
Task03					
Makespan	15	16	17	18	19
GAP scip	100.0	100.0	0.2	2.2	1.3
GAP heur	0.7	0.7	3.2	1.0	0.5
Time heur	967	994	1177	1140	957
Task04					
Makespan	21	22	23	24	25
GAP scip	100.0	100.0	2.0	0.4	3.0
GAP heur	2.4	1.8	0.7	1.4	0.6
Time heur	1371	1363	1432	1242	1461
Task05					
Makespan	20	21	22	23	24
GAP scip	100.0	100.0	100.0	2.9	100.0
GAP heur	0.5	0.8	0.3	0.4	0.2
Time heur	646	1275	487	475	464
Task06					
Makespan	15	16	17	18	19
GAP scip	3.1	5.0	0.2	2.1	0.5
GAP heur	0.8	0.1	0.8	0.4	0.4
Time heur	808	259	828	769	706
Task07					
Makespan	17	18	19	20	21
GAP scip	2.0	2.1	1.3	1.7	0.5
GAP heur	0.2	0.1	0.2	0.1	0.3
Time heur	916	899	581	296	797

Table 5. Computational results

Task08					
Makespan	24	25	26	27	28
GAP scip	100.0	5.9	100.0	22.2	100.0
GAP heur	0.0	0.6	1.2	0.9	0.6
Time heur	1570	1800	1800	1625	1800
Task09					
Makespan	16	17	18	19	20
GAP scip	3.3	3.3	100.0	4.2	4.3
GAP heur	0.3	6.2	4.1	0.8	5.8
Time heur	1058	1382	1800	1382	1800
Task10					
Makespan	8	9	10	11	12
GAP scip	2.7	1.3	100.0	1.9	1.9
GAP heur	0.8	0.2	0.0	0.3	1.2
Time heur	762	758	829	798	804
Task11					
Makespan	18	19	20	21	22
GAP scip	6.0	9.1	6.8	6.3	2.4
GAP heur	1.3	1.1	1.5	2.6	1.7
Time heur	966	973	1104	1224	1147

Acknowledgements. Dmitry Rybin was supported by Shenzhen Research Institute of Big Data.

References

1. Agiwal, M., Roy, A., Saxena, N.: Next generation 5G wireless networks: a comprehensive survey. *IEEE Commun. Surv. Tutor.* **18**(3), 1617–1655 (2016). <https://doi.org/10.1109/COMST.2016.2532458>
2. Albanese, A., Sciancalepore, V., Banchs, A., Costa-Pérez, X.: LOKO: localization-aware roll-out planning for future mobile networks (2022)
3. Artigues, C., Koné, O., Lopez, P., Mongeau, M.: Mixed-integer linear programming formulations. In: Schwindt, C., Zimmermann, J. (eds.) *Handbook on Project Management and Scheduling*. IHIS, vol. 1, pp. 17–41. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-05443-8_2
4. Ben Issa, S., Tu, Y.: A survey in the resource-constrained project and multi-project scheduling problems. *J. Project Manage.* 117–138 (2020). <https://doi.org/10.5267/j.jpm.2019.11.001>
5. Brucker, P., Drexel, A., Möhring, R., Neumann, K., Pesch, E.: Resource-constrained project scheduling: notation, classification, models, and methods. *Eur. J. Oper. Res.* **112**(1), 3–41 (1999). [https://doi.org/10.1016/S0377-2217\(98\)00204-5](https://doi.org/10.1016/S0377-2217(98)00204-5)

6. Chettri, L., Bera, R.: A comprehensive survey on internet of things (IoT) toward 5G wireless systems. *IEEE Internet Things J.* **7**(1), 16–32 (2020). <https://doi.org/10.1109/JIOT.2019.2948888>
7. Chiaraviglio, L., et al.: Planning 5G networks under EMF constraints: state of the art and vision. *IEEE Access* **6**, 51021–51037 (2018). <https://doi.org/10.1109/access.2018.2868347>, <https://doi.org/10.1109/access.2018.2868347>
8. Christofides, N., Alvarez-Valdes, R., Tamarit, J.: Project scheduling with resource constraints: a branch and bound approach. *Eur. J. Oper. Res.* **29**(3), 262–273 (1987). [https://doi.org/10.1016/0377-2217\(87\)90240-2](https://doi.org/10.1016/0377-2217(87)90240-2), <https://www.sciencedirect.com/science/article/pii/0377221787902402>
9. Geiger, M.J.: A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem. *Eur. J. Oper. Res.* **256**(3), 729–741 (2017). <https://doi.org/10.1016/j.ejor.2016.07.024>, <https://www.sciencedirect.com/science/article/pii/S0377221716305616>
10. Gonçalves, J.F., de Magalhães Mendes, J.J., Resende, M.G.C.: The basic multi-project scheduling problem. In: Schwindt, C., Zimmermann, J. (eds.) *Handbook on Project Management and Scheduling*. IHIS, vol. 2, pp. 667–683. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-05915-0_1
11. Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **207**(1), 1–14 (2010). <https://doi.org/10.1016/j.ejor.2009.11.005>
12. Mika, M., Waligóra, G., Weglarz, J.: Overview and state of the art. In: Schwindt, C., Zimmermann, J. (eds.) *Handbook on Project Management and Scheduling* Vol.1. IHIS, pp. 445–490. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-05443-8_21
13. Peteghem, V., Vanhoucke, M.: An experimental investigation of metaheuristics for the multimode resource-constrained project scheduling problem on new dataset instances. *Eur. J. Oper. Res.* **235**, 62–72 (2014). <https://doi.org/10.1016/j.ejor.2013.10.012>
14. Schwindt, C., Zimmermann, J. (eds.): *Handbook on Project Management and Scheduling*. IHIS, vol. 2. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-05915-0>
15. Schwindt, C., Zimmermann, J. (eds.): *Handbook on Project Management and Scheduling*. IHIS, vol. 1. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-05443-8>
16. Szmerekovsky, J.G., Venkateshan, P.: The discrete time-cost tradeoff problem with irregular starting time costs. In: Schwindt, C., Zimmermann, J. (eds.) *Handbook on Project Management and Scheduling*. IHIS, vol. 1, pp. 621–638. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-05443-8_29
17. Weglarz, J., Jozefowska, J., Mika, M., Waligóra, G.: Project scheduling with finite or infinite number of activity processing modes - a survey. *Eur. J. Oper. Res.* **208**, 177–205 (2011). <https://doi.org/10.1016/j.ejor.2010.03.037>

Applications



On a Nonconvex Distance-Based Clustering Problem

Tatiana V. Gruzdeva^(✉)  and Anton V. Ushakov 

Matrosov Institute for System Dynamics and Control Theory of SB RAS,
134 Lermontov Street, 664033 Irkutsk, Russia
{gruzdeva, aushakov}@icc.ru

Abstract. Clustering is one of the basic data analysis tools and an important subroutine in many machine learning tasks. Probably, the most well-known and popular clustering model is the Euclidean minimum-sum-of-squares clustering problem, also known as the k -means problem. Clustering with Bregman divergences is a generalization of the k -means problem where the distances between data items and closest cluster centers are computed according to any Bregman divergence, rather than the squared Euclidean distance. In this paper, we consider a mathematical programming problem of clustering with Bregman divergences. We propose several representations of the problem in the form of a DC (difference of convex) program and develop a DC programming approach to solve it. We provide particular DC representations and particular DC solution algorithms for several widely-known Bregman divergences.

Keywords: Clustering · Bregman divergence · Nonconvex optimization · DC programming · Local search · Global search scheme

1 Introduction

The cluster analysis problem is one of the most well-known and widely studied problems in statistics, mathematical programming, and machine learning. In the most general form, the cluster analysis problem is to divide a given set of patterns or samples into non-overlapped subsets, called clusters, such that each cluster consists of similar objects and the objects from different clusters are dissimilar. In practical settings, clustering is one of the basic data analysis tool and an important subroutine in many machine learning tasks, e.g. supervised learning (regression analysis, deep learning, etc.).

There are numerous variations and extensions of the basic clustering problem that are distinguished by the definition of similarity measure, possible constraints on the size of clusters, the form of clusters, their structure, interrelations, etc.

The research was funded by the Ministry of Education and Science of the Russian Federation (state registration No. 121041300065-9).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 139–152, 2022.
https://doi.org/10.1007/978-3-031-09607-5_10

This variety in definitions of possible clusters in a dataset actually makes the cluster analysis problem ill-posed, hence there may be many ways of formalizing it. Because of that, there are plenty of clustering algorithms based on different views on the cluster analysis problem, e.g. hierarchical algorithms, statistical approaches, density-based algorithms, etc. Nevertheless, the so-called center-based clustering algorithms, ones of the oldest clustering techniques, remain the most popular and widespread ones. They are aimed at partitioning a dataset into clusters by finding the so-called cluster representatives (or centers). Clusters are then formed by assigning data items to the closest (most similar) cluster centers. The underlying clustering model can be formulated as a nonconvex optimization problem [12]. Note, the number of representatives is often assumed to be fixed. Nevertheless, there are some general frameworks aimed at convexification of the clustering problems by introducing some convex penalties (e.g. see [19]).

The center-based clustering models can be considered as facility location problems, where one has to locate a set of facilities (representatives) in order to minimize the total cost of serving a given set of customers (data items) with respect to some constraints (for a survey, please see [25]). The well-known clustering models are k -means (minimum sum-of-squares) and closely related generalized multi-source Weber problem, k -center, and k -medoids (minimum sum-of-stars, discrete p -median).

Among the center-based clustering models, the most famous one is the minimum sum-of-squares clustering. Given m data items represented as feature vectors in a n -dimensional space, the objective is to find k cluster centers, that can be located in arbitrary points, such that the total sum of squared Euclidean distances between data items and the closest centers is minimized. The corresponding optimization problem of finding such kind of clustering is NP-hard. Note that it is NP-hard even on the plane for arbitrary number of clusters k [17] and in arbitrary space-dimension even for $k = 2$ [1]. Nowadays, the most common clustering algorithm is k -means (a.k.a. Lloyd's algorithm) [14, 16], which is a local search (alternate location-allocation) algorithm for the minimum sum-of-squares model. Its main idea is that both the problem of finding cluster centers and the problem of assigning data items to the closest centers are simple if they are considered independently. Thus, the algorithm alternates between two steps: (i) finding the best cluster centers for a given partition of a dataset into k clusters, and (ii) reassigning data items to the newly computed centers.

K -means is popular in numerous applications due to its simplicity and high speed for relatively large-scale datasets. The latter is achieved due to simplicity of finding the optimal cluster centers given a partition of data items into clusters. Indeed, for each cluster, the optimal center can explicitly be found as the mean of data items assigned to the cluster. This nice property is followed from the first order optimality conditions and makes the algorithm spent only $\mathcal{O}(m)$ time on this step. The property holds not for any distance measure. For example, in the closely related generalized multisource Weber problem that uses Euclidean distances instead of squared Euclidean distances, one has to solve a problem of finding optimal centers by an iterative procedure, known as Weiszfeld's algorithm.

As noted in [20], taking means as cluster centers in case of Euclidean distances is a widespread and hardly eradicated error.

It turns out that the mean (or center of gravity) turns out to be the optimal cluster center for a relatively large class of distance measures known as Bregman divergences, e.g. squared Euclidean distance. This result was proved in [2] and was shown that k -means type algorithms converge to a locally optimal solution in a finite number of steps if and only if the distance between data items and cluster centers is computed using a Bregman divergence.

Recent research efforts have been focused on studying the properties of k -means type algorithms with Bregman divergences and developing new ones. For example, in [26], the authors employ Bregman divergence to compute distances between data items and cluster centers in a multitask clustering setting and develop the alternate solution algorithms for the corresponding optimization problems. In [24], the authors extended the theory of Bregman divergences for the case of nondifferentiable convex functions and proposed an agglomerative clustering algorithm. Asymptotic properties of Bregman clustering and convergence of the clustering procedure as the number of centers or data items increases was studied in [15] and [5], respectively. Recently, robust Bregman clustering was introduced in [6] aimed at avoiding the main drawback for k -means type algorithms, sensitivity to noise. The authors also proposed a trimmed version of the Lloyd-type algorithm that is robust to noise and outliers.

In this paper we formulate the Bregman clustering problem as a non-convex optimization problem. We demonstrate that it can be represented as a DC (difference of convex) program. We propose several reductions of the Bregman clustering problem to a DC program where a non-convex objective function is minimized over a convex set. Using these formulations, we then develop two solution approaches based on a special global search strategy and global optimality conditions developed by A.S. Strekalovsky [21, 23] proved to be effective for practical problems, including machine learning tasks [7, 8] and industrial problems [11]. We also provide specific implementations of the developed approaches for some well-known Bregman divergences.

2 Bregman Divergences

A Bregman divergence or Bregman distance is a measure of difference between two points, defined in terms of a strictly convex function.

Let $F(\cdot) : \Omega \rightarrow \mathbb{R}$ be a continuously differentiable, strictly convex function defined on a closed convex set Ω .

Definition 1 [4]. *The Bregman distance associated with function $F(\cdot)$ for points $p, q \in \Omega$ is the difference between the value of function $F(\cdot)$ at point p and the value of the first-order Taylor expansion of $F(\cdot)$ around point q evaluated at point p :*

$$D_F(p, q) = F(p) - F(q) - \langle \nabla F(q), p - q \rangle, \quad (1)$$

where $\langle \cdot, \cdot \rangle$ stands for the scalar product of two vectors.

Squared Euclidean distance $D_F(x, y) = \|x - y\|^2$ is the simplest example of a Bregman distance induced by the convex function $F(x) = \|x\|^2$. Another example that is useful in many applications is the squared Mahalanobis distance, $D_F(x, y) = \frac{1}{2}\langle(x - y), Q(x - y)\rangle$ induced by the convex function $F(x) = \frac{1}{2}\langle x, Qx\rangle$ with positive defined matrix Q of appropriate dimension. The latter can be thought as a generalization of the squared Euclidean distance.

Bregman divergences have several well-known and useful properties [2]. In the paper, we use the following ones:

- 1) Non-negativity: $D_F(p, q) \geq 0$ for all p, q . This is a consequence of the convexity of function $F(\cdot)$.
- 2) Convexity: $D_F(p, q)$ is convex with respect to its first argument p , but not necessarily to the second argument q (see [3]).

Note that the Bregman divergence is not a proper distance metric, since it does not satisfy the triangle inequality and may also not be symmetric.

There are Necessary and Sufficient Conditions for a Bregman Divergence [2]. A divergence measure $D : \Omega \times \text{ri}(\Omega) \rightarrow [0, \infty)$, where $\text{ri}(\cdot)$ denotes the interior within the affine hull of the set Ω , is a Bregman divergence if and only if there exists $\alpha \in \text{ri}(\Omega)$ such that the function $F_\alpha(p) = D(p, \alpha)$ satisfies the following conditions:

1. $F_\alpha(\cdot)$ is strictly convex on Ω and differentiable on $\text{ri}(\Omega)$.
2. $D(p, q) = D_{F_\alpha}(p, q)$, $\forall p \in \Omega, q \in \text{ri}(\Omega)$ where $D_{F_\alpha}(\cdot)$ is the Bregman divergence associated with $F_\alpha(\cdot)$.

These conditions allow one to identify many other functions as Bregman divergences.

3 Problem Statement

Here, we formulate the Bregman clustering problem as a non-convex optimization problem. Given a finite set $J = \{1, \dots, m\}$ of data items, each of which is expressed by a feature vector $a^j \in \mathbb{R}^n$, $j \in J$. The goal is to find k cluster centers, such that the total sum of distances (dissimilarities) between data items and their closest centers is minimized. Obviously, the data items assigned to the same center form a cluster. As a measure of dissimilarity between a data item and its closest cluster center, we use a Bregman divergence.

Let us introduce the following binary variables

$$x_{ij} = \begin{cases} 1, & \text{if data item } j \text{ is assigned to cluster } i, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, \dots, k, j = 1, \dots, m.$$

which are often referred to as assignment variables.

We also suppose that the unknown locations of k cluster centers are decision variables $y^i \in \mathbb{R}^n$, $i = 1, \dots, k$. Obviously, we suppose that the number of data

items is greater than k , otherwise the problem is trivially solved. With these notations, we can formulate the following mixed integer program:

$$\sum_{i=1}^k \sum_{j=1}^m x_{ij} D_F(y^i, a^j) \downarrow \min_{(x,y)} \quad (2)$$

$$\sum_{i=1}^k x_{ij} = 1 \quad \forall j = 1, \dots, m; \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \dots, k; \forall j = 1, \dots, m. \quad (4)$$

The objective function (2) minimizes the sum of distances between data items and cluster centers, whereas constraints (3) guarantee that each data item is assigned to exactly one cluster.

For fixed centers y^i , the assignment variables x_{ij} take binary values in the corresponding optimal solution, since data items are always assigned to the closest cluster centers. Consequently, in our approach we consider a natural relaxation of (2)–(4) where the binary constraints $x_{ij} \in \{0, 1\}$ are replaced with $x_{ij} \in [0, 1]$, $i = 1, \dots, k$; $j = 1, \dots, m$. The resultant problem is to minimize a nonconvex function over a convex feasible set:

$$f(x, y) = \sum_{i=1}^k \sum_{j=1}^m x_{ij} D_F(y^i, a^j) \downarrow \min_{(x,y)} \quad x \in S, \quad y \in \mathbb{R}^{k \times n}, \quad (5)$$

where $S = \{x_{ij} \in [0, 1] : \sum_{i=1}^k x_{ij} = 1, j = 1, \dots, m\}$, $S \subset \mathbb{R}^{k \times m}$.

If we intend to solve the problem (5) by applying the global search theory [21, 23], we need an explicit DC representation of the nonconvex objective function.

4 DC Representations of the Objective Function

It is well-known that DC representation is not unique, and different DC decompositions generate various auxiliary convex problems. Here we propose several DC representations of the objective function in the problem (5).

Let us fix i and j , denote $x := x_{ij} \in \mathbb{R}$, $y := y^i \in \mathbb{R}^n$, $a := a^j \in \mathbb{R}^n$, and consider the following representation of one nonconvex term of the sum in the objective function $f(\cdot)$ of the problem (5):

$$x D_F(y, a) = \frac{1}{2}(x + D_F(y, a))^2 - \frac{1}{2}(x^2 + D_F^2(y, a)). \quad (6)$$

Since $D_F(y, a)$ is convex with respect to its first argument y , $D_F(y, a) \geq 0$ for all y, a , and $x + D_F(y, a) \geq 0$ for all $x \in [0, 1]$ then

$$\begin{aligned} g(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij} + D_F(y^i, a^j))^2, \\ h(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij}^2 + D_F^2(y^i, a^j)) \end{aligned} \quad (7)$$

are convex functions. Thus, we obtain the following d.c. representation:

$$f(x, y) = g(x, y) - h(x, y). \tag{8}$$

Another DC representation of the objective function can be obtained by using the definition of the Bregman distance associated with the continuously differentiable, strictly convex function $F(\cdot)$. Let us again fix i and j and consider one nonconvex term in the objective function of the problem (5):

$$\begin{aligned} x D_F(y, a) &= x(F(y) - F(a) - \langle \nabla F(a), y - a \rangle) \\ &= xF(y) - x\langle \nabla F(a), y \rangle - x(F(a) - \langle \nabla F(a), a \rangle). \end{aligned} \tag{9}$$

Obviously, the first two terms in the last part of (9) are nonconvex and one can decompose them in the following way ($x \in \mathbb{R}, y, a \in \mathbb{R}^n$):

$$xF(y) = \frac{1}{2}(x + F(y))^2 - \frac{1}{2}(x^2 + F^2(y)), \tag{10}$$

$$\begin{aligned} x\langle \nabla F(a), y \rangle &= \sum_{l=1}^n [\nabla F(a)]_l y_l x \\ &= \sum_{l=1}^n [\nabla F(a)]_l \left[\frac{1}{2}(x + y_l)^2 - \frac{1}{2}(x^2 + y_l^2) \right] \\ &= \frac{1}{2} \sum_{l=1}^n [\nabla F(a)]_l (x + y_l)^2 - \frac{1}{2} \sum_{l=1}^n [\nabla F(a)]_l (x^2 + y_l^2), \end{aligned} \tag{11}$$

where $[\nabla F(a)]_l$ is l th component of the gradient $\nabla F(a)$, $l = 1, \dots, n$.

Suppose that $F(y) \geq 0, x + F(y) \geq 0$. With these assumptions, we propose another representation of the objective function in (5) as the following difference of two sums of convex functions:

$$f(x, y) = \sum_{i=1}^k \sum_{j=1}^m g_{ij}(x, y) - \sum_{i=1}^k \sum_{j=1}^m h_{ij}(x, y), \tag{12}$$

where

$$\begin{aligned} g_{ij}(x, y) &= \frac{1}{2}(x_{ij} + F(y^i))^2 + \frac{1}{2} \sum_{l=1}^n [\nabla F(a^j)]_l (x_{ij}^2 + (y_l^i)^2) \\ &\quad - x_{ij}(F(a^j) - \langle \nabla F(a^j), a^j \rangle), \\ h_{ij}(x, y) &= \frac{1}{2}(x_{ij}^2 + F^2(y^i)) + \frac{1}{2} \sum_{l=1}^n [\nabla F(a^j)]_l (x_{ij} + y_l^i)^2, \\ &\quad i = 1, \dots, k, \quad j = 1, \dots, m. \end{aligned} \tag{13}$$

Remark 1. Note that functions $g_{ij}(\cdot)$ in (13) are convex functions if and only if the non-negativity condition on the value of $F(\cdot)$ is satisfied, and moreover $x_{ij} + F(\cdot) \geq 0 \forall i, j$.

The explicit DC representations (7)–(8) and (12)–(13) of the nonconvex objective function allow us to generate different auxiliary convex (linearized) problems and construct various methods for local search.

5 Local Search

In order to find a local solution of the problem (5), which is turned out to be the following DC minimization problem

$$f(x, y) = g(x, y) - h(x, y) \downarrow \min_{(x, y)}, \quad x \in S, \quad y \in \mathbb{R}^{k \times n}, \quad (\mathcal{P})$$

we apply the well-known DC Algorithm [13,21,22]. It consists of linearizing, at a current point, the function $h(\cdot)$ which defines the basic nonconvexity of Problem (P). The resultant convex approximation of the objective function $f(\cdot)$ obtained by replacing the nonconvex part with its linearization is then minimized. It is easy to see that such an approach allow finding local solutions by employing conventional convex optimization techniques [18].

The scheme of DC Algorithm for Problem (P) is the following. We start with an initial point $(x^0, y^0) : y^0 \in \mathbb{R}^{k \times n}, x^0 \in S$. Suppose a point $(x^s, y^s), x^s \in S$, is provided. Then, we find (x^{s+1}, y^{s+1}) as an approximate solution to the linearized problem

$$\Phi_s(x, y) = g(x, y) - \langle \nabla h(x^s, y^s), (x, y) \rangle \downarrow \min_{(x, y)}, \quad x \in S, \quad y \in \mathbb{R}^{k \times n}. \quad (\mathcal{P}\mathcal{L}_s)$$

It means that the next iteration (x^{s+1}, y^{s+1}) satisfies the following inequality:

$$\begin{aligned} & g(x^{s+1}, y^{s+1}) - \langle \nabla h(x^s, y^s), (x^{s+1}, y^{s+1}) \rangle \\ & \leq \inf_{\substack{x \in S \\ y \in \mathbb{R}^{k \times n}}} \{g(x, y) - \langle \nabla h(x^s, y^s), (x, y) \rangle\} + \delta_s, \end{aligned} \quad (14)$$

where $\delta_s \geq 0, s = 0, 1, 2, \dots; \sum_{s=0}^{\infty} \delta_s < \infty$.

As it was proven in [22], the point $(x^*, y^*), x^* \in S, y^* \in \mathbb{R}^{k \times n}$, of the sequence $\{(x^s, y^s)\}$ generated by the method, is a solution to Linearized Problem $(\mathcal{P}\mathcal{L}_*)$, and stationary (critical) point of Problem (P).

Note that Linearized Problem $(\mathcal{P}\mathcal{L}_s)$ is convex, whereas Problem (P) is non-convex.

As it was suggested in [21,22], one of the following inequalities can be employed as a stopping criterion:

$$\begin{aligned} & f(x^s, y^s) - f(x^{s+1}, y^{s+1}) \leq \frac{\tau}{2}, \\ & \Phi_s(x^s, y^s) - \Phi_s(x^{s+1}, y^{s+1}) \triangleq g(x^s, y^s) - g(x^{s+1}, y^{s+1}) \\ & \quad + \langle \nabla h(x^s, y^s), (x^{s+1}, y^{s+1}) - (x^s, y^s) \rangle \leq \frac{\tau}{2}, \end{aligned} \quad (15)$$

where τ is a given accuracy. Therefore, if $\delta_s \leq \frac{\tau}{2}$, the point (x^s, y^s) is a τ -solution to Problem $(\mathcal{P}\mathcal{L}_s)$.

Applying the DC representation (7)–(8), we have to solve a series of the following linearized problems:

$$\frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij} + D_F(y^i, a^j))^2 - \langle \nabla h(x^s, y^s), (x, y) \rangle \downarrow \min_{(x,y)}, x \in S, y \in \mathbb{R}^{k \times n}, \quad (16)$$

where $\nabla h(x^s, y^s) = (\nabla_x h(x^s, y^s), \nabla_y h(x^s, y^s)) \in \mathbb{R}^{k \times m + k \times n}$ with the following components

$$\begin{aligned} [\nabla_x h(x^s, y^s)]_{ij} &= x_{ij}^s, \quad i = 1, \dots, k, \quad j = 1, \dots, m; \\ [\nabla_y h(x^s, y^s)]_{il} &= \sum_{j=1}^m D_F(y^{si}, a^j) [\nabla_y D_F(y^{si}, a^j)]_l, \\ & \quad i = 1, \dots, k, \quad l = 1, \dots, n. \end{aligned} \quad (17)$$

Using second obtained DC representation (12)–(13) of the objective function $f(x, y)$ in the problem (5), we have to solve a series of the following linearized problems:

$$\sum_{i=1}^k \sum_{j=1}^m g_{ij}(x, y) - \sum_{i=1}^k \sum_{j=1}^m \langle \nabla h_{ij}(x^s, y^s), (x, y) \rangle \downarrow \min_{(x,y)}, x \in S, y \in \mathbb{R}^{k \times n}, \quad (18)$$

where the functions $g_{ij}(x, y)$, $i = 1, \dots, k$, $j = 1, \dots, m$, are given by (13), $\nabla h_{ij}(x^s, y^s) = (\nabla_x h_{ij}(x^s, y^s), \nabla_y h_{ij}(x^s, y^s)) \in \mathbb{R}^{k \times m + k \times n}$ with the following components

$$\begin{aligned} [\nabla_x h_{ij}(x^s, y^s)]_{ij} &= x_{ij}^s + \sum_{l=1}^n [\nabla F(a^j)]_l (x_{ij}^s + y_l^{si}), \\ [\nabla_y h_{ij}(x^s, y^s)]_{il} &= F(y^{si}) [\nabla F(y^{si})]_l + [\nabla F(a^j)]_l (x_{ij}^s + y_l^{si}), \\ & \quad i = 1, \dots, k, \quad j = 1, \dots, m, \quad l = 1, \dots, n. \end{aligned} \quad (19)$$

Therefore, the linearized problems (16) and (18) allow us to develop two various local search methods which may converge to different stationary points of the problem (5).

We denote the solution obtained by the local search method as $z = (x, y)$ ($z \in \text{Sol}(\mathcal{P})$), and in Sect. 7, we will show how to escape from local solutions provided by the local search.

6 Bregman Divergence Examples

Let us illustrate the obtained DC representations as well as statements of Linearized Problems ($\mathcal{P}_{\mathcal{L}_s}$) on some well-known strictly convex functions associated with Bregman distance.

6.1 Squared Euclidean Distance

Squared Euclidean distance is perhaps the simplest and most widely used Bregman divergence. The underlying function $F(p) = \langle p, p \rangle$, is strictly convex, differentiable on \mathbb{R}^n , $\nabla F(p) = 2p$, and

$$\begin{aligned} D_F(p, q) &= \langle p, p \rangle - \langle q, q \rangle - \langle \nabla F(q), p - q \rangle = \langle p, p \rangle - \langle q, q \rangle - \langle 2q, p - q \rangle \\ &= \langle p - q, p - q \rangle = \|p - q\|^2. \end{aligned}$$

In this case, according to (7)–(8) we get the following DC representation of the objective function in (5):

$$\begin{aligned} f(x, y) &= g(x, y) - h(x, y), \\ g(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij} + \|y^i - a^j\|^2)^2, \\ h(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij}^2 + \|y^i - a^j\|^4). \end{aligned} \tag{20}$$

If the DC representation (12)–(13) is used, we get the following functions

$$\begin{aligned} f(x, y) &= \sum_{i=1}^k \sum_{j=1}^m g_{ij}(x, y) - \sum_{i=1}^k \sum_{j=1}^m h_{ij}(x, y), \\ g_{ij}(x, y) &= \frac{1}{2} (x_{ij} + \|y^i\|^2)^2 + \sum_{l=1}^n a_l^j (x_{ij}^2 + (y_l^i)^2) + x_{ij} \|a^j\|^2, \\ h_{ij}(x, y) &= \frac{1}{2} (x_{ij}^2 + \|y^i\|^4) + \sum_{l=1}^n a_l^j (x_{ij} + y_l^i)^2, \\ & \quad i = 1, \dots, k, \quad j = 1, \dots, m. \end{aligned} \tag{21}$$

The obtained representations (20) and (21) are different from the decomposition used in [9, 10].

Remark 2. Since, the squared Mahalanobis distance,

$$D_F(p, q) = \frac{1}{2} \langle (p - q), Q(p - q) \rangle$$

which is generated by the convex function $F(p) = \frac{1}{2} \langle p, Qp \rangle$, is a generalization of the above squared Euclidean distance, it is easy to get DC representation in the case of the squared Mahalanobis distance based on (20) and (21).

Applying the DC representation (20), using (16) and (17), we obtain the following linearized problem:

$$\left. \begin{aligned} & \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m (x_{ij} + \|y^i - a^j\|^2)^2 - \langle x^s, x \rangle - 2 \sum_{i=1}^k \sum_{j=1}^m \|y^{si} - a^j\|^3 \sum_{l=1}^n y_l^i \downarrow \min_{(x,y)} \\ & \quad x \in S, \quad y \in \mathbb{R}^{k \times n}. \end{aligned} \right\} \tag{22}$$

The linearized problem (22) is convex, and although its objective function is more complex than the quadratic one, (22) can be solved using suitable convex optimization methods or software.

Similarly, using convex functions $g_{ij}(\cdot)$ from the DC representation (21), we get the linearized problem (18), where, according to (19), the gradients

$$\nabla h_{ij}(x^s, y^s) = (\nabla_x h_{ij}(x^s, y^s), \nabla_y h_{ij}(x^s, y^s)) \in \mathbb{R}^{k \times m + k \times n}$$

have the following components

$$\begin{aligned} [\nabla_x h_{ij}(x^s, y^s)]_{ij} &= x_{ij}^s + 2 \sum_{l=1}^n a_l^j (x_{ij}^s + y_l^{si}), \\ [\nabla_y h_{ij}(x^s, y^s)]_{il} &= 2 \|y^{si}\|^3 + 2a_l^j (x_{ij}^s + y_l^{si}), \\ i &= 1, \dots, k, \quad j = 1, \dots, m, \quad l = 1, \dots, n. \end{aligned} \tag{23}$$

6.2 Generalized I-Divergence

Another widely used Bregman divergence is the so-called generalized I-divergence.

If $p \in \mathbb{R}_+^n$, $F(p) = \sum_{l=1}^n p_l \ln p_l = \langle p, \ln p \rangle$ is a convex function, $\nabla F(p) = \ln p + \mathbb{1}$, where $\mathbb{1} = (1, \dots, 1) \in \mathbb{R}^n$. The corresponding Bregman divergence is

$$\begin{aligned} D_F(p, q) &= \langle p, \ln p \rangle - \langle q, \ln q \rangle - \langle \nabla F(q), p - q \rangle \\ &= \langle p, \ln p \rangle - \langle q, \ln q \rangle - \langle \ln p + \mathbb{1}, p - q \rangle \\ &= (\langle p, \ln p \rangle - \langle p, \ln q \rangle) - \langle \mathbb{1}, p - q \rangle = \left\langle p, \ln \frac{p}{q} \right\rangle - \langle \mathbb{1}, p - q \rangle. \end{aligned} \tag{24}$$

Therefore, using (8) we get the following functions $g(\cdot)$ and $h(\cdot)$ in DC representation (7) of the objective function $f(\cdot)$ in the problem (5):

$$\begin{aligned} g(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m \left(x_{ij} + \left\langle y^i, \ln \frac{y^i}{a^j} \right\rangle - \langle \mathbb{1}, y^i - a^j \rangle \right)^2, \\ h(x, y) &= \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^m \left(x_{ij}^2 + \left(\left\langle y^i, \ln \frac{y^i}{a^j} \right\rangle - \langle \mathbb{1}, y^i - a^j \rangle \right)^2 \right). \end{aligned} \tag{25}$$

To construct the linearized problem (16) in this case, the components of the gradient $\nabla h(x^s, y^s) = (\nabla_x h(x^s, y^s), \nabla_y h(x^s, y^s))$ can be calculated as follows:

$$\begin{aligned} [\nabla_x h(x^s, y^s)]_{ij} &= x_{ij}^s, \quad i = 1, \dots, k, \quad j = 1, \dots, m; \\ [\nabla_y h(x^s, y^s)]_{il} &= \sum_{j=1}^m \left(\left\langle y^{si}, \ln \frac{y^{si}}{a^j} \right\rangle - \langle \mathbb{1}, y^{si} - a^j \rangle \right) \ln \frac{y_l^{si}}{a_l^j}, \\ i &= 1, \dots, k, \quad l = 1, \dots, n. \end{aligned} \tag{26}$$

If the DC representation (12)–(13) is considered, the following functions are constructed:

$$\begin{aligned}
 g_{ij}(x, y) &= \frac{1}{2} (x_{ij} + \langle y^i, \ln y^i \rangle)^2 + \frac{1}{2} \sum_{l=1}^n \left[(\ln a_l^j + 1) (x_{ij}^2 + (y_l^i)^2) + 2x_{ij} a_l^j \right], \\
 h_{ij}(x, y) &= \frac{1}{2} (x_{ij}^2 + \langle y^i, \ln y^i \rangle^2) + \frac{1}{2} \sum_{l=1}^n (\ln a_l^j + 1) (x_{ij} + y_l^i)^2, \\
 & \quad i = 1, \dots, k, \quad j = 1, \dots, m.
 \end{aligned} \tag{27}$$

However, it is worth noting that functions (27) are convex if and only if $\langle y^i, \ln y^i \rangle = \sum_{l=1}^n y_l^i \ln y_l^i > 0 \quad \forall i = 1, \dots, k$. In this case only, substituting the convex functions (27) into (12), we obtain a DC representation of the objective function $f(\cdot)$ in the problem (5). The components of gradients $\nabla h_{ij}(x^s, y^s)$ for the linearized problem (18) are obtained by the following way

$$\begin{aligned}
 [\nabla_x h_{ij}(x^s, y^s)]_{ij} &= x_{ij}^s + \sum_{l=1}^n (\ln a_l^j + 1) (x_{ij}^s + y_l^{si}), \\
 [\nabla_y h_{ij}(x^s, y^s)]_{il} &= \langle y^i, \ln y^i \rangle (\ln y_l^{si} + 1) + (\ln a_l^j + 1) (x_{ij}^s + y_l^{si}), \\
 & \quad i = 1, \dots, k, \quad j = 1, \dots, m, \quad l = 1, \dots, n.
 \end{aligned} \tag{28}$$

In the next section, we describe how the aforementioned linearized problems can be used in the so-called global search scheme.

7 Global Search

According to the Global Optimality Conditions [21, 23] that form the basis of the Global Search Theory for DC optimization problems developed by A.S. Strekalovsky, whether a given point $z = (x, y)$ is a global solution to Problem (P) is determined by solving a family of the following convex linearized problems ($\mathcal{PL}(w)$)

$$g(x, y) - \langle \nabla h(w), (x, y) \rangle \downarrow \min, \quad x \in S, \quad y \in \mathbb{R}^{k \times n}, \tag{29}$$

They depend on the “perturbation” parameters (w, β) satisfying $h(w) = \beta - \zeta$ with $\zeta = f(z)$. The problem (29) can be solved by any conventional convex optimization method [18]. If the Optimality Conditions are violated at a given triple $(\tilde{w}, \tilde{\beta}, u)$, $u = (u^1, u^2) : u^1 \in S, u^2 \in \mathbb{R}^{k \times n}, h(\tilde{w}) = \tilde{\beta} - \zeta$, i.e.

$$g(u) - \tilde{\beta} < \langle \nabla h(\tilde{w}), u - \tilde{w} \rangle,$$

then due to convexity of $h(\cdot)$ we get

$$\begin{aligned}
 g(u) &< \tilde{\beta} + h(u) - h(\tilde{w}), \\
 g(u) &< h(\tilde{w}) + \zeta + h(u) - h(\tilde{w}), \\
 f(u) &= g(u) - h(u) < \zeta = f(z)
 \end{aligned}$$

and conclude that $z = (x, y) : x \in S$, is not optimal.

Moreover, it is not necessary to investigate all pairs of (w, β) : $\zeta_p = h(w) - \beta$, on each level $\zeta_p = f(z^p)$, $p = 1, 2, \dots$, but it is sufficient to determine the violation of the Optimality Conditions only for one pair $(\tilde{w}, \tilde{\beta})$ and $u = (u^1, u^2) : u^1 \in S, u^2 \in \mathbb{R}^{k \times n}$.

The properties of the Optimality Conditions allow developing an algorithm (a scheme) for solving DC minimization problems. The Global Search Scheme comprises two principal stages:

- I. Local search to find an approximate local minimizer z^p with the value corresponding to the objective function $\zeta_p = f(z^p)$;
- II. Procedures of escaping from local pits, which are based on the Optimality Conditions and can be represented as a chain of the following operations [21, 23]:
 1. Choose a number (“perturbation” parameter) β :

$$\inf(g, S) \leq \beta \leq \sup(g, S).$$

2. Construct a finite approximation

$$R_p(\beta) = \{w^1, \dots, w^{N_p} \mid h(w^t) = \beta - \zeta_p, t = 1, \dots, N_p\}$$

of the level surface $\{h(x, y) = \beta - \zeta_p\}$ of the function $h(\cdot)$.

3. Find a δ_p -solution \bar{u}^t of the following Linearized Problem:

$$g(x, y) - \langle \nabla h(w^t), (x, y) \rangle \downarrow \min_{(x, y)}, x \in S, y \in \mathbb{R}^{k \times n}. \quad (PL_t)$$

4. Starting from the point \bar{u}^t , find a local minimizer u^t with a local search method.

The procedures of escaping from local pits gave us with the triple (w^t, β, u^t) . If the Optimality Conditions are violated at the constructed triple we conclude that $f(u^t) < f(z^p)$, set $z^{p+1} := u^t$ and try to violate the Optimality Conditions again with new value $\beta + \Delta\beta$.

In the described Global Search Scheme, one can choose the local search method (for instance, from Sect. 5), methods for varying the parameter β (for example, $\Delta\beta$ is chosen by dividing a segment $[\inf(g, S), \sup(g, S)]$ into q equal parts) and constructing an approximation of the level surface of the convex function $h(\cdot)$. Therefore, the scheme can be clarified based on the properties of the problem in question.

Note that constructing an approximation is considered as one of principal steps. There are many ways and techniques to construct the approximation of the level surface of the convex function $h(\cdot)$ which generates the basic nonconvexity in Problem (P). The approximation $R_p(\beta)$ of the level surface $\{h(\cdot) = \beta - \zeta\}$ for each pair (β, ζ_p) , $\zeta_p = f(z^p)$, may be constructed, for instance, by the following rule [7, 8, 10]:

$$w^{il} = z^i + \mu_{il}e^l, \quad i = 1, \dots, k, \quad l = 1, \dots, n, \quad (30)$$

where e^l is the standard vector from the Euclidean basis of \mathbb{R}^n .

For the quadratic function $h(\cdot)$ the search of μ_{il} is simple and, actually, analytical (i.e. it is reduced to solving the quadratic equation of one variable μ_{il}). For non-quadratic functions, the search for such coefficients μ_{il} can be rather complicated. However, it is possible to suggest other approaches to choosing the points w^{il} of the approximation $R_p(\beta)$ which differ from (30). In any case, the approximation must be representative enough to decide whether the current point z^p is a global solution or not.

Using this approach, based on Optimality Conditions, we developed and tested the algorithm for finding quality clustering solutions in minimum-sum-of-squares (k-means) clustering problem [10]. In our computational experiments we demonstrated that the proposed approach is competitive with conventional k-means heuristics.

8 Conclusion

In this paper, we addressed the problem of clustering with Bregman divergences from the mathematical programming point of view. We proposed two possible reductions of the problem to a DC program, where a DC function is minimized over a convex set. We developed two variants of local search algorithms based on these formulations and considered their particular implementations for several variants of Bregman divergences. For each particular case, the algorithms may demonstrate different effectiveness and may converge to different critical points since obtaining linearized problems differ from each other.

Our further research will be focused on implementation and testing the developed algorithms on both real and synthetic datasets.

References

1. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. *Mach. Learn.* **75**, 245–248 (2009). <https://doi.org/10.1007/s10994-009-5103-0>
2. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. *J. Mach. Learn. Res.* **6**, 1705–1749 (2005)
3. Bauschke, H., Borwein, J.: Joint and separate convexity of the Bregman distance. In: *Studies in Computational Mathematics*, vol. 8, pp. 23–36 (2001). [https://doi.org/10.1016/S1570-579X\(01\)80004-5](https://doi.org/10.1016/S1570-579X(01)80004-5)
4. Bregman, L.M.: The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *Comput. Math. Math. Phys.* **7**, 200–217 (1967)
5. Fischer, A.: Quantization and clustering with Bregman divergences. *J. Multivar. Anal.* **101**(9), 2207–2221 (2010). <https://doi.org/10.1016/j.jmva.2010.05.008>
6. Fischer, A., Levrard, C., BréchetEAU, C.: Robust Bregman clustering (2020)
7. Gaudioso, M., Gruzdeva, T.V., Strelalovsky, A.S.: On numerical solving the spherical separability problem. *J. Glob. Optim.* **66**(1), 21–34 (2015). <https://doi.org/10.1007/s10898-015-0319-y>
8. Gruzdeva, T.V.: On a continuous approach for the maximum weighted clique problem. *J. Glob. Optim.* **56**(3), 971–981 (2013)

9. Gruzdeva, T.V., Ushakov, A.V.: A computational study of the DC minimization global optimality conditions applied to k-means clustering. In: Olenev, N.N., Evtushenko, Y.G., Jaćimović, M., Khachay, M., Malkova, V. (eds.) OPTIMA 2021. LNCS, vol. 13078, pp. 79–93. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91059-4_6
10. Gruzdeva, T.V., Ushakov, A.V.: K-means clustering via a nonconvex optimization approach. In: Pardalos, P., Khachay, M., Kazakov, A. (eds.) MOTOR 2021. LNCS, vol. 12755, pp. 462–476. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77876-7_31
11. Gruzdeva, T.V., Ushakov, A.V., Enkhbat, R.: A biobjective DC programming approach to optimization of rougher flotation process. *Comput. Chem. Eng.* **108**, 349–359 (2018). <https://doi.org/10.1016/j.compchemeng.2017.10.001>
12. Hansen, P., Jaumard, B.: Cluster analysis and mathematical programming. *Math. Program.* **79**(1–3), 191–215 (1997)
13. Hoai An, L.T., Tao, P.D.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Ann. Oper. Res.* **133**, 23–46 (2005)
14. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.* **31**(8), 651–666 (2010)
15. Liu, C., Belkin, M.: Clustering with Bregman divergences: an asymptotic analysis. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates Inc., New York (2016)
16. Lloyd, S.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982). <https://doi.org/10.1109/TIT.1982.1056489>
17. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. *Theor. Comput. Sci.* **442**, 13–21 (2012). <https://doi.org/10.1016/j.tcs.2010.05.034>. Special Issue on the Workshop on Algorithms and Computation (WALCOM 2009)
18. Nocedal, J., Wright, S.J.: *Numerical Optimization. Operations Research and Financial Engineering*, 2nd edn. Springer, New York (2006). <https://doi.org/10.1007/978-0-387-40065-5>
19. Pi, J., Wang, H., Pardalos, P.M.: A dual reformulation and solution framework for regularized convex clustering problems. *Eur. J. Oper. Res.* **290**(3), 844–856 (2021)
20. Plastria, F.: The Weiszfeld algorithm: proof, amendments, and extensions. In: Eiselt, H.A., Marianov, V. (eds.) *Foundations of Location Analysis. ISORMS*, vol. 155, pp. 357–389. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-7572-0_16
21. Strekalovsky, A.S.: On solving optimization problems with hidden nonconvex structures. In: Rassias, T.M., Floudas, C.A., Butenko, S. (eds.) *Optimization in Science and Engineering*, pp. 465–502. Springer, New York (2014). https://doi.org/10.1007/978-1-4939-0808-0_23
22. Strekalovsky, A.S.: On local search in d.c. optimization problems. *Appl. Math. Comput.* **255**, 73–83 (2015)
23. Strekalovsky, A.: On the minimization of the difference of convex functions on a feasible set. *Comput. Math. Math. Phys.* **43**, 380–390 (2003)
24. Telgarsky, M., Dasgupta, S.: Agglomerative Bregman clustering (2012)
25. Vasilyev, I., Ushakov, A.V.: Discrete facility location in machine learning. *Diskretn. Anal. Issled. Oper.* **28**(4), 5–60 (2021). <https://doi.org/10.33048/daio.2021.28.714>
26. Zhang, J., Zhang, C.: Multitask Bregman clustering. *Neurocomputing* **74**(10), 1720–1734 (2011). <https://doi.org/10.1016/j.neucom.2011.02.004>



On Solving One Spectral Problem

Vladimir Zubov^{1,2}(✉)  and Alla Albu¹ 

¹ Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russia

vladimir.zubov@mail.ru

² Moscow Institute of Physics and Technology (National Research University), Moscow, Russia

Abstract. When designing and optimizing modern semiconductor heterostructures, mathematical models are used that reflect the quantum mechanical nature of the behavior of charge carriers. At the nanoscale level, the applied mathematical model is a coupled system of Schrödinger and Poisson equations. As a result of solving these equations, data are obtained on the wave functions and the density distribution of charge carriers across the layered structure. The greatest computational costs when using such a model are associated with the solution of the Schrödinger equation. In this paper, we compare different methods for solving the spectral problem. A method based on the Prufer transformation of the Schrödinger equation, a variational method, and a method for solving the spectral problem for a symmetric sparse matrix of a band structure (a discrete analogue of the Schrödinger equation) are considered.

Keywords: Spectral problem · Prufer transformation · Variational method · Numerical algorithms

1 Introduction

Recently, mathematical modeling of processes in semiconductor heterostructures has become a very effective tool for determining the key parameters of such structures. This turned out to be possible due to the widespread introduction of numerical methods in materials science. In particular, numerical methods are actively used to determine the concentration profile of free charge carriers in doped semiconductor heterostructures containing a quantum well [1]. In this case, the volt-farad characteristic of a heterostructure with a quantum well is calculated using a numerical self-consistent solution of the Poisson and Schrödinger equations.

When finding a self-consistent solution of the Poisson and Schrödinger equations, the main computational costs are associated with the solution of the Schrödinger equation. Therefore, the use of an efficient algorithm for solving the spectral problem determines the effectiveness of solving the entire problem

The research was supported by Russian Science Foundation (project No. 21-71-30005).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 153–166, 2022.

https://doi.org/10.1007/978-3-031-09607-5_11

as a whole. It should also be taken into account that, in mathematical modeling of the distribution of charge carriers in semiconductor nanostructures, the main contribution to the distribution is made by the first eigenfunctions of the spectral problem. The accuracy and efficiency of determining the first eigenvalues and eigenfunctions is a key element in the mathematical modeling of modern nanostructures.

The paper studies the possibility of using variational methods in solving the spectral problem. This approach seems preferable for solving multidimensional problems. Using the example of solving a one-dimensional spectral problem, the variational method under consideration is compared with two other numerical algorithms designed to determine the first eigenvalues and eigenfunctions.

2 Formulation of the Problem

Determining the wave functions of electrons in multilayer (with different parameters in each layer) heterostructures using the stationary Schrödinger equation reduces to the following spectral problem.

Let the functions $p(x)$ and $q(x)$ be given on the interval $x \in [a, b]$ and have the following properties:

- 1) there are K points $\{X_1, X_2, \dots, X_K\}$ at which the functions $p(x)$ and $q(x)$ have discontinuities of the first kind;
- 2) these points divide the interval (a, b) into $(K + 1)$ non-intersecting subintervals $(X_0, X_1), (X_1, X_2), \dots, (X_K, X_{K+1})$, so that $[a, b] = \bigcup_{n=0}^K [X_n, X_{n+1}]$, and $X_0 = a, X_{K+1} = b$.
- 3) on each subsegment $[X_n, X_{n+1}]$, $n = 0, \dots, K$ is true $p(x) \in C^1([X_n, X_{n+1}])$, $p(x) \geq p_0 > 0$, $q(x) \in C([X_n, X_{n+1}])$, $q(x) \geq 0$.

The following spectral problem is considered: to find a number λ and a function $u(x) \in C([a, b])$, on each subsegment $[X_n, X_{n+1}]$, $n = 0, \dots, K$, satisfying the following conditions

$$u(x) \in C^2([X_n, X_{n+1}]),$$

$$-\frac{d}{dx} \left(p(x) \frac{du(x)}{dx} \right) + q(x) \cdot u(x) = \lambda \cdot u(x), \quad x \in (X_n, X_{n+1}), \quad (1)$$

$$u(a) = 0, \quad u(b) = 0, \quad (2)$$

$$u(x) \neq 0, \quad x \in [a, b], \quad (3)$$

and, at the same time, at the discontinuity points the conjugation conditions must be satisfied

$$u(x)|_{x=X_n-0} = u(x)|_{x=X_n+0}, \quad \left[p(x) \frac{du(x)}{dx} \right]_{x=X_n-0} = \left[p(x) \frac{du(x)}{dx} \right]_{x=X_n+0}, \quad (4)$$

$$(n = 1, \dots, K).$$

Let us introduce the operator $L(u)$. The domain of definition $D(L)$ of the operator $L(u)$ consists of functions of the class $u(x) \in C^2([X_n, X_{n+1}]) \cap C([a, b])$, ($n = 1, \dots, K$), satisfying conditions (2), (4). The operator $L(u)$ associates each function $u(x) \in D(L)$ with a piecewise-continuous function

$$L(u) = -\frac{d}{dx} \left(p(x) \frac{du(x)}{dx} \right) + q(x) \cdot u(x) \in L_2(a, b).$$

This operator has the following properties [2]:

- 1) the domain of definition of the operator $L(u)$ is dense in $L_2(a, b)$;
- 2) it is self-adjoint with respect to the scalar product of space $L_2(a, b)$, i.e. for all $u \in D(L)$ and $\nu \in D(L)$ the Lagrange identity $(Lu, \nu) = (u, L\nu)$ is valid;
- 3) it is positive definite, i.e. the inequality $(Lu, u) \geq \gamma(u, u)$ with $\gamma > 0$ is valid for all $u \in D(L)$.

Considering these properties of the operator $L(u)$, we construct the energy space H_L . To do this, on the set of functions from $D(L)$ we introduce the energy scalar product

$$[u, \nu] = (Lu, \nu) = \int_a^b \left(p(x) \cdot \frac{du(x)}{dx} \cdot \frac{d\nu(x)}{dx} + q(x) \cdot u(x) \cdot \nu(x) \right) dx$$

and the energy norm $\|u\|_* = [u, u]^{1/2}$. Let's replenish $D(L)$ in the norm $\|\cdot\|_*$, i.e. we add to $D(L)$ limit points of all possible fundamental sequences $\{u_k\} \in D(L)$ in the norm $\|\cdot\|_*$. Given the properties of the operator $L(u)$, we can extend it to the space H_L .

The spectral problem formulated above is equivalent to the following minimization problem (see [2–4]):

- a) first eigenvalue: among the functions $u(x) \in H_L, \|u\|_{L_2} > 0$ find the function $u_1(x)$ that minimizes the functional $\frac{[u, u]}{(u, u)}$, i.e.

$$\lambda_1 = \min_{\substack{u \in H_L, \\ \|u\|_{L_2} > 0}} \frac{(Lu, u)}{(u, u)}, \tag{5}$$

- b) k-th eigenvalue: among the functions $u(x) \in H_L$ satisfying the conditions $(u, u_j) = 0, (j = 1, \dots, k - 1), \|u\|_{L_2} > 0$, find the function $u_k(x)$ that minimizes the functional $\frac{[u, u]}{(u, u)}$, i.e.

$$\lambda_k = \min_{\substack{u \in H_L, \\ (u, u_j) = 0, j < k \\ \|u\|_{L_2} > 0}} \frac{(Lu, u)}{(u, u)}. \tag{6}$$

3 Variational Algorithm for Solving the Spectral Problem

For the numerical solution of the spectral problem, we will use the variational-difference method (modified Ritz process [5]). We will look for the minimum of the functional (5) in a subspace H_L^h of the space H_L . We define a subspace H_L^h as a linear shell stretched over a system of basis functions $\varphi_i(x) \in H_L$, ($i = 1, \dots, N - 1$). We define basis functions $\varphi_i(x)$ as follows. Let us introduce a spatial grid (generally non-uniform). On the segment $[a, b]$ we choose a system of “reference” points $\{x_i\}_{i=0}^N$ so that $x_0 = a$, $x_N = b$, $x_i < x_{i+1}$ for all $0 \leq i < N$, and each discontinuity point of the functions $p(x)$ and $q(x)$ coincides with one of the reference points. In this case $h_{i-1/2}$ is the distance between the reference points x_{i-1} and x_i , i.e. $h_{i-1/2} = x_i - x_{i-1}$, $i = \overline{1, N}$. On each segment $[x_{i-1}, x_i]$, $i = \overline{1, N}$ we define two auxiliary functions

$$\omega_{i-1/2}^R(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}}, \quad \omega_{i-1/2}^L(x) = \frac{x_i - x}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i].$$

Basis functions $\varphi_i(x) \in H_L$, ($i = 1, \dots, N - 1$) are piecewise linear functions of the form

$$\varphi_i(x) = \begin{cases} \omega_{i-1/2}^R(x) = \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x \in [x_{i-1}, x_i], \\ \omega_{i+1/2}^L(x) = \frac{x_{i+1}-x}{x_{i+1}-x_i}, & x \in [x_i, x_{i+1}], \\ 0, & x \notin [x_{i-1}, x_{i+1}]. \end{cases}$$

Then each function can be represented as

$$u(x) = \sum_{i=1}^{N-1} c_i \varphi_i(x),$$

in this case, conditions (2) will be satisfied automatically. The functionals $[u, u] = (Lu, u)$ and (u, u) are reduced to functions of $(N - 1)$ variables c_1, \dots, c_{N-1} . Let us find the form of these functions.

$$1) [u, u] = (Lu, u) = \left(L \sum_{i=1}^{N-1} c_i \varphi_i(x), \sum_{j=1}^{N-1} c_j \varphi_j(x) \right) = \sum_{i,j=1}^{N-1} c_i c_j (L\varphi_i, \varphi_j).$$

Taking into account that $(L\varphi_i, \varphi_j) = 0$ for $|i - j| \geq 2$, after simple transformations we obtain

$$[u, u] = (Lu, u) = \sum_{i=1}^N \left(\frac{c_i^2 - 2c_i c_{i-1} + c_{i-1}^2}{h_{i-\frac{1}{2}}^2} p_{i-\frac{1}{2}} + c_{i-1}^2 q_{i-\frac{1}{2}}^{LL} + 2c_{i-1} c_i q_{i-\frac{1}{2}}^{LR} + c_i^2 q_{i-\frac{1}{2}}^{RR} \right). \quad (7)$$

In relation (7), it should be assumed that the following notations are used:

$$p_{i-\frac{1}{2}} = \int_{x_{i-1}}^{x_i} p(x) dx, \quad q_{i-\frac{1}{2}}^{\alpha\beta} = \int_{x_{i-1}}^{x_i} q(x) \cdot \omega_{i-\frac{1}{2}}^\alpha(x) \cdot \omega_{i-\frac{1}{2}}^\beta(x) dx, \quad \alpha, \beta = (L, R). \quad (8)$$

$$2) (u, u) = \left(L \sum_{i=1}^{N-1} c_i \varphi_i(x), \sum_{j=1}^{N-1} c_j \varphi_j(x) \right) = \sum_{i,j=1}^{N-1} c_i c_j (\varphi_i, \varphi_j).$$

By analogy with how it was done for the functional $[u, u] = (Lu, u)$, we get

$$(u, u) = \sum_{i=1}^N \left(c_{i-1}^2 \tilde{q}_{i-\frac{1}{2}}^{LL} + 2c_{i-1}c_i \tilde{q}_{i-\frac{1}{2}}^{LR} + c_i^2 \tilde{q}_{i-\frac{1}{2}}^{RR} \right), \tag{9}$$

where $\tilde{q}_{i-\frac{1}{2}}^{\alpha\beta} = \int_{x_{i-1}}^{x_i} \omega_{i-\frac{1}{2}}^\alpha(x) \cdot \omega_{i-\frac{1}{2}}^\beta(x) dx$ is the value $q_{i-\frac{1}{2}}^{\alpha\beta}$ calculated at $q(x) \equiv 1$.

If the functions $p(x)$ and $q(x)$ are given analytically, then it is desirable to calculate exactly the integrals appearing in expressions (8). In the general case, to calculate the constants $p_{i-\frac{1}{2}}$, $q_{i-\frac{1}{2}}^{\alpha\beta}$ and $\tilde{q}_{i-\frac{1}{2}}^{\alpha\beta}$ one should use numerical methods for integrating expressions (8). In this paper, it was assumed that the functions $p(x)$ and $q(x)$ are approximated by continuous piecewise linear functions so that their values are known at the “reference” points $\{x_i\}_{i=0}^N$ of the grid (vectors $\{p_i\}_{i=0}^N$ and $\{q_i\}_{i=0}^N$), and on each segment $[x_{i-1}, x_i]$, $i = \overline{1, N}$ these functions are determined by the relations

$$p(x) = p_{i-1} \cdot \omega_{i-1/2}^L(x) + p_i \cdot \omega_{i-1/2}^R(x), \quad q(x) = q_{i-1} \cdot \omega_{i-1/2}^L(x) + q_i \cdot \omega_{i-1/2}^R(x). \tag{10}$$

In this case, the constants $p_{i-\frac{1}{2}}$ and $q_{i-\frac{1}{2}}^{\alpha\beta}$ are defined by the following equalities:

$$\begin{aligned} p_{i-\frac{1}{2}} &= \frac{p_{i-1} + p_i}{2} \cdot h_{i-\frac{1}{2}}, & q_{i-\frac{1}{2}}^{LL} &= \frac{3q_{i-1} + q_i}{12} \cdot h_{i-\frac{1}{2}}, \\ q_{i-\frac{1}{2}}^{RR} &= \frac{q_{i-1} + 3q_i}{12} \cdot h_{i-\frac{1}{2}}, & q_{i-\frac{1}{2}}^{LR} &= \frac{q_{i-1} + q_i}{12} \cdot h_{i-\frac{1}{2}}. \end{aligned}$$

After the approximation, the minimization of the functional (5) is reduced to the minimization of a function $F(c_1, \dots, c_{N-1})$ of the form

$$F(c_1, \dots, c_{N-1}) = \frac{\sum_{i=1}^N \left(\frac{c_i^2 - 2c_i c_{i-1} + c_{i-1}^2}{h_{i-\frac{1}{2}}^2} p_{i-\frac{1}{2}} + c_{i-1}^2 q_{i-\frac{1}{2}}^{LL} + 2c_{i-1}c_i q_{i-\frac{1}{2}}^{LR} + c_i^2 q_{i-\frac{1}{2}}^{RR} \right)}{\sum_{i=1}^N \left(c_{i-1}^2 \tilde{q}_{i-\frac{1}{2}}^{LL} + 2c_{i-1}c_i \tilde{q}_{i-\frac{1}{2}}^{LR} + c_i^2 \tilde{q}_{i-\frac{1}{2}}^{RR} \right)}. \tag{11}$$

To minimize the function (11), both the gradient method and the Newton method were used. The gradient $\| \frac{\partial F}{\partial c_i} \|_{i=1}^{N-1}$ of the function $F(c_1, \dots, c_{N-1})$ required for these methods and its Hessian matrix $\| \frac{\partial^2 F}{\partial c_i \partial c_j} \|_{i,j=1}^{N-1}$ were determined analytically. When calculating the first eigenvalue and the first eigenfunction, the problem of unconditional minimization of the function (11) was solved. When calculating the remaining eigenvalues and eigenfunctions, the minimization of function (11) was carried out in a subspace orthogonal to all eigenfunctions found earlier.

4 Trigonometric Tridiagonal Matrix Algorithm

The trigonometric tridiagonal matrix algorithm is based on the important change of variables proposed by Prufer [6]. It consists in replacing the differential equation (1) with an equivalent normal system of first-order differential equations

$$\frac{du(x)}{dx} = \frac{1}{p(x)} \cdot w(x), \quad \frac{dw(x)}{dx} = (q(x) - \lambda) \cdot u(x) \tag{12}$$

and in the subsequent transition in the phase plane to polar coordinates. Namely, we will introduce new functions $\varphi(x)$ and $\rho(x)$ according to the rule:

$$u(x) = \rho(x) \cdot \cos \varphi(x), \quad p(x) \cdot u'(x) = \rho(x) \cdot \sin \varphi(x), \quad x \in (a, b). \tag{13}$$

Substitution of expressions (13) into system of equations (12) leads to the following nonlinear system of differential equations (see [6])

$$\varphi'(x) = \frac{1}{p(x)} \cdot \sin^2 \varphi(x) + (\lambda - q(x)) \cdot \cos^2 \varphi(x) = 0, \quad x \in (a, b), \tag{14}$$

$$\varphi(a) = \frac{\pi}{2}, \quad \varphi(b) = \frac{\pi}{2} - k\pi, \quad (k = 1, 2, 3, \dots), \tag{15}$$

$$\frac{d \ln \rho(x)}{dx} = \frac{1}{2} \left(\frac{1}{p(x)} + q(x) - \lambda \right) \cdot \sin 2\varphi(x), \quad x \in (a, b). \tag{16}$$

The importance of the resulting system of differential equations is due to the fact that Eq. (14) contains only one unknown function $\varphi(x)$. Therefore, the spectral problem has essentially been reduced to integrating and studying one first-order equation (14). If a solution to the boundary value problem (14)–(15) is found, then the function $\rho(x)$ can be obtained by integrating Eq. (16).

To find the k -th eigenvalue λ_k , one should find a solution to Eq. (14) that satisfies conditions (15) for a given k . To do this, for fixed λ , the Cauchy problem is solved for Eq. (14) and the first of the conditions (15). As a result, some value $\varphi(b, \lambda)$ of the solution of the problem at the endpoint is obtained. The condition $\varphi(b, \lambda_k) = \frac{\pi}{2} - k\pi$ is a characteristic equation for determining the k -th eigenvalue λ_k . Solving this equation by some method (for example, the method of dividing the segment in half), we find the k -th eigenvalue λ_k . To determine the corresponding k -th eigenfunction $u_k(x)$, we solve the Cauchy problem for Eq. (16) with an arbitrary initial condition, using the first of relations (13) and the found functions $\varphi_k(x)$ and $\rho_k(x)$, construct the function $u_k(x)$ and then normalize it.

Equations (14) and (16) can be integrated using different numerical algorithms. In this work, the integration was carried out using two methods: the 4th order Runge-Kutta method and the numerical-analytical method. In both cases, as in the variational algorithm, the segment $[a, b]$ was divided into sub-segments by reference points $\{x_i\}_{i=0}^N$.

In the case of the Runge-Kutta method, described in detail in [6], on each segment $[x_{i-1}, x_i]$, ($i = \overline{1, N}$) the functions $p(x)$ and $q(x)$ were approximated by linear functions according to formulas (10).

As for the numerical-analytical method, here on each segment $[x_{i-1}, x_i]$ these functions were assumed to be constant $p_{i-1/2}$ and $q_{i-1/2}$, the values of which were equal to the average values of the functions $p(x)$ and $q(x)$ on the segment $[x_{i-1}, x_i]$, i.e.

$$p_{i-1/2} \cdot h_{i-1/2} = \int_{x_{i-1}}^{x_i} p(x)dx, \quad q_{i-1/2} \cdot h_{i-1/2} = \int_{x_{i-1}}^{x_i} q(x)dx.$$

The constancy of the functions $p(x)$ and $q(x)$ on the interval $[x_{i-1}, x_i]$ makes it possible here to analytically determine the solution of Eq. (14).

Denote by φ_i the value of the desired function $\varphi(x)$ at the point x_i , and by $\beta_{i-1/2}$ the next constant $\beta_{i-1/2} = (\lambda - q_{i-1/2}) \cdot p_{i-1/2}$. Then the value of the function $\varphi(x)$ on the interval $[x_{i-1}, x_i]$ is determined analytically by the following formulas:

a) if $[1 + (\beta_{i-1/2} - 1) \cdot \cos^2 \varphi_{i-1/2}] = 0$, then

$$\varphi(x) \equiv \varphi_{i-1/2}, \quad x \in [x_{i-1}, x_i];$$

b) if $\beta_{i-1/2} = (\lambda - q_{i-1/2}) \cdot p_{i-1/2} = 0$, then

$$\tan \varphi(x) = \frac{p_{i-1/2} \cdot \tan \varphi_{i-1/2}}{p_{i-1/2} + (x - x_i) \cdot \tan \varphi_{i-1/2}}, \quad x \in [x_{i-1}, x_i];$$

c) if $\beta_{i-1/2} = (\lambda - q_{i-1/2}) \cdot p_{i-1/2} > 0$, then

$$\tan \varphi(x) = \sqrt{\beta_{i-1/2}} \cdot \tan \left[-\frac{\sqrt{\beta_{i-1/2}}}{p_{i-1/2}} \cdot (x - x_i) + \arctan \left(\frac{\tan \varphi_{i-1/2}}{\sqrt{\beta_{i-1/2}}} \right) \right], \quad x \in [x_{i-1}, x_i];$$

d) if $\beta_{i-1/2} = (\lambda - q_{i-1/2}) \cdot p_{i-1/2} < 0$, then

$$\tan \varphi(x) = \sqrt{-\beta_{i-1/2}} \cdot \frac{1 + A}{1 - A}, \quad x \in [x_{i-1}, x_i],$$

$$A = \frac{\tan \varphi_{i-1/2} - \sqrt{-\beta_{i-1/2}}}{\tan \varphi_{i-1/2} + \sqrt{-\beta_{i-1/2}}} \cdot \exp \left(-\frac{2\sqrt{-\beta_{i-1/2}}}{p_{i-1/2}} \cdot (x - x_i) \right).$$

The calculation starts from the interval $[x_0, x_1]$. Using one of the above formulas and condition $\varphi_0 = 0$, the value φ_1 of the function $\varphi(x)$ at the point x_1 is determined. Taking into account the continuity of the function $\varphi(x)$, we solve the Cauchy problem for Eq. (14) with the initial condition $\varphi(x_1) = \varphi_1$ on the interval $[x_1, x_2]$ and find the value φ_2 of the function $\varphi(x)$ at the point x_2 , and so on. As a result, we obtain the value φ_N of the function $\varphi(x)$ at the point $x_N = b$, and select the parameter λ so that the condition is satisfied for the fixed k

$$\varphi_N = \varphi(x_N) = \frac{\pi}{2} - k\pi.$$

After the eigenvalue λ_k is found, Eq. (16) is integrated and the eigenfunction $u_k(x)$ is constructed using the first of relations (13).

5 Results of Numerical Calculations

A large number of computational experiments were carried out concerning the numerical solution of the spectral problem (1)–(4) with different input data. The same problem was solved by the variational method (VM), the trigonometric tridiagonal matrix algorithm (TMRK—version using the 4th order Runge-Kutta method for the numerical integration of Eq. (12), the numerical-analytical version—TMNA) and using the standard software package (SSP), which determines the eigenvalues and eigenvectors of the matrix.

The criterion for comparing the algorithms was the accuracy of calculating the first eigenvalues. The most characteristic of the results obtained are presented in this section and are based on two series of calculations.

In the **first series** of calculations, the following spectral problem was considered:

$$\begin{aligned} -u''(x) &= \lambda \cdot u(x), & x \in (0, \pi), \\ u(0) &= 0, & u(\pi) = 0, \\ u(x) &\neq 0, & x \in (0, \pi). \end{aligned}$$

The solution to this problem is well known:

$$\lambda_k = k^2, \quad u_k(x) = \sin kx, \quad k = 1, 2, 3, \dots$$

The segment $[0, \pi]$ was divided uniformly into N sub-segments, and the calculations were carried out for $N = 25, 50, 100, 500, 1000, 2000$.

We note at once that the numerical-analytical method of TMNA makes it possible to obtain the first twenty eigenvalues with relative accuracy $\approx 5 \cdot 10^{-12}\%$, regardless of the number N of partitions of the segment $[0, \pi]$. As for other methods, some of the results obtained for $N = 50, 100, 500$ are presented in Tables 1, 2 and 3. The numbers in the table indicate the relative deviation of the calculated eigenvalues of the spectral problem from the theoretical ones in percent.

Table 1. The relative deviation of the calculated eigenvalues of the spectral problem from the theoretical ones in percent, $N = 50$.

$N = 50$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
SSP	$3.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$3.0 \cdot 10^{-1}$	$5.3 \cdot 10^{-1}$	$8.2 \cdot 10^{-1}$	$3.2 \cdot 10^0$	$7.2 \cdot 10^0$	$1.2 \cdot 10^1$
VM	$3.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$3.0 \cdot 10^{-1}$	$5.3 \cdot 10^{-1}$	$8.3 \cdot 10^{-1}$	$3.3 \cdot 10^0$	$7.6 \cdot 10^0$	$1.4 \cdot 10^1$
TMRK	0.0	$1.2 \cdot 10^{-3}$	$3.7 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$	$1.1 \cdot 10^0$	$5.9 \cdot 10^1$	$7.4 \cdot 10^1$	$8.4 \cdot 10^1$

Table 2. The relative deviation of the calculated eigenvalues of the spectral problem from the theoretical ones in percent, $N = 100$.

$N=100$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
SSP	$8.2 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$2.1 \cdot 10^{-1}$	$8.2 \cdot 10^{-1}$	$1.8 \cdot 10^0$	$3.3 \cdot 10^0$
VM	$8.2 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$	$2.1 \cdot 10^{-1}$	$8.3 \cdot 10^{-1}$	$1.9 \cdot 10^0$	$3.3 \cdot 10^0$
TMRK	0.0	$7.8 \cdot 10^{-5}$	$2.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$2.7 \cdot 10^1$	$6.4 \cdot 10^1$	$7.7 \cdot 10^1$

Table 3. The relative deviation of the calculated eigenvalues of the spectral problem from the theoretical ones in percent, $N = 500$.

$N=500$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
SSP	$3.3 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	$5.3 \cdot 10^{-3}$	$8.2 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$
VM	$3.3 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	$5.3 \cdot 10^{-3}$	$8.2 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$
TMRK	0.0	$1.3 \cdot 10^{-7}$	$4.0 \cdot 10^{-6}$	$3.8 \cdot 10^{-5}$	$2.0 \cdot 10^{-4}$	$2.9 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$	$1.2 \cdot 10^1$

Analysis of the results showed that the SSP and VM methods behave almost identically (the difference is observed in 3–4 significant figures). For all considered N , the qualitative dependence of the relative deviation of the eigenvalue is preserved. This dependence for $N = 100$ is shown in Fig. 1.

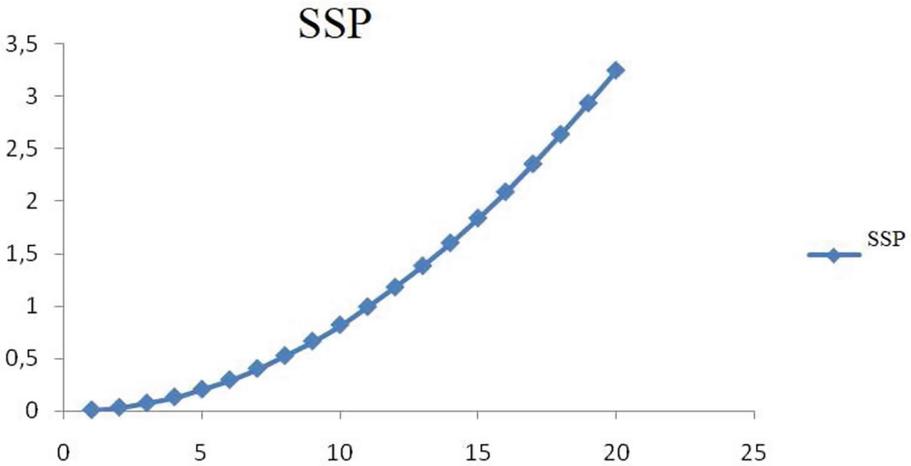


Fig. 1. The qualitative dependence of the relative deviation of the eigenvalue (SSP method, $N = 100$).

As for the TMRK method, in all examples the first eigenvalues (their number depends on N) are calculated more accurately than in the case of using the SSP and VM methods, but the remaining eigenvalues are calculated noticeably

rougher. The dependence of the relative deviation of the eigenvalue at $N = 100$ is shown in Fig. 2.

Finally, Table 4 shows the dependence on the number N of the maximum relative deviation (in percent) of the calculated eigenvalues from their theoretical values for the first 20 eigenvalues.

Judging by the latest results, we can say that the SSP and VM methods turn out to be preferable when it comes to determining the first 20 eigenfunctions of the spectral problem. At the same time, it should be recalled that the TMNA method restores these eigenvalues with an accuracy of $\approx 5 \cdot 10^{-12}\%$.

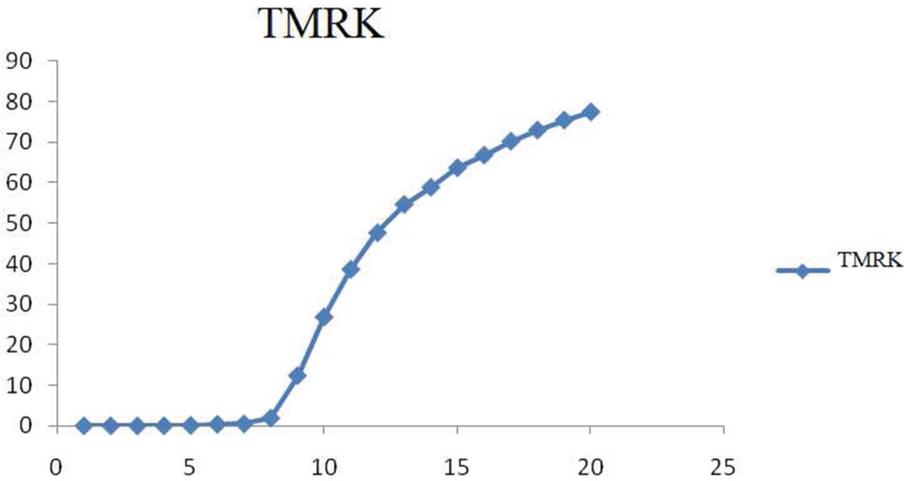


Fig. 2. The qualitative dependence of the relative deviation of the eigenvalue (TMRK method, $N = 100$).

Table 4. The dependence on the number N of the maximum relative deviation (in percent) of the calculated eigenvalues from their theoretical values for the first 20 eigenvalues.

	$N = 25$	$N = 50$	$N = 100$	$N = 500$	$N = 1000$	$N = 2000$
SSP	$4.3 \cdot 10^{+1}$	$1.2 \cdot 10^{+1}$	$3.2 \cdot 10^0$	$1.3 \cdot 10^{-1}$	$3.3 \cdot 10^{-2}$	$8.2 \cdot 10^{-3}$
VM	$4.4 \cdot 10^{+1}$	$1.4 \cdot 10^{+1}$	$3.3 \cdot 10^0$	$1.3 \cdot 10^{-1}$	$3.3 \cdot 10^{-2}$	$8.2 \cdot 10^{-3}$
TMRK	$8.4 \cdot 10^{+1}$	$8.4 \cdot 10^{+1}$	$7.7 \cdot 10^{+1}$	$1.2 \cdot 10^{+1}$	$2.1 \cdot 10^{-1}$	$1.5 \cdot 10^{-2}$

In the **second series** of calculations, a more interesting spectral problem was considered:

$$-u''(x) + x \cdot (x - \pi) \cdot u(x) = \lambda \cdot u(x), \quad x \in (0, \pi),$$

$$\begin{aligned} u(0) &= 0, & u(\pi) &= 0, \\ u(x) &\neq 0, & x &\in (0, \pi). \end{aligned}$$

As the “exact” solution of this problem, we chose the solution constructed by the numerical-analytical version of the trigonometric tridiagonal matrix algorithm for $N = 2000$. The value $N = 2000$ was chosen on the basis that a further increase in this number leads to a change in the eigenvalues in the 9th significant digit.

As in the previous example, the segment $[0, \pi]$ was divided evenly into N sub-segments, and calculations were carried out for $N = 25, 50, 100, 500, 1000, 2000$ using all the described methods.

Note that in the case of the second example, the numerical-analytical method of trigonometric tridiagonal matrix algorithm no longer allows obtaining the first twenty eigenvalues with relative accuracy $\approx 5 \cdot 10^{-12}\%$, as it was in the case of the first example.

Some of the results obtained for $N = 50, 100, 500$ are presented in Tables 5, 6 and 7. As in the first example, the analysis of the results showed that the SSP and VM methods behave almost the same way. Therefore, Tables 5, 6 and 7 present the results of using only the variational method. In addition, a row with the results obtained using the TMNA method appeared in these tables. The numbers in the tables, as before, indicate the relative deviation of the computed eigenvalues of the spectral problem from the theoretical values in percent.

Table 5. The relative deviation of the computed eigenvalues of the spectral problem from the theoretical values in percent, $N = 50$.

$N=50$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
VM	$1.6 \cdot 10^{-2}$	$3.5 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$	$6.8 \cdot 10^{-1}$	$9.7 \cdot 10^{-1}$	$3.4 \cdot 10^0$	$7.3 \cdot 10^0$	$1.3 \cdot 10^1$
TMRK	$5.7 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$5.7 \cdot 10^{-2}$	$3.5 \cdot 10^{-1}$	$9.9 \cdot 10^{-1}$	$5.9 \cdot 10^1$	$7.9 \cdot 10^1$	$8.5 \cdot 10^1$
TMNA	$1.1 \cdot 10^{-1}$	$5.9 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	$9.2 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	$6.1 \cdot 10^{-4}$	$3.5 \cdot 10^{-4}$

The results presented in Tables 5, 6 and 7 allow us to draw the following conclusions. First, the SSP and VM methods behave almost identically. The qualitative dependence of the relative deviation of the eigenvalue is also preserved here and it coincides with the one shown in Fig. 1. Secondly, as in the previous example, when using the TMRK method, the first eigenvalues (their number depends on N) are calculated more accurately than in the case of using the SSP and VM methods, but the remaining eigenvalues are calculated noticeably rougher. The qualitative behavior of the relative deviation of the eigenvalues is similar to that shown in Fig. 2. Thirdly, when the relative error of its calculation by the TMNA method decreases. The dependence of the relative deviation of the eigenvalue at $N = 100$ is shown in Fig. 3.

Table 6. The relative deviation of the computed eigenvalues of the spectral problem from the theoretical values in percent, $N = 100$.

$N=100$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
VM	$4.3 \cdot 10^{-3}$	$8.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$8.6 \cdot 10^{-1}$	$1.9 \cdot 10^0$	$3.3 \cdot 10^0$
TMRK	$1.4 \cdot 10^{-2}$	$7.6 \cdot 10^{-3}$	$5.4 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$2.8 \cdot 10^1$	$6.4 \cdot 10^1$	$7.7 \cdot 10^1$
TMNA	$2.8 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	$3.3 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$8.4 \cdot 10^{-5}$

Table 7. The relative deviation of the computed eigenvalues of the spectral problem from the theoretical values in percent, $N = 500$.

$N=500$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
VM	$7.3 \cdot 10^{-4}$	$3.7 \cdot 10^{-3}$	$4.4 \cdot 10^{-3}$	$6.8 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$3.4 \cdot 10^{-2}$	$7.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$
TMRK	$5.0 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	$8.4 \cdot 10^{-5}$	$8.3 \cdot 10^{-5}$	$2.4 \cdot 10^{-4}$	$3.0 \cdot 10^{-2}$	$2.9 \cdot 10^{-1}$	$1.1 \cdot 10^1$
TMNA	$1.1 \cdot 10^{-3}$	$5.6 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$8.6 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	$1.3 \cdot 10^{-5}$	$5.5 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$

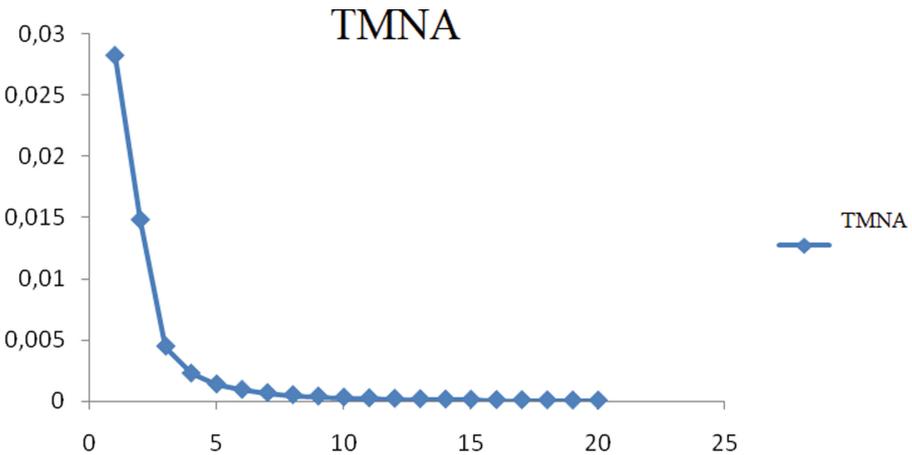


Fig. 3. The dependence of the relative deviation of the eigenvalue at $N = 100$.

Finally, Table 8 shows the dependence on the number N of the maximum for the first 20 eigenvalues of the relative deviation (in percent) of the calculated eigenvalues from their theoretical values.

Table 8. The dependence on the number N of the maximum relative deviation (in percent) of the calculated eigenvalues from their theoretical values for the first 20 eigenvalues.

	$N = 25$	$N = 50$	$N = 100$	$N = 500$	$N = 1000$	$N = 2000$
VM	$4.3 \cdot 10^{+1}$	$1.3 \cdot 10^{+1}$	$3.3 \cdot 10^0$	$1.3 \cdot 10^{-1}$	$3.3 \cdot 10^{-2}$	$8.3 \cdot 10^{-3}$
TMRK	$8.8 \cdot 10^{+1}$	$8.5 \cdot 10^{+1}$	$7.7 \cdot 10^{+1}$	$1.1 \cdot 10^{+1}$	$2.2 \cdot 10^{-1}$	$1.6 \cdot 10^{-2}$
TMNA	$4.5 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$2.8 \cdot 10^{-2}$	$1.1 \cdot 10^{-3}$	$2.1 \cdot 10^{-4}$	0.0

Recent results show that the SSP and VM methods are preferable to the TMRK method, but lose to the TMNA method.

In the **third series** of calculations, an example borrowed from [7] was considered. In this example, the stationary states of an electron in the simplest rectangular quantum well formed by a three-layer $Al_{0.3}Ga_{0.7}As/GaAs$ heterostructure are studied. The study of the electron states is reduced to solving the spectral problem (1)–(4) for the following values of dimensionless parameters

$$\begin{aligned}
 a &= -38.4, & b &= +38.4, \\
 p(x) &= \begin{cases} 0.414128487, & 2.8 \leq |x| \leq 38.4, \\ 0.568654042, & |x| \leq 2.8, \end{cases} \\
 q(x) &= \begin{cases} 0.23, & 2.8 \leq |x| \leq 38.4, \\ 0, & |x| \leq 2.8. \end{cases}
 \end{aligned}$$

Since the functions $p(x)$ and $q(x)$ are piecewise constant, the numerical-analytical version of the trigonometric tridiagonal matrix method TMNA allows one to obtain exact eigenvalues for any ($N \geq 3$) partition of the segment $[a, b]$, provided that the break points of the functions $p(x)$ and $q(x)$ coincide with the reference points.

The segment $[-38.4, +38.4]$ was divided evenly into sub-segments, and the calculations were carried out using all the methods described.

Some of the results obtained are presented in Table 9. As in the previous examples, the analysis of the results showed that the VM and SSP methods behave almost the same. Therefore, Table 9 presents the results of using only the VM variational method. The numbers in the tables, as before, indicate the relative deviation of the calculated eigenvalues of the spectral problem from the theoretical values in percent.

Table 9. The relative deviation of the calculated eigenvalues of the spectral problem from the theoretical ones in percent, $N = 768$.

$N = 768$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{10}	λ_{15}	λ_{20}
VM	$1.8 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$4.7 \cdot 10^{-5}$	$7.3 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$4.2 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$
TMRK	$1.6 \cdot 10^{-6}$	$9.5 \cdot 10^{-5}$	$4.3 \cdot 10^{-6}$	$1.3 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$	$2.1 \cdot 10^{-4}$	$4.7 \cdot 10^{-4}$	$7.1 \cdot 10^{-4}$

The results presented in Table 9 allow us to draw the following conclusions. First, recall that the VM and SSP methods behave almost identically. Secondly, the trigonometric tridiagonal matrix algorithm using the Runge-Kutta method (TMRK method) determines the first 20 eigenvalues approximately two orders of magnitude more accurately than the VM variational method and the SSP matrix method. Thirdly, there is a non-monotonicity in the accuracy of determining the eigenvalues both by the VM method and by the TMRK method. But in general, as the number of the eigenvalue increases, the relative error of its calculation by the VM, SSP, and TMRK methods increases.

6 Conclusion

Analysis of the results of the computational experiments, some of which are presented above, allows us to draw the following conclusions.

When solving the spectral problem (1)–(4) in the case when it is necessary to obtain several first eigenvalues and eigenfunctions, it is advisable to use the numerical-analytical version of the trigonometric tridiagonal matrix algorithm. In this case, to obtain the eigenvalues, one can use a not too detailed partition of the segment $[a, b]$, since for $N \approx 25\text{--}50$ the relative error of the calculated eigenvalues does not exceed 1%. After finding the required eigenvalues, the determination of the corresponding eigenfunctions is carried out without iteration by solving Eq. (16) by the Runge-Kutta method on a sufficiently detailed grid ($N \approx 500\text{--}1000$).

If we are talking about a multidimensional case, then it seems reasonable to use the variational method described above.

References

1. Abgaryan, K.K., Reviznikov, D.L.: Numerical simulation of the distribution of charge carrier in nanosized semiconductor heterostructures with account for polarization effects. *Comput. Math. Math. Phys.* **56**(1), 161–172 (2016)
2. Marchuk, G.I., Agoshkov, V.I.: Introduction to projection-grid methods. Nauka. The main editorial office of the physical and mathematical literature (1981)
3. Rectoris, K.: Variational methods in mathematical physics and engineering. Mir (1985)
4. Marchuk, G.I.: Methods of computational mathematics. Study guide. 3rd edn. Nauka. The main editorial office of the physical and mathematical literature (1989)
5. Mikhlin, S.G.: Variational methods in mathematical physics. Nauka. The main editorial office of the physical and mathematical literature (1970)
6. Fedorenko, R.P.: Introduction to Computational Physics. Publishing House of the Moscow Institute of Physics and Technology, Moscow (1994)
7. Tan, I.-H., Snider, G.L., Chang, L.D., Hu, E.L.: A self-consistent solution of Schrödinger-Poisson equations using a nonuniform mesh. *J. Appl. Phys.* **68**(8), 4071–4076 (1990)

Mathematical Economy



Optimal Arrivals to Preemptive Queueing System

Julia V. Chirkova¹^(✉) and Vladimir V. Mazalov^{1,2}

¹ Institute of Applied Mathematical Research of Karelian Research Centre of RAS,
Pushkinskaya st. 11, Petrozavodsk, Russia
{julia,vmazalov}@krc.karelia.ru

² Saint-Petersburg State University, 7/9, Universitetskaya nab.,
Saint-Petersburg 199034, Russia

Abstract. This paper considers a single-server queueing system with strategic users in which customers (players) enter the system with preemptive access. As soon as the customer request enters the system, the server immediately starts the service. But when the next request arrives in the system, the previous one leaves the system even he has not finished his service yet. We study the following non-cooperative game for this service system. Each player decides when to arrive at the queueing system within a certain period of time. The objective of the player is to maximize the probability of receiving service. We show that there exists a unique symmetric Nash equilibrium in this game. Finally, some numerical experiments are carried out to compare the equilibria under different values of the model parameters.

Keywords: Queueing system · Preemptive access · Strategic users · Optimal arrivals · Kolmogorov backward equations · Nash equilibrium

1 Introduction

We consider a single-server queueing system with strategic users. The customers (players) log into the system with preemptive access during a fixed time interval $[0, T]$. As soon as a customer arrives in the system, the server immediately starts his service. But when the next customer arrives in the system, the current one leaves the system even it has not finished its service yet, unlike the model in [2] where the current request is moved to a queue to resume its service later.

Such service discipline looks strange and unfair, but it is often found in real life. In nature, animals mark their territory and exchange tags during the mating season - the last one is the owner until the next one leaves his mark. The situation is similar with graffiti on the walls. Another example is an access to social shared objects. For example, observing some kind of an art object the user leaves it to

This research was supported by the Russian Science Foundation: grant No. 22-11-20015, <https://rscf.ru/project/22-11-20015/>, jointly with support of the authorities of the Republic of Karelia with funding from the Venture Investment Foundation of the Republic of Karelia; grant No. 22-11-00051, <https://rscf.ru/en/project/22-11-00051/>.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 169–181, 2022.
https://doi.org/10.1007/978-3-031-09607-5_12

make room for the next visitor. Or we can consider an open access webcam (such as i.g. <http://webcam.anw.at/>) which can change its angle of view according to user's commands. When one user interact with such service, another user can take a control.

There are data transmission systems in which it is important not the integrity of the information, but its relevance. Such are audio and video streaming applications. Data exchange in such applications is implemented on the basis of the UDP protocol. The transmitted data stream is divided into fragments – data packets, possibly of different sizes. They are sent over the network asynchronously and without delivery confirmation. In the process of transmission, packets may be delayed, and the next packet may not be fully delivered when the next packet is received. When analysing such systems, it is important to understand that high outgoing losses are the norm, since UDP traffic does not require acknowledgement. The use of reliable protocols, such as TCP, would inevitably lead to large delays in data exchange, which is unacceptable when it is necessary to exchange video and audio information in real time. Therefore, UDP is used, which is an unreliable data transfer protocol. This means that the party sending the packets can send as much traffic as the network system can receive without worrying about losses due to network delays. The addressee processes that part of the information that was received in a timely manner. Since there are many packets, successfully delivered packets are sufficient for continuous video playback at the receiving point. Video quality depends on the percentage of delivery losses. UDP loss determines the degree of user comfort when working with such applications. A high percentage of losses leads to severe jitter and delays in audio and video. If there are several devices (cameras, sensors etc.) sending streaming information, they compete to provide their packets delivering. This data transmission system can also be considered as a system with preemptive access.

In conventional queueing theory, the structure of the input process is usually assumed to be predefined and specified by the input rate of the customers. However, there exists a different approach to the queueing which is based on the assumption that the customers logging into the system are strategic [1, 3, 5–16]. Namely, it is assumed that the user strategy is to select the arrival instant to the system on a time interval $[0, T]$. In this setting, the queue in the system is determined after each player selects their random arrival instant in the system. Thus, each user spends some time in the system, and this time is their personal utility function. As a result, a non-zero-sum game is obtained, in which we need to find the *Nash equilibrium*. The paper [6] is the first work that considers the queue as a result of the user's behavior. They further formulate a non-cooperative game in which a Poisson-distributed number of the customers determines their arrival instances in the queue of a single-server system, over a (limited) admission interval $[0, T]$. The purpose of the customers is to minimize their waiting time in the system. It is shown in [6] that the *symmetric Nash equilibrium strategy* is mixed. In particular, it was revealed that this strategy is the (absolutely continuous) uniform distribution over time interval $[0, T]$, except a singularity at zero, and the density function decreases between zero and T . A similar model

with $m \geq 1$ identical (exponential) servers and the buffer size $c \geq 0$ for the waiting customers is considered in the paper [9]. Note that the arrival times game with the *batch service* has been investigated in [7]. A single-server bufferless system in which the customers have a time-sensitivity function that they want to minimize, instead of their own waiting costs, has been studied in [14]. A model where the customers may incur tardiness costs in addition to the waiting costs is considered in [12]. The paper [11] considers a model combining the tardiness costs, waiting costs, and restrictions on the opening and closing times. The paper [2] presents a queueing system where a single server opens and serves users according to the last-come first-served discipline with preemptive-resume. A recent paper [4] is devoted to finding an equilibrium in a single-server queueing system with retrievals and strategic timing of arrivals.

In this paper, we apply a game-theoretic approach to a preemptive single-server queueing system. The queue is formed by the strategic players. The player's strategy is to choose a moment to enter the system. It is possible to assume that each player tries to maximize the probability to perform his request completely or maximizes a service time or a degree of completion for his request. In this paper we use the first form of payoff, the second one is a direction for the following development. The paper is organized as follows. In Sect. 2, we describe the generalized model in details. In the Sect. 3 we assume that the number of players is fixed. We demonstrate that there exists a unique symmetric equilibrium. Finally, some numerical experiments are performed to compare the equilibria under different values of the model parameters.

2 Description of the Model

2.1 Service System

Now we describe our model in more details in a general setting. We assume that there exists a single server which serves $N + 1$ customers presented in the system at the initial instant $t = 0$. Unlike the conventional queueing theory setting, these customers use some strategy to choose an instant on a time interval $[0, T]$ to enter the server. By a symmetry, this strategy is the same for each user. This strategy is determined by a distribution function which is the main purpose of the analysis, and it determines the instant of the attempt to enter the server.

When some customer arrives then he captures server for the exponentially distributed time with parameter μ . But when the next customer arrives in the system the current one must leave the system even it has not finished its service yet. The system has no queues. The server may simultaneously perform only one request. It may happen that several customers arrive to the system at the same time. Then the server chooses with equal probability one of the currently arrived requests for further servicing.

In the queueing theory, the structure of the input process and the service process are usually assumed to be predefined and specified by the input rate and service times of the customers. Here the input process is actually formed by

the strategic customers who like to maximize the probability to be served in the system.

2.2 Game-Theoretic Model

Consider the optimal request discipline problem in the system as a non-cooperative game. Here the players are the system users sending their requests for servicing. Denote by S the player set. The number of players is $N + 1 = |S|$, which can be a fixed or a random value. Each player chooses the time to send his request to the system, seeking to maximize the probability of servicing for his requests. The pure strategy of player i is the arrival time t_i of his request in the system. The mixed strategy of player i is the distribution function $F_i(t)$ (having density $f_i(t)$) of the arrival times in the system on the time interval $[0, T]$. Let $F = \{F_i(t), i \in S\}$ be the strategy profile.

All players are identical, independent and demonstrate the selfish behavior without cooperation. As the optimality criterion we choose the symmetric Nash equilibrium. In this case, the strategies of all players coincide, i.e., $F_i(t) = F(t)$ for all i .

Definition 1. *A distribution function $F(t)$ of the arrival times t in the system is a symmetric Nash equilibrium if there exists a constant C such that at any time $t \in [0, T]$ the probability of service does not exceed C and is equal to C on the support of $F(t)$.*

To find the Nash equilibrium in this game we use the following approach. Assume that all players $\{1, 2, \dots, N\}$ use the same mixed strategy $F(t), t \in [0, T]$. Then we find the best response of player $N + 1$ to the described strategy $F(t)$. As a payoff function of player $N + 1$, we will consider the probability of servicing of his request. Thus, the objective of player $N + 1$ is to choose a strategy that will maximize his payoff function. Due to the symmetry of the problem, in equilibrium the optimal strategy of player $N + 1$ must coincide with the chosen strategy of his opponents. To do this, it is sufficient that the strategy of player $N + 1$ is chosen in such a way that the payoff function of player $N + 1$ takes a constant value over the support of the distribution function $F(t)$ (see [13]). It yields that the best response of player $N + 1$ for the mixed strategies of his opponents $F_i(t) = F(t), i = 1, \dots, N$ coincides with $F(t)$. Thus, $F(t)$ is the Nash equilibrium in this game.

Lemma 1. *The support of the equilibrium strategy contains an atom at the point $t = T$, i.e., the equilibrium probability p of request arrivals at the instant T is strictly positive. In addition, there exists a time interval (t_e, T) without receiving requests in the system.*

Proof. Really, the probability of request arrivals in the system at the instant $t = T$ has a positive value. Assume that it is not true, i.e., no one sends his requests to the system at this time. Then any player deviating from the equilibrium and sending his request to the system at the instant T receives servicing with

probability 1. Consequently, there is a positive probability to arrive at the instant T in the equilibrium.

Assume that X_p be a random variable that represents the number of requests received at the instant T . The probability that a request arriving at the instant T receives service is

$$E \left[\frac{1}{X_p + 1} \right] = P(X_p = 0) + E \left[\frac{1}{X_p + 1} | X_p > 0 \right].$$

Consider the instant t such that there are no arrivals at the interval (t, T) . The probability that a request arriving at the instant t receives service is

$$P(X_p = 0) + P(X_p > 0)(1 - e^{-\mu(T-t)}) = 1 - P(X_p > 0)e^{-\mu(T-t)},$$

it obviously decreases by t .

Consider the instant $t = T^-$. The probability that a request arriving at the instant T^- receives service is

$$\lim_{t \rightarrow T^-} P(X_p = 0) + P(X_p > 0)(1 - e^{-\mu(T-t)}) = P(X_p = 0),$$

which is less than if the request had arrived at instant T .

Thus, the player's payoff decreases up to the moment T^- , and remain less than at the moment T , provided that no one sent requests to the system during this period. This explains the existence of the time interval $[t_e, T)$ without requests coming before the instant T . \square

Suppose we know the equilibrium probability p of a request arrival at the instant T , where $0 < p \leq 1$. We show that the player's payoff is a decreasing function on the interval (t_e, T) . Hence, there exists an instant t_e (perhaps, negative) when the payoffs at the instants t_e and T coincide. It yields

$$E \left[\frac{1}{X_p + 1} \right] = 1 - P(X_p > 0)e^{-\mu(T-t)}. \tag{1}$$

Lemma 2. *In the queueing game with two players the Eq. (1) defines t_e that is independent of p . In game with $N + 1 \geq 3$ players the Eq. (1) defines a function $t_e(p)$ that strictly increases in p .*

Proof. Consider the game with two players, so $N = 1$. The Eq. (1) is

$$(1 - p) \cdot 1 + p \frac{1}{2} = 1 - p \left(1 - e^{-\mu(T-t_e)} \right).$$

It yields

$$t_e = T - \frac{1}{\mu} \log 2.$$

We see that t_e doesn't depend on p .

Let $N = n \geq 2$. The Eq. (1) can be presented as

$$\sum_{k=0}^n \frac{1}{k+1} \binom{n}{k} p^k (1-p)^{n-k} = 1 - (1 - (1-p)^n) e^{-\mu(T-t_e)}. \tag{2}$$

Rewrite (2) as

$$e^{-\mu(T-t_e)} = \frac{1 - \sum_{k=0}^n \frac{1}{k+1} \binom{n}{k} p^k (1-p)^{n-k}}{1 - (1-p)^n}. \tag{3}$$

Differentiating (3) in p we obtain

$$\mu e^{-\mu(T-t_e)} \frac{dt_e}{dp} = \frac{(1 - (1-p)^n)^2 - n^2 p^2 (1-p)^{n-1}}{(n+1)(1 - (1-p)^n)^2 p^2}. \tag{4}$$

From the Cauchy inequality

$$\frac{\sum_{i=0}^{n-1} (1-p)^i}{n} \geq \left(\prod_{i=0}^n (1-p)^i \right)^{\frac{1}{n}} = (1-p)^{\frac{n-1}{2}}$$

we obtain

$$(1 - (1-p)^n)^2 = p^2 \left(\sum_{i=0}^{n-1} (1-p)^i \right)^2 \geq n^2 p^2 (1-p)^{n-1}.$$

Consequently, the right side of (4) is non-negative (in fact it is positive for all $p \in [0, 1)$). It yields that $dt_e/dp > 0, \forall p \in [0, 1)$. It means that function $t_e(p)$ strictly increases in p . □

As follows from the Lemma 2, the higher the probability p of requests entering the system at the instant T , the larger the interval $[0, t_e]$ where players send their requests to the system with a positive density. Also, note that for a given p , the value of t_e may even be less than 0. In this case the probability of p should be increased. Even if $t_e(0) \leq 0$, then the equilibrium strategy is pure, i.e. sending requests to the system at instant $t = T$ with probability 1. Further, we assume that $t_e(0) > 0$.

Remark 1. The expected value $E \left[\frac{1}{X_p+1} \right]$ decreases in p (see [13]).

The Remark 1 means that with increasing the arrival probability at the instant T , the probability of loss at this moment increases. It can be explained by the fact that if more requests arrive in the system at the instant T with increasing of p , then more requests are lost because only one of them is served.

Lemma 3. *If $t_e > 0$, then at the interval $[0, t_e]$ there exists a strictly positive density function $f(t) > 0$ of the arrival times in the system. This interval has no atoms or discontinuities.*

Proof. Consider the interval $[0, t_e]$ where the players enter to the system. We show that in equilibrium, the distribution density function is strictly positive over the entire interval. Let's assume the opposite, i.e. on the interval $[0, t_e]$ there is some interval (t_1, t_2) where none of the players arrives in the system. Then, if one of the players decides to come to the system at the instant t_1 , he will receive service with probability

$$1 - e^{-\mu(t_2-t_1)} + \int_{t_2}^T (1 - e^{-\mu(\theta-t_1)})dP(\theta),$$

where $dP(\theta)$ is a probability that another request arrives to the system at the instant θ . But if this player arrives in the system at the instant t_2 , he will receive service with probability

$$\int_{t_2}^T (1 - e^{-\mu(\theta-t_2)})dP(\theta),$$

which is less than probability to receive service at the instant t_1 . This means that the strategy support $[0, t_e]$ does not contain such discontinuities.

Now, show that the strategy support $[0, t_e]$ has no atoms. Suppose such an atom exists at a point $t \in [0, t_e]$ and the probability that a request arrives at the instant t is $p > 0$. Consider the instant $t+$, which is infinitesimally close on the right to the instant t . Let's take a certain player who is trying to send his request to the system at the instant t . Let the random variable X_p represent the number of his opponents whose requests entered the system at the instant t . Due to the strict positivity of the probability p , this random variable must also be positive. The probability that this player receives service at the instant t is

$$E \frac{1}{X_p + 1} \int_{t+}^T (1 - e^{-\mu(\theta-t+)})dP(\theta),$$

which is smaller than such probability at the instant $t+$:

$$\int_{t+}^T (1 - e^{-\mu(\theta-t+)})dP(\theta).$$

In other words, if the distribution of request arrivals before the instant t_e contains an atom at some point, it is better to send the request to the system immediately after this instant. Unlike the instant T (when the system is closed for arrivals and the player just needs to get a service opportunity), here the service may be interrupted by another request, and the chance of not being selected reduces the likelihood of service. □

3 The Nash Equilibrium in the Queueing Game

Assume that the number of players sending their requests to the system is equal to $N + 1$. Each of them has N opponents that can prevent them from receiving service. For the sake of certainty, let's consider player $N + 1$. Let's assume that at time $t = T$ each of his N opponents sends his request to the system with probability p . Denote X_p the number of players who sent their requests to the server at time T . Then, for player $N + 1$, the probability of receiving service in the system at time T is defined as

$$C(T) = E \left[\frac{1}{X_p + 1} \right].$$

Note that if the number of players is fixed, the random variable X_p obeys the binomial distribution $Bin(N, p)$. So, the payoff function of player $N + 1$ at the instant T is

$$C(T) = \sum_{i=0}^N \binom{N}{i} p^i (1-p)^{N-i} \frac{1}{i+1} = \frac{1 - (1-p)^{N+1}}{p(N+1)}. \tag{5}$$

The probability that player $N + 1$ receives a service at the instant $t_e < T$ in case there is no customers arriving at the interval (t_e, T) is defined by

$$C(t_e) = 1 - (1 - (1-p)^N) e^{-\mu(T-t_e)}. \tag{6}$$

So, in the equilibrium p and t_e must satisfy the equation

$$\frac{1 - (1-p)^{N+1}}{p(N+1)} = 1 - (1 - (1-p)^N) e^{-\mu(T-t_e)},$$

implying

$$t_e = T - \frac{1}{\mu} \log \frac{p(N+1)(1 - (1-p)^N)}{p(N+1) - 1 + (1-p)^{N+1}}. \tag{7}$$

Our objective now is to find the equilibrium density function $f(t)$ for the arrival time in the system on the interval $[0, t_e]$. Define a Markov process with system states (i) at each instant $t \in [0, t_e]$, where $i \in \{0, \dots, N\}$ indicates the number of players who have sent their requests to the system before the time t . This process is inhomogeneous in time, since the request rate in the system decreases in jumps as soon as a new request is received from a successive player. The arrival rate at instant t depends on the chosen strategy and the number k of customers who have already entered the system up to instant t . These rates has the following form $\lambda_k(t) = (N - k) \frac{f(t)}{1 - F(t)}$. Now we can write down the corresponding Kolmogorov backward equations for state probabilities $p_i(t)$

$$\begin{aligned} p'_0(t) &= -\lambda_0(t)p_0(t), \\ p'_i(t) &= -\lambda_i(t)p_i(t) + \lambda_{i-1}(t)p_{i-1}(t) \text{ for } i = 1, \dots, N - 1, \\ p'_N(t) &= \lambda_{N-1}(t)p_{N-1}(t), \end{aligned} \tag{8}$$

which can be resolved to

$$p_i(t) = \binom{N}{i} F(t)^i (1 - F(t))^{N-i} \text{ for } i = 0, \dots, N. \quad (9)$$

The initial state probabilities are $p_0(0) = 1$ and $p_i(0) = 0$ for $i = 1, \dots, N$.

Then the payoff function of player $N + 1$ at the instant $t \in [0, t_e]$ is

$$C(t) = \sum_{i=0}^{N-1} p_i(t) C_{N-i}(t) + p_N(t),$$

where $C_j(t)$ is a probability that player $N + 1$ arriving at the instant $t \in [0, t_e]$ will be served, provided that j customers have not arrived into the system yet before the instant t .

For $j = 1$ let τ_1 is an instant of the arriving request to the system. Then

$$\begin{aligned} C_1(t) &= E(1 - e^{-\mu(\tau_1-t)} | t \leq \tau_1 \leq t_e) + P(\tau_1 = T)(1 - e^{-\mu(T-t)}) \\ &= \frac{1}{1 - F(t)} \left(\int_t^{t_e} dF(\tau)(1 - e^{-\mu(\tau-t)}) + p(1 - e^{-\mu(T-t)}) \right). \end{aligned}$$

For $j = 2$ let τ_1, τ_2 be the instants of the two arriving requests to the system. We obtain

$$\begin{aligned} C_2(t) &= 2E(1 - e^{-\mu(\tau_1-t)} | t \leq \tau_1 \leq \tau_2 \leq t_e) + 2E(1 - e^{-\mu(\tau_1-t)} | t \leq \tau_1 \leq t_e, \tau_2 = T) \\ &\quad + P(\tau_1 = T, \tau_2 = T)(1 - e^{-\mu(T-t)}) \\ &= \frac{1}{(1 - F(t))^2} \left(2 \int_t^{t_e} dF(t_1) \int_{t_1}^{t_e} dF(t_2)(1 - e^{-\mu(t_1-t)}) + 2p \int_t^{t_e} dF(t_1)(1 - e^{-\mu(t_1-t)}) \right. \\ &\quad \left. + p^2(1 - e^{-\mu(T-t)}) \right) \\ &= \frac{1}{(1 - F(t))^2} \left(2 \int_t^{t_e} dF(t_1)(1 - F(t_1))(1 - e^{-\mu(t_1-t)}) + p^2(1 - e^{-\mu(T-t)}) \right). \end{aligned}$$

Arguing the same way we obtain for $k = 2, \dots, N$

$$\begin{aligned} C_k(t) &= kE(1 - e^{-\mu(\tau_1-t)} | t \leq \tau_1 \leq t_e, \tau_1 \leq \tau_j, j = 2, \dots, N) \\ &\quad + P(\tau_j = T, j = 1, \dots, N)(1 - e^{-\mu(T-t)}) \\ &= \frac{1}{(1 - F(t))^k} \left(k \int_t^{t_e} dF(t_1)(1 - F(t_1))^{k-1}(1 - e^{-\mu(t_1-t)}) + p^k(1 - e^{-\mu(T-t)}) \right). \end{aligned}$$

Substituting $C_j(t)$ into $C(t)$ we obtain

$$\begin{aligned} C(t) &= F(t)^N + \sum_{i=0}^{N-1} \binom{N}{i} F(t)^i p^{N-i}(1 - e^{-\mu(T-t)}) \\ &\quad + \sum_{i=0}^{N-1} \binom{N}{i} F(t)^i (N - i) \int_t^{t_e} (1 - e^{-\mu(s-t)})(1 - F(s))^{N-i-1} dF(s). \end{aligned}$$

The first sum equals to

$$(F(t) + p)^N(1 - e^{-\mu(T-t)}) - F(t)^N(1 - e^{-\mu(T-t)}).$$

Simplifying the second sum

$$\begin{aligned} & \sum_{i=0}^{N-1} \binom{N}{i} F(t)^i (N-i) \int_t^{t_e} (1 - e^{-\mu(s-t)})(1 - F(s))^{N-i-1} dF(s) \\ &= N \sum_{i=0}^{N-1} \binom{N-1}{i} F(t)^i \int_t^{t_e} (1 - e^{-\mu(s-t)})(1 - F(s))^{N-i-1} dF(s) \\ &= N \int_t^{t_e} (1 - e^{-\mu(s-t)})(F(t) + 1 - F(s))^{N-1} dF(s). \end{aligned}$$

Hence, we obtain

$$\begin{aligned} C(t) &= (F(t) + p)^N(1 - e^{-\mu(T-t)}) + F(t)^N e^{-\mu(T-t)} \\ &+ N \int_t^{t_e} (1 - e^{-\mu(s-t)})(F(t) + 1 - F(s))^{N-1} dF(s). \end{aligned} \tag{10}$$

The equilibrium payoff function must be constant on the interval $[0, t_e]$, so the distribution $F(t)$ must satisfy the equation $C(t) = C(t_e)$ for $t \in [0, t_e]$, that is

$$\begin{aligned} & (F(t) + p)^N(1 - e^{-\mu(T-t)}) \\ &+ N \int_t^{t_e} (1 - e^{-\mu(s-t)})(F(t) + 1 - F(s))^{N-1} dF(s) + F(t)^N e^{-\mu(T-t)} \\ &= 1 - (1 - (1 - p)^N)e^{-\mu(T-t_e)}. \end{aligned} \tag{11}$$

The probability of arrival at the instant $t = T$ can be found from the normalization condition

$$\int_0^{t_e} dF(t) + p = 1. \tag{12}$$

So, we have established the following:

Theorem 1. *The symmetric Nash equilibrium in the $N + 1$ -person queueing game with preemptive access is described by the distribution function $F(t)$ on the interval $[0, T]$, which has the following properties.*

1. *There is a non-zero probability p of a request entering the system at the instant T .*
2. *At the interval $[t_e, T]$, where t_e is determined by (7), the players do not enter the service system.*
3. *Let the solution of Eq. (7) be negative for $p = 1$, then in equilibrium all players send their requests to the system at instant T . Otherwise, $p < 1$, and t_e is greater than 0; in addition, the PDF $f(t)$ on the support $[0, t_e]$ is determined from Eqs. (11).*
4. *The probability p of a request entering the system at the instant T is determined from Eq. (12).*

5. In equilibrium, the probability that a player receives service is equal to $C(T) = \frac{1-(1-p)^{N+1}}{p(N+1)}$.

Lemma 4. *The distribution function $F(t)$ representing the solution of (11) with the boundary condition $F(t_e) = 1 - p$, where t_e is defined by (7), increases in p at any point of the interval $[0, T]$.*

Proof. Consider two given probabilities $0 < p < q \leq 1$ of request arrival at the instant T that define the boundary conditions for constructing the two distribution functions $F_p(t)$ and $F_q(t)$ as the solutions of (11). The corresponding probabilities of receiving service $C_p(t)$ and $C_q(t)$ are constant on the whole distribution support. By Remark 1 the function $C(\cdot)$ decreases by the probability to arrive at the instant T . Then the probability of loss must be smaller for p than for q on the whole distribution support.

By Lemma 2, we have $t_q = t(q) > t_p = t(p)$ for the corresponding starting points of the intervals where the requests again arrive in the system. That is, the function $F_q(t)$ continues to increase to the value $1 - q$ at the instant when $F_p(t)$ becomes the constant $1 - p > 1 - q$. For $t \in [t_p, T]$ the lemma is true, since in this case $F_p(t) = 1 - p > 1 - q \geq F_q(t)$.

Assume there exists a certain instant $s < t_p$ such that $F_p(t) < F_q(t)$ and $F_p(s) = F_q(s)$. Then $f_p(s) > f_q(s)$, as both distribution functions do not decrease in t , and at the point s the function $F_p(t)$ must cross $F_q(t)$ upwards. Hence, the slope of $F_p(t)$ exceeds that of $F_q(t)$ and therefore $\frac{f_p(s)}{1-F_p(s)} > \frac{f_q(s)}{1-F_q(s)}$. This means that the request rate at the time s is higher for the probability p than for q , while the service rates are the same in both cases. Then the probability of loss at the instant s must be greater for p than for q , which obviously contradicts the fact that the probability of loss is smaller for p than for q on the whole distribution support. \square

Theorem 2. *The symmetric equilibrium distribution F of the arrival times that is defined by Theorem 1 actually exists and is unique.*

Proof. The uniqueness of the equilibrium follows from Lemma 4. The equilibrium condition (12) represents an equation whose left-hand side increases monotonically in p . For $p \approx 0$, the left-hand side equals the probability of request arrival on the interval $[0, t_e]$, which does not exceed 1. For $p = 1$, the left-hand side is not smaller than 1. Therefore, there exists the unique solution p that is associated with the unique value t_e and density function $f(t)$ on $[0, t_e]$. \square

Example 1. Let $T = 4$, $\mu = 0.25$. Here the average service time is $1/\mu = T$. The computations gives the optimal values in the equilibrium for different N which are given in Table 1. The PDFs for the optimal arrival time at the interval $[0, t_e]$ are presented at Fig. 1a.

Example 2. Let $T = 4$, $\mu = 2$. Here the average time to serve request is small comparing with T . The computations give the optimal values in the equilibrium for different N which are given in Table 1. The PDFs for the optimal arrival time at the interval $[0, t_e]$ are presented at Fig. 1b.

Table 1. Optimal p , t_e and payoff $T = 4$.

N	$\mu = 0.25$			$\mu = 2$		
	Payoff $C(t)$	p	t_e	Payoff $C(t)$	p	t_e
2	0.43916	0.74669	1.95199	0.80403	0.21078	3.67268
5	0.25996	0.63972	2.82015	0.62724	0.19150	3.71850
10	0.15600	0.58271	3.32223	0.46092	0.17274	3.77238
20	0.08718	0.54618	3.63511	0.30191	0.15289	3.83874
100	0.01918	0.51629	3.92255	0.07662	0.12923	3.96014

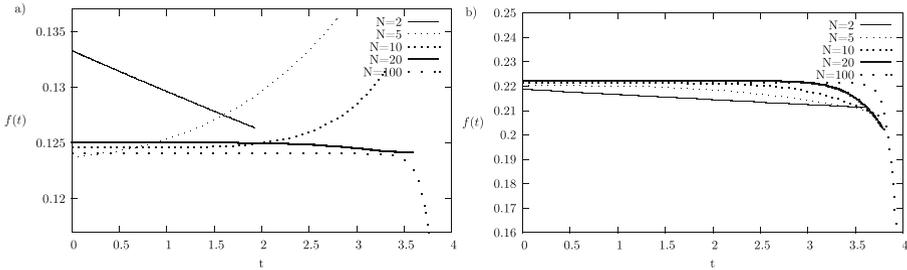


Fig. 1. Equilibrium PDF $f(t)$ for $T = 4$.

4 Conclusion

This paper has studied a game-theoretic model for a single-server queueing system with strategic users in which customers (players) enter the system with preemptive access on a time interval $[0, T]$. As it has been demonstrated, there exists a unique symmetric equilibrium with the following features. The non-zero density function of the arrival times is defined at the time interval $[0, t_e]$. On a time interval $[t_e, T]$ there are no arrivals. At the instant T the players send their requests to the system with a certain positive probability p . Some numerical experiments have been performed to compare the equilibria under different values of the model parameters.

For our future work, we plan to incorporate this approach for the model with unknown (random) number of players and make a comparison between competitive and cooperative behavior in the service system based on the concept of price of anarchy.

References

1. Altman, E., Shimkin, N.: Individually optimal dynamic routing in a processor sharing system. *Oper. Res.* **46**, 776–784 (1998)

2. Breinbjerg, J., Platz, T.T., Østerdal, L.P.: Equilibrium arrivals to a last-come first-served preemptive-resume queue. Working Papers 17–2020, Copenhagen Business School, Department of Economics (2020). https://ideas.repec.org/p/hhs/cbsnow/2020_017.html. Accessed 28 Apr 2022
3. Chirkova, Y.V.: Optimal arrivals in a two-server random access system with loss. *Autom. Remote Control.* **78**(3), 557–580 (2017). <https://doi.org/10.1134/S0005117917030146>
4. Chirkova, J., Mazalov, V., Morozov, E.: Equilibrium in a queueing system with retrials. *Mathematics* **10**(3), 428 (2022)
5. Dimitriou, I.: A queueing system for modeling cooperative wireless networks with coupled relay nodes and synchronized packet arrivals. *Perform. Eval.* **114**(C), 16–31 (2017)
6. Glazer, A., Hassin, R.: $M/M/1$: on the equilibrium distribution of customer arrivals. *Eur. J. Oper. Res.* **13**, 146–150 (1983)
7. Glazer, A., Hassin, R.: Equilibrium arrivals in queues with bulk service at scheduled times. *Transp. Sci.* **21**, 273–278 (1987)
8. Hassin, R., Kleiner, Y.: Equilibrium and optimal arrival patterns to a server with opening and closing times. *IIE Trans.* **43**, 164–175 (2011)
9. Haviv, M.: When to arrive at a queue with tardiness costs. *Perform. Eval.* **70**, 387–399 (2013)
10. Haviv, M., Kella, O., Kerner, Y.: Equilibrium strategies in queues based on time or index of arrival. *Prob. Eng. Inform. Sci.* **24**, 13–25 (2010)
11. Haviv, M., Ravner, L.: A survey of queueing systems with strategic timing of arrivals. *Queueing Syst.* **99**, 163–198 (2021)
12. Jane, R., Juneja, S., Shimkin, N.: The concert queueing game: to wait or to be late. *Discret. Event Dyn. Syst.* **21**, 103–138 (2011)
13. Mazalov, V., Chirkova, J.: *Networking Games: Network Forming Games and Games on Networks*, 322 p. Academic Press, Cambridge (2019)
14. Mazalov, V.V., Chuiko, J.V.: Nash equilibrium in the optimal arrival time problem. *Comput. Technol.* **11**, 60–71 (2006)
15. Haviv, M., Ravner, L.: Strategic timing of arrivals to a finite queue multi-server loss system. *Queueing Syst.* **81**(1), 71–96 (2015). <https://doi.org/10.1007/s11134-015-9453-y>
16. Ravner, L., Haviv, M.: Equilibrium and socially optimal arrivals to a single server loss system. In: *International Conference on Network Games Control and Optimization 2014 (NETGCOOP 2014)*, Trento, Italy, October 2014



Multistage Inventory Model with Probabilistic and Quantile Criteria

Sergey V. Ivanov^(✉)  and Aleksandra V. Mamchur

Moscow Aviation Institute (National Research University),
Volokolamskoe Shosse, 4, Moscow 125993, Russia
sergeyivanov89@mail.ru

Abstract. We consider a multistage inventory model. At each stage, a company determines order quantities of several products to satisfy a demand. The demand is described by a discrete random process. If the demand on a product is more than inventory level of this product, the company has to make additional ordering by market price. Otherwise, the company has to hold the rest of this product by a known price. The company can use a storage. The capacity of the storage is bounded. We consider two objective functions in this model. The first objective function is the probability that the losses are less or equal to a desirable level. The second one is the quantile of losses. To solve the considered problem we reduce them to mixed integer linear programming problems. This reduction is based on introducing auxiliary variables. We suggest conditions ensuring the equivalence of the original problems and the reduced ones. Also, we consider the problems when the distribution of the random process is unknown. For this case, we prove the convergence of the sample approximation method. Numerical results are discussed.

Keywords: Stochastic programming · Multistage problem · Inventory model · Sample approximation · Quantile criterion · Probabilistic criterion

1 Introduction

Stochastic programming problems are optimization problems with loss function depending on random parameters. In these problems, the objective function is defined as a functional of the random loss function. The theory of stochastic programming is described in [1–3]. The most widely used case of the objective function in stochastic programming is the expectation of losses. To take into account risks, the probabilistic and quantile criteria are used [4]. The probabilistic objective function is defined as the probability that the losses are less than or equal to a desirable level. The quantile objective function is the losses that cannot be exceeded with a fixed probability.

To model sequential decisions based on realizations of several random parameters, multistage stochastic programming problems are used. In these problems, the decision of each stage depends on realizations of random parameters on

previous stages. For solving multistage problem with expectation criterion, the Bellman Principle can be applied (see, e.g., [2]). It can be shown that the Bellman Principle is not valid for multistage stochastic problems with quantile criterion. For this reason, other methods should be developed for these problems. By using auxiliary variables, these problems can be reduced to mixed integer programming problems [5,6].

In this paper, a multistage inventory model is considered. The model is based on the model described in [3]. The review on inventory models can be found in [7]. Unlike the model in [3], we consider probabilistic and quantile objective functions. Also, we take into account the capacity of the storage. Based on ideas suggested in [3,8,9] for single-stage and two-stage problems, we suggest a method to reduce the considered problems to mixed integer linear programming problems. We solve these problems by using Gurobi solver [10].

In practice, the distribution of random parameters can be unknown. To solve problems with unknown distribution, sample approximation method can be applied [3]. The method was suggested for expectation optimization problems in [11], for problems with probabilistic constraints in [12], and for problems with probabilistic and quantile criteria in [13]. Application of the sample approximation method for multistage problems with expectation criterion is described in [14]. The convergence of approximations of multistage problems with expectation criterion is proved in [15,16]. In this paper, the convergence of the sample approximation method is proved for the considered problems with discrete distribution of random process.

2 Problem Statement

We consider a company that has to satisfy demand on n types of products. The demand is described by a random process $X = (X_t), t = \overline{1, T}$, taking values in $\mathcal{X} \subset \mathbb{R}^n$ for each t . In this paper, we assume that the set \mathcal{X} is finite. At each stage t , the company determines order quantities $u_t \in \mathbb{R}^n$ of n products. The unit prices of ordering are known and given by a vector c ($c_i > 0$). If the demand on the i -th product is more than inventory level of this product, the company has to make additional ordering by market unit price $b_i > 0, b = (b_1, \dots, b_n)$. If the demand on the i -th product is less than inventory level of this product, the company has to hold the rest of this product by price $h_i > 0, h = (h_1, \dots, h_n)$. The inventory level is equal to

$$z_t = [u_t - x_t + z_{t-1}]_+, \quad t = \overline{1, T}. \tag{1}$$

where $[a]_+ \in \mathbb{R}^n$ is a vector such that $([a]_+)_i = \max\{a_i, 0\}$, x_t is the realization of the random vector X_t . The initial values of the inventory level $z_0 \in \mathbb{R}^n$ is known. Let us define the loss function

$$\Phi(u_1, \dots, u_T, x) = \sum_{t=1}^T (c^\top u_t + b^\top [x_t - u_t - z_{t-1}]_+ + h^\top z_t).$$

We take into account a constraint on the capacity of the storage:

$$(u_t + z_{t-1})^\top v \leq V, \tag{2}$$

where V is the capacity of the storage, $v = (v_1, \dots, v_n)$ is the vector consisting of the volumes of the product units.

The decision vector of each stage is considered as a function of realizations of the random process X_t on previous stages. Let us denote this function by $\mathbf{u}_t: \mathcal{X}^{t-1} \rightarrow \mathbb{R}^n$, $t = \overline{2, T}$. Note that the first stage strategy is deterministic because this is selected before the realization of the demand X_1 is known. Let us introduce the notation

$$X_{[t]} = (X_1, \dots, X_t), \quad x_{[t]} = (x_1, \dots, x_t), \quad \mathbf{u} = (u_1, \mathbf{u}_2, \dots, \mathbf{u}_T).$$

Let us introduce the probability function:

$$P_\varphi(\mathbf{u}) = \mathbf{P} \{ \Phi(u_1, \mathbf{u}_2(X_{[1]}), \dots, \mathbf{u}_T(X_{[T-1]}), X) \leq \varphi \}, \quad (3)$$

where \mathbf{P} is a probability, $\varphi \in \mathbb{R}$ is a fixed value of losses. Notice that the probability function is the probability that the losses are less than or equal to the value φ .

The quantile function is defined by

$$\varphi_\alpha(\mathbf{u}) = \min \{ \varphi \in \mathbb{R} \mid P_\varphi(\mathbf{u}) \geq \alpha \},$$

where $\alpha \in (0, 1)$ is a fixed value of probability. The quantile function shows the minimal value of losses that cannot be exceeded with probability α .

The set of feasible strategies \mathbf{U} is defined as the set of \mathbf{u} with non-negative values such that constraints (1) and (2) are satisfied for $u_t = \mathbf{u}_t(x_{[t-1]})$, $t = \overline{2, T}$,

Thus, the probability maximization problem

$$P_\varphi(\mathbf{u}) \rightarrow \max_{\mathbf{u} \in \mathbf{U}} \quad (4)$$

and the quantile minimization problem

$$\varphi_\alpha(\mathbf{u}) \rightarrow \min_{\mathbf{u} \in \mathbf{U}} \quad (5)$$

are considered.

Let us notice that problems (4) and (5) can be considered for the random process X with arbitrary distribution of the random parameters. In the next section these problems will be rewritten for the discrete distribution.

3 Equivalent Mixed Integer Problems

We suppose that each random vector X_t has a finite number of realizations belonging to the set $\mathcal{X} = \{x^1, \dots, x^M\}$. Then we can use the notation $u_t^{i_1 \dots i_{t-1}} = \mathbf{u}_t(x^{i_1}, \dots, x^{i_{t-1}})$, $t = \overline{2, T}$. To simplify the notation, we denote by I_t the tuple of indices (i_1, \dots, i_t) . Thus,

$$u_t^{I_{t-1}} = u_t^{i_1 \dots i_{t-1}} = \mathbf{u}_t(x^{i_1}, \dots, x^{i_{t-1}}), \quad I_t \in \{1, \dots, M\}^t.$$

We use the notation $u_1^{I_0} = u_1, z_0^{I_0} = z_0$. Let us denote by u the tuple consisting of the variable u_1, M variables $u_2^{i_1}, M^2$ variables $u_3^{i_1 i_2}, \dots, M^{T-1}$ variables $u_T^{i_1 \dots i_{T-1}}$. Let U be the set of tuples u . Let us denote by $p_I = p_{i_1 \dots i_T}$ the probability of the realization $x^I = (x^{i_1}, \dots, x^{i_T})$ of the random process X , where $I = I_T$. Also, we introduce variables $z_t^{I_t} = z_t^{i_1 \dots i_t}$ corresponding to inventory levels for known realizations of the random process $X_{[t]}, t = \overline{1, T}$.

Let us notice that the objective functions in problems (4) and (5) can be considered as functions of u :

$$P_\varphi(\mathbf{u}) = \tilde{P}_\varphi(u) = \mathbf{P} \left\{ \tilde{\Phi}(u, X) \leq \varphi \right\},$$

$$\varphi_\alpha(\mathbf{u}) = \tilde{\varphi}_\alpha(u) = \min \left\{ \varphi \in \mathbb{R} \mid \tilde{P}_\varphi(u) \geq \alpha \right\},$$

where

$$\tilde{\Phi}(u, x) = \Phi \left(u_1, u_2^{i_1}, \dots, u_T^{i_1 \dots i_{T-1}}, x^I \right) \text{ if } x = x^I, I \in \{1, \dots, M\}^T.$$

This allows us to replace problems (4) and (5) by problems

$$\tilde{P}_\varphi(u) \rightarrow \max_{u \in U}, \tag{6}$$

$$\tilde{\varphi}_\alpha(u) \rightarrow \min_{u \in U} \tag{7}$$

subject to

$$z_t^{I_t} = \left[u_t^{I_{t-1}} - x_t^{i_t} + z_{t-1}^{I_{t-1}} \right]_+, \quad t = \overline{1, T}, \tag{8}$$

$$(u_t^{I_{t-1}} + z_{t-1}^{I_{t-1}})^\top v \leq V, \tag{9}$$

$$u_t^{I_{t-1}} \geq 0, \quad I \in \{1, \dots, M\}^T. \tag{10}$$

Constraints (8) and (9) follow from (1) and (2).

We will reduce problems (6) and (7) to mixed integer programming problems by using a method described in [8] for the discrete distribution. Let us introduce variables $\delta_I, I \in \{1, \dots, M\}^T$, corresponding to realizations of the random process X . The value $\delta_I = 1$ if $\tilde{\Phi}(u, x^I) \leq \varphi$, otherwise $\delta_I = 0$.

The ordered set of δ_I is denoted by δ . Then the probability maximization problem (6) can be reduced to the problem:

$$\sum_{I \in \{1, \dots, M\}^T} p_I \delta_I \rightarrow \max_{u, \delta} \tag{11}$$

subject to

$$\tilde{\Phi}(u, x^I) \leq \varphi + L(1 - \delta_I), \quad I \in \{1, \dots, M\}^T,$$

where L is a sufficiently large constant.

The quantile minimization problem (7) is reduced to the problem:

$$\varphi \rightarrow \min_{\varphi, u, \delta} \tag{12}$$

subject to

$$\begin{aligned} \tilde{\Phi}(u, x^I) &\leq \varphi + L(1 - \delta_I), \quad I \in \{1, \dots, M\}^T, \\ \sum_{I \in \{1, \dots, M\}^T} p_I \delta_I &\geq \alpha. \end{aligned} \tag{13}$$

Let z be the vector consisting of z_t^I . Then, it follows from (11) that probability maximization problem (6) is equivalent to the problem:

$$\sum_{I \in \{1, \dots, M\}^T} p_I \delta_I \rightarrow \max_{u, z, \delta} \tag{14}$$

subject to

$$\sum_{t=1}^T \left(c^\top u_t^{I_{t-1}} + b^\top \left[x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \right]_+ + h^\top z_t^{I_t} \right) \leq \varphi + L(1 - \delta_I), \tag{15}$$

and (8)–(10).

By using the variables z , we obtain from (12) that problem (5) is equivalent to the problem

$$\varphi \rightarrow \min_{\varphi, u, z, \delta} \tag{16}$$

subject to (8)–(10), (15), and (13).

Let us introduce an auxiliary vector y consisting of auxiliary variables $y_t^I \in \mathbb{R}^n$ to transform the considered problems into linear ones. Under the conditions given below in Theorem 1, y_t^I is equal to the vector whose i -th coordinate is equal to the additional ordering quantity of the i -th product. Let us consider the problem

$$\sum_{I \in \{1, \dots, M\}^T} p_I \delta_I \rightarrow \max_{u, z, y, \delta} \tag{17}$$

subject to

$$\sum_{t=1}^T \left(c^\top u_t^{I_{t-1}} + b^\top y_t^{I_t} + h^\top z_t^{I_t} \right) \leq \varphi + L(1 - \delta_I), \tag{18}$$

$$x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \leq y_t^{I_t}, \quad t = \overline{1, T}, \tag{19}$$

$$u_t^{I_{t-1}} - x_t^{i_t} + z_{t-1}^{I_{t-1}} \leq z_t^{I_t}, \tag{20}$$

$$(u_t^{I_{t-1}} + z_{t-1}^{I_{t-1}})^\top v \leq V, \tag{21}$$

$$u_t^{I_{t-1}} \geq 0, \quad z_t^{I_t} \geq 0, \quad y_t^{I_t} \geq 0, \quad \delta_I \in \{0, 1\}, \quad I \in \{1, \dots, M\}^T. \tag{22}$$

Here and below, we write $a \leq b$, where a and b are vectors, if and only if $a_i \leq b_i$ for all i . Problem (17) contains $(n + 2nM)(1 + M + \dots + M^{T-1})$ real variables, M^T integer variables, and $M^T + (2nM + 1)(1 + M + \dots + M^{T-1})$ constraints.

Theorem 1. *Suppose that $b \leq h$, each random vector X_t has a finite number of realizations. Then problems (4) and (17) are equivalent.*

Proof. It has been proved above that problem (14) is equivalent to problem (6). Let us notice that a solution to (14) is feasible in problem (17) for $y_t^{I_t} = \left[x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \right]_+$ because $z_t^{I_t} = \left[u_t^{I_{t-1}} - x_t^{i_t} + z_{t-1}^{I_{t-1}} \right]_+$. Since the objective functions (14) and (17) are the same, the optimal objective value in (17) is more than or equal to the optimal one in (14).

Now, we show that there exists a solution to (17) satisfying the equalities

$$z_t^{I_t} = \left[u_t^{I_{t-1}} - x_t^{i_t} + z_{t-1}^{I_{t-1}} \right]_+, \quad (23)$$

$$y_t^{I_t} = \left[x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \right]_+. \quad (24)$$

From this, it will follow that the optimal objective value in (17) cannot be more than the optimal one in (14) because in this case the values of $z_t^{I_t}$ and $u_t^{I_{t-1}}$ are feasible in (14).

Suppose that (24) is not valid, i.e., $y_t^{I_t} > \left[x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \right]_+$. Then replacing $y_t^{I_t}$ by

$$\tilde{y}_t^{I_t} = \left[x_t^{i_t} - u_t^{I_{t-1}} - z_{t-1}^{I_{t-1}} \right]_+$$

does not reduce the optimal objective value in problem (17). Thus, we can consider that constraint (24) is valid. Now, suppose that (23) is not valid. Let us replace $z_t^{I_t}$ by $\tilde{z}_t^{I_t} = z_t^{I_t} - \Delta$, where

$$\Delta = z_t^{I_t} - \left[u_t^{I_{t-1}} - x_t^{i_t} + z_{t-1}^{I_{t-1}} \right]_+.$$

Then, due to (19), we need to change $y_{t+1}^{I_{t+1}}$ by $\tilde{y}_{t+1}^{I_{t+1}} = y_{t+1}^{I_{t+1}} + \Delta$. Since $b \leq h$, from (18) it follows that this changing does not reduce the optimal objective value in problem (17). Therefore, it is proved that there exists a solution to problem (17) satisfying the equalities (23) and (24). Thus, the theorem is proved.

Remark 1. If the condition $b \leq h$ is not satisfied, we cannot guarantee that equality (23) is true. In this case, $z_t^{I_t}$ is not equal to the inventory level. This implies that the solution to problem (17) is not feasible in problem (11).

In the same way, the quantile minimization problem (5) can be reduced to the mixed integer linear programming problem

$$\varphi \rightarrow \min_{\varphi, u, z, y, \delta} \quad (25)$$

subject to (18)–(22) and (13).

Compared to problem (17), problem (25) has one additional variable φ and one additional constraint (13).

Theorem 2. *Suppose that $b \leq h$, each random vector X_t has a finite number of realizations. Then problems (5) and (25) are equivalent.*

Proof. We have shown that problem (5) is equivalent to problem (16). Unlike Theorem 1, minimization problems are considered. Since problems (17) and (25) have the same constraints (18)–(19), we conclude that the optimal objective value in (25) is less than or equal to the optimal one in (16). As in the proof of Theorem 2, it can be noticed that there exists a solution to (25) satisfying the equalities (23) and (24). This proves Theorem 2.

4 Sample Approximation

Suppose that the probabilities p_I are unknown, but there is a sample $X(1), \dots, X(N)$ of realizations of the random process X . By using the sample, the probabilities p_I can be estimated:

$$\hat{p}_I^N = \frac{m_I}{N},$$

where N is the sample size, m_I is number of realization x^I in the sample, $I \in \{1, \dots, M\}^T$. Then the probability function (3) can be estimated by the function

$$P_\varphi^N(\mathbf{u}) = \frac{1}{N} \sum_{\nu=1}^N \chi_{(-\infty, \varphi]} (\Phi(u_1, \mathbf{u}_2(X_{[1]}(\nu)), \dots, \mathbf{u}_T(X_{[T-1]}(\nu)), X(\nu))),$$

where

$$\chi_A(a) = \begin{cases} 1 & \text{if } a \in A, \\ 0 & \text{if } a \notin A. \end{cases}$$

The estimator of the quantile function is defined by the rule:

$$\varphi_\alpha^N(\mathbf{u}) = \min \{ \varphi \in \mathbb{R} \mid P_\varphi^N(\mathbf{u}) \geq \alpha \}.$$

Let us consider the sample approximation of the probability maximization problem

$$P_\varphi^N(\mathbf{u}) \rightarrow \max_{\mathbf{u} \in \mathbf{U}} \tag{26}$$

and the sample approximation of the quantile minimization problem

$$\varphi_\alpha^N(\mathbf{u}) \rightarrow \min_{\mathbf{u} \in \mathbf{U}}. \tag{27}$$

From Theorem 1 and Theorem 2, it follows that problems (26) and (27) can be reduced to mixed integer problems if $h \geq b$. Thus problem (26) is reduced to the problem

$$\sum_{I \in \{1, \dots, M\}^T} \hat{p}_I^N \delta_I \rightarrow \max_{u, z, y, \delta} \tag{28}$$

subject to (18)–(22).

By Theorem 2, problem (27) is equivalent to the problem

$$\varphi \rightarrow \min_{\varphi, u, z, y, \delta} \tag{29}$$

subject to (18)–(22) and

$$\sum_{I \in \{1, \dots, M\}^T} \hat{p}_I^N \delta_I \geq \alpha.$$

Let us notice that the number of variables in the approximation problems (28) and (29) is equal to the number of variables in the mixed integer problems (17) and (25). This number does not depend on the sample size.

5 Convergence of Sample Approximations

Let U_φ be the set of solutions to problem (6), and let V_α be the set of solutions to problem (7). Let us denote by U_φ^N and V_α^N the sets of solutions to approximation problems (28) and (29), respectively. The optimal objective values of problems (6), (7), (28), (29) are denoted by α^* , φ^* , α_N^* , φ_N^* , respectively. Let

$$D(A, B) = \sup_{a \in A} \inf_{b \in B} \|a - b\|$$

be the deviation of the set A from the set B .

Theorem 3. *Suppose that each random vector X_t has a finite number of realizations. Then*

$$\begin{aligned} \lim_{N \rightarrow \infty} \alpha_N^* &= \alpha^*, \\ \lim_{N \rightarrow \infty} D(U_\varphi^N, U_\varphi) &= 0 \end{aligned}$$

almost surely.

Proof. It easily seen that the function $\tilde{\Phi}$ is continuous. From this, it follows [4] that there exists a solution to problem (6). Since $\lim_{\|u\| \rightarrow \infty} \Phi(u, x) = +\infty$, we can suppose that the set of feasible strategies is compact. It was proved in [13] that, under these conditions, the statement of Theorem 3 is valid. The convergence of the set deviations is proved in [17].

Theorem 4. *Suppose that each random vector X_t has a finite number of realizations. Let $\max_{u \in U} P_{\varphi^*}(u) > \alpha$; then*

$$\begin{aligned} \lim_{N \rightarrow \infty} \varphi_N^* &= \varphi^*, \\ \lim_{N \rightarrow \infty} D(V_\alpha^N, V_\alpha) &= 0. \end{aligned}$$

almost surely.

Proof. As in the proof of Theorem 3, we can consider the U being compact. Since the function $\tilde{\Phi}$ is continuous and $\max_{u \in U} P_{\varphi^*}(u) > \alpha$, the conditions of the convergence given in [13] are satisfied for the sample approximations.

Let us notice that the condition $\max_{u \in U} P_{\varphi^*}(u) > \alpha$ is satisfied if

$$\alpha \neq \sum_{i \in I} p_I$$

for all $I \in \{1, \dots, M\}^T$. This means that the conditions of Theorem 4 are not satisfied only for a finite number of values α .

6 Numerical Results

Let us consider the model for 5 types of products with the following data:

$$\begin{aligned} c &= (1.0, 1.5, 2.0, 1.9, 2.1)^\top, \\ b &= (1.2, 1.7, 2.2, 2.4, 2.6)^\top, \\ h &= (1.3, 1.8, 3.0, 2.7, 3.1)^\top, \\ v &= (1, 3, 1, 2, 3)^\top, \quad V = 80 \\ z_0 &= (5, 7, 0, 4, 0)^\top. \end{aligned}$$

The number of stages is equal to 2. We assume that random vectors X_1 and X_2 are independent and identically distributed. The probability maximization problem and the quantile minimization problems have been solved for different number of realizations of the random vectors X_1, X_2 . Let us notice that the random process X has M^2 realizations, where M is the number of realizations of X_1 . We studied the dependence of the solution on the number M . We considered the following values of M : 5, 7, 9, 12, 13, 15. Realizations are given in Table 1. All realizations are assumed to be equiprobable.

Table 1. Realizations of X_1, X_2 .

$x^1 = (5.14, 7.78, 12.87, 8.28, 9.23)^\top$	$x^2 = (8.88, 8.97, 11.83, 5.14, 14.51)^\top$
$x^3 = (9.77, 10.11, 11.07, 10.96, 11.51)^\top$	$x^4 = (8.88, 10.60, 6.48, 10.38, 11.63)^\top$
$x^5 = (12.01, 14.22, 12.38, 10.48, 12.63)^\top$	$x^6 = (11.78, 5.21, 14.06, 10.48, 10.87)^\top$
$x^7 = (14.81, 10.67, 14.20, 10.94, 8.37)^\top$	$x^8 = (9.27, 6.84, 13.08, 10.86, 10.89)^\top$
$x^9 = (10.44, 11.28, 5.47, 14.43, 10.53)^\top$	$x^{10} = (11.12, 14.64, 13.62, 6.86, 12.46)^\top$
$x^{11} = (13.33, 7.22, 8.67, 6.48, 14.83)^\top$	$x^{12} = (11.34, 7.97, 9.99, 14.62, 7.74)^\top$
$x^{13} = (9.84, 10.28, 9.78, 13.64, 10.20)^\top$	$x^{14} = (5.57, 6.33, 14.18, 11.03, 13.83)^\top$
$x^{15} = (11.54, 10.10, 6.99, 5.13, 9.00)^\top$	

For the computations we used M first values x^1, \dots, x^M . The computations were made on computer with Intel Core i9-10900 (16 GB RAM, 2.80 GHz) by using Gurobi optimization solver.

Table 2. Probability maximization.

M	u_1^*	α^*	τ
5	(0.140000, 0.000000, 6.885221, 2.120427, 10.047668)	0.88	0.04
7	(3.880000, 1.233714, 9.026856, 2.141001, 8.370000)	0.8367347	0.06
9	(3.880000, 0.000000, 5.740765, 4.859966, 8.886434)	0.8765432	0.27
12	(6.0023178, 0.2528767, 6.9803990, 3.0925865, 7.7400000)	0.8680556	0.87
13	(5.076347, 0.000000, 5.470000, 3.010110, 9.027598)	0.8757396	1.24
15	(2.690324, 0.000000, 6.640171, 2.963143, 8.454006)	0.8711111	212.59

Results of solving the probability maximization problem with $\varphi = 180$ are given in Table 2. We present the optimal objective values α^* , optimal solutions of the first stage u_1^* , and computation time τ (seconds).

We can see that the probability maximization problem can be successfully solved for 225 realizations of the random process X . Notice that the equivalent mixed integer problem (17) has 2642 constraints and 2706 variables.

Results for the quantile minimization problem with $\alpha = 0.9$ are given in Table 3, where φ^* is the optimal objective value, u_1^* is an optimal solution of the first stage, τ is computation time in seconds.

Table 3. Quantile minimization.

M	u_1^*	φ^*	τ
5	(3.880000, 0.000000, 10.046562, 3.484375, 8.368229)	182.0202	0.11
7	(3.880000, 0.000000, 10.491562, 6.224718, 6.393000)	183.5875	0.25
9	(3.880000, 0.1189006, 6.1382983, 5.2575000, 8.3700000)	181.2283	2.01
12	(5.245558, 0.000000, 6.076668, 3.742764, 8.370000)	181.7486	6.20
13	(5.245558, 0.000000, 6.076668, 3.616778, 8.370000)	181.8116	223.22
15	(3.694711, 0.000000, 7.327970, 2.601477, 8.385202)	181.3834	3739.25

Comparing the results for the two problems, we can see that the quantile minimization requires more computations. It can be noticed that the structure of the optimal solutions are similar for different number of realizations.

Also, the problem of expected losses minimization [3] was solved for these data. The results of solving the problem are presented in Table 4, where m^* is the minimal expectation of the losses, u_1^* is an optimal solution of the first stage. The problem of expected losses minimization is reduced to a linear programming problem [3], and it does not require such complex calculations as the quantile and probability ones. From Table 4, it can be seen that the solution differs slightly from the solution of probabilistic and quantile problems. However, the minimal expected losses is less than the minimal quantiles of losses.

Table 4. Expected losses minimization.

M	u_1^*	m^*
5	(0.14, 0.78, 6.48, 1.14, 9.23)	164.754
7	(0.14, 0.00, 6.48, 4.28, 8.37)	170.2391
9	(3.88, 0.00, 5.47, 4.28, 9.23)	169.6113
12	(3.88, 0.00, 5.47, 2.48, 8.37)	170.4867
13	(3.88, 0.00, 5.47, 2.86, 8.37)	170.7813
15	(3.88, 0.00, 5.47, 2.48, 9.00)	168.6431

7 Conclusion

We suggested a method to solve optimization problems with probabilistic and quantile criteria in the multistage inventory level problem. For the case of discrete distribution, the problem was reduced to a mixed integer problem. However, due to the large dimension of the equivalent problem, the problems are hard for numerical solving. The growth of size is exponential with the number of stages. The numerical results have shown that the problem can be solved for 225 realizations of the process. Therefore, approximation methods for solving these problems can be a topic of additional research. The convergence of the sample approximation method was proved for the discrete distribution of the random parameters, although the continuous distribution is more common for the random demand. Unfortunately, the case of arbitrarily distribution is more complicated because this requires research on the convergence in functional spaces of strategies. This can be studied in future research. Also, the suggested methods can be applied for a wider class of multistage stochastic programming problems.

Acknowledgements. The study was supported by the Russian Science Foundation (project No. 22-21-00213), <https://rscf.ru/project/22-21-00213/>.

References

1. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, New York (2011). <https://doi.org/10.1007/978-1-4614-0237-4>
2. Kall, P., Wallace, S.W.: Stochastic Programming. Wiley, Chichester (1994)
3. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming. Modeling and Theory, SIAM, Philadelphia (2014)
4. Kibzun, A.I., Kan, Y.S.: Stochastic Programming Problems with Probability and Quantile Functions. Wiley, Chichester (1996)
5. Kibzun, A.I., Khromova, O.M.: On reduction of the multistage problem of stochastic programming with quantile criterion to the problem of mixed integer linear programming. Autom. Remote Control **75**(4), 688–699 (2014). <https://doi.org/10.1134/S0005117914040092>

6. Kibzun, A.I., Ignatov, A.N.: Reduction of the two-step problem of stochastic optimal control with bilinear model to the problem of mixed integer linear programming. *Autom. Remote Control* **77**(12), 2175–2192 (2016). <https://doi.org/10.1134/S0005117916120079>
7. Zipkin, P.H.: *Foundations of Inventory Management*. McGraw-Hill (2000)
8. Kibzun, A.I., Naumov, A.V., Norkin, V.I.: On reducing a quantile optimization problem with discrete distribution to a mixed integer programming problem. *Autom. Remote Control* **74**(6), 951–967 (2013). <https://doi.org/10.1134/S0005117913060064>
9. Norkin, V.I., Kibzun, A.I., Naumov, A.V.: Reducing two-stage probabilistic optimization problems with discrete distribution of random data to mixed-integer programming problems*. *Cybern. Syst. Anal.* **50**(5), 679–692 (2014). <https://doi.org/10.1007/s10559-014-9658-9>
10. Gurobi Optimization. <https://www.gurobi.com/>. Accessed 27 Feb 2022
11. Artstein, Z., Wets, R.J.-B.: Consistency of minimizers and the SLLN for stochastic programs. *J. Convex Anal.* **2**, 1–17 (1996)
12. Pagnoncelli, B.K., Ahmed, S., Shapiro, A.: Sample average approximation method for chance constrained programming: theory and applications. *J. Optim. Theory Appl.* **142**, 399–416 (2009) <https://doi.org/10.1007/s10957-009-9523-6>
13. Ivanov, S.V., Kibzun, A.I.: On the convergence of sample approximations for stochastic programming problems with probabilistic criteria. *Autom. Remote Control* **79**(2), 216–228 (2018). <https://doi.org/10.1134/S0005117918020029>
14. Shapiro, A.: Inference of statistical bounds for multistage stochastic programming problems. *Math. Methods Oper. Res.* **58**, 57–68 (2003). <https://doi.org/10.1007/s001860300280>
15. Pennanen, T.: Epi-convergent discretizations of multistage stochastic programs. *Math. Oper. Res.* **30**(1), 245–256 (2005). <https://doi.org/10.1287/moor.1040.0114>
16. Pennanen, T.: Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Math. Program. Ser. B.* **116**, 461–479 (2009) <https://doi.org/10.1007/s10107-007-0113-9>
17. Ivanov, S.V., Ignatov, A.N.: Sample approximations of bilevel stochastic programming problems with probabilistic and quantile criteria. In: Pardalos, P., Khachay, M., Kazakov, A. (eds.) *MOTOR 2021*. LNCS, vol. 12755, pp. 221–234. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77876-7_15



Pricing in Two-Sided Markets on the Plain with Different Agent Types

Elena Konovalchikova^{1,3}  and Anna Ivashko^{2,3} 

¹ Laboratory of Digital Technologies in Regional Development,
Karelian Research Centre of the Russian Academy of Sciences,
Pushkinskaya Street, 11, Petrozavodsk 185910, Russia

konovalchikova.en@mail.ru

² Institute of Applied Mathematical Research, Karelian Research Centre of the
Russian Academy of Sciences, Pushkinskaya Street, 11, Petrozavodsk 185910, Russia
aivashko@krc.karelia.ru

³ Petrozavodsk State University, 33, Lenina Street, Petrozavodsk 185910, Russia

Abstract. The article investigates the price equilibrium in two-sided markets of platforms with cross-side network effects for users from different groups. The focus is on the problem of optimal pricing in two-sided markets where the location of platforms for different types of agents is taken into account.

The model deals with agents belonging to two groups, who are evenly distributed on the plane of the circle. Agents from both groups choose between two platforms, their rationale being the utility they can derive from visiting the platforms. The agents' utility function is constructed with Hotelling's specification involved, and therefore includes the value of the network effect from the interaction of one group with members of the other and the total costs of visiting the platforms, including transport costs. The payoff of each platform depends on the number of agents of both groups on the platform, the entry fee, and the costs of servicing the users.

We find the optimal two-sided market pricing strategies for symmetrically located platforms for two scenarios. In the first case agents from both groups can join only one platform, whereas in the second case members of the second group can join both platforms simultaneously. Numerical results for different parameters of the problem are compared.

Keywords: Two-sided platform market · Network externalities · Pricing · Hotelling's duopoly · Nash equilibrium · Optimal location of platforms

1 Introduction

The papers [1, 3, 9] have drawn much attention to the study of two-sided markets involving platforms. Interest in such studies has been growing lately as digital technologies develop and spread, and as digital platforms emerge as a new business model. The platforms in such studies are firms that act as mediators facilitating interactions between members of two or more different groups present in

the market. A key feature of two-sided markets involving platforms is that interactions between members of different groups on a platform are accompanied by network externalities, i.e., an increase in the number of members of one group on the platform causes an increase in the individual utility of members of the other group. Platforms are common in many markets. In media markets, e.g., the platforms are the various media services (Netflix, Okko, etc.), which bring together two groups of users: content viewers and advertisers, who gain the viewer's contact information. The platforms in the medical services market are medical insurance companies - intermediaries between customers and providers of medical services. Computer operating systems, payment card systems, supermarkets, dating services, taxi and food delivery services, media outlets - this is an incomplete listing of platforms existing in two-sided markets.

A fundamental challenge in two-sided markets is to attract members of different groups to the platform. The main tool for attracting users to a platform is the user fee. Proper pricing in a platform is a key to its successful development. The literature considers two approaches to setting the fee for the use of platform services. Thus, the price of using a payment card platform is proportional to the number of transactions made, and the price of using the platform in the case of supermarkets and dating services is proportional to the number of agents on the platform. A thing to remember when studying the pricing strategies on platforms is that members of groups in the market may be able to use the services of two or more platforms simultaneously. Agents who join only one platform are commonly referred to as *single-homing* agents, while those joining two or more platforms at a time are called *multi-homing* agents. Depending on the type of market group members, platforms can apply different pricing schemes, which are analyzed in [2,4], and the papers [5,8,10] specifically deal with discriminative pricing strategies. The above papers assume the market to be linear. Hotelling's duopoly model with the Euclidian distance where the market lies on the plane of the circle is considered in [7].

The two-sided market model in this paper is a generalization of the well-known Armstrong model under the assumption that the market lies on the plane. Agents' utility is determined including Hotelling's specification with Euclidian metric. Members of one group are supposed to be single-homing, and members of the other group can be either single-homing or multi-homing. Assuming that the market size is fixed and agents from both groups take part in transactions, we find the optimal pricing strategies and compare the results for two cases. In the first case agents in both groups are single-homing, and in the second case agents from the second group can be multi-homing.

The article below is structured as follows. The second section describes the basic model for a two-sided market on the plane of the circle. The third section analyzes the price equilibrium for a model with platforms symmetrically located on the plane under the assumption that members of both groups are single-homing. Analysis of the model where agents from the second group can enter both platforms simultaneously and comparison of the results are given in the

fourth section. The Conclusions give the general takeaway and the possible vectors for further research.

2 Basic Setting

We consider a model of a duopoly in a two-sided market where two non-intersecting groups of agents interact. Suppose group **1** consists of customers and group **2** — consists of sellers. Agents from both groups are distributed evenly over the circle S and their location is defined by the points $A_i(x_i, y_i)$, where $x_i, y_i \in [-1, 1]$, $i = 1, 2$. The sellers and the customers interact on the platforms **I** and **II**, which are located, respectively, in the points $(a, 0)$ and $(b, 0)$, where $-1 \leq b < a \leq 1$ (see Fig. 1). In choosing the platform, agents of both groups are governed by the utility they can derive from joining one or the other platform. The utility of agents depends on cross-side network externalities, i.e., an increase in the size of one group of agents on the platform leads to an increase in the individual utility of agents in the other group on the platform, and vice versa. The Hotelling’s specification is taken into account when determining the agents’ utility function.

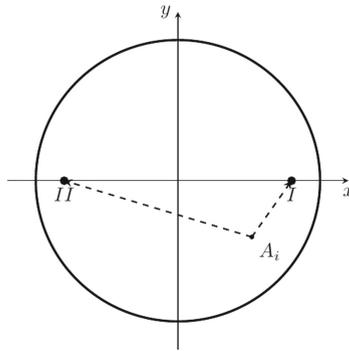


Fig. 1. Two-sided platform market on a plane.

The following notations are introduced: $n_i^{(j)}$ is the size of the group i on the platform j ($i = 1, 2, j = I, II$); $p_i^{(j)}$ is the price that an i -th group agent pays to visit the platform j ($i = 1, 2, j = I, II$); α, β is the degree to which the number of agents in the second (first) group present on the platform influences the payoff of the first (second) group; t_i is the strength of the effect of the transport costs of visiting both platforms by i -th group agents ($i = 1, 2$).

The utility function for customers visiting the platform **I** located in the point $(a, 0)$ has the form:

$$u_1^{(I)} = \alpha \cdot n_2^{(I)} - p_1^{(I)} - \sqrt{(x_1 - a)^2 + y_1^2} \cdot t_1, \quad (1)$$

and the utility of visiting the platform *II* in the point $(b, 0)$ has

$$u_1^{(II)} = \alpha \cdot n_2^{(II)} - p_1^{(II)} - \sqrt{(x_1 - b)^2 + y_1^2} \cdot t_1. \quad (2)$$

The utility function for sellers visiting the platforms *I* and *II* is determined similarly, being, respectively

$$u_2^{(I)} = \beta \cdot n_1^{(I)} - p_2^{(I)} - \sqrt{(x_2 - a)^2 + y_2^2} \cdot t_2, \quad (3)$$

$$u_2^{(II)} = \beta \cdot n_1^{(II)} - p_2^{(II)} - \sqrt{(x_2 - b)^2 + y_2^2} \cdot t_2. \quad (4)$$

Observe that the formulas (1)–(4) are true for the case where all agents are *single-homing*. In other words, each agent joins only one of the platforms. This means that if the market size is fixed, the following equality takes place $n_i^{(I)} + n_i^{(II)} = \pi$, where $i = 1, 2$. There is, however, another variant of agent interaction with platforms. In some cases, agents can join different platforms simultaneously. Such agents are usually referred to as *multi-homing*. For customers, multi-homing is obviously disadvantageous as it increases their costs of visiting two platforms simultaneously. Hence, we will assume in the following that only sellers can join both platforms. Considering that part of the sellers can be single-homing, and another part multi-homing, the following applies to the sellers group: $n_2^{(I)} + n_2^{(II)} \geq \pi$. The utility derived by multi-homing agents from group **2** from joining the platforms **I** and **II** will have the form:

$$u_2^{(I+II)} = \beta \cdot (n_1^{(I)} + n_1^{(II)}) - p_2^{(I)} - p_2^{(II)} - \left(\sqrt{(x_2 - a)^2 + y_2^2} + \sqrt{(x_2 - b)^2 + y_2^2} \right) \cdot t_2.$$

With all agents in the buyers group being single-homing, the utility of multi-homing sellers can be written in the following form:

$$u_2^{(I+II)} = \beta \cdot \pi - p_2^{(I)} - p_2^{(II)} - \left(\sqrt{(x_2 - a)^2 + y_2^2} + \sqrt{(x_2 - b)^2 + y_2^2} \right) \cdot t_2. \quad (5)$$

Setting the fees $p_i^{(I)}$ and $p_i^{(II)}$ ($i = 1, 2$) for agents from both groups, the platforms **I** and **II** wish to maximize the profit, which is calculated from the formulas:

$$H^{(I)}(p_1^{(I)}, p_2^{(I)}) = n_1^{(I)}(p_1^{(I)} - g_1) + n_2^{(I)}(p_2^{(I)} - g_2), \quad (6)$$

$$H^{(II)}(p_1^{(II)}, p_2^{(II)}) = n_1^{(II)}(p_1^{(II)} - g_1) + n_2^{(II)}(p_2^{(II)} - g_2), \quad (7)$$

where g_1 and g_2 are the platform's costs of servicing users belonging to the respective groups.

A similar problem was solved in [6], where equilibrium in a pricing game was found for the case of a market lying on the plane of the square and with

members of both groups being single-homing agents. The current paper solves the optimal pricing problem for a market lying on the plane of the circle and analyzes equilibria for the cases where members of both groups are single-homing agents and where members of only one group can be multi-homing.

3 Model with Single-Homing Agents

We assume that the platforms are situated in points located symmetrically relative to the center of the circle S . For definiteness, the platforms **I** and **II** are said to be located in the points $(a, 0)$ and $(-a, 0)$, respectively. Suppose that members of both groups are single-homing agents and join one of the platforms, i.e., the number of agents of groups **1** and **2** meets the condition $n_i^{(I)} + n_i^{(II)} = \pi$, $i = 1, 2$. Thus, according to (1) and (2), the utility of agents from group **1** from visiting the platforms **I** and **II** has the form

$$u_1^{(I)} = \alpha \cdot n_2^{(I)} - p_1^{(I)} - \sqrt{(x_1 - a)^2 + y_1^2} \cdot t_1,$$

and

$$u_1^{(II)} = \alpha \cdot n_2^{(II)} - p_1^{(II)} - \sqrt{(x_1 + a)^2 + y_1^2} \cdot t_1.$$

Similarly, (3) and (4) indicate that the utility of agents from group **2** from visiting both platforms **I** and **II** is

$$u_2^{(I)} = \beta \cdot n_1^{(I)} - p_2^{(I)} - \sqrt{(x_2 - a)^2 + y_2^2} \cdot t_2,$$

$$u_2^{(II)} = \beta \cdot n_1^{(II)} - p_2^{(II)} - \sqrt{(x_2 + a)^2 + y_2^2} \cdot t_2.$$

When i -th group agents choose one of the platforms, the boundary between market regions for this group is shaped by the set of its agents who have equal utilities from visiting the platforms **I** and **II**. Hence, the market boundary for the i -th group is found from the equation $u_i^{(I)} = u_i^{(II)}$, $i = 1, 2$, from which we get that

$$\frac{x_i^2}{s_i^2} - \frac{y_i^2}{a^2 - s_i^2} = 1, \tag{8}$$

where

$$s_1 = \frac{\pi\alpha - 2\alpha n_2^{(I)} + p_1^{(I)} - p_1^{(II)}}{2t_1}, \tag{9}$$

$$s_2 = \frac{\pi\beta - 2\beta n_1^{(I)} + p_2^{(I)} - p_2^{(II)}}{2t_2}. \tag{10}$$

We further assume for definiteness that the market boundary coincides with the right-hand branch of the hyperbola (8) (see Fig. 2). Considering the market

boundaries for both groups, the number of i -th group agents visiting the platforms **I** (the area of the shaded region of Fig. 2) and **II** is calculated from the formulas:

$$n_i^{(I)} = \frac{\pi}{2} - 2 \left[s_i \int_0^{d_i} \sqrt{1 + \frac{y_i^2}{a^2 - s_i^2}} dy_i + \int_{d_i}^1 \sqrt{1 - y_i^2} dy_i \right], \quad (11)$$

$$n_i^{(II)} = \pi - n_i^{(I)}, \quad (12)$$

where $d_i = d_i(s_i) = \sqrt{1 - \frac{s_i^2(1 + a^2 - s_i^2)}{a^2}}$, and s_i are derived from (9) and (10).

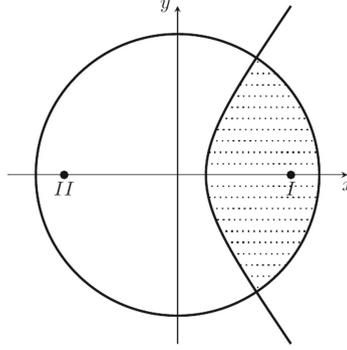


Fig. 2. The distribution of 2^{nd} group users between the two platforms.

While symmetric location of platforms in the market implies they are identical, the platforms **I** and **II** presumably also have identical costs $g_1 = g_2 = g$ of servicing agents of both groups. Let us consider a case where customers and sellers have different parameters of influence on each other. In this case, it suffices to find the optimal solution for one (e.g., the first) platform. The price equilibrium can be obtained from the first order optimality condition $\frac{\partial H^{(I)}}{\partial p_i^{(I)}} = 0$, $i = 1, 2$,

which gives us

$$\begin{cases} \frac{\partial H^{(I)}}{\partial p_1^{(I)}} = \frac{\partial n_1^{(I)}}{\partial p_1^{(I)}} (p_1^{(I)} - g) + n_1^{(I)} + \frac{\partial n_2^{(I)}}{\partial p_1^{(I)}} (p_2^{(I)} - g) = 0, \\ \frac{\partial H^{(I)}}{\partial p_2^{(I)}} = \frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} (p_1^{(I)} - g) + \frac{\partial n_2^{(I)}}{\partial p_2^{(I)}} (p_2^{(I)} - g) + n_2^{(I)} = 0. \end{cases} \quad (13)$$

The derivatives $\frac{\partial n_1^{(I)}}{\partial p_i^{(I)}}$ and $\frac{\partial n_2^{(I)}}{\partial p_i^{(I)}}$ ($i = 1, 2$) are taken from the Eqs. (9)–(12) and are

$$\frac{\partial n_1^{(I)}}{\partial p_1^{(I)}} = \frac{1}{2t_1} \frac{\partial n_1^{(I)}}{\partial s_1} \left(1 - \frac{\alpha\beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2} \right)^{-1}, \quad (14)$$

$$\frac{\partial n_2^{(I)}}{\partial p_1^{(I)}} = -\frac{\beta}{2t_1t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2} \left(1 - \frac{\alpha\beta}{t_1t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2}\right)^{-1}, \quad (15)$$

$$\frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} = -\frac{\alpha}{2t_1t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2} \left(1 - \frac{\alpha\beta}{t_1t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2}\right)^{-1}, \quad (16)$$

$$\frac{\partial n_2^{(I)}}{\partial p_2^{(I)}} = \frac{1}{2t_2} \frac{\partial n_2^{(I)}}{\partial s_2} \left(1 - \frac{\alpha\beta}{t_1t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \frac{\partial n_2^{(I)}}{\partial s_2}\right)^{-1}. \quad (17)$$

The platforms being symmetrically located relative to the center of the circle S , in equilibrium, the platforms are to set identical prices for each group: $p_1^{(I)} = p_1^{(II)} = p_1$ and $p_2^{(I)} = p_2^{(II)} = p_2$. Furthermore, both groups must have the same number of agents, i.e., $n_1^{(I)} = n_1^{(II)} = \frac{\pi}{2}$ and $n_2^{(I)} = n_2^{(II)} = \frac{\pi}{2}$.

Hence,

$$\frac{\partial n_1^{(I)}}{\partial s_1} = \frac{\partial n_2^{(I)}}{\partial s_2} = -2 \int_0^1 \sqrt{1 + \frac{y^2}{a^2}} dy = -2I,$$

where $I = \int_0^1 \sqrt{1 + \frac{y^2}{a^2}} dy$. Then, (14)–(17) take the following form

$$\begin{aligned} \frac{\partial n_1^{(I)}}{\partial p_1^{(I)}} &= -\frac{I}{t_1} \left(1 - \frac{4\alpha\beta I^2}{t_1t_2}\right)^{-1}, & \frac{\partial n_2^{(I)}}{\partial p_1^{(I)}} &= -\frac{2\beta I^2}{t_1t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1t_2}\right)^{-1}, \\ \frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} &= -\frac{2\alpha I^2}{t_1t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1t_2}\right)^{-1}, & \frac{\partial n_2^{(I)}}{\partial p_2^{(I)}} &= -\frac{I}{t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1t_2}\right)^{-1}. \end{aligned}$$

Substituting these derivatives into the system of Eq. (13), we find the equilibrium prices, which are

$$\begin{cases} p_1 = g + \frac{\pi t_1}{2I} - \pi\beta, \\ p_2 = g + \frac{\pi t_2}{2I} - \pi\alpha. \end{cases} \quad (18)$$

Test the sufficient conditions for the maximum of the function $H^{(I)}(p_1^{(I)}, p_2^{(II)})$. To this end, find the following expressions

$$\begin{aligned} A &= \frac{\partial^2 H^{(I)}}{\partial^2 p_1^{(I)}} = \frac{\partial^2 n_1^{(I)}}{\partial^2 p_1^{(I)}}(p_1 - g) + \frac{\partial^2 n_2^{(I)}}{\partial^2 p_1^{(I)}}(p_2 - g) + 2 \frac{\partial n_1^{(I)}}{\partial p_1^{(I)}}, \\ B &= \frac{\partial^2 H^{(I)}}{\partial p_1^{(I)} \partial p_2^{(I)}} = \frac{\partial^2 n_1^{(I)}}{\partial p_1^{(I)} \partial p_2^{(I)}}(p_1 - g) + \frac{\partial^2 n_2^{(I)}}{\partial p_1^{(I)} \partial p_2^{(I)}}(p_2 - g) + \frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} + \frac{\partial n_2^{(I)}}{\partial p_1^{(I)}}, \\ C &= \frac{\partial^2 H^{(I)}}{\partial^2 p_2^{(I)}} = \frac{\partial^2 n_1^{(I)}}{\partial^2 p_2^{(I)}}(p_1 - g) + \frac{\partial^2 n_2^{(I)}}{\partial^2 p_2^{(I)}}(p_2 - g) + 2 \frac{\partial n_2^{(I)}}{\partial p_2^{(I)}}. \end{aligned}$$

Considering the symmetry of the problem we have

$$\begin{aligned}
 A &= 2 \frac{\partial n_1^{(I)}}{\partial p_1^{(I)}} = -\frac{2I}{t_1} \left(1 - \frac{4\alpha\beta I^2}{t_1 t_2}\right)^{-1}, \\
 B &= \frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} + \frac{\partial n_2^{(I)}}{\partial p_1^{(I)}} = -\frac{2I^2(\alpha + \beta)}{t_1 t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1 t_2}\right)^{-1}, \\
 C &= 2 \frac{\partial n_2^{(I)}}{\partial p_2^{(I)}} = -\frac{2I}{t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1 t_2}\right)^{-1}.
 \end{aligned}$$

Since the expression

$$AC - B^2 = \frac{4I^2}{t_1 t_2} \left(1 - \frac{4\alpha\beta I^2}{t_1 t_2}\right)^{-2} \left(1 - \frac{I^2(\alpha + \beta)^2}{t_1 t_2}\right) > 0$$

when $\frac{\alpha\beta}{t_1 t_2} < \frac{1}{4I^2}$ and since $A < 0$ the function $H^{(I)}(p_1^{(I)}, p_2^{(II)})$ has a maximum at (p_1, p_2) , where p_1 and p_2 are found from (18).

Thus, for identical platforms and for single-homing agents of both groups which have different parameters of influence on each other and different transport costs of visiting the platforms the following theorems are true.

Theorem 1. *In the pricing problem for a two-sided market with platforms located symmetrically on the plane, competitive service will take place given that $\frac{\alpha\beta}{t_1 t_2} < \frac{1}{4I^2}$, where $I = \int_0^1 \sqrt{1 + \frac{y^2}{a^2}} dy$ and the equilibrium prices of visiting platforms for heterogeneous single-homing agents are (18).*

In a particular case, where agents of both groups are identical, i.e., $\alpha = \beta$ and $t_1 = t_2 = t$, the price of visiting the two platforms coincides for both groups, so the payoffs of the platforms **I** and **II** are equal. Competitive servicing in the case of identical agents is possible if $\frac{\alpha}{t} < \frac{1}{2I}$. Note that the value of I depends on the location of the platforms and, as demonstrated in the Table 1, the closer the platforms are to the center of the circle S , the closer the ratio $\frac{\alpha}{t}$ is to zero.

Tables 2 and 3 show the payoffs of the platforms and the prices of visiting the platforms depending on their location on the plane for different α , given that the transport costs are $t = 1$ and the costs of servicing are zero ($g = 0$). Numerical modeling results show that a decrease in the parameter α causes the platforms' payoff and the price of visiting to increase for both groups. Where the value of α is the same, the payoffs of both platforms decline with decreasing distance to the center of the circle S , and there are some positions that are disadvantageous for both platforms for their payoffs will be negative.

Table 1. The value of $\frac{1}{2I}$ depending on the location of the platforms.

a	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
$\frac{1}{2I}$	0.4356	0.4243	0.4100	0.3919	0.3685	0.3381	0.2984	0.2466	0.1798	0.0966

Table 2. Platforms' payoff values for different α for $t = 1, g = 0$.

a	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
$\alpha = 0.4$	0.3515	0.2398	0.0990	<0	0	<0	<0	<0	<0	<0
$\alpha = 0.3$	1.3385	1.2267	1.0859	0.9066	0.6757	0.3758	<0	<0	<0	<0
$\alpha = 0.2$	2.3255	2.2137	2.0729	1.8936	1.6627	1.3628	0.9710	0.4596	<0	<0
$\alpha = 0.1$	3.3124	3.2006	3.0599	2.8806	2.6497	2.3498	1.9579	1.4465	0.7877	<0
$\alpha = 0.01$	4.2007	4.0889	3.9481	3.7688	3.5380	3.2380	2.8462	2.3348	1.6759	0.8549
$\alpha = 0$	4.2994	4.1876	4.0468	3.8675	3.6367	3.3367	2.9449	2.4335	1.7746	0.9536

4 Model with Single-Homing and Multi-homing Agents

Suppose agents from one of the groups can join two platforms simultaneously, i.e., the agents are multi-homing. The assumption in our model is that sellers are multi-homing, and customers are single-homing. Importantly, this assumption is aligned with reality since in situations where sellers can be present in two platforms simultaneously the utility of customers, which depends on the number of sellers on the chosen platform, will not increase significantly if they visit the other platform at the same time. We assume that the platforms apply the same pricing to both single-homing and multi-homing agents. This means that the platforms are either unaware of the type of agents of both groups or do not care whether an agent is multi-homing or single-homing.

It is obvious in these settings that the utility of group **1** agents (customers) from visiting the platforms **I** and **II** located symmetrically relative to the center of the circle S is derived from (1) and (2). The number of agents of this group on both platforms is

Table 3. Prices of services for different α for $t = 1, g = 0$.

a	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
$\alpha = 0.4$	0.1119	0.0763	0.0315	<0	<0	<0	<0	<0	<0	<0
$\alpha = 0.3$	0.4261	0.3905	0.3457	0.2886	0.2151	0.1196	<0	<0	<0	<0
$\alpha = 0.2$	0.7402	0.7046	0.6598	0.6028	0.5293	0.4338	0.3091	0.1463	<0	<0
$\alpha = 0.1$	1.0544	1.0188	0.9740	0.9169	0.8434	0.7479	0.6232	0.4604	0.2507	<0
$\alpha = 0.01$	1.3371	1.3015	1.2567	1.1997	1.1262	1.0307	0.9060	0.7432	0.5335	0.2721
$\alpha = 0$	1.3685	1.3330	1.2881	1.2311	1.1576	1.0621	0.9374	0.7746	0.5649	0.3035

$$n_1^{(I)} = \frac{\pi}{2} - 2 \left[s_1 \int_0^{d_1} \sqrt{1 + \frac{y_1^2}{a^2 - s_1^2}} dy_1 + \int_{d_1}^1 \sqrt{1 - y_1^2} dy_1 \right],$$

$$n_1^{(II)} = \pi - n_1^{(I)},$$

where s_1 is calculated from (9) and $d_1 = \sqrt{1 - \frac{s_1^2(1+a^2-s_1^2)}{a^2}}$.

In platform selection by group 2 agents (sellers) it may happen so that some agents benefit more from single-homing while other agents gain more from joining both platforms simultaneously. Hence, the utilities from visiting both platforms are given by (3) and (4) for single-homing agents from group 2, and by (5) for its multi-homing agents. Where the platforms apply uniform pricing for group 2, market-boundary positions will be occupied by the agents whose utility from joining one platform or joining both platforms is zero. Thus, the set of multi-homing agents from group 2 is determined from the system of inequalities

$$\begin{cases} u_2^{(I)} \leq u_2^{(I+II)}, \\ u_2^{(II)} \leq u_2^{(I+II)}, \end{cases}$$

solving which we get the equations for the 2 group’s market boundaries between the platforms I and II (see Fig. 3):

$$\begin{aligned} (x_2 - a)^2 + y_2^2 &= (s_{21})^2, \\ (x_2 + a)^2 + y_2^2 &= (s_{22})^2, \end{aligned}$$

where $s_{21} = \frac{\beta \cdot n_1^{(I)} - p_2^{(I)}}{t_2}$ and $s_{22} = \frac{\beta \cdot n_1^{(II)} - p_2^{(II)}}{t_2}$; we assume for definiteness that $s_{21} \leq s_{22}$. Note that if $s_{21} + s_{22} < 2a$ that the set of multi-homing agents is empty.

Where the price of visiting the platforms prevents group 2 agents from joining both platforms simultaneously, the market is divided by the right-hand branch of the hyperbola (8) (see Fig. 3), the equation for which takes the form:

$$x = s_2 \sqrt{1 + \frac{y^2}{a^2 - (s_2)^2}},$$

where $s_2 = \frac{\pi\beta - 2\beta n_1^{(I)} + p_2^{(I)} - p_2^{(II)}}{2t_2}$. The relationship between the parameters s_2 , s_{21} and s_{22} is

$$s_2 = \frac{s_{22} - s_{21}}{2} \geq 0,$$

which implies that the market boundary in the case where all agents in the group 2 are single-homing intersects the market boundaries for the case where agents can be multi-homing in points with the coordinates

$$x = \frac{s_2(s_2 + s_{21})}{a},$$

$$k = y = \pm \sqrt{(s_{21})^2 - \left(\frac{s_2(s_2 + s_{21})}{a} - a \right)^2}.$$

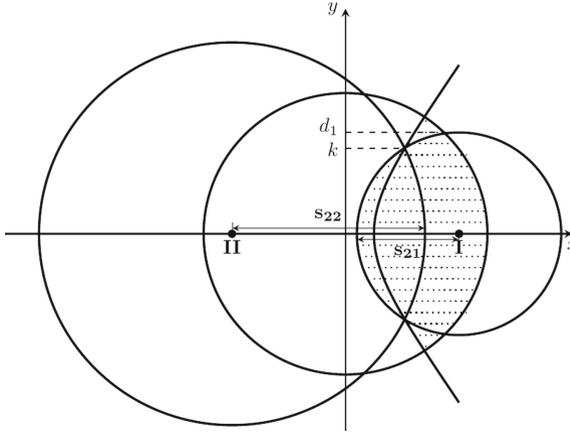


Fig. 3. The distribution of 2^{nd} group users between the two platforms.

Consider the case, where $|k| \leq 1$. In this case, $s_{21} \leq \sqrt{a^2 + 1}$ and the market division for the group **2** if there are multi-homing agents is mapped in Fig. 3, where multi-homing agents are situated in the shaded region. Considering the market boundaries for the group **2**, the numbers of agents visiting the platforms **I** (the area of the shaded region of Fig. 3) and **II** are, respectively,

$$n_2^{(I)} = 2 \left[\int_0^{d_2} \sqrt{1 - y^2} dy + \int_0^k \left(\sqrt{(s_{21})^2 - y^2} - a \right) dy - \int_k^{d_2} s_2 \sqrt{1 + \frac{y^2}{a^2 - (s_2)^2}} dy \right], \quad (19)$$

$$n_2^{(II)} = \pi - 2 \left[\int_0^{d_2} \sqrt{1 - y^2} dy + \int_k^{d_2} s_2 \sqrt{1 + \frac{y^2}{a^2 - (s_2)^2}} dy + \int_0^k \left(\sqrt{(s_{22})^2 - y^2} - a \right) dy \right], \quad (20)$$

where $k = \sqrt{(s_{21})^2 - \left(\frac{s_2(s_2 + s_{21})}{a} - a \right)^2}$ and $d_2 = \sqrt{1 - \frac{s_2^2(1 + a^2 - s_2^2)}{a^2}}$.

The locations of the platforms **I** and **II** being symmetrical, equilibrium in the pricing game with multi-homing agents on one side of the market can be found by simply studying the first-order conditions for the payoff function for one platform. First-order conditions for the payoff function for the platform **I** have the form (13), where the derivatives are obtained from the formulas

$$\frac{\partial n_1^{(I)}}{\partial p_1^{(I)}} = \frac{1}{2t_1} \frac{\partial n_1^{(I)}}{\partial s_1} \left(1 + \frac{\alpha\beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \right)^{-1}, \quad (21)$$

$$\frac{\partial n_2^{(I)}}{\partial p_1^{(I)}} = \frac{\beta}{2t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \left(1 + \frac{\alpha\beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \right)^{-1}, \quad (22)$$

$$\frac{\partial n_1^{(I)}}{\partial p_2^{(I)}} = \frac{\alpha}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \cdot \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{1}{2} \frac{\partial n_2^{(I)}}{\partial s_2} \right) \cdot \left(1 + \frac{\alpha \beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \right)^{-1}, \quad (23)$$

$$\frac{\partial n_2^{(I)}}{\partial p_2^{(I)}} = -\frac{1}{t_2} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{1}{2} \frac{\partial n_2^{(I)}}{\partial s_2} \right) \left(1 + \frac{\alpha \beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \right)^{-1}. \quad (24)$$

The symmetry of the problem suggests that, in equilibrium, the prices for the same group of agents should be the same on each of the platforms, i.e., $p_1^{(I)} = p_1^{(II)} = p_1$ and $p_2^{(I)} = p_2^{(II)} = p_2$, and the size of each group should be the same on the two platforms, i.e., $n_1^{(I)} = n_1^{(II)} = \frac{\pi}{2}$ and $n_2^{(I)} = n_2^{(II)} = n_2 > \frac{\pi}{2}$. Hence, we have

$$s_2 = 0, \quad s_{21} = s_{22} = \frac{\beta \pi - p_2}{2 t_2}.$$

The price p_1 and the value of s_{21} can be found by solving the system of Eq. (13), which, after the substitution of the derivatives (21)–(24) into it, will take the form

$$\begin{cases} -\frac{I}{t_1} (p_1 - g) - \frac{\beta I(D-C)}{t_1 t_2} (p_2 - g) + \frac{1}{A} n_1 = 0, \\ -\frac{2\alpha I}{t_1 t_2} \left(D - \frac{1}{2} C \right) (p_1 - g) - \frac{1}{t_2} \left(D - \frac{1}{2} C \right) (p_2 - g) + \frac{1}{A} n_2 = 0, \end{cases}$$

where $\frac{\partial n_1^{(I)}}{\partial s_1} = -2I$,

$$A = \left(1 + \frac{\alpha \beta}{t_1 t_2} \frac{\partial n_1^{(I)}}{\partial s_1} \left(\frac{\partial n_2^{(I)}}{\partial s_{21}} - \frac{\partial n_2^{(I)}}{\partial s_2} \right) \right)^{-1} = \left(1 - \frac{2\alpha \beta}{t_1 t_2} I(D - C) \right)^{-1},$$

$$D = \frac{\partial n_2^{(I)}}{\partial s_{21}} = 2 \left[\int_0^k \frac{s_{21}}{\sqrt{(s_{21})^2 - y^2}} dy + k'_{s_{21}} \left(\sqrt{(s_{21})^2 - k^2} - a \right) \right],$$

$$C = \frac{\partial n_2^{(I)}}{\partial s_2} = 2 \left[k'_{s_2} \left(\sqrt{(s_{21})^2 - k^2} - a \right) - \int_k^{d_2} \sqrt{1 + \frac{y^2}{a^2}} dy \right],$$

$$k = \sqrt{(s_{21})^2 - a^2}, \quad k'_{s_{21}} = k'_{s_2} = \frac{s_{21}}{\sqrt{(s_{21})^2 - a^2}}, \quad d_2 = 1.$$

The result will be the optimal prices for each group of users:

$$p_1 = g + \frac{t_1}{I} n_1 - \frac{\beta(D - C)}{D - \frac{1}{2}C} n_2, \quad (25)$$

$$p_2 = g + \frac{t_2}{D - \frac{1}{2}C} n_2 - 2\alpha n_1. \quad (26)$$

Observe that the Eq. (26) for calculating the price p_2 depends on the parameter s_{21} , which is equal to $\frac{\beta\pi - 2p_2}{2t_2}$. It follows that

$$p_2 = \frac{\beta\pi}{2} - s_{21}t_2. \tag{27}$$

Hence, the parameter s_{21} can be found from the equation

$$\frac{\beta\pi}{2} - t_2s_{21} = g + \frac{t}{D - \frac{1}{2}C}n_2 - 2\alpha n_1.$$

In the case $|k| > 1$ (or $s_{21} > \sqrt{1 + a^2}$), the market division for the group **2** if there are multi-homing agents is mapped in Fig. 4, where multi-homing agents are situated in the shaded region.

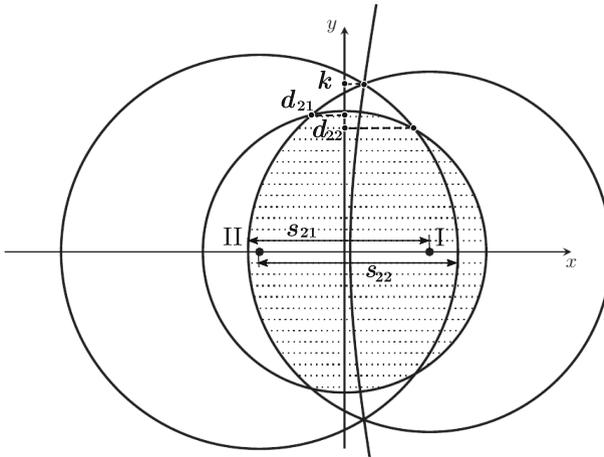


Fig. 4. The distribution of 2^{nd} group users between the two platforms.

The numbers of agents visiting the platforms **I** and **II** are, respectively,

$$n_2^{(I)} = \frac{\pi}{2} + 2 \left[\int_0^{d_{21}} \left(\sqrt{(s_{21})^2 - y^2} - a \right) dy + \int_{d_{21}}^1 \sqrt{1 - y^2} dy \right], \tag{28}$$

$$n_2^{(II)} = \frac{\pi}{2} + 2 \left[\int_0^{d_{22}} \left(\sqrt{(s_{22})^2 - y^2} - a \right) dy + \int_{d_{22}}^1 \sqrt{1 - y^2} dy \right], \tag{29}$$

where $d_{21} = \sqrt{1 - \left(\frac{1 - (s_{21})^2 + a^2}{2a} \right)^2}$, $d_{22} = \sqrt{1 - \left(\frac{(s_{22})^2 - a^2 - 1}{2a} \right)^2}$.

In a similar way, we get that the optimal prices for each group of users are

$$p_1 = g + \frac{\pi t_1}{2I} - \beta n_2, \tag{30}$$

$$p_2 = g + \frac{n_2 t_2}{D_1} - \alpha \pi, \tag{31}$$

where the parameter s_{21} can be found from the equation

$$\frac{\beta \pi}{2} - t_2 s_{21} = g + \frac{n_2 t_2}{D_1} - \alpha \pi$$

and $n_2 = n_2^{(I)} = n_2^{(II)}$, $D_1 = \frac{\partial n_2}{\partial s_{21}}$.

Thus, the following theorem is true for a symmetric arrangement of platforms relative to the center of the circle S with multi-homing agents on one side of the market.

Theorem 2. *In the pricing problem for a two-sided market with platforms symmetrically located on the plane, the equilibrium price of visiting the platforms for the group with single-homing agents with different parameters is (25) for $a \leq s_{21} \leq \sqrt{a^2 + 1}$ (or (30) for $s_{21} > \sqrt{1 + a^2}$), and that for the group with multi-homing agents is (26) for $a \leq s_{21} \leq \sqrt{a^2 + 1}$ (or (31) for $s_{21} > \sqrt{1 + a^2}$).*

Numerical simulation shows that platforms set equal prices for agents within the same group, but the fee for the group with multi-homing agents is lower than for groups with single-homing agents. Given certain parameters of the degree of influence and the transport costs, platforms may choose to set a negative price of visiting for groups with multi-homing agents, which can be interpreted as discount offers. Platforms benefit from offering discounts to the group with multi-homing agents only if the profit gained from the other side of the market offsets the costs of these offers and the platform’s total profit is therefore non-negative (see Table 4).

Table 4. Platforms’ payoff values for different s_{21} for $a = 1$, $t_1 = 1$, $t_2 = 0.5$ and $g = 0$.

s_{21}	α	n_1	n_2	p_1	p_2	$H^{(I)} = H^{(II)}$
1.1	0.2237	1.5708	1.6325	0.8514	-0.1986	1.0131
1.2	0.2281	1.5708	1.7509	0.8743	-0.2418	0.9500
1.3	0.2378	1.5708	1.9116	0.8627	-0.2765	0.8265
$\sqrt{2}$	0.2523	1.5708	2.1416	0.8281	-0.3107	0.635
1.5	0.2732	1.5708	2.3301	0.7320	-0.3209	0.4021
1.6	0.3007	1.5708	2.5420	0.6040	-0.3276	0.1161
1.7	0.3345	1.5708	2.7398	<0	<0	<0

E.g., when $\alpha = 0.25$, platforms set negative prices for the group with multi-homing agents and gain positive profit when occupying the positions $a > 0.4$. Starting from the position $a = 0.4$, the profit of both platforms becomes negative (see Table 4).

Similarly to the case of single-homing agents on both sides of the market, if α is the same, the platforms' profit declines in positions closer to the center of the circle, the reasons being either lowering of the price of visiting for both groups of agents or lowering of the price for the group with single-homing agents and an increased discount for the group with multi-homing agents. It is noteworthy that the platforms' profit decreases where there are multi-homing agents on one side of the market.

Table 5. The price of visiting the platforms and their payoffs depending on the location of the platforms for $t_1 = 1$, $t_2 = 0.5$, $\alpha = 0.25$ and $g = 0$.

a	s_{21}	n_1	n_2	p_1	p_2	$H^{(I)} = H^{(II)}$
1	1.3973	1.5708	2.1045	0.8343	-0.3059	0.6667
0.9	1.3637	1.5708	2.2288	0.7758	-0.2888	0.5750
0.8	1.3216	1.5708	2.3338	0.7047	-0.2694	0.4782
0.7	1.2821	1.5708	2.4433	0.6202	-0.2483	0.3675
0.6	1.2415	1.5708	2.5491	0.5203	-0.2281	0.2359
0.5	1.2011	1.5708	2.6533	0.3988	-0.2078	0.0750
0.4	1.1607	1.5708	2.7558	0.2484	-0.1876	-0.1268

5 Conclusion

The paper has investigated the structure of prices in equilibrium in a two-sided market of platforms for different agent types in the presence of external network effects. Assuming that the market has a fixed size and lies on the plane of the circle, we studied a duopoly model with the platforms located symmetrically in relation to the center of the circle. Two cases were analyzed: 1) members of both groups are single-homing agents, i.e., join only one of the platforms; 2) one group consists of single-homing agents and the other group - of multi-homing agents, i.e., ones capable of joining both platforms. The platforms were assumed to apply the same pricing to different types of agents, i.e., the platforms were unaware whether an agent was single- or multi-homing. We solved the optimal pricing problem for the platforms and produced the analytical expressions for equilibrium prices, which depend on the structure of costs and the external network effects. The output of numerical simulation of the values of equilibrium prices and the platforms' profit functions for agents with different parameters is presented.

Analysis of the problem of optimal pricing in a two-sided market of platforms showed that when the platforms applied uniform pricing for different groups of

agents, their profit decreased if multi-homing agents appeared on one side of the market. Hence, the platforms need to apply other pricing methods to raise their profit, e.g., differentiated pricing for single-homing and multi-homing agents. We found also that when the platforms were situated close to the center of the circle, their profit declined, and could become negative at some values of network effect parameters.

In the future, we plan to study the optimal pricing problem for non-symmetric location of platforms on the plane with different metrics. Having found that the presence of multi-homing agents is disadvantageous for the platforms, we shall also study the pricing problem for the case where different types of agents are charged different prices for using the platforms' services.

References

1. Armstrong, M.: Competition in two-sided markets. *RAND J. Econ.* **37**(2), 668–691 (2006)
2. Armstrong, M., Wright, J.: Two-sided markets, competitive Bottlenecks and exclusive contracts. *Econ. Theory* **32**, 353–380 (2007)
3. Caillaud, B., Jullien, B.: Chicken & egg: competition among intermediation service providers. *RAND J. Econ.* **32**(2), 309–328 (2003)
4. GlenWeyl, E.: A price theory of multi-sided platforms. *Am. Econ. Rev.* **100**(4), 1642–1672 (2010)
5. Kodera, Ô.: Discriminatory pricing and spatial competition in two-sided media markets. *B.E. J. Econ. Anal. Policy* **15**(2), 891–926 (2015)
6. Mazalov, V., Konovalchikova, E.: Hotelling's Duopoly in a two-sided platform market on the plane. *Mathematics* **8**, 865 (2020)
7. Mazalov, V.V., Sakaguchi, M.: Location game on the plane. *Int. Game Theory Rev.* **5**(1), 13–25 (2003)
8. Liu, Q., Serfes, K.: Price discrimination in two-sided markets. *RAND J. Econ. Manag. Strat.* **22**(4), 768–786 (2013)
9. Rochet, C., Tirole, J.: Platform competition in two-sided markets. *J. Eur. Econ. Assoc.* **4**(6), 990–1029 (2003)
10. Zhang, K., Weiqi, L.: Price discrimination in two-sided markets. *South Afr. J. Econ. Manag. Sci.* **19**(1), 1–17 (2016)



On the Existence of a Fuzzy Core in an Exchange Economy

Valeriy Marakulin^(✉) 

Sobolev Institute of Mathematics, Russian Academy of Sciences,
4 Acad. Koptuyug avenue, 630090 Novosibirsk, Russia
marakulv@gmail.com

<http://www.math.nsc.ru/mathecon/marakENG.html>

Abstract. The fuzzy core is widely used in theoretical economics for modeling perfect competition. However, in modern literature, the proof of its existence is presented indirectly and applies a cumbersome construction. It usually is based on the idea of replicated economies, via standard core existence, followed by passing to the limits, allowing the number of replicas tends to infinity. The result is also proved under additional restrictive assumptions. We present a direct proof based on two well-known theorems: Michael's theorem on the existence of a continuous selector for a point-to-set mapping and Brouwer's fixed point theorem. This new direct proof is efficient and shortest among others. Moreover, now the existence of a fuzzy core is stated under other weakest assumptions: agent preferences can be incomplete, non-transitive, satiated, and so on.

Keywords: Fuzzy core · Edgeworth equilibria · Perfect competition · Existence theorems

1 Introduction

The concept of the core of the economy is one of the key equilibrium notions of modern economic theory, closely related to the concept of competitive (Walrasian) equilibrium. The core embodies the idea of a cooperatively stable resources allocation: such that no group of individuals has clear incentives to change it (does not dominate). It has long been known that the Walrasian allocation is always an element of the core and, thus, besides the market balance of interests, it implements a purely cooperative principle of stability. Since the time of Edgeworth, there appeared a hypothesis, first formally proved many years later by Debreu and Scarf, that in conditions of perfect competition, the core and the equilibria coincide. In the work of Debreu and Scarf, a model of a replicated economy was proposed, in which economic agents function as copies of themselves, and their number tends to infinity. In their famous theorem, Debreu and Scarf

The study was supported by the Program of Basic Scientific Research of the Siberian Branch of the Russian Academy of Sciences (Grant no. FWNF-2022-0019).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 210–217, 2022.
https://doi.org/10.1007/978-3-031-09607-5_15

showed that the core of a replicated economy shrinks towards equilibria. So, the first model of perfect competition appeared in the theory. Subsequently, following [1], allocations from the limit core of replicated economies were called the Edgeworth equilibria. If one allows the participation of the agents in a coalition with a rate belonging to the rational interval $[0, 1]$, an Edgeworth equilibrium can also be defined as a feasible allocation that cannot be blocked by a coalition with rational rates of participation. A fuzzy coalition is a coalition whose rates of participation can take any value in the real interval $[0, 1]$. The fuzzy core (first introduced in [4]) is the set of all attainable allocations which cannot be blocked by a fuzzy coalition.

Nowadays due to Debreu–Scarff theorem on limit coincidence of core and equilibria in the replicated economy¹ the fuzzy core is widely used in theoretical economics not only to model the conditions of perfect competition but also to state the existence of competitive equilibrium, e.g. see [2, 9]. This notion plays a key role in modern economic theory and the conditions under which it exists have a high theoretical meaning. The problem of the non-emptiness of a fuzzy core was a subject of a variety of studies, but by now the most advanced results still are presented in [5, 6] (proved in the context of an economy with an infinite-dimensional commodity space via passing to limits). The idea of the proof was based on the theorem on the non-emptiness of the ordinary core and the consideration of the asymptotic limit of the core of replica economics that coincides with the fuzzy core. By enlarging the feasible payoff sets for coalitions, [10] provides an alternative proof of the non-emptiness of the fuzzy core. Notwithstanding the novelty of this approach, the result still relies on a conventional limit argument. Only in [3] there is first appeared the proof based on fixed point arguments (Fans coincidence theorem is applied), but the result was stated under rather strong model assumptions: preferences are presented via utility functions, and so on. In this paper I fill this gap: the problem of the non-emptiness of the fuzzy core in the exchange economy is stated under very weak conditions, even weaker of [6] (we do not need agents' upper preference sections to be open). The proof of the result is based on two well-known theorems, they are Michael's theorem on the existence of a continuous selector for a point-to-set mapping and Brouwer's fixed point theorem. So, though the non-emptiness of the fuzzy core is a well-known fact, however, we present a new direct efficient, and shortest proof among others. This result can be efficiently incorporated in the proving of Edgeworth's conjecture [2] or even to state the existence of Walrasian equilibrium in economies with infinite-dimensional commodity spaces [9].

¹ Using the density of rational numbers among real ones, one can easily prove, under weak assumptions (one needs the set of preferred consumption bundles to be open for each agent and every allocation), that the elements of the fuzzy core coincide with the Edgeworth equilibria. So the non-domination via fuzzy coalition with rational rates of participation is equivalent to non-domination for coalition with real rates of participation.

2 An Economic Model, Definitions, and Fuzzy Core

I consider a typical exchange economy in which L denotes the (finite-dimensional) *space of commodities*. Let $\mathcal{I} = \{1, \dots, n\}$ be a set of agents (traders or consumers). A consumer $i \in \mathcal{I}$ is characterized by a consumption set $X_i \subset L$, an initial endowment $\mathbf{e}_i \in L$, and a preference relation described by a point-to-set mapping $\mathcal{P}_i : X \rightrightarrows X_i$ where $X = \prod_{j \in \mathcal{I}} X_j$ and $\mathcal{P}_i(x)$ denotes the set of all consumption bundles strictly preferred by the i -th agent to the bundle x_i relative to allocation $x \in X$. It is also can be applied the notation $y_i \succ_i x_i$ which is equivalent to $y_i \in \mathcal{P}_i(x)$ (to simplify notations; preferences can indirectly depend on other agents consumption $x_j \in X_j, j \in \mathcal{I}, j \neq i$). So, the pure exchange model may be represented as a triplet

$$\mathcal{E} = \langle \mathcal{I}, L, (X_i, \mathcal{P}_i, \mathbf{e}_i)_{i \in \mathcal{I}} \rangle.$$

Let us denote by $\mathbf{e} = (\mathbf{e}_i)_{i \in \mathcal{I}}$ the vector of initial endowments of all traders of the economy. Denote $X = \prod_{i \in \mathcal{I}} X_i$ and let

$$\mathcal{A}(X) = \left\{ x \in X \mid \sum_{i \in \mathcal{I}} x_i = \sum_{i \in \mathcal{I}} \mathbf{e}_i \right\}$$

be the set of all *feasible allocations*. Now let us recall some definitions.

A pair (x, p) is said to be a *quasi-equilibrium* of \mathcal{E} if $x \in \mathcal{A}(X)$ and there exists a linear functional $p \neq 0$ onto L such that

$$\langle p, \mathcal{P}_i(x) \rangle \geq px_i = p\mathbf{e}_i, \quad \forall i \in \mathcal{I}.$$

A quasi-equilibrium such that $x'_i \in \mathcal{P}_i(x)$ actually implies $px'_i > px_i$ is a *Walrasian or competitive equilibrium*.

An allocation $x \in \mathcal{A}(X)$ is said to be dominated (blocked) by a nonempty coalition $S \subseteq \mathcal{I}$ if there exists $y^S \in \prod_{i \in S} X_i$ such that $\sum_{i \in S} y_i^S = \sum_{i \in S} \mathbf{e}_i$ and $y_i^S \in \mathcal{P}_i(x) \forall i \in S$.

The *core* of \mathcal{E} , denoted by $\mathcal{C}(\mathcal{E})$, is the set of all $x \in \mathcal{A}(X)$ that are blocked by no (nonempty) coalition.

One more important notion, fruitfully working in the theory of economic equilibrium, is the concept of the fuzzy core. Recall that any vector

$$t = (t_1, \dots, t_n) \neq 0, \quad 0 \leq t_i \leq 1, \quad \forall i \in \mathcal{I}$$

maybe identified with a fuzzy coalition, where the real number t_i is interpreted as the measure of agent i participation in the coalition. A coalition t is said to dominate (block) an allocation $x \in \mathcal{A}(X)$ if there exists $y^t \in \prod_{i \in \mathcal{I}} X_i$ such that

$$\sum_{i \in \mathcal{I}} t_i y_i^t = \sum_{i \in \mathcal{I}} t_i \mathbf{e}_i \iff \sum_{i \in \mathcal{I}} t_i (y_i^t - \mathbf{e}_i) = 0 \tag{1}$$

and

$$y_i^t \succ_i x_i, \quad \forall i \in \text{supp}(t) = \{i \in \mathcal{I} \mid t_i > 0\}. \tag{2}$$

The set of all feasible allocations which cannot be dominated by fuzzy coalitions is called the *fuzzy core* of the economy \mathcal{E} and is denoted by $\mathcal{C}^f(\mathcal{E})$.

Everywhere below, we assume that model \mathcal{E} satisfies the following assumption.

(A) For each $i \in \mathcal{I}$, $X_i \subset L$ is a convex closed subset, $\mathbf{e}_i \in X_i$, and for every $x = (x_j)_{j \in \mathcal{I}} \in \mathcal{A}(X)$ the set $\mathcal{P}_i(x) \subset X_i$ is convex and $x_i \notin \mathcal{P}_i(x)$.

Notice that due to (A) preferences may be satiated, i.e., $\mathcal{P}_i(x) = \emptyset$ is possible for some agent i and $x \in X$.

For the existence of objects we are interested in, we apply the following weakest requirement of preferences continuity.

(C) For each $i \in \mathcal{I}$ for every $x \in \mathcal{A}(X)$, $\forall y_i \in \mathcal{P}_i(x)$ the set

$$\mathcal{P}_i^{-1}(y_i) = \{z \in X \mid y_i \in \mathcal{P}_i(z)\}$$

is open in X .

2.1 Fuzzy Core Specification

We begin with a study of the specific properties of the fuzzy core allocations. The elements of fuzzy core are defined via conditions (1), (2) which for non-satiated preferences, i.e., when $\mathcal{P}_i(x) \neq \emptyset, \forall i \in \mathcal{I}$, may be equivalently rewritten in the form²

$$0 \notin \sum_{i \in \mathcal{I}} t_i(\mathcal{P}_i(x) - \mathbf{e}_i).$$

Thus, in this case, condition $x \in \mathcal{C}^f(\mathcal{E})$ is equivalent to³

$$0 \notin \text{co}[\bigcup_{\mathcal{I}} (\mathcal{P}_i(x) - \mathbf{e}_i)], \tag{3}$$

that, after applying the separation theorem, allows concluding that the elements of the fuzzy core are quasi-equilibria. Below we describe another useful in applications characterization (first proposed in [8]) of fuzzy core points presented in “geometrical” terms. Let us consider the sets

$$\Upsilon_i(x) = \text{co}(\mathcal{P}_i(x) \cup \{\mathbf{e}_i\}), \quad i \in \mathcal{I}.$$

Due to the convexity of $\mathcal{P}_i(x)$, for $\mathcal{P}_i(x) \neq \emptyset$, conclude

$$\text{co}(\mathcal{P}_i(x) \cup \{\mathbf{e}_i\}) = \bigcup_{0 \leq \lambda \leq 1} [\lambda \mathcal{P}_i(x) + (1 - \lambda)\mathbf{e}_i] = \bigcup_{0 \leq \lambda \leq 1} \lambda(\mathcal{P}_i(x) - \mathbf{e}_i) + \mathbf{e}_i, \quad i \in \mathcal{I}.$$

This implies that the condition $z + \mathbf{e} \in \prod_{\mathcal{I}} \Upsilon_i(x)$, where $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_n)$, is equivalent to the existence of $0 \leq \lambda_i \leq 1$ and $[y_i \in \mathcal{P}_i(x) \neq \emptyset$ and $y_i = \mathbf{e}_i$, if $\mathcal{P}_i(x) = \emptyset]$, $i \in \mathcal{I}$ such that

$$z = (\lambda_1(y_1 - \mathbf{e}_1), \dots, \lambda_n(y_n - \mathbf{e}_n)).$$

² Admitting some inaccuracy in formulas here and below, we identify a vector with a one-element set containing it.

³ Clearly, for a dominating fuzzy coalition t one may always think that $\sum_{i \in \mathcal{I}} t_i = 1$.

Hence, due to (1), (2)

$$\begin{aligned}
 x \in \mathcal{C}^f(\mathcal{E}) &\iff \nexists z \in L^{\mathcal{I}}, z \neq 0: z + \mathbf{e} \in \prod_{\mathcal{I}} \mathcal{Y}_i(x) \quad \& \quad \sum_{i \in \mathcal{I}} z_i = 0 \\
 &\iff \prod_{\mathcal{I}} \mathcal{Y}_i(x) \cap \mathcal{A}(L^{\mathcal{I}}) = \{\mathbf{e}\}, \tag{4}
 \end{aligned}$$

where $\mathcal{A}(L^{\mathcal{I}})$ is a subspace defined by the balance constraints of a pure exchange economy:

$$\mathcal{A}(L^{\mathcal{I}}) = \{(z_1, \dots, z_n) \in L^{\mathcal{I}} \mid \sum_{i \in \mathcal{I}} z_i = \sum_{i \in \mathcal{I}} \mathbf{e}_i\}.$$

Notice that characterization (4) is also valid for satiated preferences. In doing so, we have proven the following

Proposition 1. *An allocation $x \in \mathcal{A}(X)$ is the element of fuzzy core if and only if relation (4) is true.*

In the case of a 2-agent economy with autonomous preferences, condition (4) may be rewritten in the form

$$\mathcal{Y}_1(x_1) \cap (\bar{\mathbf{e}} - \mathcal{Y}_2(\bar{\mathbf{e}} - x_1)) = \{\mathbf{e}_1\}, \quad \bar{\mathbf{e}} = \mathbf{e}_1 + \mathbf{e}_2.$$

Hence,

$$\begin{aligned}
 (x_1, x_2) \notin \mathcal{C}^f(\mathcal{E}) &\iff \exists \text{ray starting at the point } \mathbf{e}_1, \text{ which intersects} \\
 &\quad \text{both sets, } \mathcal{P}_1(x_1) \text{ and } \bar{\mathbf{e}} - \mathcal{P}_2(\bar{\mathbf{e}} - x_1) = \tilde{\mathcal{P}}_2(x_2).
 \end{aligned}$$

Figure 1 presents a graphic illustration of conducted analysis in Edgeworth’s box for a 2-goods economy. In this case, an allocation x lying in the fuzzy core is equivalent to the convex hulls of $\mathcal{P}_1(x_1) \cup \{\mathbf{e}_1\}$ and of $[\bar{\mathbf{e}} - \mathcal{P}_2(\bar{\mathbf{e}} - x_1)] \cup \{\mathbf{e}_1\}$ having only one point, \mathbf{e}_1 , in common. In modern literature, allocations from the fuzzy core are interpreted as Edgeworth’s equilibria and served as a technical tool more than an economic concept. Moreover, the fact that every element of the fuzzy core is a quasi-equilibrium (this is why the fuzzy core is so popular in existence theory) can be also easily derived from formula (4).

3 The Result: Non-emptiness of Fuzzy Core

The existence of an ordinary and a fuzzy core in an economy can be established by applying Brouwer fixed point theorem and Michael theorems [7] on the existence of a continuous selector. Our main result is presented below.

Theorem 1. *Under imposed assumptions (A), (C) and if $\mathcal{A}(X)$ is bounded, fuzzy core is non-empty, i.e. $\mathcal{C}^f(\mathcal{E}) \neq \emptyset$.*

Remark 1. An analysis of the proof below shows that the assumption of (C) that $\mathcal{P}_i(x), x \in \mathcal{A}(X)$ are convex can be replaced by the standard and a formally weaker one $x_i \notin \text{co } \mathcal{P}_i(x) \forall x = (x_j)_{j \in \mathcal{I}} \in \mathcal{A}(X)$.

Now I am presenting proofs.

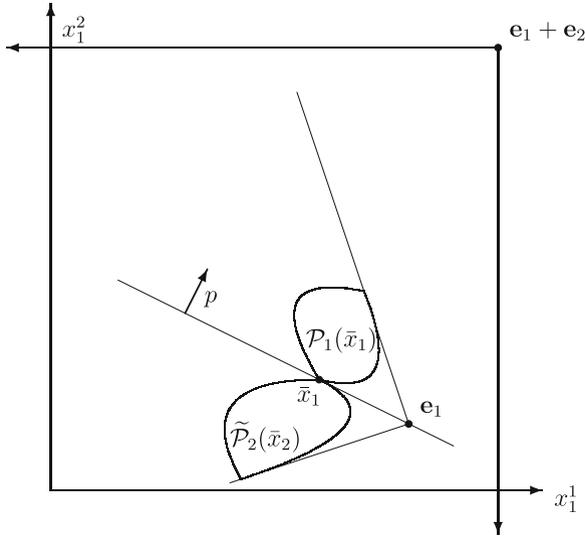


Fig. 1. Specification of (\bar{x}_1, \bar{x}_2) as a fuzzy core point.

3.1 Proof

For the analysis, we need the following auxiliary lemmas. Let Ω be a subset of $\mathcal{A}(X)$ consisting of the points $x \in \mathcal{A}(X)$ for which (4) is false. Now I study the properties of these allocations.

Let $x \in \Omega$. Consider the set $\varphi(x)$ of all contracts that fuzzily block this allocation:

$$\varphi(x) = \{(v_i, t_i)_{\mathcal{I}} \in (L \times [0, 1])^{\mathcal{I}} \mid \sum_{\mathcal{I}} v_i = 0, v \neq 0 : \forall i \in \mathcal{I} \mathcal{P}_i(x) \neq \emptyset$$

$$\Rightarrow \exists g_i(x) \in \mathcal{P}_i(x) : v_i = t_i(g_i(x) - e_i) \ \& \ \mathcal{P}_i(x) = \emptyset \Rightarrow v_i = 0, t_i = 0.\} \quad (5)$$

The following lemma presents crucial properties of the point-to-set mapping $\varphi(\cdot)$. First, I recall the definition of lower hemicontinuous⁴ point-to-set mapping.

Definition 1. *Let Y, Z be topological spaces. A point-to-set mapping $\psi : Y \rightrightarrows Z$ is called lower hemicontinuous (l.h.c.) iff*

$$\psi^{-1}(V) = \{y \in Y \mid \psi(y) \cap V \neq \emptyset\}$$

is open for every open $V \subset Z$. For a metric spaces Y, Z a l.h.c. mapping can be equivalently characterized as follows:

⁴ According to the modern views, the term semi-continuous mapping is specifically applied for a function—point-to-point map—and hemicontinuous for a correspondence.

For every $y \in Y$, $z \in \psi(y) \subset Z$ and every sequence $y_m \rightarrow y$ there is a subsequence y_{m_k} , $m, k \in \mathbb{N}$ and a sequence $z_k \in \psi(y_{m_k})$ such that $z_k \rightarrow z$ for $k \rightarrow \infty$.

Lemma 1. *Let $x \in \mathcal{A}(X)$. The set $\varphi(x)$ is convex, and $\varphi(x) \neq \emptyset$ if (4) is false. Moreover, the point-to-set mapping $\varphi : \Omega \rightrightarrows (L \times [0, 1])^{\mathcal{I}}$ is lower hemicontinuous.*

Proof. To show the convexity of $\varphi(x)$ one takes any $(w', t'), (w'', t'') \in \varphi(x)$ and $\alpha \in (0, 1)$. Now for $i \in \mathcal{I}$ such that $\mathcal{P}_i(x) \neq \emptyset$ & $(t'_i, t''_i) \neq 0$ one has: $\exists g'_i, g''_i \in \mathcal{P}_i(x)$ such that

$$\begin{aligned} \alpha w'_i + (1 - \alpha)w''_i &= \alpha t'_i(g'_i - \mathbf{e}_i) + (1 - \alpha)t''_i(g''_i - \mathbf{e}_i) \\ &= [\alpha t'_i + (1 - \alpha)t''_i] \left[\frac{\alpha t'_i}{\alpha t'_i + (1 - \alpha)t''_i} g'_i + \frac{(1 - \alpha)t''_i}{\alpha t'_i + (1 - \alpha)t''_i} g''_i - \mathbf{e}_i \right]. \end{aligned}$$

Thus, for $t_i = \alpha t'_i + (1 - \alpha)t''_i$ and

$$g_i = \frac{\alpha t'_i}{\alpha t'_i + (1 - \alpha)t''_i} g'_i + \frac{(1 - \alpha)t''_i}{\alpha t'_i + (1 - \alpha)t''_i} g''_i \in \mathcal{P}_i(x)$$

we have $w_i = t_i(g_i - \mathbf{e}_i) = \alpha w'_i + (1 - \alpha)w''_i$. If $t'_i = t''_i = 0$ one has $w'_i = w''_i = 0$ and we obtain the same result for any $g_i \in \mathcal{P}_i(x)$. Therefore, for $\alpha \in [0, 1]$ relative to $t = \alpha t' + (1 - \alpha)t''$ contract $w = \alpha w' + (1 - \alpha)w''$ is so that $(w, t) \in \varphi(x)$, as we wanted to prove.

Further, we show that the point-to-set mapping $\varphi(\cdot)$ defined in (5) is lower hemicontinuous. Indeed, let $(v, t) \in \varphi(x)$ and let $x^m \in \mathcal{A}(X)$, $x^m \rightarrow x$ for $m \rightarrow \infty$. Due to (C) for every $i \in \mathcal{I}$ preferences \mathcal{P}_i have open low sections in X and hence for sufficiently large m we have $x^m \in \mathcal{P}_i^{-1}(g_i(x)) \forall i \in \mathcal{I}$: $\mathcal{P}_i(x) \neq \emptyset$. If $\mathcal{P}_i(x) = \emptyset$ for some i then $v_i = 0$, $t_i = 0$ and $t_i(g_i(x_m) - \mathbf{e}_i) = 0$ for any $g_i(x_m) \in \mathcal{P}_i(x_m)$. So, via (5) one concludes $(v, t) \in \varphi(x^m)$ for all $m \in \mathbb{N}$ big enough. This proves, by definition, $\varphi(\cdot)$ is lower hemicontinuous in $x \in \Omega \subset \mathcal{A}(X)$. \square

In the proof of the lemma below I apply the following Michael theorem (see [7] p. 368, Th 3.1''', (c)) on the existence of a continuous selector in its simplified finite-dimensional presentation.⁵

Theorem 2 (Michael, 1956). *Let Y and Z be subsets of finite-dimensional linear spaces. Then every l.h.c. point-to-set mapping $\psi : Y \rightrightarrows Z$ having nonempty convex images $\psi(y) \subset Z \forall y \in Y$ has a continuous selector.*

Lemma 2. *There is a continuous function $h : \Omega \rightarrow \mathcal{A}(X)$ such that for every $x \in \Omega$ and $\forall i \in \mathcal{I}$ $h_i(x) \in \text{co}(\mathcal{P}_i(x) \cup \{\mathbf{e}_i\})$ with $h_j(x) \in \mathcal{P}_j(x)$ for some $j \in \mathcal{I}$.*

⁵ Note that in original paper item (c) has a typo for the range of $\phi : X \rightarrow \mathcal{K}(Y)$. Author denoted $\mathcal{K}(Y)$ as a set of all convex subsets of Y , but speak and prove the result for a narrower class of sets $\mathcal{D}(Y) \subset \mathcal{K}(Y)$, see p. 372. Here I present a less general result, to avoid a cumbersome specification of $\mathcal{D}(Y)$.

Proof. According to assumptions and Lemma 1, the correspondence $\varphi(\cdot)$ specified in (5) obeys requirements of Michael’s theorem on the existence of continuous selector: a lower hemicontinuous correspondence having domain $\Omega \subset \mathcal{A}(X)$, and with convex non-empty images. Thus, there is a continuous mapping satisfying

$$(v, t)(\cdot) : \Omega \rightarrow (L \times [0, 1])^{\mathcal{I}} \text{ such that } (v(x), t(x)) \in \varphi(x) \quad \forall x \in \Omega.$$

By construction one has $\sum_{\mathcal{I}} v_i(x) = 0, v(x) \neq 0$ and $\forall i \in \mathcal{I} v_i(x) = t_i(x)(g_i(x) - e_i), t_i(x) \in [0, 1]$ and $g_i(x) \in \mathcal{P}_i(x) \neq \emptyset$. Now one specifies

$$\eta(x) = \max_{i \in \mathcal{I}} t_i(x) > 0$$

and due to (A) ($g_i(x) \in \mathcal{P}_i(x) \subset X_i, X_i$ is convex and $e_i \in X_i \quad \forall i \in \mathcal{I}$) one concludes

$$h_i(x) = \frac{v_i(x)}{\eta(x)} + e_i \in X_i \quad \forall i \in \mathcal{I} \Rightarrow h(x) = (h_1(x), \dots, h_n(x)) \in \mathcal{A}(X).$$

So, by construction we have $h(x) \neq e, h_i(x) = \frac{t_i(x)}{\eta(x)}(g_i(x) - e_i) + e_i$ with $\frac{t_j(x)}{\eta(x)} = 1$ for some $j \in \mathcal{I}$ and $h(\cdot)$ is a function that we needed to find. \square

Proof of Theorem 1. Assume (4) is false for every $x \in \mathcal{A}(X)$, i.e. $\Omega = \mathcal{A}(X)$ and therefore $C^f(\mathcal{E}) = \emptyset$. Now applying Lemma 2 one can find a continuous function $h : \mathcal{A}(X) \rightarrow \mathcal{A}(X)$ such that for every $x \in \mathcal{A}(X)$ one has $h_j(x) \in \mathcal{P}_j(x)$ for some $j \in \mathcal{I}$. Since $\mathcal{A}(X)$ is a convex compact set then due to Brouwer’s fixed point theorem, this function has to have a fixed point $\bar{x} \in \mathcal{A}(X)$. At this point, there is $j \in \mathcal{I}$ for which one has $\bar{x}_j = h_j(\bar{x}) \in \mathcal{P}_j(\bar{x})$ that is impossible. So supposition $\Omega = \mathcal{A}(X)$ is false and therefore there is a point $x \in \mathcal{A}(X)$ such that (4) is true and $C^f(\mathcal{E}) \neq \emptyset$. \square

References

1. Aliprantis, C.D., Brown, D.J., Burkinshaw, O.: Edgeworth equilibria in production economies. *J. Econ. Theory* **43**, 252–91 (1989)
2. Aliprantis, C.D., Brown, D.J., Burkinshaw, O.: Existence and Optimality of Competitive Equilibria, p. 284. Springer, Heidelberg (1989). <https://doi.org/10.1007/978-3-642-61521-4>
3. Allouch, N., Predtetchinski, A.: On the non-emptiness of the fuzzy core. *Int. J. Game Theory* **37**, 203–10 (2008). <https://doi.org/10.1007/s00182-007-0105-2>
4. Aubin, J.P.: Mathematical methods of game and economic theory. North-Holland, Amsterdam/New York/Oxford (1979)
5. Florenzano, M.: On the non-emptiness of the core of a coalitional production economy without ordered preferences. *J. Math. Anal. Appl.* **141**, 484–90 (1989)
6. Florenzano, M.: Edgeworth equilibria, fuzzy core and equilibria of a production economy without ordered preferences. *J. Math. Anal. Appl.* **153**, 18–36 (1990)
7. Michael, E.: Continuous selections I. *Ann. Math.* **63**(2), 361–82 (1956)
8. Marakulin, V.M.: Contracts and domination in competitive economies. *J. New Econ. Assoc.* **9**, 10–32 (2011). (in Russian)
9. Marakulin, V.M.: Abstract equilibrium analysis in mathematical economics, 348p. SB Russian Academy of Science Publisher, Novosibirsk (2012). (in Russian)
10. Predtetchinski, A.: The fuzzy core and the (II, β) -balanced core. *Econ. Theory* **26**, 717–724 (2005)

Game Theory



Value of Cooperation in a Differential Game of Pollution Control

Angelina Chebotareva¹, Shimai Su¹✉, Elizaveta Voronina¹,
and Ekaterina Gromova²

¹ Saint Petersburg State University, St. Petersburg, Russia
st073379@student.spbu.ru

² National Research University Higher School of Economics, St. Petersburg, Russia
egromova@hse.ru

Abstract. The paper studies a linear-quadratic pollution control model for which a term named value of cooperation (VC) subordinated to value of information (VI) is introduced. Namely, the quantification of possible benefit or loss occurring under the circumstance where the players choose to play the game in a cooperative way or refuse to act as a coalition is being presented. In the paper, the construction of VC consists of the Shapley value involving four various types of characteristic functions which steer the performance of normalized value of cooperation (NVC) toward quite similar direction and an elaborate analysis related to the actual scenario is ensuing. Both of characteristics (VC and NVC) are new in the field of game theory and can be further applied to a wide class of problems. Theoretical results are demonstrated with a numerical example on the basis of pollution data contributed by three local enterprises in Penza (City of Russia).

Keywords: Differential game · Pollution control · Value of cooperation · Value of information

1 Introduction

In retrospect, we have been prominently investing our enthusiasm into the field of value of information since recent years. So far, the value of information (VI) regarding the possible changes of structure of model which cover the feasible adjustment of the boundary of control and the existence of terminal cost in [1], the estimation of initial stock in [5] has been explored. While we are still deep into the research of the current knowledge, expanding the width of the application of VI is also underway. In this paper, we would like to present value of cooperation (VC) which is fresh in the field of game theory. On the level of subordination, we take VC as a subset of VI.

The reported study was funded by RFBR and DFG, project number 21-51-12007.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 221–234, 2022.
https://doi.org/10.1007/978-3-031-09607-5_16

When it comes to the definition of VC, the meaning has already told us by itself, i.e., the potential benefit or loss through the cooperation. As we know, the player is always able to obtain higher payoff when they opt to cooperate instead of acting solo. Therefore, the 'loss' here does not refer to the case we just mentioned. The highlight is encircled in the difference between the payoff which player could achieve in the cooperative case and the Nash equilibrium case. A description of the model considered in this paper can be found in [6, 7]. The methods for constructing characteristic functions during the process of finding the Shapley value under the cooperative condition are described in [2–4, 8, 10, 11, 15].

As to the reason of bringing game theory into the environmental management is that the environmental problem is one of the most striking topic in the contemporary world. Obviously, there are many factors which contribute to it, in particular, the radical development of high-polluted industry which could be shown in the way of reckless exploitation of natural resources and lack of environmental protection measures. Meanwhile, one of the most important environmental regulations for the whole society is to keep a robust balance between the volume of pollution and the profit. In this paper, we will consider three leading enterprises in Penza. The relevant production data of three enterprises in 2016 can be found in the sources [16–18].

The structure of this paper is well organized in the following way. In Sect. 2, we formulate our main problem and determine our objective. The theoretical part concerning the selection of characteristic functions and construction of the Shapley value [13] are demonstrated in Sect. 3. The detailed procedure for formalizing value of cooperation is explained in Sect. 4 and a numerical example with support of actual production data is added in Sect. 5. In the end, we make our conclusion in Sect. 6.

2 Problem Formulation

Let us consider a pollution control model formulated in a linear-quadratic way over the time interval $[t_0, T]$ in correlation with [6, 7]. It is assumed that there are n stationary pollution contributors within a targeted region, i.e., the game has n players (firms) who fulfill their regular production at their own pace. Suppose the volume of pollutant u_i generated by player i at instant time t is proportional to the amount of its production,

$$u_i \in [0, b_i], \quad b_i > 0, \quad i = \overline{1, n}, \quad (1)$$

where b_i symbolizes the rate of total production income of the player i corresponding to its total pollution amount. Then the strategy of player i can be described as the plan of the pollution rate per unit time.

The total pollution level is given by $x(t)$ with the initial condition $x(t_0) = x_0$ so that the dynamics of pollution volume can be expressed by the differential equation

$$\dot{x}(t) = \sum_{i=1}^n u_i(t), \quad t \in [t_0, T]. \tag{2}$$

The purpose of the player is to choose the control path u_i in an optimal way. Generally, we assume that the player has the objective functional

$$K_i(x_0, T - t_0, u) = \int_{t_0}^T ((b_i - \frac{1}{2}u_i(t))u_i(t) - d_i x) dt, \tag{3}$$

where the cost of the player i to eliminate a unit of pollution is represented by $d_i \geq 0$. Each player in the game is expecting to maximize the functional as indicated in (3), i.e.,

$$K_i(x_0, T - t_0, u_1(t), u_2(t), \dots, u_n(t)) \rightarrow \max \tag{4}$$

In the paper, the game under the cooperative cases will be explored which means the players multilaterally agree on the use of optimal control $u^*(t) = (u_1^*(t), u_2^*(t), \dots, u_n^*(t))$ which satisfies (1) while conforming to the system dynamics (2) and the initial condition to realize

$$\sum_{i=1}^n K_i(x_0, T - t_0, u_1(t), u_2(t), \dots, u_n(t)) \rightarrow \max_{u_1^*, u_2^*, \dots, u_n^*}. \tag{5}$$

Subsequently, the optimal trajectory $x^*(t)$ can be obtained by integrating (1) with optimal control u^* .

3 Construction of the Shapley Value

3.1 Characteristic Functions

It is known that there are assorted types of characteristic functions. In general, we pay our attention to α -, δ -, ζ -, η -characteristic functions whose detailed descriptions of their constructions are explained in [3, 15].

α -characteristic Functions. The classical approach formulated by J. Neumann and O. Morgenstern in 1944 in [8] is used in the construction of α -characteristic function. According to this approach, $V(S)$ refers to the maximally guaranteed gain of the coalition S and the value of $V(S)$ can be calculated on the basis of an auxiliary antagonistic game between the coalition S and the anti-coalition $N \setminus S$. Thus, the coalition S acts as the maximizing module and the coalition $N \setminus S$ as the minimizing module:

$$V^\alpha(x_0, t_0, T, S) = \begin{cases} 0, & S = \{\emptyset\}, \\ \max_{\substack{u_i, i \in S \\ j \in N \setminus S}} \min_{i \in S} \sum_{i \in S} K_i(x_0, T - t_0, u_S, u_{N \setminus S}) & S \subset N, \\ \max_{u_1, \dots, u_n} \sum_{i=1}^n K_i(x_0, T - t_0, u_1, \dots, u_n) & S = N. \end{cases} \tag{6}$$

It is proved in [10] that $V^\alpha(x_0, t_0, T, S)$ is a superadditive function, i.e. it satisfies the following property:

$$V(x_0, t_0, S_1 \cup S_2) \geq V(x_0, t_0, T, S_1) + V(x_0, t_0, T, S_2), \tag{7}$$

$$\forall S_1, S_2 \in N, S_1 \cap S_2 = \emptyset.$$

However, this type of characteristic function has significant computational difficulties.

δ - characteristic Functions. In the work of L.A. Petrosyan, G. Zaccour [11], another constructive approach for the build of δ -characteristic function) was proposed.

$V(x_0, t_0, T, S)$ can be calculated as follows: The players from S maximize their total gain $K_i(u_S, u_{N \setminus S}^N)$ while the remaining players from the set $N \setminus S$ take strategies from Nash equilibrium $u_{N \setminus S}^N = \{u_j^N\}_{j \in N \setminus S}$. Thus, we have a 2-step procedure for constructing the characteristic function:

- 1) find the Nash equilibrium $\{u_i^N\}$ for all players $i \in N$;
- 2) stick the strategies from the Nash equilibrium u_j^N for player $j \in N \setminus S$ and for players from the coalition S we find the maximum of their total gain through $u_S = \{u_i\}_{i \in S}$. Then the formal definition is as follows:

$$V^\alpha(x_0, t_0, T, S) = \begin{cases} 0, & S = \{\emptyset\}, \\ \max_{u_i, i \in S} \min_{j \in N \setminus S} \sum_{i \in S} K_i(u_S, u_{N \setminus S}^N) & S \subset N, \\ \max_{u_1, \dots, u_n} \sum_{i=1}^n K_i(x_0, T - t_0, u_1, \dots, u_n) & S = N. \end{cases} \tag{8}$$

The characteristic function made in this way has the following advantages. First of all, it requires less computational effort. Secondly, the calculated value of V^δ is based on the already calculated Nash equilibrium, which greatly simplifies further calculations. What’s more, the definition of δ characteristic function has a clear economic interpretation, namely that players who do not join coalition S will not form anti-coalition $N \setminus S$, which corresponds to their non-aggressive behaviour.

In general, the δ -characteristic function is not superadditive, i.e., it does not satisfy condition (7) in contrast to (6). The question of existence and uniqueness of the Nash equilibrium also becomes relevant in this case.

ζ - characteristic Functions. A two-step procedure is also being applied in order to build ζ characteristic function. We choose a set of optimal rules that maximize the total payoff of all players. Next, we utilize the optimal rules obtained in the previous step for the players in the coalition S , while the players in the set $N \setminus S$ minimize the payoff of players in the coalition S . Then we have

$$V^\zeta(x_0, t_0, T, S) = \begin{cases} 0, & S = \{\emptyset\}, \\ \min_{u_j, j \in N \setminus S} \sum_{i \in S} K_i(x_0, T - t_0, u_S^*, u_{N \setminus S}) & S \subset N, \\ \max_{u_1, \dots, u_n} \sum_{i=1}^n K_i(x_0, T - t_0, u_1, \dots, u_n) & S = N. \end{cases} \tag{9}$$

It is confirmed [4] that V^ζ is a superadditive characteristic function compared with (6). Also ζ -characteristic function can be calculated in two stages by using expressions for optimal control, which dramatically simplifies the calculation process with comparison to the construction of α -characteristic function.

η - characteristic Functions. This type of characteristic function [2] has a benefit that the computational process is succinct because the players from the coalition S use the previously formed optimal strategies $u_S^* = \{u_i^*\}_{i \in S}$, while the players from $N \setminus S$ act similarly to the δ -characteristic function, i.e., using u_i^{NE} ,

$$V^\eta(x_0, t_0, T, S) = \begin{cases} 0, & S = \{\emptyset\}, \\ \sum_{i \in S} K_i(x_0, T - t_0, u_S^*, u_{N \setminus S}^{NE}) & S \subset N, \\ \max_{u_1, \dots, u_n} \sum_{i=1}^n K_i(x_0, T - t_0, u_1, \dots, u_n) & S = N. \end{cases} \tag{10}$$

3.2 The Shapley Value

Given the Shapley value as a cooperative principle of optimality in a game,

$$Sh_i(x_0, t_0, T) = \sum_{s \subset N, i \in S} \frac{(n-s)!(s-1)!}{n!} (V(S, x_0, t_0, T) - V(S \setminus \{i\}, x_0, t_0, T)). \tag{11}$$

It represents a division that satisfies the properties of individual and collective rationality. For a game of three participants, the Shapley value is as follows:

$$\begin{aligned} Sh_1(x_0, t_0, T) &= \frac{1}{3}[V(\cdot, \{1, 2, 3\}) - V(\cdot, \{2, 3\})] + \frac{1}{3}V(\cdot, \{1\}) \\ &+ \frac{1}{6}[V(\cdot, \{1, 2\}) - V(\cdot, \{2\}) + V(\cdot, \{1, 3\}) - V(\cdot, \{3\})], \\ Sh_2(x_0, t_0, T) &= \frac{1}{3}[V(\cdot, \{1, 2, 3\}) - V(\cdot, \{1, 3\})] + \frac{1}{3}V(\cdot, \{2\}) \\ &+ \frac{1}{6}[V(\cdot, \{1, 2\}) - V(\cdot, \{1\}) + V(\cdot, \{2, 3\}) - V(\cdot, \{3\})], \\ Sh_3(x_0, t_0, T) &= \frac{1}{3}[V(\cdot, \{1, 2, 3\}) - V(\cdot, \{1, 2\})] + \frac{1}{3}V(\cdot, \{3\}) \\ &+ \frac{1}{6}[V(\cdot, \{1, 3\}) - V(\cdot, \{1\}) + V(\cdot, \{2, 3\}) - V(\cdot, \{2\})]. \end{aligned} \tag{12}$$

4 Construction of Value of Cooperation

4.1 Cooperative Solution to the Model

Let us discuss a game-theoretic model with 3 players, i.e., $n = 3$ and the initial condition $x(t_0) = x_0$. The Pontryagin maximum principle [9] is being applied to solve the problem (5). Out of this purpose, it is necessary to construct the Hamiltonian function:

$$H(x, u, \psi) = \sum_{i=1}^3 ((b_i - \frac{1}{2}u_i)u_i - d_i x) + \psi(t)(u_1 + u_2 + u_3) \rightarrow \max.$$

In accordance with derivative rule, we could determine u^* where the maximal value of Hamiltonian function can be reached. Furthermore, we have the canonical system

$$\begin{cases} \dot{x} = \frac{\partial H}{\partial \psi} \\ \dot{\psi} = -\frac{\partial H}{\partial x} \end{cases} \tag{13}$$

Since there is no terminal cost in this case, then $\psi(T) = 0$. Combined with $\dot{\psi} = -\frac{\partial H}{\partial x} = d_s$ which is obtained from (13), we present $d_1 + d_2 + d_3 = d_s$, $b_1 + b_2 + b_3 = b_s$, therefore

$$\psi(t) = d_s(t - T). \tag{14}$$

Correspondingly, the optimal control goes:

$$u^*(t) = \begin{pmatrix} b_1 - d_s(T - t) \\ b_2 - d_s(T - t) \\ b_3 - d_s(T - t) \end{pmatrix} \tag{15}$$

The additional condition on the parameters of the model under which the optimal controls are admissible $u_i \in [0 : b_i]$:

$$d_i \in [0, \frac{\min\{b_1, b_2, b_3\}}{T} - d_s], \tag{16}$$

Now turning to optimal trajectory, from (2), (15) and the initial condition $x(0) = x_0$, naturally

$$x^*(t) = \frac{3d_s}{2}(t^2 - t_0^2) + (b_s - 3Td_s)(t - t_0) + x_0. \tag{17}$$

4.2 Nash Equilibrium Case

Similar to the case above, the Hamiltonian function is

$$H_i(x, u, \psi) = (b_i - \frac{1}{2}u_i)u_i - d_i x + \psi(t)(u_1 + u_2 + u_3), \tag{18}$$

The general method of pinpointing the optimal control through the derivative in this case proceeds in the same way. Hence, the optimal control:

$$u_i^{NE} = b_i - d_i(T - t), \quad i = 1, 2, 3.$$

The optimal trajectory:

$$x^{NE}(t) = \frac{d_s}{2}(t^2 - t_0^2) + (b_s - Td_s)(t - t_0) + x_0. \tag{19}$$

4.3 Expressions for Characteristic Functions

Replying on definitions (6)-(10) we generate expressions for all types of characteristic functions in the frame of a 3-player model above.

Construction of the α -characteristic Function. Let us construct an α -characteristic function for the grand coalition $S = N$. In this case, each player uses a control that maximizes the total payoff of all players, i.e., the controls are defined by expression (15). Thus,

$$\begin{aligned} V^\alpha(N, T - t_0) &= \sum_{i=1}^3 K_i(t_0, u_1^*, u_2^*, u_3^*) \\ &= \frac{1}{2}(T - t_0)^3 d_s^2 + \frac{1}{2}(T - t_0)(B_s^2 - 2d_s x_0) - \\ &\quad - \frac{1}{2}(T - t_0)^2 b_s d_s \end{aligned} \tag{20}$$

where $B_s^2 = b_1^2 + b_2^2 + b_3^2$, $d_s^2 = (d_1 + d_2 + d_3)^2$.

For constructing $V^\alpha(\{i\}, T - t_0)$, $i = \overline{1, 3}$, we use definition (6). In advance, we need to find such controls $u_j, u_k, i \neq j \neq k \in N$ which lead to $\min_{u_j, u_k} K_i(t_0, u_i, u_j, u_k)$ by means of the Pontryagin maximum principle [14]. They will take the form:

$$u_j = b_j, \quad u_k = b_k.$$

Later, utilizing the same method, it is expected to find the control u_i under which $\max_{u_i} \min_{u_j, u_k} K_i(t_0, u_i, u_j, u_k)$ is achieved. It will take the form:

$$u_i = b_i - d_i(T - t),$$

The characteristic function for single player

$$\begin{aligned} V^\alpha(\{i\}, T - t_0) &= \frac{1}{6}(T - t_0)^3 d_i^2 - \frac{1}{2}(T - t_0)^2 b_s d_i + \frac{1}{2}(T - t_0) b_i^2 \\ &\quad - (T - t_0) d_i x_0. \end{aligned} \tag{21}$$

For constructing $V^\alpha(\{i, j\}, T - t_0)$, it is necessary to find a control $u_k, i \neq j \neq k \in N$ with which we get $\min_{u_k} (K_i(t_0, u_i, u_j, u_k) + K_j(t_0, u_i, u_j, u_k))$. In this case, $u_k = b_k$. Thereafter, the u_i, u_j have to be determined by making $\max_{u_i, u_j} \lim_{u_k} (K_i(t_0, u_i, u_j, u_k) + K_j(t_0, u_i, u_j, u_k))$,

$$u_i = b_i - (d_i + d_j)(T - t).$$

Then the characteristic function for coalition $S = \{i, j\}$ is

$$\begin{aligned} V^\alpha(\{i, j\}, T - t_0) &= \frac{1}{3}(T - t_0)^3 d_{ij}^2 - \frac{1}{2}(T - t_0)^2 b_s d_{ij} \\ &\quad + \frac{1}{2}(T - t_0)(b_i^2 + b_j^2) - (T - t_0) d_{ij} x_0. \end{aligned} \tag{22}$$

where $d_{ij} = d_i + d_j, d_{ij}^2 = (d_i + d_j)^2$.

Construction of the δ -characteristic Function. Hinging on definition (8), we can decide δ -characteristic function. The procedure is succinct because the players who do not belong to the coalition S use Nash equilibria as strategies. Then, we will have the problem of finding controls $u_S, S \in N$ that maximize $\sum_{i \in S} K_i(t_0, u_S, u_{N \setminus S})$.

$$V^\delta(\{i\}, T - t_0) = \frac{1}{6}(T - t_0)^3(2d_s d_i - d_i^2) - \frac{1}{2}(T - t_0)^2 b_s d_i + \frac{1}{2}(T - t_0) b_i^2 - (T - t_0) x_0 d_i. \tag{23}$$

$$V^\delta(\{i, j\}, T - t_0) = \frac{1}{3}(T - t_0)^3(d_k d_{ij} + d_{ij}^2) - \frac{1}{2}(T - t_0)^2 b_s d_{ij} + \frac{1}{2}(T - t_0)(b_i^2 + b_j^2) - (T - t_0) d_{ij} x_0. \tag{24}$$

Construction of the ζ -characteristic Function. As presented in definition (9), the ζ -characteristic function in our case can be formulated with the optimal controls which are mentioned in (15) used as $u_{s \in N}$, and $u_{S \setminus N}$ to satisfy $\min_{u_j, j \in S \setminus N} (\sum_{i \in S} K_i(t_0, u_S^*, u_{S \setminus N}))$.

$$V^\zeta(\{i\}, T - t_0) = \frac{1}{6}(T - t_0)^3(2d_i d_s - d_s^2) - \frac{1}{2}(T - t_0)^2 b_s d_i + \frac{1}{2}(T - t_0) b_i^2 - (T - t_0) x_0 d_i. \tag{25}$$

$$V^\zeta(\{i, j\}, T - t_0) = \frac{1}{3}(T - t_0)^3(2d_{ij} d_s - d_s^2) - \frac{1}{2}(T - t_0)^2 b_s d_{ij} + \frac{1}{2}(T - t_0)(b_i^2 + b_j^2) - (T - t_0) d_{ij} x_0. \tag{26}$$

Construction of the η -characteristic Function. By definition (10), the computational process for the η -characteristic function is the most simplified compared with the characteristic functions described previously, there is no additional calculations required after finding $u^*(t), u^{NE}(t)$.

$$V^\eta(\{i\}, T - t_0) = \frac{1}{6}(T - t_0)^3(2d_i(d_s + d_{jk}) - d_s^2) - \frac{1}{2}(T - t_0)^2 b_s d_i + \frac{1}{2}(T - t_0) b_i^2 - (T - t_0) x_0 d_i. \tag{27}$$

where $d_{jk} = d_j + d_k$.

$$V^\eta(\{i, j\}, T - t_0) = \frac{1}{3}(T - t_0)^3(2d_{ij} d_s + d_k d_{ij} - d_s^2) - \frac{1}{2}(T - t_0)^2 b_s d_{ij} + \frac{1}{2}(T - t_0)(b_i^2 + b_j^2) - (T - t_0) d_{ij} x_0. \tag{28}$$

4.4 Value of Cooperation

Definition 1. *The value of cooperation referring to the cooperative case and Nash equilibrium one is defined as*

$$VC_i = Sh_i - K_i(x_0, T, u^{NE}). \tag{29}$$

Furthermore, we define the normalized value of cooperation for player i as

$$NVC_i = \frac{VC_i}{Sh_i} = \frac{Sh_i - K_i(x_0, T, u^{NE})}{Sh_i} \times 100\%. \tag{30}$$

The Shapley value with respect to player i is determined by expression (11). To find the vector of the Shapley we can use a characteristic function of any type. We take the construction of the Shapley value regarding α -characteristic function as an example, VC_i^α and NVC_i^α can be expressed as

$$VC_i^\alpha = Sh_i^\alpha - K_i(x_0, T, u^{NE}),$$

$$NVC_i^\alpha = \frac{VC_i^\alpha}{Sh_i^\alpha} = \frac{Sh_i^\alpha - K_i(x_0, T, u^{NE})}{Sh_i^\alpha} \times 100\%.$$

4.5 Value of Information

Now suppose that the information about the initial level of pollution x_0 is not available to the players, and they overestimate the initial level of pollution, i.e., $\hat{x}_0 > x_0$ and the initial condition will be altered to $x(t_0) = \hat{x}_0$.

In this case, apart from the change of initial level of pollution, the rest procedure will be identical to what we describe in Subsect. 4.1. Therefore, we have our new optimal control $\hat{u}(t)$,

$$\hat{u}(t) = \begin{pmatrix} b_1 - d_s(T - t) \\ b_2 - d_s(T - t) \\ b_3 - d_s(T - t) \end{pmatrix} \tag{31}$$

And the optimal trajectory $\hat{x}(t)$ and the total payoff of three players under this condition are

$$\hat{x}(t) = \hat{x}_0 + (t - t_0)b_s - 3(t - t_0)Td_s + \frac{3d_s}{2}(t^2 - t_0^2)$$

$$= \hat{x}_0 + (t - t_0)(b_s - 3Td_s) + \frac{3d_s}{2}(t^2 - t_0^2), \tag{32}$$

$$\sum_{i=1}^3 K_i(t_0, \hat{u}_i) = \frac{1}{2}(T - t_0)^3 d_s^2 + \frac{1}{2}(T - t_0)(B_s^2 - 2d_s \hat{x}_0)$$

$$- \frac{1}{2}(T - t_0)^2 b_s d_s. \tag{33}$$

Definition 2. *The value of information here referring to the estimation of the initial pollution level is defined as*

$$VI = \sum_{i=1}^3 K_i(t_0, u^*) - \sum_{i=1}^3 K_i(t_0, \hat{u}) \tag{34}$$

where $\sum_{i=1}^3 K_i(t_0, u^*)$ can be taken from (20). Furthermore, we define the normalized value of information for players as

$$NVI = \frac{\sum_{i=1}^3 K_i(t_0, u^*) - \sum_{i=1}^3 K_i(t_0, \hat{u})}{\sum_{i=1}^3 K_i(t_0, u^*)} \times 100\%. \tag{35}$$

5 Numerical Example of Actual Scenarios in Penza

In this paper, the three leading enterprises which we take into account are specialist in different realms in Penza and Penza Oblast. With currently available data [16–18], we formalize the problem and formulate a differential game of three players named as Foton, Penzadieselmash and Penza Bread Plant.

5.1 Parameters of the Model

To calculate the parameters of model - b_i, d_i , we will look for the data on pollution sources. The coefficient $b_i > 0$ equals the ratio of the total income from the production of the i -th company (P_i) to the total amount of pollution by the right company (V_i):

$$b_i = \frac{P_i}{V_i}. \tag{36}$$

The parameter $d_i > 0$ defines the costs of the player i to eliminate a unit of total pollution:

$$d_i = \frac{L_i}{V_1 + V_2 + V_3}. \tag{37}$$

The consequent values of b_i, d_i are indicated in Table 1. Besides, for better demonstration, we assume that $t_0 = 0, x_0 = 0, T \in [1, 10]$ in the game.

Table 1. Real value of b_i, d_i for three enterprises

Company	b_i	d_i
Foton	148534, 48	421, 61
Penzadieselmash	1619303, 9	426, 29
Penza Bread Plant	1752944, 9	876, 62

5.2 Analysis of Value of Information and Cooperation

Value of Cooperation. Applying the production data and combining the result we get in Figs. 1 and 2, the first point we would like to discuss is whether the player should cooperate or not and which kind of characteristic functions the player ought to choose if cooperation is ongoing. In Fig. 1, it's crystal clear that the cooperation is beneficial to all players especially to the player - 'Foton', although the magnitude of NVC is not larger than 10^{-3} , we have to take the total amount of profit into account. Therefore, it's reasonable for them to take cooperative solution.

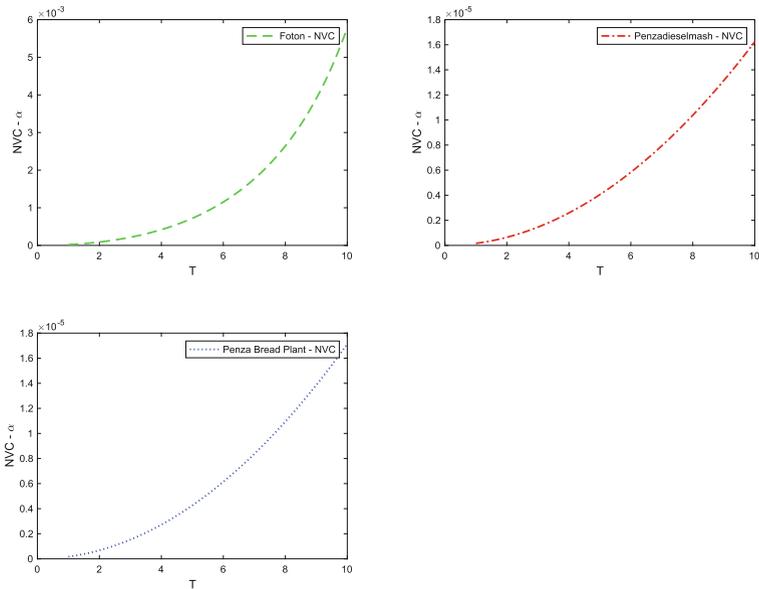


Fig. 1. The performance of NVC for three players i, j, k varied in α characteristic function(cf) with changing terminal time T . (Since the performances of NVC varied in four characteristic functions are approximately same, only one of them shown here)

Now suppose the cooperation is settled, then what's the comparatively best choice of characteristic functions for each player? In fact, the result shows in Fig. 2 that there is no unanimous selection of characteristic function for all three players to attain their optimal goal. In this case, we believe there's further discussion need to be made.

Value of Information. Suppose $\hat{x}_0 = x_0 + \theta, x_0 = 0, \theta \in [-0, 5, 0, 5]$, as shown in Fig. 3, We can observe that when $\theta = 0$, the decision-making players have accurate information about the original pollution and are not willing to pay anything to improve this knowledge. However, as the information becomes

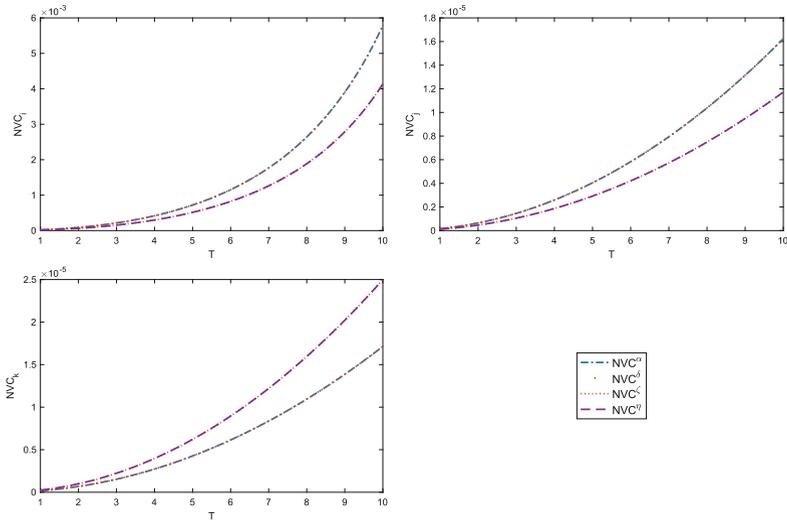


Fig. 2. The performance of NVC corresponding to four characteristic functions(cf) with changing terminal time T for each player i, j, k , up-left: Foton - i , up-right: Penzadieselmash - j , bottom-left: Penza Bread Plant - k .

increasingly inaccurate, the value of the information increases. And the more θ deviates from zero, the more value NVC takes on. This is true both in the case of overestimation of the initial contamination level x_0 and in the case of underestimation.

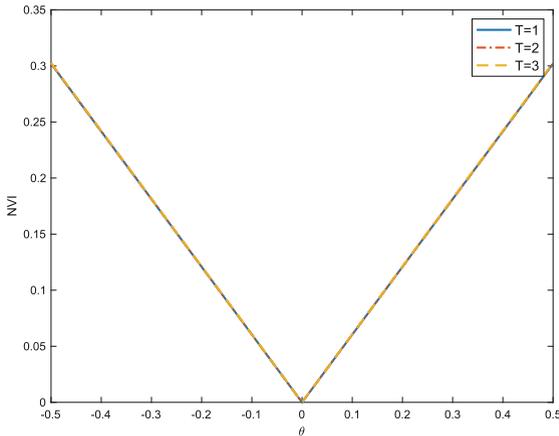


Fig. 3. The performance of NVI corresponding to different terminal time $T = 1, 2, 3$ with varied estimation of initial pollution stock.

6 Conclusion

The limelight of the paper is the comparison of payoff between the cooperative and Nash equilibrium case. With help of four different types of characteristic function, we separately construct the Shapley value which is used to evaluate the outcome of NVC and an actual example is being brought to vividly demonstrate our intention. The result informs the enterprises of the benefit of cooperation and the explicit proportion of the benefit the player will get if he comply with the strategies, which we think would greatly reduce their risk of making unprofitable decision. In addition, the analysis of the impact of estimation of initial stock is being provided to the enterprises to complement their knowledge over incomplete information.

References

1. Chebotareva, A., Su, S., Tretyakova, S., Gromova, E.: On the value of the preexisting knowledge in an optimal control of pollution emissions. In: Contributions to Game Theory and Management, vol. 14, pp. 48–57 (2021). <https://doi.org/10.13140/RG.2.2.34428.87686>
2. Gromova, E., Marova, E.: Coalition and anti-coalition interaction in cooperative differential games. In: IFAC-PapersOnLine, vol. 51, pp. 479–483 (2018). <https://doi.org/10.1016/j.ifacol.2018.11.466>
3. Gromova, E., Marova, E.: On the characteristic function construction technique in differential games with prescribed and random duration. In: Contributions to Game Theory and Management, vol. 11, pp. 53–66 (2018)
4. Gromova, E., Petrosyan, L.: On an approach to constructing a characteristic function in cooperative differential games. Autom. Remote Control **78**(9), 1680–1692 (2017). <https://doi.org/10.1134/S0005117917090120>
5. Gromova, E., Tur, A., Gromov, D.: On the estimation of the initial stock in the problem of resource extraction. Mathematics **9**(23), 3099 (2021). <https://doi.org/10.3390/math9233099>
6. Gromova, E.: The Shapley value as a sustainable cooperative solution in differential games of three players. In: Recent Advances in Game Theory and Applications, Static and Dynamic Game Theory: Foundations and Applications, pp. 67–91 (2016). https://doi.org/10.1007/978-3-319-43838-2_4
7. Haurie, A., Zaccour, G.: Differential game models of global environmental management. In: Annals of Dynamic Games, Boston, pp. 124–132 (1994). https://doi.org/10.1007/978-1-4612-0841-9_1
8. Neumann, J., Morgenstein, O.: Game Theory and Economic Behavior. Princeton University Press (1944)
9. Pontryagin, L., Boltyansky, V., Gamkrelidze, R., Mishchenko, E.: Mathematical theory of optimal processes. Interscience, New York (1962)
10. Petrosyan, L., Danilov, N.: Cooperative differential games and their applications. Tomsk University Press (1985)
11. Petrosyan, L., Zaccour, G.: Time-consistent Shapley value allocation of pollution cost reduction. J. Econ. Dyn. Control **27**(3), 381–398 (2003) [https://doi.org/10.1016/S0165-1889\(01\)00053-7](https://doi.org/10.1016/S0165-1889(01)00053-7)
12. Raiffa, H., Schlaifer, R.: Applied Statistical Decision Theory. Wiley, New York (1961)

13. Shapley, L.: Notes on the n-Person Game - II: The Value of an n-Person Game, Santa Monica, California (1951). <https://doi.org/10.1515/9781400881970-018>
14. Savin, K., Gromova, E.: On properties of characteristic functions in a game with multilateral external influences. In: Control and Stability Processes (2021)
15. Vikulova, A.: On non-standard characteristic function construction in the cooperative game of harmful emissions control. In: Control Processes and Stability, pp. 617–621 (2016)
16. Ecological and economic efficiency of measures to reduce emissions into the atmosphere at the enterprise CJSC “Foton” (Penza). No. 1 (2019)
17. Education and science in the modern world. Innovations **5**, 226–232 (2018)
18. Education and science in the modern world. Innovations **5**, 233–240 (2018)



A Cooperation Scheme in Multistage Game of Renewable Resource Extraction with Asymmetric Players

Denis Kuzyutin^{1,2} , Yulia Skorodumova¹ , and Nadezhda Smirnova² 

¹ Saint Petersburg State University, Universitetskaya nab. 7/9,
199034 St. Petersburg, Russia

d.kuzyutin@spbu.ru, st054724@student.spbu.ru

² HSE University, Soyuza Pechatnikov ul., 16, 190008 St. Petersburg, Russia
nvsmirnova@hse.ru

Abstract. We derive a non-cooperative and cooperative strategies and state trajectories for a finite-horizon multistage game of renewable resource extraction with asymmetric players. Assuming transferable utility we extend the subgame perfect core concept introduced for extensive-form games to the class of n -person multistage games and specify an algorithm for choosing a unique payoff distribution procedure from the core in a two-player game. This quasi proportional payment schedule satisfies several good properties and could be applied to implement a cooperative solution based on the maximization of the relative benefit from cooperation (or the value of cooperation). We provide a numerical example to demonstrate the properties of the obtained solutions and the algorithm implementation.

Keywords: Multistage game · Subgame-perfect equilibrium · Payoff distribution procedure · Cooperative solution · Renewable resource extraction · Fishery-management model

1 Introduction

In the paper, we consider a competitive model of renewable resource extraction as a finite-horizon multistage game with feedback information structure. This model, in particular, could be interpreted as a fishery-management model (see, e.g., seminal paper [18] on the so-called fish wars, the related papers [1–3, 12, 20–23, 28, 29]) and the review [30]. We adopt a rather general assumption that each player's stage performance criterion is *log* of the current extraction level and focus on the finite-horizon game when the players value differently the resource residual stock after the extraction process ends. This is the only source for asymmetry of the players accepted in the paper (see, e.g., [2, 5, 23, 29] for other reasons and aspects of the players' asymmetry).

The reported study was funded by RFBR and DFG, project number 21-51-12007.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 235–249, 2022.
https://doi.org/10.1007/978-3-031-09607-5_17

As it is known, the non-cooperative (selfish) behavior in dynamic models of renewable resource extraction under fairly general assumptions leads to worse results (in particular, more extensive resource exploitation) than the cooperative behavior (see, e.g. [2, 3, 12, 20–23, 29], bearing in mind the possible exceptions [9, 19]). Hence, a problem how to guarantee the sustainability of cooperation (especially, from the long-term perspective) arises. In the paper, we assume that the payoffs are transferable (between the players) and explore the payoff distribution procedure (PDP) based approach to reach and implement the cooperative agreement. Such approach was firstly introduced in [25] for differential games and then was successfully applied to different classes of dynamic games (see, e.g., [4, 5, 11, 13–16, 21, 26, 27, 30, 32]).

To derive non-cooperative and cooperative feedback strategies we use standard dynamic programming method. Then we extend the novel β -subgame perfect core (β -S-P Core) concept (see [6, 7, 17]) to the class multistage games under consideration. Further, we introduce a refinement of the β -S-P Core based on maximization of the relative benefit from cooperation (see [17]) and constructing a specific PDP meeting several advantageous properties. Finally, we provide a numerical example of the two-person multistage game to demonstrate the properties of the obtained solutions. The contributions of the paper is twofold:

- we derive analytical solution for specific finite-horizon multistage game of renewable resource extraction with asymmetric players;
- we extend the β -S-P Core concept to n -person multistage games with transferable utility and provide an algorithm for the constructing of quasi proportional PDP which belongs to non-empty β -S-P Core of a two-player game.

The remainder of the paper is organized as follows. In Sect. 2, we introduce the model and derive non-cooperative solution (subgame perfect feedback-equilibrium strategies). In Sect. 3, we obtain a cooperative strategy and trajectory and define the β -S-P Core for multistage games. An algorithm for selecting a unique PDP from β -S-P Core is specified in Sect. 4. We provide a numerical example in Sect. 5 and briefly conclude in Sect. 6.

2 The Model Non-cooperative Behavior

We consider the following finite-horizon discrete time model of renewable resource extraction. Suppose that n players exploit a common renewable resource. Let $x(t)$ be a measure of the resource at time $t = 0, 1, \dots, T$ (state variable), while $u_j(t)$ denote player j 's extraction level in that period (control variable). Player $j \in N = \{1, \dots, n\}$ aims to maximize an objective function or performance criterion of the form

$$H_j(\cdot) = \sum_{\tau=0}^{T-1} \delta^\tau \ln u_j(\tau) + K_j \delta^T \ln x(T), \quad (1)$$

where $\delta \in (0, 1)$ is a discount factor, and $K_j > 0$ is a parameter that specifies the player j 's valuation of the resource residual stock after the extraction process ends.

As it was noted in [2, 5, 23, 29] there are several sources of the players asymmetry in the renewable resource extraction models (in particular, fishery management models) as well as in dynamic environmental models. The players may have different costs, different discount rates, they may value the residual stock differently, e.t.c. We focus in the paper on the case when the players have the same discount factor δ , and the only source for asymmetry is that the players value differently the resource residual stock (after the fishery process ends). Namely, we assume that coefficients K_j in (1) could be different. Note that similar assumption is accepted in [5, 11, 17].

We adopt in the paper the linear dynamics of the resource stock evolution when there is exploitation, namely:

$$x(t + 1) = \alpha \cdot x(t) - \sum_{j=1}^n u_j(t), \quad x(0) = x_0, \tag{2}$$

where $\alpha \geq 1$ denotes the natural growth rate, and the feedback information structure, i.e. $u_j(\cdot) = u_j(t, x(t))$, $j = 1, \dots, n$; $t = 0, \dots, T - 1$.

Denote by $G^0(n, x_0, T)$ multistage n -player game starting at time instant $t = 0$ with discrete dynamics (2), objective functions (1) and feedback information structure. Each intermediate state $x(t)$, $t = 0, \dots, T - 1$ determines a subgame $G^t(n, x(t), T)$ starting at time instant $\tau = t$ and initial state $x(t)$ with the subgame objective functions

$$H_j^t(\cdot) = \sum_{\tau=t}^{T-1} \delta^{\tau-t} \ln u_j(\tau) + K_j \delta^{T-t} \ln x(T), \quad j = 1, \dots, n. \tag{3}$$

The concept of subgame perfect equilibrium [31] is now accepted as a standard non-cooperative solution in a dynamic game.

Definition 1. A feedback strategy profile $u(t, x) = (u_1(t, x), \dots, u_n(t, x))$ constitutes a Nash equilibrium (NE) in $G^0(n, x(0), T)$, if

$$H_j(v_j(\cdot), u_{-j}(\cdot)) \leq H_j(u_j(\cdot), u_{-j}(\cdot))$$

for any admissible feedback strategy $v_j(\cdot)$ of every player $j = 1, \dots, n$.

Definition 2. A feedback strategy profile u forms a subgame perfect equilibrium (SPE) in $G^0(n, x(0), T)$ if for each intermediate time instant $t = 1, \dots, T - 1$ and state $x(t)$ the restriction of u in the subgame $G^t(n, x(t), T)$ still constitutes a NE in that subgame.

We employ the dynamic-programming algorithm to determine the feedback-equilibrium strategies of the players in multistage game $G^0(n, x(0), T)$.

Let $(u_j^{SPE}(t, x), u_{-j}^{SPE}(t, x))$, $t = 0, 1, \dots, T - 1$, denote a feedback SPE solution. Then the (present-valued) value function for player j in the subgame $G^t(n, x(t), T)$ takes the form

$$V_j(t, x) = \max_{u_j} \{ \ln u_j + \delta \cdot V_j(t + 1, \alpha x - u_j - \sum_{i \neq j} u_i^{SPE}(t, x)) \}, \quad (4)$$

$$V_j(T, x) = K_j \cdot \ln x(T). \quad (5)$$

For the sake of simplicity hereinafter we will consider a game of two players (denoted j and $-j$) to derive equilibrium and cooperative solutions. It is worth noting that the same approach is applicable for n -player multistage game, implying that one can obtain similar results for the case $n > 2$.

We guess the following functional form of the value functions:

$$V_j(t, x) = A_j(t) \ln x + B_j(t), \quad t = 0, 1, \dots, T, \quad j = 1, 2. \quad (6)$$

Proposition 1. *A multistage finite-horizon game $G^0(n = 2, x(0), T)$ possesses a unique SPE*

$$u_j(x) = \alpha \frac{A_{-j}(t + 1)}{\varphi(t + 1)} \cdot x, \quad j = 1, 2; \quad t = 0, \dots, T - 1, \quad (7)$$

where $\varphi(t + 1) = A_j(t + 1) + A_{-j}(t + 1) + \delta A_j(t + 1)A_{-j}(t + 1)$, while coefficients $A_j(t)$ satisfy the recurrence formula

$$A_j(t) = 1 + \delta A_j(t + 1), \quad A_j(T) = K_j. \quad (8)$$

The SPE state trajectory is

$$x(t + 1) = \frac{\alpha \delta A_j(t + 1) \cdot A_{-j}(t + 1)}{\varphi(t + 1)} \cdot x(t), \quad t = 0, \dots, T - 1. \quad (9)$$

The value functions (6) represent the SPE payoffs in the subgame $G^t(n = 2, x(t), T)$, $t = 0, \dots, T - 1$, while coefficients $B_j(t)$ satisfy the recurrence formula

$$B_j(t) = \Phi_j(\alpha, \delta, A_j(t + 1), A_{-j}(t + 1), B_j(t + 1)) = \ln \frac{\alpha A_{-j}(t + 1)}{\varphi(t + 1)} + \delta [A_j(t + 1) \cdot \ln \frac{\alpha \delta A_j(t + 1) \cdot A_{-j}(t + 1)}{\varphi(t + 1)} + B_j(t + 1)], \quad B_j(T) = 0. \quad (10)$$

Proof (Proof Sketch). We use the standard technique based on the dynamic programming (see, e.g., [11] for details) and the value functions in the form (6). Substituting value functions (6) in (4) we get

$$\begin{aligned} V_j(t, x) &= A_j(t) \ln x + B_j(t) \\ &= \max \{ \ln u_j + \delta (A_j(t + 1) \cdot \ln(\alpha x - u_j - u_{-j}^{SPE}(t, x)) + B_j(t + 1)) \}. \end{aligned} \quad (11)$$

Using first order conditions for the interior solution we obtain linear system

$$\frac{1}{u_j} = \delta \frac{A_j(t+1)}{\alpha x - (u_j + u_{-j})}, \quad j = 1, 2,$$

which has a unique solution (7). Hence, corresponding state trajectory is given by (9).

Further, we substitute functions (7) in (11) and compare the coefficients in the left- and right-hand sides. Straightforward calculation yields the recurrence formulae (8) and (10), that could be used to compute (backward in time) all the characteristics of the SPE scenario. \square

Remark 1. One can prove that multistage finite-horizon game $G^0(n, x(0), T)$, $n > 2$, still possesses a unique SPE, and moreover, the feedback equilibrium strategies $u_j(x)$ are proportional to x .

Remark 2. Recurrence formulae (8) and (10) are sufficient and convenient to calculate all the value functions (6) coefficients. However, we can provide explicit formulae for $A_j(t)$ and $B_j(t)$, $t = 0, 1, \dots, T$, $j = 1, 2$. Namely,

$$A_j(t) = \delta^{T-t} \cdot K_j + \sum_{\tau=t+1}^T \delta^{T-\tau}, \quad t = 0, 1, \dots, T-1, \quad A_j(T) = K_j. \quad (12)$$

To simplify (10) we'll use the following notations:

$$L_1(t+1) = \ln \frac{\alpha A_{-j}(t+1)}{\varphi(t+1)}, \quad L_2(t+1) = \ln \frac{\alpha \delta A_j(t+1) \cdot A_{-j}(t+1)}{\varphi(t+1)}.$$

Then, having all the coefficients $A_j(t)$, $t = 0, 1, \dots, T$, $j = 1, 2$, one can use the following explicit formulae to calculate $B_j(t)$:

$$B_j(t) = \sum_{\tau=t+1}^T \delta^{\tau-(t+1)} \cdot [L_1(\tau) + \delta A_j(\tau) \cdot L_2(\tau)], \quad t = 0, \dots, T-1, \quad B_j(T) = 0. \quad (13)$$

3 Cooperative Behavior β -S-P Core

Given nonempty coalition $S \subset N$, the induced multistage game $G_S^0(n - |S| + 1, x(0), T)$ describes the case when coalition S becomes a new player, i.e. all the players in S fully coordinate their strategies to maximize the total payoff of S

$$H_S(\cdot) = \sum_{\tau=0}^{T-1} \delta^\tau \ln \sum_{j \in S} u_j(\tau) + \sum_{j \in S} K_j \cdot \delta^T \ln x(T). \quad (14)$$

Denote by $\gamma(S, t, x)$ the SPE payoff of coalition S in the induced subgame $G_S^t(n - |S| + 1, x(t), T)$, $t = 0, \dots, T-1$. Note that for $n = 2$ the values

$\gamma(\{j\}, t, x)$, $j = 1, 2$; $t = 0, \dots, T-1$ are given by (6), (8) and (10) in accordance to Prop. 1.

Now consider the fully cooperative solution when all the players cooperate to reach the maximal total payoff

$$H_N(\cdot) = \sum_{\tau=0}^{T-1} \delta^\tau \ln u(\tau) + K \cdot \delta^T \ln x(T), \tag{15}$$

where

$$u(\tau) = \sum_{j \in N} u_j(\tau), \quad K = \sum_{j \in N} K_j.$$

Again we suppose the log-linear form of the value function:

$$V(t, x) = A(t) \cdot \ln x + B(t), \quad t = 0, 1, \dots, T. \tag{16}$$

Proposition 2. *A multistage finite-horizon game $G^0(n = 2, x(0), T)$ possesses a cooperative solution*

$$u(x) = \frac{\alpha}{1 + \delta A(t+1)} \cdot x, \quad t = 0, \dots, T-1, \tag{17}$$

while coefficients $A(t)$ satisfy recurrence formula

$$A(t) = 1 + \delta A(t+1), \quad A(T) = K. \tag{18}$$

The cooperative state trajectory is

$$x(t+1) = \frac{\alpha \delta A(t+1)}{1 + \delta A(t+1)} \cdot x(t), \quad t = 0, \dots, T-1. \tag{19}$$

The value function (16) determines the cooperative payoff in the subgame $G^t(n = 2, x(t), T)$, $t = 0, \dots, T-1$, while coefficients $B(t)$ are given by the recurrence formula

$$B(t) = \ln \frac{\alpha}{1 + \delta A(t+1)} + \delta A(t+1) \ln \frac{\alpha \delta A(t+1)}{1 + \delta A(t+1)} + \delta B(t+1), \quad B(T) = 0. \tag{20}$$

The proof based on the dynamic-programming method is similar to the proof of Prop. 1.

Note that the explicit formulae for $A(t)$ and $B(t)$ similar to (12) and (13) could be provided, although recurrence formulae (18) and (20) are more convenient to calculate value functions (16).

Remark 3. By construction, the cooperative payoff in any subgame $G^t(n = 2, x(t), T)$, $t = 0, \dots, T-1$, is greater than or equal to the sum of players' SPE payoffs in this subgame.

We assume in the paper the transferable utility case, i.e., any payoff transfers between the players are allowed. Let $\bar{\omega} = (x(0) = \bar{x}(0), \dots, \bar{x}(t), \dots, \bar{x}(T))$ denote a cooperative trajectory (19) whereas $\bar{u}(\bar{x}(t))$ denote a cooperative total extraction level in period t which is determined by (17), (18).

A vector (p_1^t, \dots, p_n^t) such that

$$\sum_{i \in N} p_i^t = V(t, \bar{x}(t)) \tag{21}$$

specifies a possible sharing rule to distribute the total cooperative (subgame) payoff between the players and could be considered as a cooperative solution for the subgame $G^t(n, \bar{x}(t), T)$.

Definition 3. Vectors $\beta_i(\bar{\omega}) = (\beta_i(\bar{x}(\tau)))$, $\tau = 0, \dots, T$; $i = 1, \dots, n$ denote the Payoff Distribution Procedure (PDP) for cooperative solution (p_1^0, \dots, p_n^0) if

$$p_i^0 = \sum_{\tau=0}^T \delta^\tau \beta_i(\bar{x}(\tau)). \tag{22}$$

The PDP based approach firstly introduced in [25] for differential games implies that all the players have agreed to distribute the total cooperative payoff in $G^0(n, x(0), T)$ according to vector (p_1^0, \dots, p_n^0) and, in addition, to allocate each player’s cooperative payoff p_i^0 along the cooperative trajectory $\bar{\omega}$ in accordance with some payment schedule (namely, PDP β). Then, $\beta_i(\bar{x}(\tau))$ denotes the actual current payment that the i -th player should get at time τ when the players use PDP β under cooperative scenario.

We adopt in the paper the following assumptions about the players non-cooperative behavior if a cooperative agreement is broken down at some intermediate time constant $t = 0, \dots, T - 1$, because of some coalition S deviation from cooperative scenario (17):

- all the players $j \in N \setminus S$ form singletons and switch (immediately and forever) to non-cooperative (that is, SPE) behavior scheme in a subgame $G^t(n, \bar{x}(t), T)$ - see, e.g. [7, 10] for discussion;
- the maximal guaranteed payoff a coalition S could expect in $G^t(n, \bar{x}(t), T)$ in case of its deviation equals to $\gamma(S, t, \bar{x}(t))$ instead of $\sum_{\tau=t}^T \delta^{\tau-t} \beta_S(\bar{x}(\tau))$, where $\beta_S(\bar{x}(\tau)) = \sum_{j \in S} \beta_j(\bar{x}(\tau))$.

Definition 4. A PDP $\beta = (\beta_i(\bar{x}(\tau)))$, $i = 1, \dots, n$; $\tau = 0, \dots, T$ belongs to the β -Subgame-Perfect Core (β -S-P Core) of the multistage finite-horizon game $G^0(n, x(0), T)$ if for each nonempty coalition $S \subset N$ and each intermediate time instant $t = 0, \dots, T - 1$ the following inequality holds

$$\sum_{\tau=t}^T \delta^{\tau-t} \beta_S(\bar{x}(\tau)) \geq \gamma(S, t, \bar{x}(t)). \tag{23}$$

Inequality (23) means that no coalition $S \subset N$ has an incentive to deviate from cooperative agreement (i.e. cooperative strategies (17) and PDP β) at each subgame $G^t(n, \bar{x}(t), T)$, $t = 0, \dots, T - 1$ along the cooperative trajectory $\bar{\omega}$. Moreover, as it follows from (21), (22) and Prop. 2 constraint (23) is binding for $S = N$, and

$$\sum_{\tau=t}^T \delta^{\tau-t} \beta_N(\bar{x}(\tau)) = V(t, \bar{x}(t)) = A(t) \ln \bar{x}(t) + B(t), \tag{24}$$

where coefficients $A(t)$, $B(t)$ meet (18), (20).

Remark 3 implies that the following proposition holds (at least for two-person game).

Proposition 3. *β -Subgame-Perfect Core of a multistage finite-horizon game $G^0(2, x(0), T)$ is non-empty.*

Remark 4. The $n \times (T + 1)$ components $\beta_j(\bar{x}_\tau)$ of the PDP β from β -S-P Core have to satisfy a system of non-strict linear inequalities (23) and linear equations (22). Hence, a non-empty β -S-P Core for multistage finite-horizon game $G^0(n, x(0), T)$ is a convex closed polytope Δ in $R^{n \times (T+1)}$.

The next advantageous property ensures that PDP β could be implemented without any loans or credits since at each stage the players redistribute exactly what they have gained at this stage in accordance with the cooperative scenario (see, e.g., [13, 16, 26]).

Definition 5. *A payoff distribution procedure β satisfies the strict balance constraints if*

$$\sum_{j \in N} \beta_j(\bar{x}(\tau)) = \ln \bar{u}(\bar{x}(\tau)), \quad \tau = 0, \dots, T - 1; \quad \sum_{j \in N} \beta_j(\bar{x}(T)) = K \cdot \ln \bar{x}(T). \tag{25}$$

4 An Algorithm for Choosing Unique PDP from β -S-P Core

To choose a unique PDP β from the β -S-P Core one can adopt, for instance, maxmin relative benefit from cooperation (maxmin RBC) approach introduced in [17].

If we apply this approach to multistage finite-horizon game $G^0(n, x(0), T)$ and focus on the case when $\gamma(\{i\}, 0, \bar{x}(0)) > 0$, $i \in N$, we need to solve the following optimization problem

$$\max_{\beta \in \Delta} \min_{i \in N} \frac{p_i^0 - \gamma(\{i\}, 0, \bar{x}(0))}{\gamma(\{i\}, 0, \bar{x}(0))} \tag{26}$$

and then distribute each player i 's cooperative payoff $p_i^0 = \sum_{\tau=0}^T \delta^\tau \beta_i(\bar{x}(\tau))$ along the cooperative trajectory in such a way that PDP β meets (23) and (25). One can use the relative benefit from cooperation in (26) to measure the so-called Value of Cooperation (see, e.g., [8]).

Remark 5. Note that for two-player game problem (26) takes the following simple form

$$\frac{p_1^0 - \gamma(\{1\}, 0, \bar{x}(0))}{\gamma(\{1\}, 0, \bar{x}(0))} = \frac{p_2^0 - \gamma(\{2\}, 0, \bar{x}(0))}{\gamma(\{2\}, 0, \bar{x}(0))}. \tag{27}$$

Vector (p_1^0, p_2^0) implies that both players reach maximal (in a sense of (26)) and equal relative benefit from the cooperation.

To determine a distribution of the players' cooperative payoff along $\bar{\omega}$ in a 2-person game $G^0(2, x(0), T)$ let us specify the following algorithm.

Algorithm (quasi proportional PDP from β -S-P Core):

1. Using Prop. 2 find a cooperative trajectory $\bar{\omega} = (\bar{x}(0), \bar{x}(1), \dots, \bar{x}(T - 1), \bar{x}(T))$ and corresponding sequence of cooperative extraction levels $\bar{u}(\bar{x}(t)), t = 0, \dots, T - 1$.
2. Calculate $\gamma(\{j\}, t, \bar{x}(t)), t = 0, \dots, T - 1; j = 1, 2$ using (6), (8) and (10) in accordance with Prop. 1.
3. Solve (27) and (21) to obtain p_1^0 and p_2^0 .
4. Using strict balance constraints (25) and inequalities (23) write the system of double inequalities for $\sum_{\tau=t}^T \delta^{\tau-t} \beta_1(\bar{x}(\tau)), t = T - 1, T - 2, \dots, 1$ in the form:

$$\begin{cases} c_1^{T-1} \leq \beta_1(\bar{x}(T - 1)) + \delta \cdot \beta_1(\bar{x}(T)) \leq C_1^{T-1} \\ \vdots \\ c_1^t \leq \beta_1(\bar{x}(t)) + \delta \cdot \beta_1(\bar{x}(t + 1)) + \dots + \delta^{T-t} \cdot \beta_1(\bar{x}(T)) \leq C_1^t, \\ \vdots \\ c_1^1 \leq \sum_{\tau=1}^T \delta^{\tau-1} \cdot \beta_1(\bar{x}(\tau)) \leq C_1^1 \end{cases} \tag{28}$$

where $c_1^t \leq C_1^t$, for all $t = 1, \dots, T - 1$.

5. Denote by μ the first player's part of the total cooperative payoff $\frac{p_1^0}{p_1^0 + p_2^0}$. Then $\frac{p_2^0}{p_1^0 + p_2^0} = 1 - \mu$. Accept $\beta_1(\bar{x}(T)) = \mu \cdot K \cdot \ln \bar{x}(T)$.
6. Solve (28) in series assuming that in each subgame $G^t(2, \bar{x}(t), T), t = T - 1, T - 2, \dots, 1$, player 1 receives exactly part μ of the admissible range $(C_1^t - c_1^t)$ of the subgame payment. Namely,

$$\begin{cases} \beta_1(\bar{x}(T - 1)) = c_1^{T-1} + \mu(C_1^{T-1} - c_1^{T-1}) - \delta \beta_1(\bar{x}(T)) \\ \vdots \\ \beta_1(\bar{x}(1)) = c_1^1 + \mu(C_1^1 - c_1^1) - \sum_{\tau=2}^T \delta^{\tau-1} \beta_1(\bar{x}(\tau)) \end{cases} \tag{29}$$

7. Take

$$\beta_1(\bar{x}(0)) = p_1^0 - \sum_{\tau=1}^T \delta^\tau \beta_1(\bar{x}(\tau)). \tag{30}$$

8. Calculate $\beta_2(\bar{x}(t)), t = 0, \dots, T$, from the strict balance constraints (25).

Remark 6. Payoff distribution procedure β specified above satisfies the following properties:

- it belongs to the β -S-P Core of multistage game $G^0(2, \bar{x}(0), T)$;
- the resulting cooperative solution (p_1^0, p_2^0) maximizes the relative benefit from cooperation (26) of the least winning player;
- it meets the strict balance constraints (25);
- PDP β implements a reasonable and subgame-consistent sharing rule in a sense that in each intermediate state $\bar{x}(t), t = 1, \dots, T$, the first player receives the same share of the current range $(C_1^t - c_1^t)$ of the admissible subgame $G^t(2, \bar{x}(t), T)$ payment $\sum_{\tau=2}^T \delta^{\tau-1} \beta_1(\bar{x}(\tau))$ as she/he is expected to obtain in the whole game $G^0(2, \bar{x}(0), T)$ in accordance with cooperative solution (p_1^0, p_2^0) .

5 Numerical Example

To demonstrate the above theoretical results with a simple numerical example let us consider a two-player multistage game of renewable resource extraction with the following parameters values: $T = 2$ ($t = 0, 1, 2$), $\alpha = 1.5$, $\delta = 0.95$, $K_1 = 1$, $K_2 = 0.5$, $K = K_1 + K_2 = 1.5$. The SPE strategies of the players are given by formulae (7) and (8). Cooperative strategy \bar{u} is defined by (17) and (18). All strategies linearly depend on the initial state x_0 (see Table 1). The relative values of these strategies at time instants $t = 0$ and $t = 1$ (current extraction levels divided by \bar{x}_0) are presented in Fig. 1 and connected by a dashed lines for visual clarity. The state trajectory generated by the subgame perfect equilibrium and the cooperative state trajectory are calculated using formulae (9) and (19), respectively. The results (current values of the resource divided by x_0) are given in Table 2 and presented in Fig. 2.

Table 1. SPE strategies and cooperative strategy

t	u_1^{SPE}	u_2^{SPE}	u^{Coop}
0	$0.3593 \cdot x_0$	$0.475 \cdot x_0$	$0.454 \cdot x_0$
1	$0.2528 \cdot x_0$	$0.5056 \cdot x_0$	$0.647 \cdot x_0$

To compare the sum of the players' SPE payoffs (6) and the cooperative payoff (16) in the whole game $G^0(2, \bar{x}(0), T)$ and in the subgames $G^t(2, \bar{x}(t), T), t = 1, 2$, along the cooperative trajectory we fix initial state $x_0 = e^{1.5} \approx 4.4817$. The results are presented in Table 3.

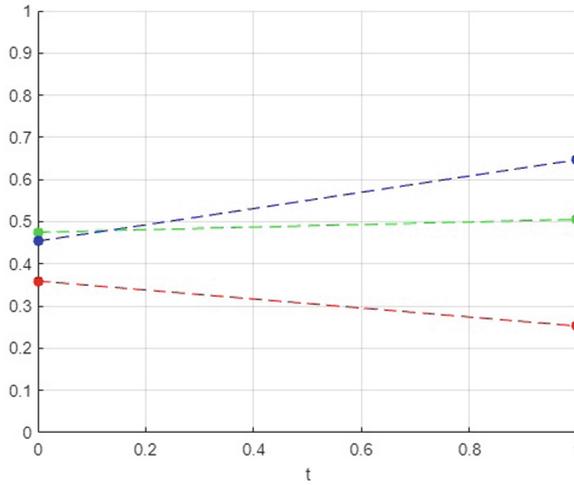


Fig. 1. SPE strategies – for the first player (red), for the second player (green), cooperative strategy (blue). (Color figure online)

Table 2. SPE trajectory and cooperative trajectory

t	x^{SPE}/x_0	x^{Coop}/x_0
0	1	1
1	0.6656	1.046
2	0.2401	0.922

Table 3. Sum of the players’ SPE payoffs versus cooperative payoff along cooperative trajectory

t	$V_1(t, \bar{x}(t)) + V_2(t, \bar{x}(t))$	$V(t, \bar{x}(t))$
0	2.227	3.642
1	2.595	3.086
2	2.128	2.128

Following the algorithm and using (21) and (27) we receive conditions on p_1^0, p_2^0 :

$$\frac{p_1^0 - 0.66}{0.66} = \frac{p_2^0 - 1.56}{1.56}, \quad p_1^0 + p_2^0 = 3.64,$$

from where we get $p_1^0 = 1.08, p_2^0 = 2.56$. Then, system of inequalities (28) takes the following form:

$$1.076 \leq \beta_1(\bar{x}(1)) + 0.95 \cdot \beta_1(\bar{x}(2)) \leq 1.567.$$

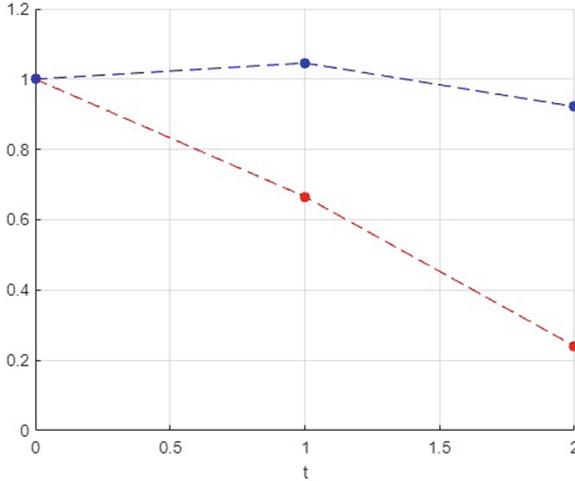


Fig. 2. SPE (red) and cooperative (blue) state trajectories. (Color figure online)

Coefficient μ is equal to 0,297 for this game. Further, using (29), (30) and the strict balance constraints (25) we obtain quasi proportional PDP from β -S-P Core:

	$\beta_i(\bar{x}(0))$	$\beta_i(\bar{x}(1))$	$\beta_i(\bar{x}(2))$
$i = 1$	-0.079	0.621	0.632
$i = 2$	0.789	0.443	1.496

Note that a negative payment to some player in accordance with (30) can only arise in the initial state of a multistage game (when the players just enter into a cooperative agreement).

6 Concluding Remarks

It is worth noting that if we multiply both sides of inequality (23) by δ^t and then add $\sum_{\tau=0}^{t-1} \delta^\tau \beta_S(\bar{x}(\tau))$, the LHS of the resulting inequality represents the total payoff of coalition S (estimated at the initial time instant $t = 0$ under assumption that all the players will use cooperative strategies and implement PDP β throughout the game evolution). Whereas the RHS is an estimation of the coalition S payoff that corresponds to the specific combined type of the players' behavior (namely, all the players cooperate from the beginning till some intermediate time instant t and then switch to non-cooperative mode in the induced subgame $G_S^t(n - |S| + 1, \bar{x}(t), T)$). Hence, inequalities (23) in the β -S-P Core

definition for multistage games could be considered a condition for subgame consistency (see, e.g., [14, 16, 26, 32]) of cooperative agreement as well as of this agreement implementation process via PDP β .

A novel quasi proportional PDP introduced in the paper always belongs to the β -S-P Core and has several good properties. However, it is surely of interest to consider other approaches for the β -S-P Core refinement as well as to study and compare properties of these cooperative solutions. An open question is whether the β -S-P Core concept could be adapted to the analysis of formalized dynamic models of ideological controversy and conflicts.

References

1. Breton, M., Dahmouni, I., Zaccour, G.: Equilibria in a two-species fishery. *Math. Biosci.* **309**, 78–91 (2019)
2. Breton, M., Keoula, M.Y.: A great fish war model with asymmetric players. *Ecol. Econ.* **97**, 209–223 (2014)
3. Breton, M., Keoula, M.Y.: Farsightedness in a coalitional great fish war. *Environ. Resour. Econ.* **51**, 297–315 (2012). <https://doi.org/10.1007/s10640-011-9501-y>
4. Bulgakova, M.A., Petrosyan, L.A.: Multistage games with pairwise interactions on complete graph. *Autom. Remote Control* **81**(8), 1519–1530 (2020). <https://doi.org/10.1134/S0005117920080135>
5. Cabo, F., Tidball, M.: Cooperation in a dynamic setting with asymmetric environmental valuation and responsibility. *Dyn. Games Appl.* (2021). <https://doi.org/10.1007/s13235-021-00395-y>
6. Chander, P.: Subgame-perfect cooperative agreements in a dynamic game of climate change. *J. Environ. Econ. Manag.* **84**, 173–188 (2017)
7. Chander, P., Wooders, M.: Subgame-perfect cooperation in an extensive game. *J. Econ. Theory* **187**, 105017 (2020)
8. Chebotareva, A., Shimai, S., Tretyakova, S., Gromova, E.: On the value of the preexisting knowledge in an optimal control of pollution emissions. *Contrib. Game Theory Manag.* **14**, 49–58 (2021)
9. Crettez, B., Hayek, N., Zaccour, G.: Do charities spend more on their social programs when they cooperate than when they compete? *Eur. J. Oper. Res.* **283**, 1055–1063 (2020)
10. Gromova, E., Marova, E., Gromov, D.: A substitute for the classical Neumann–Morgenstern characteristic function in cooperative differential games. *J. Dyn. Games* **7**(2), 105–122 (2020). <https://doi.org/10.3934/jdg.2020007>
11. Haurie, A., Krawczyk, J.B., Zaccour, G.: *Games Dyn. Games*. Scientific World, Singapore (2012)
12. Kaitala, V.T., Lindroos, M.: Game theoretic applications to fisheries. In: Weintraub, A., Romero, C., Bjørndal, T., Epstein, R., Miranda, J. (eds.) *Handbook of Operations Research in Natural Resources*. International Series In Operations Research & Mana, vol. 99. Springer, Boston (2007). https://doi.org/10.1007/978-0-387-71815-6_11
13. Kuzytin, D., Nikitina, M.: An irrational behavior proof condition for multistage multicriteria games. In: *Construtive Nonsmooth Analysis and Related Topics (Dedic. to the Memory of V.F.Demyanov)*, CNSA 2017, pp. 178–181. IEEE, New York (2017)

14. Kuzyutin, D., Gromova, E., Smirnova, N.: On the cooperative behavior in multistage multicriteria game with chance moves. In: Kononov, A., Khachay, M., Kalyagin, V.A., Pardalos, P. (eds.) *MOTOR 2020*. LNCS, vol. 12095, pp. 184–199. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49988-4_13
15. Kuzyutin, D., Lipko, I., Pankratova, Y., Tantlevskij, I.: Cooperation enforcing in multistage multicriteria game: new algorithm and its implementation. In: Petrosyan, L., Mazalov, V., Zenkevich, N. (eds.) *Frontiers of Dynamic Games. Static & Dynamic Game Theory: Foundations & Applications*. Birkhäuser, Cham (2020). https://doi.org/10.1007/978-3-030-51941-4_10
16. Kuzyutin, D., Smirnova, N.: Subgame consistent cooperative behavior in an extensive form game with chance moves. *Mathematics* **8**(7), 1061 (2020). <https://doi.org/10.3390/math8071061>
17. Kuzyutin, D., Smirnova, N., Skorodumova, Y.: Implementation of subgame-perfect cooperative agreement in an extensive-form game. In: Petrosyan, L.A., Zenkevich, N.A. (eds.) *Contributions to Game Theory and Management*, pp. 257–272. St. Petersburg State University, St. Petersburg (2021). <http://hdl.handle.net/11701/33701>
18. Levhari, D., Mirman, L.J.: The great fish war: an example using a dynamic Cournot-Nash solution. *Bell J. Econ.* **11**(1), 322–334 (1980)
19. Masoudi, N., Zaccour, G.: Adapting to climate change: is cooperation good for the environment? *Econo. Lett.* **153**, 1–5 (2017). <https://doi.org/10.1016/j.econlet.2017.01.018>
20. Mazalov, V., Parilina, E., Zhou, J.: Altruistic-like equilibrium in a differential game of renewable resource extraction. In: Pardalos, P., Khachay, M., Kazakov, A. (eds.) *MOTOR 2021*, LNCS, vol. 12755, pp. 326–339. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77876-7_22
21. Mazalov, V.V., Rettiyeva, A.N.: The discrete-time bioresource sharing model. *J. Appl. Math. Mech.* **75**, 180–188 (2011)
22. Mazalov, V. V., Rettieva, A. N.: Cooperation maintenance in fishery problems. In: *Fishery Management*, pp. 151–198. Nova Science Publishers (2012)
23. Mazalov, V.V., Rettieva, A.N.: Asymmetry in a cooperative bioresource management problem. In: *Game-Theoretic Models in Mathematical Ecology*, pp. 113–152. Nova Science Publishers (2015)
24. Nash, J.F.: Equilibrium points in n -person games. *Proc. Natl. Acad. Sci. USA* **36**, 48–49 (1950)
25. Petrosyan, L.A., Danilov, N.N.: Stability of solutions in non-zero sum differential games with transferable payoffs. *Astronomy* **1**, 52–59 (1979)
26. Petrosyan, L., Kuzyutin, D.: *Games in Extensive Form: Optimality and Stability*. Saint Petersburg University Press, St. Petersburg (2000). (in Russian)
27. Petrosian, O., Zakharov, V.: IDP-core: novel cooperative solution for differential games. *Mathematics* **8**, 721 (2020)
28. Pintassilgo, P., Finus, M., Lindroos, M., Munro, G.R.: Stability and success of regional fisheries management organizations. *Environ. Resour. Econ.* **46**, 377–402 (2010)
29. Rettieva, A.N.: A discrete-time bioresource management problem with asymmetric players. *Autom. Remote Control* **75**(9), 1665–1676 (2014). <https://doi.org/10.1134/S0005117914090124>
30. Rettieva, A.: Cooperation in bioresource management problems. In: Petrosyan, L.A., Zenkevich, N.A. (eds.) *Contributions to Game Theory and Management*, vol. 10, pp. 245–286. St. Petersburg State University, St. Petersburg (2017)

31. Selten, R.: Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games. *Int. J. Game Theory* **4**, 25–55 (1975)
32. Yeung, D., Petrosyan, L.: *Subgame Consistent Economic Optimization: An Advanced Cooperative Dynamic Game Analysis*. Springer (2012)



Two Level Cooperation in Dynamic Network Games with Partner Sets

Leon Petrosyan  and Yaroslavna Pankratova ^(✉) 

St Petersburg State University, Saint-Petersburg, Russia
{l.petrosyan,y.pankratova}@spbu.ru

<http://www.apmath.spb.ru/en/staff/petrosjan/index.html>

Abstract. In the presented paper, we consider dynamic network games with partner sets in which players cooperate to get the best outcomes. Using the game structure the two-level cooperative scheme is introduced. On the first level, the partner sets are considered as players, and the cooperative behavior is used in the game with partner sets, it is assumed that partners intend to maximize their joint payoff and then distribute it using a given optimality principle as usual in cooperative game theory. On the second level, the gain obtained by each player (partner set) is distributed among members of this partner set. The distribution of this gain is also made based on solution concepts from classical cooperative game theory. Since the game is dynamic the problem of time-consistency (dynamic stability) of the proposed two-level solution arises. To simplify the calculations the new characteristic function is introduced based on the possibility of cutting connections by players outside the coalition. Also, this newly defined characteristic function allows construction of time-consistent (dynamically stable) solutions.

Keywords: Dynamic network game · Partner set · Shapley value

1 Introduction

Dynamic network games with partner sets in which players cooperate to get the best outcomes is a topic of ongoing research (see Cao et al. (1963) [2], Pai (2010) [7], Zhang et al. (2018) [18], Meza and Lopez-Barrrientos (2016) [5], Bulgakova, Petrosyan (2019) [1]). Cooperation in dynamic network games and different solutions of cooperative dynamic network games are also considered in papers Petrosyan (2010) [9], Gao and Pankratova (2017) [3], and the papers of Petrosyan and Yeung (2016), (2020) [15,17] where the new characteristic function in differential cooperative network game was introduced in a special case when the payoffs of players depend only upon their actions and actions of neighbors in the network. Different properties of the cooperative solutions of dynamic network games are investigated in [13,14,16]. In the paper [19], the differential games on networks with partner sets are considered. In such games,

Supported by the Russian Science Foundation grant No. 22-11-00051, <https://rscf.ru/en/project/22-11-00051/>.

player’s payoff depend upon the payoffs of players from his partner set. It is supposed that one player can belong to many partner sets.

In this paper, we consider the differential network game with a new cooperation structure, namely the two-level cooperative scheme. On the first level, the partner sets are considered as players. It is assumed that partners intend to maximize their joint payoff and then distribute it using a given optimality principle. On the second level, the gain obtained by each player (partner set) is distributed among members of this partner set. We find the Shapley value on both levels of the game. Based on this the new solution concept is introduced.

2 Class of Differential Network Games

Consider a class of n -person differential games on network with game horizon $[t_0, T]$. The players are connected in a network system. We use $N = \{1, 2, \dots, n\}$ to denote the set of players in the network. The nodes of the network are used to represent the players from the set N . We also denote the set of nodes by N and denote the set of all arcs in network N by L . The arcs in L are the $arc(i, j) \in L$ for players $i, j \in N, i \neq j$. For notational convenience, we denote the set of players connected to player i as $\tilde{K}(i) = \{j : arc(i, j) \in L\}$, for $i \in N$.

We suppose also that a family of subsets $M_1, \dots, M_k, \dots, M_l, k = 1, \dots, l, M_l \cap M_r = \emptyset, r \neq l$ of the set N is given. It is supposed that $|M_k| \geq 2$, and for all $i \in N$ there exist $l \in N$, such that $i \in M_l$. Also for each two nodes $z_1 \in M_k, z_2 \in M_k$ there exist a path connecting z_1 and z_2 in M_k . The sets $M_1, \dots, M_k, \dots, M_l$ are called “partner” sets.

Let $x^i(\tau) \in R^m$ be the state variable of player $i \in N$ at time τ , and $u^i(\tau) \in U^i \subset R^k$ the control variable of player $i \in N$.

Every player $i \in N$ can cut the connection with any other player from the set M_k at any instant of time.

The state dynamics of the game is

$$\dot{x}^i(\tau) = f^i(x^i(\tau), u^i(\tau)), \quad x^i(t_0) = x_0^i, \quad \text{for } \tau \in [t_0, T] \text{ and } i \in N. \tag{1}$$

The function $f^i(x^i, u^i)$ is continuously differentiable in x^i and u^i .

The payoff function of player i depends upon his state variable, his own control variable and the state variables of players from the sets $\tilde{K}(i)$.

In particular, the payoff of player i is given as

$$H_i(x_0^1, \dots, x_0^n, u^1, \dots, u^n) = \sum_{j \in \tilde{K}(i)} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau. \tag{2}$$

The term $h_i^j(x^i(\tau), x^j(\tau))$ is the instantaneous gain that player i can obtain through network links with player $j \in \tilde{K}(i)$ (note that the pair $(i, i) \notin L$).

The functions $h_i^j(x^i(\tau), x^j(\tau))$, for $j \in M_k$ are non-negative. For notational convenience, we use $x(t)$ to denote the vector $(x^1(t), x^2(t), \dots, x^n(t))$.

Since the set N is finite the sum in (2) contains a finite number of summands $\leq |N|$.

2.1 Game with Players (Partner Sets) $M = (M_1, \dots, M_k, \dots, M_l)$

In this section, we consider the game between players $M_k \subset N$ (subsets of players from the player set N , or partner sets), where $M_k \cap M_j = \emptyset, k \neq j$ and $\cup_{k=1}^l M_k = N$.

Consider the cooperative version of this game. The control variable of player $M_k, k = 1, \dots, l$ is defined as vector $u_i(t) = (u_i(t), i \in M_k)$, and the state variable of M_k is defined as

$$x_k(t) = (x^i(t), i \in M_k). \tag{3}$$

The payoff function of M_k is given as

$$\begin{aligned} H^k(x_0^1, \dots, x_0^n, u^1, \dots, u^n) &= \sum_{i \in M_k} H_i(x_0^1, \dots, x_0^n, u^1, \dots, u^n) \\ &= \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i)} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau \right) \end{aligned}$$

To achieve group optimality, the players maximize their joint payoff

$$\sum_{k=1}^l \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i)} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau \right) \tag{4}$$

subject to dynamics (1).

Denote by $\bar{x}(t) = (\bar{x}^1(t), \bar{x}^2(t), \dots, \bar{x}^n(t))$ and by $\bar{u}(t) = (\bar{u}^1(t), \bar{u}^2(t), \dots, \bar{u}^n(t))$ the optimal cooperative trajectory and the optimal cooperative control in the problem of maximization (4) subject to (1). The maximized joint cooperative payoff $V(x_0, t_0, N)$ involving all players can then be expressed as

$$\begin{aligned} &\sum_{k=1}^l \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i)} \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right) \\ &= \max_{u^1, u^2, \dots, u^n} \sum_{k=1}^l \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i)} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau \right) \end{aligned} \tag{5}$$

subject to dynamics (1).

Next, we consider distributing the cooperative payoff to the participating partner sets $(M_1, \dots, M_k, \dots, M_l)$ under an agreeable scheme. Given that the contributions of an individual player to the joint payoff through linked players can be diverse, the Shapley value (1953) [11] provides one of the best solutions in attributing a fair gain to each player in a complex network. One of the contentious issues in using the Shapley value is the determination of the worth of subsets of players (characteristic function).

In this section, we present a new formulation of the worth of coalition $S \subset M = \{M_1, \dots, M_k, \dots, M_l\}$. Let $L = \{1, \dots, k, \dots, l\}$ and $S \subset L$. In computing the values of characteristic function for coalitions, we evaluate contributions of the players in the process of cooperation and maintain the cooperative strategies for all players along the cooperative trajectory. In particular, we evaluate the worth of the coalitions $(S \subset L)$ along the cooperative trajectory as

$$V(S; x_0, T - t_0) = \sum_{k \in S} \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i) \cap (\cup_{k \in S} M_k)} \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right). \quad (6)$$

Note that the worth of coalition S is measured by the sum of payoffs of the players M_k in the coalition in the cooperation process with the exclusion of the gains from players outside coalition S . Thus, the characteristic function reflecting the worth of coalition S in (6) is formulated along the cooperative trajectory $\bar{x}(t)$.

Similarly, the characteristic function at time $t \in [t_0, T]$ can be evaluated as

$$V(S; \bar{x}(t), T - t) = \sum_{k \in S} \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i) \cap (\cup_{k \in S} M_k)} \int_t^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right). \quad (7)$$

Proposition 1. *The characteristic function defined by (6) and (7) is convex.*

The proof is similar to one in [19,20] This also means that the core of the game is not empty and the Shapley value belongs to the core.

From (6), (7) we get

$$\begin{aligned} V(S; x_0, T - t_0) &= \sum_{k \in S} \sum_{i \in M_k} \sum_{\tilde{K}(i) \cap (\cup_{k \in S} M_k)} \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \\ &\quad + \sum_{k \in S} \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i) \cap (\cup_{k \in S} M_k)} \int_t^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right) \\ &= \sum_{k \in S} \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i) \cap (\cup_{k \in S} M_k)} \int_{t_0}^t h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right) + V(S; \bar{x}(t), T - t) \end{aligned} \quad (8)$$

The Eq. (8) can be interpreted as time-consistency property of introduced characteristic function.

In our case, the worth of coalitions is measured under the process of cooperation instead of under min-max confrontation or Nash non-cooperative stance. And, any individual player or coalition attempting to act independently will have the links to other players in the network being cut off.

Because of this players outside S in worst case will cut connection with players from S , and players from S will get positive payoffs only interacting with other players from S .

3 Dynamic Shapley Value

In this section, we develop a dynamic Shapley value imputation using the defined characteristic function.

Now, we consider allocating the grand coalition cooperative network gain to individual players according to the Shapley value imputation. Player i 's payoff under cooperation would become

$$Sh_i(x_0, T - t_0) = \sum_{\substack{S \subset N, \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!} \times [V(S; x_0, T - t_0) - V(S \setminus \{i\}; x_0, T - t_0)], \tag{9}$$

for $i \in N$.

Invoking (6), in our case, we can obtain the cooperative payoff of player $i \in L = \{1, \dots, k, \dots, l\}$ under the Shapley value as

$$Sh_i(x_0, T - t_0) = \sum_{\substack{S \subset L \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!} \times \left\{ \sum_{m \in S} \sum_{k \in M_m} \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S} M_m)} \int_{t_0}^T h_k^j(\bar{x}^k(\tau), \bar{x}^j(\tau)) d\tau \right) - \sum_{m \in S \setminus \{i\}} \sum_{k \in M_m} \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S \setminus \{i\}} M_m)} \int_{t_0}^T h_k^j(\bar{x}^k(\tau), \bar{x}^j(\tau)) d\tau \right) \right\}, \quad S \subset L. \tag{10}$$

However, in a dynamic framework, the agreed upon optimality principle for sharing the gain has to be maintained throughout the cooperation duration (see Yeung and Petrosyan (2004 and 2016) [14, 15]) for a dynamically consistent solution. Applying the Shapley value imputation in (11) to any time instance $t \in [t_0, T]$, we obtain:

$$Sh_i(\bar{x}(t), T - t) = \sum_{\substack{S \subset L \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!}$$

$$\begin{aligned}
 & \times \left\{ \sum_{m \in S} \sum_{k \in M_m} \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S} M_m)} \int_t^T h_k^j(\bar{x}^k(\tau), \bar{x}^j(\tau)) d\tau \right) \right. \\
 & \left. - \sum_{m \in S \setminus \{i\}} \sum_{k \in M_m} \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S \setminus \{i\}} M_m)} \int_t^T h_k^j(\bar{x}^k(\tau), \bar{x}^j(\tau)) d\tau \right) \right\} \quad (11)
 \end{aligned}$$

The Shapley value imputation in (10)–(11) is based on characteristic function evaluates along the optimal cooperative trajectory, and it attributes the contributions of the players under the optimal cooperation process. Indeed, it can be regarded as optimal trajectory dynamic Shapley value. In addition, this Shapley value imputation (10)–(11) fulfils the property of time consistency.

Proposition 2. *The Shapley value imputation in (10)–(11) satisfies the time consistency property.*

Proof. By direct computation we get.

$$\begin{aligned}
 Sh_i(x_0, T - t_0) &= \sum_{\substack{S \subseteq L \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!} \\
 & \times \left\{ \sum_{m \in S} \sum_{k \in M_m} \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S} M_m)} \int_{t_0}^t h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right) \right. \\
 & \left. - \sum_{l \in S \setminus \{i\}} \sum_{k=1}^l \left(\sum_{j \in \tilde{K}(k) \cap (\cup_{m \in S \setminus \{i\}} M_m)} \int_{t_0}^t h_k^j(\bar{x}^k(\tau), \bar{x}^j(\tau)) d\tau \right) \right\} \\
 & + Sh_i(\bar{x}(t), T - t) = i \in N,
 \end{aligned}$$

which exhibits the time consistency property of the Shapley value imputation $Sh_i(\bar{x}(t), T - t)$, for $t \in [t_0, T]$.

We see that a Shapley value measure itself in a dynamic framework fulfils the property of time consistency (see existing dynamic Shapley value measures which do not share this property in Gromova (2016) [4], Petrosyan and Zaccour (2003) [10], Yeung (2010) [13], Yeung and Petrosyan (2016 and 2018) [15, 16]).

4 Game Inside the Partner Set $M_k, k = 1, \dots, l$

Consider now cooperative game with players $i \in M_k$ belonging to one partner set M_k . In this game, it is supposed that player $i \in M_k$ interacts only with players $j \in M_k \setminus \{i\}$.

Thus the payoff function of player $i \in M_k$ has the form

$$\sum_{j \in \bar{K}(i) \cap M_k} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau,$$

where $x^i(\tau), x^j(\tau)$ are solutions of (1).

In this case, the cooperative behaviour of players from M_k means the maximization of the sum

$$\sum_{i \in M_k} \sum_{j \in \bar{K}(i) \cap M_k} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau$$

Denote by $\bar{x}(t) = (\bar{x}^i(t), i \in M_k)$ the corresponding optimal cooperative trajectory

$$\begin{aligned} & \sum_{i \in M_k} \sum_{j \in \bar{K}(i) \cap M_k} \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \\ &= \max_{u^i, i \in M_k} \sum_{i \in M_k} \sum_{j \in \bar{K}(i) \cap M_k} \int_{t_0}^T h_i^j(x^i(\tau), x^j(\tau)) d\tau = W(M_k, T - t_0) \end{aligned}$$

subject to dynamic (1).

Next, we consider distributing the cooperative payoff to the participating players $i \in M_k$ under an agreeable scheme. Given that the contributions of an individual player to the joint payoff through linked players can be diverse, the Shapley value provides one of the best solutions in attributing a fair gain to each player in a complex network. One of the contentious issues in using the Shapley value is the determination of the worth of subsets of players (characteristic function). We shall use the Shapley value as in previous case.

In computing the values of characteristic function for coalitions, before we evaluate contributions of the players in the process of cooperation and maintain the cooperative strategies for all players along the cooperative trajectory. In particular, we evaluate the worth of the coalitions $S \subset M_k$ along the cooperative trajectory as

$$V^{M_k}(S; x_0, T - t_0) = \sum_{i \in S} \left(\sum_{j \in \bar{K}(i) \cap S} \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right). \tag{12}$$

Note that the worth of coalition S is measured by the sum of players payoffs from the set $S \subset M_k$ in the cooperation process with the exclusion of gains from players outside coalition S . Thus, the characteristic function reflecting the worth of coalition S in (12) is formulated along the cooperative trajectory $\bar{x}(t)$.

Similarly, the characteristic function at time $t \in [t_0, T]$ can be evaluated as

$$V^{M_k}(S; \bar{x}(t), T - t) = \sum_{i \in S} \left(\sum_{j \in \bar{K}(i) \cap S} \int_t^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right). \tag{13}$$

An important property of the above characteristic function as a measure of the worth of coalition in the Shapley value is given below.

Proposition 3. *The characteristic function defined by (12) and (13) is convex.*

From (12), (13) we get

$$\begin{aligned}
 V^{M_k}(S; x_0, T - t_0) &= \sum_{i \in S} \sum_{i \in \bar{K}(i) \cap S} \int_{t_0}^t h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \\
 &\quad + \sum_{i \in S} \left(\sum_{j \in \bar{K}(i) \cap S} \int_t^T h_i^j(\bar{x}(\tau), \bar{x}^j(\tau)) d\tau \right) \\
 &= \sum_{i \in S} \left(\sum_{j \in \bar{K}(i) \cap S} \int_{t_0}^t h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau)) d\tau \right) + V^{M_k}(S; \bar{x}(t), T - t) \quad (14)
 \end{aligned}$$

The Eq. (14) can be interpreted as time-consistency property of introduced characteristic function.

As before in our case the worth of coalitions is measured under the process of cooperation instead of under min-max confrontation or Nash non-cooperative stance. Players outside S in worst case will cut connection with players from S , and players from S will get positive payoffs only interacting with other players from S .

5 Dynamic Shapley Value in Game Inside the Partner Set M_k , $k = 1, \dots, l$

In this section, we develop a dynamic Shapley value imputation using the defined characteristic function.

Now, we consider allocating the grand coalition cooperative network gain $V(N; x_0, T - t_0)$ to individual players according to the Shapley value imputation. Player i 's payoff under cooperation would become

$$\begin{aligned}
 Sh_i^{M_k}(x_0, T - t_0) &= \sum_{\substack{S \subset M_k, \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!} \\
 &\times [V^{M_k}(S; x_0, T - t_0) - V^{M_k}(S \setminus \{i\}; x_0, T - t_0)], \quad (15)
 \end{aligned}$$

for $i \in N$.

Invoking (14), in our case, we can obtain the cooperative payoff of player i under the Shapley value as

$$Sh_i^{M_k}(x_0, T - t_0) = \sum_{\substack{S \subset M_k, \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!}$$

$$\begin{aligned} & \times \left\{ \sum_{m \in S} \left(\sum_{j \in \bar{K}(m) \cap S} \int_{t_0}^T h_m^j(\bar{x}^m(\tau), \bar{x}^j(\tau)) d\tau \right) \right. \\ & \left. - \sum_{m \in S \setminus \{i\}} \left(\sum_{j \in \bar{K}(m) \cap (S \setminus \{i\})} \int_{t_0}^T h_m^j(\bar{x}^m(\tau), \bar{x}^j(\tau)) d\tau \right) \right\}. \end{aligned} \tag{16}$$

However, in a dynamic framework, the agreed upon optimality principle for sharing the gain has to be maintained throughout the cooperation duration (see Yeung and Petrosyan (2004 and 2016) [14, 15]) for a dynamically consistent solution. Applying the Shapley value imputation in (17) to any time instance $t \in [t_0, T]$, we obtain:

$$\begin{aligned} Sh_i^{M_k}(\bar{x}(t), T - t) &= \sum_{\substack{S \subset M_k \\ S \ni i}} \frac{(|S| - 1)!(n - |S|)!}{n!} \\ & \times \left\{ \sum_{m \in S} \left(\sum_{j \in \bar{K}(m) \cap S} \int_t^T h_m^j(\bar{x}^m(\tau), \bar{x}^j(\tau)) d\tau \right) \right. \\ & \left. - \sum_{m \in S \setminus \{i\}} \left(\sum_{j \in \bar{K}(m) \cap (S \setminus \{i\})} \int_t^T h_m^j(\bar{x}^m(\tau), \bar{x}^j(\tau)) d\tau \right) \right\} \end{aligned} \tag{17}$$

6 The Solution Under the Two Level Cooperation

Let $Sh_k(x_0, T - t_0)$ defined by (11) be the amount of the joint gain of partner sets $\{M_1, \dots, M_l\}$ given to the partner set (player on the first level) M_k . And let $Sh_i^{M_k}$ be the amount given to the player $i \in M_k$ under the cooperation when players from M_k play independent of players from other partner sets.

Introduce the following notation

$$\beta_{ik} = \frac{Sh_i^{M_k}(x_0, T - t_0)}{\sum_{i \in M_k} Sh_i^{M_k}(x_0, T - t_0)}. \tag{18}$$

Let the amount prescribed to player $i \in N \cap M_k$ under two level cooperation be equal to

$$\gamma_i(x_0, T - t_0) = \beta_{ik} Sh_k(x_0, T - t_0). \tag{19}$$

Similarly, the amount prescribed to player $i \in N \cap M_k$ at time t under two level cooperation be equal to

$$\gamma_i(\bar{x}(t), T - t) = \beta_{ik} Sh_k(\bar{x}(t), T - t).$$

If we keep β_{ik} independent from $t \in [t_0, T]$ the proposed solution concept will be time-consistent as $Sh = \{Sh_k\}$, $k = 1, \dots, l$ is time-consistent solution in the game between partner sets.

7 Example

For simplicity in notation, we denote the gain that player i can obtain through the network link with player $j \in \tilde{K}(i)$ as

$$\alpha_{ij}(x_0, T - t_0) = \int_{t_0}^T h_i^j(\bar{x}^i(\tau), \bar{x}^j(\tau))d\tau, \tag{20}$$

$$\alpha_{ij}^k(x_0, T - t_0) = \int_{t_0}^T h_i^j(\bar{\bar{x}}^i(\tau), \bar{\bar{x}}^j(\tau))d\tau. \tag{21}$$

Using (20), (21) we can rewrite the expression of characteristic functions in the following form

$$V(S; x_0, T - t_0) = \sum_{k \in S} \sum_{i \in M_k} \left(\sum_{j \in \tilde{K}(i) \cap (\cup_{k \in S} M_k)} \alpha_{ij}(x_0, T - t_0) \right), \tag{22}$$

$$V^{M_k}(S; x_0, T - t_0) = \sum_{i \in S} \left(\sum_{j \in \tilde{K}(i) \cap S} \alpha_{ij}^k(x_0, T - t_0) \right). \tag{23}$$

Introduce additional notations before considering the example.

$$\alpha_{ij}(x_0, T - t_0) + \alpha_{ji}(x_0, T - t_0) = \alpha_{ij} + \alpha_{ji} = A_{ij} = A_{ji} = A(i, j)$$

and

$$\alpha_{ij}^k(x_0, T - t_0) + \alpha_{ji}^k(x_0, T - t_0) = \alpha_{ij}^k + \alpha_{ji}^k = A_{ij}^k = A_{ji}^k = A^k(i, j)$$

Example. Consider the following 5 player network game (see Fig. 1). First, consider the cooperative game between partner sets $M_1 = \{1, 5\}$, $M_2 = \{2, 3, 4\}$, and construct the characteristic function in this 2-player game using (6) and (20). We get

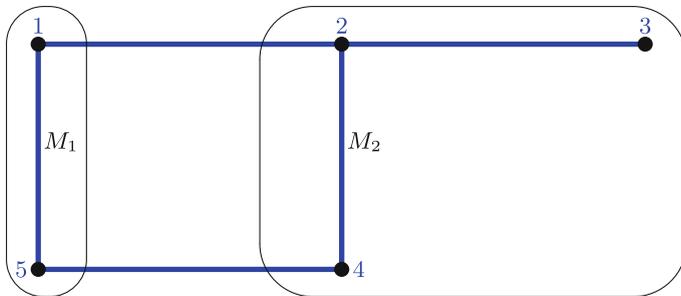


Fig. 1. 5 player network game

$$\begin{aligned} V(\{1\}) &= \alpha_{15} + \alpha_{51} = A(1, 5), \\ V(\{2\}) &= \alpha_{23} + \alpha_{24} + \alpha_{42} + \alpha_{32} = A(2, 3) + A(2, 4), \\ V(\{1, 2\}) &= \alpha_{15} + \alpha_{51} + \alpha_{12} + \alpha_{21} + \alpha_{23} + \alpha_{24} + \alpha_{42} + \alpha_{32} + \alpha_{45} + \alpha_{54} \\ &= A(1, 2) + A(1, 5) + A(2, 3) + A(2, 4) + A(4, 5). \end{aligned}$$

Computing the Shapley value we obtain

$$Sh_1 = A(1, 5) + \frac{A(1, 2) + A(5, 4)}{2}, \quad Sh_2 = A(2, 3) + A(2, 4) + \frac{A(1, 2) + A(5, 4)}{2}. \tag{24}$$

Now, consider cooperative game with players $i \in M_k$, $k = 1, 2$ belonging to one partner set M_k and construct characteristic function using (12) and (21).

For $M_1 = \{1, 5\}$ characteristic function has the following form

$$\begin{aligned} V^{M_1}(\{1\}) &= 0, \\ V^{M_1}(\{5\}) &= 0, \\ V^{M_1}(\{1, 5\}) &= \alpha_{15}^1 + \alpha_{51}^1 = A^1(1, 5). \end{aligned}$$

The Shapley value in this game is equal to

$$Sh_1^{M_1} = \frac{A^1(1, 5)}{2}, \quad Sh_5^{M_1} = \frac{A^1(1, 5)}{2} \tag{25}$$

For $M_2 = \{2, 3, 4\}$ characteristic function has the following form

$$\begin{aligned} V^{M_2}(\{2\}) &= V^{M_2}(\{3\}) = V^{M_2}(\{4\}) = 0, \\ V^{M_2}(\{2, 3\}) &= \alpha_{23}^2 + \alpha_{32}^2 = A^2(2, 3), \\ V^{M_2}(\{2, 4\}) &= \alpha_{24}^2 + \alpha_{42}^2 = A^2(2, 4), \\ V^{M_2}(\{3, 4\}) &= 0. \\ V^{M_2}(\{2, 3, 4\}) &= \alpha_{23}^2 + \alpha_{32}^2 + \alpha_{24}^2 + \alpha_{42}^2 = A^2(2, 3) + A^2(2, 4). \end{aligned}$$

The Shapley value in this game is equal to

$$Sh_2^{M_2} = \frac{A^2(2, 3) + A^2(2, 4)}{2}, \quad Sh_3^{M_2} = \frac{A^2(2, 3)}{2}, \quad Sh_4^{M_2} = \frac{A^2(2, 4)}{2} \tag{26}$$

Using (18), (25), (26) we can compute β_{ik} .

$$\begin{aligned} \beta_{11} &= \frac{Sh_1^{M_1}}{\sum_{i \in M_1} Sh_i^{M_1}} = \frac{Sh_1^{M_1}}{V^{M_1}(\{1, 5\})} = \frac{\frac{A^1(1, 5)}{2}}{A^1(1, 5)} = \frac{1}{2}, \\ \beta_{22} &= \frac{Sh_2^{M_2}}{\sum_{i \in M_2} Sh_i^{M_2}} = \frac{Sh_2^{M_2}}{V^{M_2}(\{2, 3, 4\})} = \frac{\frac{A^2(2, 3) + A^2(2, 4)}{2}}{A^2(2, 3) + A^2(2, 4)} = \frac{1}{2}, \\ \beta_{32} &= \frac{Sh_3^{M_2}}{\sum_{i \in M_2} Sh_i^{M_2}} = \frac{Sh_3^{M_2}}{V^{M_2}(\{2, 3, 4\})} = \frac{\frac{A^2(2, 3)}{2}}{A^2(2, 3) + A^2(2, 4)} \\ &= \frac{1}{2} \frac{A^2(2, 3)}{A^2(2, 3) + A^2(2, 4)}, \\ \beta_{42} &= \frac{Sh_4^{M_2}}{\sum_{i \in M_2} Sh_i^{M_2}} = \frac{Sh_4^{M_2}}{V^{M_2}(\{2, 3, 4\})} = \frac{\frac{A^2(2, 4)}{2}}{A^2(2, 3) + A^2(2, 4)} \\ &= \frac{1}{2} \frac{A^2(2, 4)}{A^2(2, 3) + A^2(2, 4)}, \\ \beta_{51} &= \frac{Sh_5^{M_1}}{\sum_{i \in M_1} Sh_i^{M_1}} = \frac{Sh_5^{M_1}}{V^{M_1}(\{1, 5\})} = \frac{1}{2}. \end{aligned}$$

The payoff of player i , $i \in N$ under two-level cooperation is equal to (see (19)).

$$\gamma_1(x_0, T - t_0) = \beta_{11}Sh_1(x_0, T - t_0) = \frac{1}{2} \cdot \left(A(1, 5) + \frac{A(1, 2) + A(5, 4)}{2} \right),$$

$$\gamma_2(x_0, T - t_0) = \beta_{22}Sh_2(x_0, T - t_0) = \frac{1}{2} \cdot \left(A(2, 3) + A(2, 4) + \frac{A(1, 2) + A(5, 4)}{2} \right),$$

$$\gamma_3(x_0, T - t_0) = \beta_{32}Sh_2(x_0, T - t_0)$$

$$= \frac{1}{2} \frac{A^2(2, 3)}{A^2(2, 3) + A^2(2, 4)} \cdot \left(A(2, 3) + A(2, 4) + \frac{A(1, 2) + A(5, 4)}{2} \right),$$

$$\gamma_4(x_0, T - t_0) = \beta_{42}Sh_2(x_0, T - t_0)$$

$$= \frac{1}{2} \frac{A^2(2, 4)}{A^2(2, 3) + A^2(2, 4)} \cdot \left(A(2, 3) + A(2, 4) + \frac{A(1, 2) + A(5, 4)}{2} \right),$$

$$\gamma_5(x_0, T - t_0) = \beta_{51}Sh_1(x_0, T - t_0) = \frac{1}{2} \cdot \left(A(1, 5) + \frac{A(1, 2) + A(5, 4)}{2} \right).$$

8 Conclusion

Differential cooperative network games with partner sets are considered. As optimality principle two level cooperative solution is proposed. The proposed solution is similar to one introduced by G. Owen [6] but is easier to compute. On the first level, players (partner sets) cooperate to get maximal joint payoff. This payoff is then allocated between partner sets according to the Shapley value. On the second level, players from each partner set allocate the payoff prescribed by the corresponding component of the Shapley value between players members from this partner set. This allocation is proportional to the components of the Shapley value defined as a cooperative solution between players in the partner set. The example is provided.

References

1. Bulgakova, M., Petrosyan, L.: About one multistage non-antagonistic network game (in Russian). *Vestnik S.-Petersburg Univ. Ser. 10. Prikl. Mat. Inform. Prots. Upr.* **5**(4), 603–615 (2019). <https://doi.org/10.21638/11702/spbu10.2019.415>
2. Cao, H., Ertin, E., Arora, A.: MiniMax equilibrium of networked differential games. *ACM TAAS* **3**(4), 1–21 (1963). <https://doi.org/10.1145/1452001.1452004>
3. Gao, H., Pankratova, Y.: Cooperation in dynamic network games. *Contrib. Game Theory Manage.* **10**, 42–67 (2017)
4. Gromova, E.: The Shapley value as a sustainable cooperative solution in differential games of three players. In: Petrosyan, L., Mazalov, V. (eds.) *Recent Advances in Game Theory and Applications. Static & Dynamic Game Theory: Foundations & Applications*, pp. 67–89. Birkhäuser, Cham (2016). https://doi.org/10.1007/978-3-319-43838-2_4
5. Petrosyan, L.A., Mazalov, V.V. (eds.): *Recent Advances in Game Theory and Applications*. SDGTFA, Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-43838-2>
6. Owen, G.: Values of games with a priori unions. In: Henn, R., Moeschlin, O. (eds.) *Essays in Mathematical Economics and Game Theory*, pp. 76–88. Springer, Heidelberg (1977). https://doi.org/10.1007/978-3-642-45494-3_7
7. Pai, H.M.: A differential game formulation of a controlled network. *Queueing Syst.* **64**(4), 325–358 (2010). <https://doi.org/10.1007/s11134-009-9161-6>
8. Petrosian, O.L., Gromova, E.V., Pogozhev, S.V.: Strong time-consistent subset of core in cooperative differential games with finite time horizon (in Russian). *Mat. Teor. Igr Pril.* **8**(4), 79–106 (2016)
9. Petrosyan, L.A.: Cooperative differential games on networks (in Russian). *Trudy Inst. Mat. i Mekh. UrO RAN* **16**(5), 143–150 (2010)
10. Petrosyan, L., Zaccour, G.: Time-consistent Shapley value allocation of pollution cost reduction. *J. Econ. Dyn. Control.* **27**, 381–398 (2003). [https://doi.org/10.1016/S0165-1889\(01\)00053-7](https://doi.org/10.1016/S0165-1889(01)00053-7)
11. Shapley, L.S.: A value for N-person games. In: Kuhn, H., Tucker, A. (eds.) *Contributions to the Theory of Games*, pp. 307–317. Princeton University Press, Princeton (1953)
12. Wie, B.W.: A differential game approach to the dynamic mixed behavior traffic network equilibrium problem. *Eur. J. Oper. Res.* **83**(1), 117–136 (1995). <https://doi.org/10.1002/net.3230230606>
13. Yeung, D.W.K.: Time consistent Shapley value imputation for cost-saving joint ventures. *Mat. Teor. Igr Pril.* **2**(3), 137–149 (2010)
14. Yeung, D.W.K., Petrosyan, L.A.: Subgame consistent cooperative solution in stochastic differential games. *J. Optimiz. Theory. App.* **120**(3), 651–666 (2004). <https://doi.org/10.1023/B:JOTA.0000025714.04164.e4>
15. Yeung, D.W.K., Petrosyan, L.A.: *Subgame Consistent Cooperation: A Comprehensive Treatise*. Springer, Cham (2016)
16. Yeung, D.W.K., Petrosyan, L.A.: *Dynamic Shapley Value and Dynamic Nash Bargaining*. Nova Science, New York (2018)
17. Petrosyan, L.A., Yeung, D.W.K.: Shapley value for differential network games: theory and application. *JDG* **8**(2), 151–166 (2020). <https://doi.org/10.3934/jdg.2020021>
18. Zhang, H., Jiang, L.V., Huang, S., Wang, J., Zhang, Y.: Attack-defense differential game model for network defense strategy selection. *IEEE Access* (2018). <https://doi.org/10.1109/ACCESS.2018.2880214>

19. Petrosyan, L., Yeung, D., Pankratova, Y.: Dynamic cooperative games on networks. In: Strekalovsky, A., Kochetov, Y., Gruzdeva, T., Orlov, A. (eds.) MOTOR 2021. CCIS, vol. 1476, pp. 403–416. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86433-0_28
20. Petrosyan, L., Yeung, D.W.K., Pankratova, Y.: Cooperative differential games with partner sets on networks. Trudy Instituta Matematiki i Mekhaniki UrO RAN **27**(3), 286–295 (2021). <https://doi.org/10.21538/0134-4889-2021-27-3-286-295>



Multicriteria Dynamic Games with Asymmetric Horizons

Anna Rettieva^{1,2} 

¹ Institute of Applied Mathematical Research, Karelian Research Center of RAS,
11 Pushkinskaya Street, Petrozavodsk 185910, Russia

annaret@krc.karelia.ru

² Saint Petersburg State University, 7/9 Universitetskaya nab.,
Saint Petersburg 199034, Russia

Abstract. We consider a dynamic, discrete-time, game model where many players use a common resource and have different criteria to optimize. Moreover, the participants planning horizons are assumed to be different. Multicriteria Nash and cooperative equilibria are defined via modified bargaining schemas. To stabilize the multicriteria cooperative solution a time-consistent payoff distribution procedure is constructed. To illustrate the presented approaches, a dynamic bi-criteria bioresource management problem with many players and asymmetric planning horizons is investigated.

Keywords: Dynamic games · Multicriteria games · Nash bargaining solution · Asymmetric horizons

1 Introduction

Game-theoretic models with the presence of more than one players' objective [20] are closer to real problems. Participants often aim to optimize several criteria, which can be incomparable, simultaneously. For example, in renewable resource management problems the players wish both to maximize the profit from the resource exploitation and to minimize the costs or the harm to the environment. The multicriteria approach helps to determine an optimal behavior in such situations.

In static multicriteria games, the solution concepts are usually based on the Pareto set [2, 20] or some convolutions of the criteria [1]. Some other approaches for the solution have been proposed recently, including, e.g., the ideal Nash equilibrium [22] and the E-equilibrium [13]. However, Pareto equilibrium is the most studied solution for static multicriteria game theory.

The methods of static multicriteria games are not applicable to the dynamic version. The construction of equilibria in dynamic games with vector payoffs

This research was supported by the Russian Science Foundation grant No. 22-11-00051, <https://rscf.ru/en/project/22-11-00051/>.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 264–278, 2022.
https://doi.org/10.1007/978-3-031-09607-5_19

is a little-studied problem. In the series of papers [14–17], new approaches to obtain players' optimal behavior in dynamic multicriteria games have been suggested. The concept of multicriteria noncooperative equilibrium was formalized in [14] combining the ideas of multi-objective optimization (nadir points [24]) and the classical concept of Nash equilibrium. The concept of Pareto optimality, most widespread in multi-objective optimization, and the Nash bargaining approach [8, 9, 21] were applied to determine the cooperative strategies and payoffs of the players in [15]. Another method to define cooperative behavior in dynamic multicriteria games that guarantees the fulfillment of individual rationality conditions was presented in [16, 19]. For different game-theoretic resource management problems with vector payoffs [15, 17], it was shown that the cooperative behavior determined according to presented approaches is beneficial for the players and, that is more important, improves the ecological situation.

As is well known, the Nash bargaining scheme is not dynamically stable [4]. The concept of time-consistency (dynamic stability) and the notion of time-consistent imputation distribution procedure have been introduced by Petrosyan L.A. [11, 12]. Different payoff distribution procedures, including the time-consistent ones, for multicriteria multistage games were presented in [5–7]. The idea of payoff distribution procedure was applied in [16, 17, 19] for dynamic multicriteria games with finite and random horizons.

The aim of this paper is to adopt the developed approaches to the multicriteria dynamic game with asymmetric planning horizons. In renewable resource management problems when the exploitation time of the participant is smaller than that of others, the player under consideration is interested in gaining more from exploitation process than the players that continue operation. Hence, the solution concept should capture the possibility of the players' leaving the game.

In [19] the dynamic multicriteria game with random planning horizon was investigated under assumption that the exploitation time is identical for all the players. Here the extension of this model where the players have asymmetric planning horizons is presented.

We consider a dynamic, discrete-time, game model where many players use a common resource and have different criteria to optimize. Moreover, the participants planning horizons are assumed to be different. To construct a multicriteria Nash equilibrium the bargaining solution is adopted [14]. To design a multicriteria cooperative equilibrium a modified bargaining scheme [16] is applied. To stabilize the multicriteria cooperative solution a time-consistent payoff distribution procedure [17] is constructed. To illustrate the presented approaches a dynamic bi-criteria bioresource management problem with many players and asymmetric planning horizons is investigated.

Further exposition has the following structure. Section 2 describes the non-cooperative and cooperative solution concepts for a multicriteria dynamic game with many players and asymmetric horizons. The time-consistent payoff distribution procedure is presented in Sect. 2.3. A bi-criteria discrete-time game-theoretic bioresource management model (harvesting problem) is treated in Sect. 3. Finally, Sect. 4 provides the basic results and their discussion.

2 Dynamic Multicriteria Game with Asymmetric Horizons

Consider a multicriteria dynamic game in discrete time. Let $N = \{1, \dots, n\}$ players exploit a common resource and each of them wishes to optimize k different criteria. The state dynamics is in the form

$$x_{t+1} = f(x_t, u_{1t}, \dots, u_{nt}), \quad x_1 = x, \tag{1}$$

where $x_t \geq 0$ is the resource size at time $t \geq 0$, $f(x_t, u_{1t}, \dots, u_{nt})$ denotes the growth function, and $u_{it} \geq 0$ gives the exploitation rate of player i at time t , $i \in N$.

We explore a model where the players possess heterogeneous planning horizons. By assumption, the players exploit the resource during m_1, \dots, m_n steps, respectively. In [19] the planning horizon was assumed to be identical for all the players. Here, the asymmetric case is considered that can be interpreted as the situation when the players conclude exploitation agreements with the resource owner for different time periods. For example, the regional government gives the license to exploit the resource in Karelian lakes for the period from one month to several years.

For simplicity, renumber the players according to $m_1 \leq \dots \leq m_n$ and $m_1 > m_0 = 0$. Therefore, during the time period $[m_{i-1}, m_i]$ $n + 1 - i$ players exploit the same resource stock, and the problem consists in evaluating their optimal strategies.

Each player has k goals to optimize. The payoff functions of the players are defined by

$$\begin{aligned}
 J_1 &= \begin{pmatrix} J_1^1 = \sum_{t=m_0+1}^{m_1} \delta^t g_1^1(x_t, u_{1t}, \dots, u_{nt}) \\ \dots \\ J_1^k = \sum_{t=m_0+1}^{m_1} \delta^t g_1^k(x_t, u_{1t}, \dots, u_{nt}) \end{pmatrix}, \\
 J_2 &= \begin{pmatrix} J_2^1 = \sum_{t=m_0+1}^{m_1} \delta^t g_2^1(x_t, u_{1t}, \dots, u_{nt}) + \sum_{t=m_1+1}^{m_2} \delta^t g_2^1(x_t, u_{2t}, \dots, u_{nt}) \\ \dots \\ J_2^k = \sum_{t=m_0+1}^{m_1} \delta^t g_2^k(x_t, u_{1t}, \dots, u_{nt}) + \sum_{t=m_1+1}^{m_2} \delta^t g_2^k(x_t, u_{2t}, \dots, u_{nt}) \end{pmatrix}, \\
 &\dots, \\
 J_n &= \begin{pmatrix} J_n^1 = \sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^1(x_t, u_{it}, \dots, u_{nt}) \\ \dots \\ J_n^k = \sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^k(x_t, u_{it}, \dots, u_{nt}) \end{pmatrix}, \tag{2}
 \end{aligned}$$

where $g_i^j(\cdot) \geq 0$ gives the instantaneous utility, $i \in N$, $j = 1, \dots, k$, $\delta \in (0, 1)$ denotes a common discount factor and x_t possesses the dynamics

$$x_{t+1} = f(x_t, u_{it}, \dots, u_{nt}), \quad t \in [m_{i-1} + 1, m_i], \quad i \in N, \quad x_1 = x. \tag{3}$$

Here the asymmetry of the players is captured by different planning horizons; another approach was presented in [16] where the participants possessed different discount factors. The results of the presented paper can be extended to such a case, but to simplify the analysis we assume that the discount factor is common.

2.1 Multicriteria Nash Equilibrium

We design the noncooperative behavior in dynamic multicriteria game with asymmetric horizons combining the methods of multi-objective optimization and the classical concept of Nash equilibrium [14, 15]. To adopt the ideas of nadir points or status quo points for the Nash products some worst or guaranteed objective values of all the criteria should be determined. The possible concepts to construct the guaranteed payoffs for the game with two players were presented in [14]. As it was demonstrated, the variant where the guaranteed payoffs are determined as the Nash equilibrium solutions is the best for the state of the exploited system and profitable for the players. Therefore, for dynamic multicriteria game with asymmetric horizons we adopt this concept. Namely,

$G_i^{1m_j}, i \in N, j = 1, \dots, i$, are the Nash equilibrium payoffs in the dynamic game $\langle x_t, \{j, \dots, n\}, \{U_l\}_{l=j}^n, \{J_l^1\}_{l=j}^n \rangle, t \in [m_{j-1} + 1, m_j]$,

$\dots,$
 $G_i^{km_j}, i \in N, j = 1, \dots, i$, are the Nash equilibrium payoffs in the dynamic game $\langle x_t, \{j, \dots, n\}, \{U_l\}_{l=j}^n, \{J_l^k\}_{l=j}^n \rangle, t \in [m_{j-1} + 1, m_j]$,

where the state dynamics has the form (3). It is assumed that on the last period $[m_{n-1}, m_n]$ where the player n exploit the stock alone the guaranteed payoffs points are equal to zero.

The players' payoff functions in the dynamic multicriteria game are constructed adopting the Nash products with the guaranteed payoffs playing the role of the status quo points:

$$\begin{aligned}
 H_1(u_{1t}, \dots, u_{nt}) &= (J_1^1 - G_1^{1m_1}) \cdot \dots \cdot (J_1^k - G_1^{km_1}) \\
 &= \left(\sum_{t=m_0+1}^{m_1} \delta^t g_1^1(x_t, u_{1t}, \dots, u_{nt}) - G_1^{1m_1} \right) \cdot \dots \cdot \\
 &\cdot \left(\sum_{t=m_0+1}^{m_1} \delta^t g_1^k(x_t, u_{1t}, \dots, u_{nt}) - G_1^{km_1} \right), t \in [m_0 + 1, m_1], \\
 H_2(u_{1t}, \dots, u_{nt}) &= (J_2^1 - G_2^{1m_1} - G_2^{1m_2}) \cdot \dots \cdot (J_2^k - G_2^{km_1} - G_2^{km_2}) \\
 &= \left(\sum_{i=1}^2 \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_2^1(x_t, u_{it}, \dots, u_{nt}) - G_2^{1m_1} - G_2^{1m_2} \right) \cdot \dots \cdot \\
 &\cdot \left(\sum_{i=1}^2 \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_2^k(x_t, u_{it}, \dots, u_{nt}) - G_2^{km_1} - G_2^{km_2} \right), t \in [m_0 + 1, m_2], \\
 &\dots \\
 H_n(u_{1t}, \dots, u_{nt}) &= (J_n^1 - \sum_{j=1}^{n-1} G_n^{1m_j}) \cdot \dots \cdot (J_n^k - \sum_{j=1}^{n-1} G_n^{km_j})
 \end{aligned}$$

$$\begin{aligned}
 &= \left(\sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^1(x_t, u_{it}, \dots, u_{nt}) - \sum_{j=1}^{n-1} G_n^{1m_j} \right) \cdot \dots \cdot \\
 &\cdot \left(\sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^k(x_t, u_{it}, \dots, u_{nt}) - \sum_{j=1}^{n-1} G_n^{km_j} \right), \quad t \in [m_0 + 1, m_n].
 \end{aligned}$$

The multicriteria Nash equilibrium strategies are constructed in the feedback form $u_{it}^N = u_{it}^N(x_t)$, $i \in N$, $t \in [m_0 + 1, m_i]$.

Definition 1. A strategy profile $u_t^N = u_t^N(x_t) = (u_{1t}^N, \dots, u_{nt}^N)$ is called a multicriteria Nash equilibrium [14] of the problem (2), (3) if

$$H_i(u_i^N) \geq H_i(u_{1t}^N, \dots, u_{i-1t}^N, u_{it}, u_{i+1t}^N, \dots, u_{nt}^N) \quad \forall u_{it} \in U_i, i \in N, t \in [m_0 + 1, m_i]. \quad (4)$$

Hence, in accordance with the proposed noncooperative solution concept, the players maximize the product of all deviations of their payoffs from the guaranteed ones.

2.2 Multicriteria Cooperative Equilibrium

An approach to determine cooperative strategies in dynamic multicriteria game with asymmetric players was presented in [16]. This solution concept guarantees the rationality of cooperative behavior as the cooperative payoffs of the players are greater than or equal to the multicriteria Nash payoffs. More specifically, the cooperative strategies and payoffs of the players are determined from the modified bargaining solution that combines compromise programming [24] and the Nash bargaining scheme [8, 9]. The status quo points are the noncooperative payoffs obtained by the players applying the multicriteria Nash equilibrium strategies u_t^N :

$$\begin{aligned}
 J_1^N &= \left(\begin{array}{c} J_1^{1N} = \sum_{t=m_0+1}^{m_1} \delta^t g_1^1(x_t^N, u_{1t}^N, \dots, u_{nt}^N) \\ \dots \\ J_1^{kN} = \sum_{t=m_0+1}^{m_1} \delta^t g_1^k(x_t^N, u_{1t}^N, \dots, u_{nt}^N) \end{array} \right), \\
 &\dots, \\
 J_n^N &= \left(\begin{array}{c} J_n^1 = \sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^1(x_t^N, u_{it}^N, \dots, u_{nt}^N) \\ \dots \\ J_n^k = \sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^k(x_t^N, u_{it}^N, \dots, u_{nt}^N) \end{array} \right), \quad (5)
 \end{aligned}$$

where noncooperative trajectory x_t^N possesses the dynamics

$$x_{t+1}^N = f(x_t^N, u_{it}^N, \dots, u_{nt}^N), \quad t \in [m_{i-1} + 1, m_i], \quad i = 1, \dots, n, \quad x_1 = x.$$

The cooperative strategies in the feedback form $u_{it}^c = u_{it}^c(x_t)$, $i \in N$, $t \in [m_0 + 1, m_i]$, and payoffs are obtained as the solution of the following problem:

$$\begin{aligned}
 & (V_1^{1c} - J_1^{1N}) \cdot \dots \cdot (V_1^{kc} - J_1^{kN}) + \dots + (V_n^{1c} - J_n^{1N}) \cdot \dots \cdot (V_n^{kc} - J_n^{kN}) \\
 &= \left(\sum_{t=m_0+1}^{m_1} \delta^t g_1^1(x_t, u_{1t}^c, \dots, u_{nt}^c) - J_1^{1N} \right) \cdot \dots \cdot \\
 & \quad \cdot \left(\sum_{t=m_0+1}^{m_1} \delta^t g_1^k(x_t, u_{1t}^c, \dots, u_{nt}^c) - J_1^{kN} \right) + \dots \\
 & \quad + \left(\sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^1(x_t, u_{it}^c, \dots, u_{nt}^c) - J_n^{1N} \right) \cdot \dots \cdot \\
 & \quad \cdot \left(\sum_{i=1}^n \sum_{t=m_{i-1}+1}^{m_i} \delta^t g_n^k(x_t, u_{it}^c, \dots, u_{nt}^c) - J_n^{kN} \right) \rightarrow \max_{u_{1t}^c, \dots, u_{nt}^c}, \quad (6)
 \end{aligned}$$

where J_i^{jN} are the noncooperative payoffs given by (5), $i \in N$, $j = 1, \dots, k$, cooperative trajectory x_t^c is defined by (3) where $u_{it} = u_{it}^c$, $i \in N$.

Definition 2. A strategy profile $u_i^c = u_i^c(x_t) = (u_{1t}^c, \dots, u_{nt}^c)$ is a rational multicriteria cooperative equilibrium [16] of problem (2), (3) if it is the solution of problem (6).

This approach is similar to the classical cooperative solution as the players seek to maximize the sum of their individual payoffs. The goal of each player is to maximize the distance to the noncooperative payoffs, and to behave cooperatively the players aim to do it jointly.

2.3 Time-consistent Payoff Distribution Procedure

The players' cooperative payoffs for the duration of the game can be calculated as

$$J_i^c(1, x) = \begin{pmatrix} J_i^{1c}(1, x) = \sum_{j=1}^i \sum_{t=m_{j-1}+1}^{m_j} \delta^t g_i^1(x_t^c, u_{it}^c, \dots, u_{nt}^c) \\ \dots \\ J_i^{kc}(1, x) = \sum_{j=1}^i \sum_{t=m_{j-1}+1}^{m_j} \delta^t g_i^k(x_t^c, u_{it}^c, \dots, u_{nt}^c) \end{pmatrix}, \quad i \in N, \quad (7)$$

where $u_t^c = (u_{1t}^c, \dots, u_{nt}^c)$ are the cooperative strategies obtained from (6).

Similarly we determine the cooperative payoffs $J_i^c(t, x_t^c)$, $i = 1, \dots, n$, $t \in [m_0 + 1, m_i]$, for every subgame started from the state x_t^c at time t .

To stabilize the cooperative solution in multicriteria dynamic game with asymmetric horizons we adopt the time-consistent payoff distribution procedure [5, 11, 12, 17] which main idea is to distribute the cooperative gain along the game path. The payment to player i , $i \in N$, in all criteria at time t is defined from the following definitions.

Definition 3. A vector

$$\beta(t, x_t) = (\beta_1(t, x_t), \dots, \beta_n(t, x_t)),$$

where

$$\beta_i(t, x_t) = \begin{pmatrix} \beta_i^1(t, x_t) \\ \dots \\ \beta_i^k(t, x_t) \end{pmatrix}, \quad i \in N, \quad t \in [m_0 + 1, m_i],$$

is a payoff distribution procedure (PDP) for the dynamic multicriteria game with asymmetric horizons (2), (3), if

$$J_i^{jc}(1, x) = \sum_{t=m_0+1}^{m_i} \delta^t \beta_i^j(t, x_t), \quad i \in N, \quad j = 1, \dots, k. \quad (8)$$

Definition 4. A vector $\beta(t, x_t) = (\beta_1(t, x_t), \dots, \beta_n(t, x_t))$ is a time-consistent PDP [11, 12] for dynamic multicriteria game with asymmetric horizons (2), (3), if for every $t \in [m_0 + 1, m_i]$

$$J_i^{jc}(1, x) = \sum_{l=m_0+1}^t \delta^l \beta_i^j(l, x_l) + J_i^{jc}(t + 1, x_{t+1}), \quad i \in N, \quad j = 1, \dots, k. \quad (9)$$

Theorem 1. A vector $\beta(t, x_t) = (\beta_1(t, x_t), \dots, \beta_n(t, x_t))$, where

$$\beta_i(t, x_t) = J_i^c(t, x_t) - \delta J_i^c(t + 1, x_{t+1}), \quad i \in N, \quad t \in [m_0 + 1, m_i], \quad (10)$$

is a time-consistent payoff distribution procedure for dynamic multicriteria game with asymmetric horizons (2), (3).

Proof. is similar to [17].

Next, we consider a dynamic bi-criteria model with many players and different planning horizons related with the bioresource management problem (harvesting) to illustrate the suggested concepts.

3 Dynamic Bi-Criteria Resource Management Problem with Asymmetric Horizons

Consider a bi-criteria discrete-time dynamic resource management model. Let n players (countries or firms) exploit a bioresource during m_1, \dots, m_n steps, $m_1 \leq \dots \leq m_n$ and $m_1 > m_0 = 0$. The bioresource evolves according to the equation

$$x_{t+1} = \varepsilon x_t - u_{it} - \dots - u_{nt}, \quad t \in [m_{i-1} + 1, m_i], \quad i \in N, \quad x_1 = x, \quad (11)$$

where $x_t \geq 0$ is the resource size at time $t \geq 0$, $\varepsilon \geq 1$ denotes the natural birth rate, and $u_{it} = u_{it}(x_t) \geq 0$ specifies the exploitation strategy of player i at time $t \geq 0$, $i \in N = \{1, \dots, n\}$.

Each player wishes to achieve two goals: to maximize the revenue from resource sales and to minimize the exploitation costs. Assume that the players have different market prices but the same costs that depend quadratically on the exploitation rate. The vector payoff functions of the players take the forms

$$J_1 = \left(\begin{array}{l} J_1^1 = \sum_{t=m_0+1}^{m_1} \delta^t p_1 u_{1t}(x_t) \\ J_1^2 = - \sum_{t=m_0+1}^{m_1} \delta^t c u_{1t}^2(x_t) \end{array} \right), \dots, J_n = \left(\begin{array}{l} J_n^1 = \sum_{t=m_0+1}^{m_n} \delta^t p_n u_{nt}(x_t) \\ J_n^2 = - \sum_{t=m_0+1}^{m_n} \delta^t c u_{nt}^2(x_t) \end{array} \right), \tag{12}$$

where $p_i \geq 0$ is the market price of the resource for player i , $i \in N$, $c \geq 0$ indicates the exploitation cost, and $\delta \in (0, 1)$ denotes the discount factor.

3.1 Multicriteria Nash Equilibrium

First, we construct the guaranteed payoff points $G_i^{1m_j}$, $i \in N$, $j = 1, \dots, i$, as the Nash equilibrium in the game $\langle x_t, \{j, \dots, n\}, \{U_l\}_{l=j}^n, \{J_l^1\}_{l=j}^n \rangle$, $t \in [m_{j-1} + 1, m_j]$.

Applying the Bellman principle and assuming the linear form of the strategies and value functions, we obtain the Nash equilibrium strategies

$$u_{jt} = \dots = u_{nt} = \frac{\varepsilon - 1}{n - j} x_t,$$

and the dynamics becomes

$$x_{t+1} = \frac{n - j + 1 - \varepsilon}{n - j} x_t.$$

Hence, on the time interval $[m_{j-1} + 1, m_j]$ we can define the resource size as

$$x_t = \left(\frac{n - j + 1 - \varepsilon}{n - j} \right)^t x_{m_{j-1}+1}$$

and $x_{m_{j-1}+1}$ can be expressed via the initial stock size $x_1 = x$ as

$$x_{m_{j-1}+1} = \prod_{l=1}^{j-1} \left(\frac{n - l + 1 - \varepsilon}{n - l} \right)^{m_l} x.$$

Then the guaranteed payoff points take the form

$$G_i^{1m_j} = p_i \sum_{t=m_{j-1}+1}^{m_j} \delta^t u_{jt} = p_i A_{m_j} x, \tag{13}$$

where

$$A_{m_j} = \frac{\varepsilon - 1}{n - j} \frac{(\delta Q_j)^{m_j+1} - (\delta Q_j)^{m_j-1}}{\delta Q_j - 1} \prod_{l=1}^{j-1} Q_l^{m_l}, \quad Q_j = \frac{\delta(n - j + 1 - \varepsilon)}{n - j}.$$

Similarly, determining the Nash equilibrium in the game with the second criteria of all players $\langle x_t, \{j, \dots, n\}, \{U_l\}_{l=j}^n, \{J_l^2\}_{l=j}^n \rangle, t \in [m_{j-1}, m_j]$, yields the guaranteed payoffs points

$$G_i^{2m_j} = -cD_{m_j}x^2, \tag{14}$$

where

$$D_{m_j} = \left(\frac{\delta\varepsilon^2 - 2(n-j+1) + \varepsilon^2 S}{(n-j+1)(\delta\varepsilon^2 + \varepsilon S)} \right)^2 \frac{(\delta L_j)^{2m_j+1} - (\delta L_j)^{m_j-1}}{L^2 - 1} \left(\prod_{l=1}^{j-1} L_l^{m_l} \right)^2,$$

$$L_j = \frac{2\varepsilon(n-j+1)}{\delta\varepsilon^2 + \varepsilon S}, \quad S = \sqrt{\delta(\delta\varepsilon^2 - 4(n-j+1) + 4(n-j+1)^2)}.$$

According to Definition 1, to determine the multicriteria Nash equilibrium of the game (11) (12) the following problem has to be solved:

$$p_1 c \left(\sum_{t=m_0+1}^{m_1} \delta^t u_{1t} - A_{m_1} x \right) \left(- \sum_{t=m_0+1}^{m_1} \delta^t u_{1t}^2 + D_{m_1} x^2 \right) \rightarrow \max_{u_{1t}} \dots$$

$$p_n c \left(\sum_{t=m_0+1}^{m_n} \delta^t u_{nt} - \sum_{l=1}^{n-1} A_{m_l} x \right) \left(- \sum_{t=m_0+1}^{m_n} \delta^t u_{nt}^2 + \sum_{l=1}^{n-1} D_{m_l} x^2 \right) \rightarrow \max_{u_{nt}}. \tag{15}$$

Proposition 1. *The multicriteria Nash equilibrium strategies in problem (11), (12) have the form $u_{it}^N = \gamma_{it}^N x_t, i \in N, t \in [m_0 + 1, m_i]$*

$$\gamma_{jt}^N = \frac{\gamma_{j1}^N}{\varepsilon^{t-1} - \sum_{k=1}^{i-1} \gamma_{k1}^N \sum_{l=t-m_k-1}^{t-2} \varepsilon^l - \sum_{k=i}^n \gamma_{k1}^N \sum_{l=0}^{t-2} \varepsilon^l}, \quad j=i, \dots, n, t \in [m_{i-1}+1, m_i], \tag{16}$$

where the players' strategies at the first stage are

$$\gamma_{j1}^N = \frac{- \sum_{l=1}^j A_{m_l} + \sqrt{(\sum_{l=1}^j A_{m_l})^2 - 3 \sum_{k=0}^{m_j-1} \delta^k \sum_{l=1}^j D_{m_l}}}{3 \sum_{k=0}^{m_j-1} \delta^k}.$$

Proof. We start with the last time interval $[m_{n-1} + 1, m_n]$. Here player n exploits the stock alone. Lets consider the one-step game and as usual we seek the strategies in linear form $u_{nm_n}^N = \gamma_{nm_n} x$. The players n 's payoff for the first criterium is

$$V_{nm_n}^1(\gamma_{nm_n}; x) = \gamma_{nm_n} x,$$

and for the second one

$$V_{nm_n}^2(\gamma_{nm_n}; x) = -\gamma_{nm_n}^2 x^2.$$

Hence, to determine the strategies for this one-step game we solve the following problem

$$\begin{aligned} & (V_{nm_n}^1(\gamma_{nm_n}; x) - A_{m_n})(V_{nm_n}^2(\gamma_{nm_n}; x) - G_{m_n}) \\ & = (\gamma_{nm_n} x - A_{m_n} x)(-\gamma_{nm_n}^2 x^2 - G_{m_n} x^2) \rightarrow \max_{\gamma_{nm_n}}. \end{aligned}$$

We can now consider problem (15) for the two-step game. The player n 's objective function for the two-step game is

$$\begin{aligned} V_{nm_{n-1}}^1(\gamma_{nm_{n-1}}, \gamma_{nm_n}; x) & = \gamma_{nm_{n-1}} x + \delta V_{nm_n}^1(\gamma_{nm_n}; (\varepsilon - \gamma_{nm_{n-1}})x), \\ V_{nm_{n-1}}^2(\gamma_{nm_{n-1}}, \gamma_{nm_n}; x) & = -\gamma_{nm_{n-1}}^2 x^2 + \delta V_{nm_n}^2(\gamma_{nm_n}; (\varepsilon - \gamma_{nm_{n-1}})x). \end{aligned}$$

To determine the strategies for this two-step game we solve the following problem

$$\begin{aligned} & (V_{nm_{n-1}}^1(\gamma_{nm_{n-1}}, \gamma_{nm_n}; x) - A_{m_n} x) \cdot \\ & \cdot (V_{nm_{n-1}}^2(\gamma_{nm_{n-1}}, \gamma_{nm_n}; x) - G_{m_n} x^2) \\ & = (\gamma_{nm_{n-1}} x + \delta \gamma_{nm_n} (\varepsilon - \gamma_{nm_{n-1}})x - A_{m_n} x) \cdot \\ & \cdot (-\gamma_{nm_{n-1}}^2 x^2 + \delta \gamma_{nm_n}^2 (\varepsilon - \gamma_{nm_{n-1}})^2 x^2 - G_{m_n} x^2) \rightarrow \max_{\gamma_{nm_{n-1}}, \gamma_{nm_n}}. \end{aligned} \tag{17}$$

By continuing the process for the $m_n - m_{n-1} + 1$ -step game we get the relations for the player n 's strategies

$$\gamma_{nt} = \frac{\gamma_{nm_{n-1}}^N}{\varepsilon^{t-1} - \gamma_{nm_{n-1}}^N \sum_{l=0}^{t-2} \varepsilon^l}, \quad j = i, \dots, n, \quad t \in [m_{n-1} + 1, m_n].$$

Lets move to the time interval $[m_{n-2} + 1, m_{n-1}]$. Here two players (n and $n - 1$) exploit the resource stock.

The player $n - 1$'s objective function for the $m_n - m_{n-2} + 1$ -step game and the first criterium is

$$V_{n-1m_{n-1}}^1(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) = \gamma_{n-1m_{n-1}} x,$$

and for the second one

$$V_{n-1m_{n-1}}^2(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) = -\gamma_{n-1m_{n-1}}^2 x^2.$$

The player n 's objective functions are

$$\begin{aligned} & V_{nm_{n-1}}^1(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) = \gamma_{nm_{n-1}} x \\ & + \delta V_{nm_{n-1}+1}(\gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; (\varepsilon - \gamma_{nm_{n-1}} - \gamma_{n-1m_{n-1}})x) \\ & = \gamma_{nm_{n-1}} + (\varepsilon - \gamma_{nm_{n-1}} - \gamma_{n-1m_{n-1}})x \cdot \\ & \cdot (\delta \gamma_{nm_{n-1}+1} + \delta^2 \gamma_{nm_{n-1}+2} (\varepsilon - \gamma_{nm_{n-1}+1}) + \dots + \delta^{m_n} \gamma_{nm_n} \prod_{j=m_{n-1}+1}^{m_n-1} (\varepsilon - \gamma_{nj})), \end{aligned}$$

$$\begin{aligned}
 &V_{nm_{n-1}}^2(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) = -\gamma_{nm_{n-1}}^2 x^2 \\
 &+ \delta V_{nm_{n-1}+1}(\gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; (\varepsilon - \gamma_{nm_{n-1}} - \gamma_{n-1m_{n-1}})x) \\
 &= -\gamma_{nm_{n-1}}^2 x^2 + (\varepsilon - \gamma_{nm_{n-1}} - \gamma_{n-1m_{n-1}})^2 x^2 \cdot \\
 &\cdot (-\delta \gamma_{nm_{n-1}+1}^2 - \delta^2 \gamma_{nm_{n-1}+2}^2 (\varepsilon - \gamma_{nm_{n-1}+1})^2 - \dots - \delta^{m_n} \gamma_{nm_n}^2 \prod_{j=m_{n-1}+1}^{m_n-1} (\varepsilon - \gamma_{nj})^2).
 \end{aligned}$$

To determine noncooperative strategies for this $m_n - m_{n-2} + 1$ -step game we solve the following problem

$$\begin{aligned}
 &(V_{n-1m_{n-1}}^1(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) - A_{m_{n-1}}x) \cdot \\
 &\cdot (V_{n-1m_{n-1}}^2(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) - G_{m_{n-1}}x^2) \rightarrow \max_{\gamma_{n-1m_{n-1}}}, \\
 &(V_{nm_{n-1}}^1(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) - (A_{m_{n-1}} + A_{m_n})x) \cdot \\
 &\cdot (V_{nm_{n-1}}^2(\gamma_{n-1m_{n-1}}, \gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}; x) - (G_{m_{n-1}} + G_{m_n})x^2) \rightarrow \max_{\gamma_{nm_{n-1}}, \gamma_{nm_{n-1}+1}, \dots, \gamma_{nm_n}}. \tag{18}
 \end{aligned}$$

Continuing the process from the first-order conditions we obtain the relations (16) for the players n and $n - 1$'s noncooperative strategies on time interval $[m_{n-2} + 1, m_{n-1}]$.

By applying the described scheme on each time interval $[m_{j-1} + 1, m_j]$, $j = 1, \dots, n - 2$, we obtain the relations (16).

Corollary 1. *The players with smaller planning horizon get more gain from the exploitation process.*

Proof. As the players' strategies on each time interval $[m_{i-1} + 1, m_i]$, $i \in N$, are expressed via the strategies at the first stage (16) let us compare γ_{j1} and γ_{j+11} , $j = 1, \dots, n - 1$.

γ_{j1} is the solution of the next equation

$$3\gamma_{j1}^2 \sum_{k=0}^{m_j-1} \delta^k + 2\gamma_{j1} \sum_{l=1}^j A_{m_l} + \sum_{l=1}^j G_{m_l} = 0, \tag{19}$$

while γ_{j+11} is defined from

$$3\gamma_{j+11}^2 \sum_{k=0}^{m_{j+1}-1} \delta^k + 2\gamma_{j+11} \sum_{l=1}^{j+1} A_{m_l} + \sum_{l=1}^{j+1} G_{m_l} = 0. \tag{20}$$

Subtracting (20) from (19) we get

$$3(\gamma_{j1}^2 - \gamma_{j+11}^2) \sum_{k=0}^{m_j-1} \delta^k - 3\gamma_{j+11}^2 \sum_{k=m_j}^{m_{j+1}-1} \delta^k - 2\gamma_{j+11} A_{m_{j+1}} - G_{m_{j+1}} = 0,$$

that yields

$$\gamma_{j1} - \gamma_{j+11} = \frac{3\gamma_{j+11}^2 \sum_{k=m_j}^{m_{j+1}-1} \delta^k + 2\gamma_{j+11}A_{m_{j+1}} + G_{m_{j+1}}}{3(\gamma_{j1} + \gamma_{j+11}) \sum_{k=0}^{m_j-1} \delta^k} > 0.$$

Hence, on each time interval $t \in [m_{i-1} + 1, m_i]$, for $i \in N, j = i, \dots, n$,

$$u_{jt} - u_{j+1t} = (\gamma_{jt} - \gamma_{j+1t})x_t = \frac{\gamma_{j1} - \gamma_{j+11}}{\varepsilon^{t-1} - \sum_{k=1}^{i-1} \gamma_{k1}^N \sum_{l=t-m_k-1}^{t-2} \varepsilon^l - \sum_{k=i}^n \gamma_{k1}^N \sum_{l=0}^{t-2} \varepsilon^l} > 0, \tag{21}$$

that means that the player j with planning horizon m_j exploit the stock more intensively that the player $j + l$ with larger planning horizon $m_{j+l} > m_j, l = 1, \dots, n - j$.

Corollary 2. *The players exploitation rates decrease in time.*

Proof. Let us consider time interval $[m_{i-1}, m_i], i \in N$, and compare u_{jt} and $u_{j,t+1}, j = i, \dots, n$:

$$\begin{aligned} u_{jt} - u_{j,t+1} &= \gamma_{jt}x_t - \gamma_{j,t+1}x_{t+1} = \gamma_{jt}x_t - \gamma_{j,t+1}(\varepsilon x_t - \sum_{l=i}^n \gamma_{lt}x_t) \\ &= x_t(\gamma_{jt} - \varepsilon\gamma_{j,t+1} + \gamma_{j,t+1} \sum_{l=i}^n \gamma_{lt}) = x_t(-\gamma_{j,t+1} \sum_{l=i}^n \gamma_{lt} + \gamma_{j,t+1} \sum_{l=i}^n \gamma_{lt}) = 0 \end{aligned} \tag{22}$$

as

$$\begin{aligned} & \gamma_{jt} - \varepsilon\gamma_{j,t+1} \\ &= \frac{-\gamma_{j1} \sum_{k=i}^n \gamma_{k1}}{(\varepsilon^{t-1} - \sum_{k=1}^{i-1} \gamma_{k1}^N \sum_{l=t-m_k-1}^{t-2} \varepsilon^l - \sum_{k=i}^n \gamma_{k1}^N \sum_{l=0}^{t-2} \varepsilon^l)(\varepsilon^t - \sum_{k=1}^{i-1} \gamma_{k1}^N \sum_{l=t-m_k}^{t-1} \varepsilon^l - \sum_{k=i}^n \gamma_{k1}^N \sum_{l=0}^{t-1} \varepsilon^l)} \\ &= -\gamma_{j,t+1} \sum_{l=i}^n \gamma_{lt}. \end{aligned}$$

Hence,

$$\gamma_{jt} - \gamma_{j,t+1}(\varepsilon - \sum_{l=i}^n \gamma_{lt}) = 0$$

and

$$\gamma_{j,t+1} < \gamma_{jt}. \tag{23}$$

The same reasoning for the neighboring intervals $[m_{i-1} + 1, m_i]$ and $[m_i + 1, m_{i+1}]$ leads to

$$\gamma_{j,t+m_i} < \gamma_{jt}, j = i + 1, \dots, n, t \in [m_i - m_{i-1}, m_{i+1} - m_i].$$

3.2 Cooperative Equilibrium

To construct the cooperative payoffs and strategies the modified bargaining scheme [16] is applied. First, we have to determine the noncooperative payoffs as the ones gained by the players using the multicriteria Nash strategies. Then, we construct the sum of the Nash products with the noncooperative payoffs of players acting as the status quo points.

In view of Proposition 1, the noncooperative payoffs have the forms

$$J_i^{1N}(x) = \sum_{t=m_0+1}^{m_i} \delta^t p_i \gamma_{it}^N x,$$

$$J_i^{2N}(x) = -c \sum_{t=m_0+1}^{m_i} \delta^t (\gamma_{it}^N)^2 x^2, \quad i \in N.$$

In accordance with Definition 2, for designing the multicriteria cooperative equilibrium the following problem has to be solved:

$$p_1 \left(\sum_{t=m_0+1}^{m_1} \delta^t u_{1t}^c - B_{m_1} x \right) \left(- \sum_{t=m_0+1}^{m_1} \delta^t (u_{1t}^c)^2 + K_{m_1} x^2 \right) \dots$$

$$+ p_n \left(\sum_{t=m_0+1}^{m_n} \delta^t u_{nt}^c - \sum_{j=1}^n B_{m_j} x \right) \left(- \sum_{t=m_0+1}^{m_n} \delta^t (u_{nt}^c)^2 + \sum_{j=1}^n K_{m_j} x^2 \right) \rightarrow \max_{u_{1t}^c, \dots, u_{nt}^c},$$

where $B_{m_j} = \sum_{t=m_{j-1}+1}^{m_j} \delta^t \gamma_{jt}^N$, $K_{m_j} = \sum_{t=m_{j-1}+1}^{m_j} \delta^t (\gamma_{jt}^N)^2$, $j = 1, \dots, n$.

Considering the process starting from one-stage game to m_n -stage, similarly to the Proposition 1 we construct cooperative behavior.

Proposition 2. *The multicriteria cooperative equilibrium strategies in problem (11), (12) take the form $u_{it}^c = \gamma_{it}^c x_t$, $i \in N$, $t \in [m_0 + 1, m_n]$*

$$\gamma_{jt}^c = \frac{\gamma_{j1}^c}{\varepsilon^{t-1} - \sum_{k=1}^{i-1} \gamma_{k1}^c \sum_{l=t-m_k-1}^{t-2} \varepsilon^l - \sum_{k=i}^n \gamma_{k1}^c \sum_{l=0}^{t-2} \varepsilon^l}, \quad j = i, \dots, n, \quad t \in [m_{i-1} + 1, m_i]. \quad (24)$$

where the players' strategies at the first stage are

$$\gamma_{j1}^c = \frac{- \sum_{l=1}^j B_{m_l} + \sqrt{\left(\sum_{l=1}^j B_{m_l} \right)^2 - 3 \sum_{h=0}^{m_j-1} \delta^h \sum_{l=1}^j K_{m_l}}}{3 \sum_{h=0}^{m_j-1} \delta^h}.$$

Remark 1. As the cooperative strategies possess the similar to the noncooperative ones relations (16) the Corollaries 1 and 2 are also fulfilled for cooperative behavior.

Proposition 3. *The time-consistent payoff distribution procedure in the problem (11), (12) takes the form*

$$\beta_i(t, x_t) = \begin{pmatrix} \beta_i^1(t, x_t) \\ \beta_i^2(t, x_t) \end{pmatrix}, \quad i \in N, \quad t \in [m_0 + 1, m_i],$$

where

$$\beta_i^1(t, x_t) = \delta^t p_i \gamma_{it}^c x_t + p_i (1 - \delta) \sum_{\tau=t+1}^{m_i} \delta^\tau \gamma_{i\tau}^c x_\tau,$$

$$\beta_i^2(t, x) = -c \delta^t (\gamma_{it}^c)^2 x_t^2 - c (1 - \delta) \sum_{\tau=t+1}^{m_i} \delta^\tau (\gamma_{i\tau}^c)^2 x_\tau^2.$$

Proof. Follows from Theorem 1 and the form of cooperative payoffs.

4 Conclusions

The multicriteria dynamic game with asymmetric planning horizons has been investigated. The multicriteria Nash and cooperative equilibria have been constructed via the modified bargaining schemas. The proposed approaches capture the possibility of the players' leaving the game. We have adopted the concept of dynamic stability for multicriteria dynamic games with different planning horizons and have constructed the time-consistent payoff distribution procedure.

To illustrate the presented approaches, we have studied a bi-criteria discrete-time bioresource management problem with asymmetric planning horizons. Multicriteria Nash and cooperative equilibria strategies as well as the time-consistent payoff distribution procedure have been derived analytically. It was proved that the proposed solution concepts give the players with smaller planning horizons more gain from resource exploitation during their operation times.

The results presented in this paper can be applied in biological, economic and other game-theoretic models with vector payoffs.

References

1. Ghose, D., Prasad, U.R.: Solution concepts in two-person multicriteria games. *J. Optim. Theory Appl.* **63**, 167–189 (1989)
2. Ghose, D.: A necessary and sufficient condition for Pareto-optimal security strategies in multicriteria matrix games. *J. Optim. Theory Appl.* **68**, 463–481 (1991)
3. Gromova, E.V., Plekhanova, T.M.: On the regularization of a cooperative solution in a multistage game with random time horizon. *Discret. Appl. Math.* **255**, 40–55 (2019)
4. Haurie, A.: A note on nonzero-sum differential games with bargaining solution. *J. Optim. Theory Appl.* **18**, 31–39 (1976)
5. Kuziyutin, D., Nikitina, M.: Time consistent cooperative solutions for multistage games with vector payoffs. *Oper. Res. Lett.* **45**(3), 269–274 (2017)

6. Kuzyutin, D., Smirnova, N., Gromova, E.: Long-term implementation of the cooperative solution in a multistage multicriteria game. *Oper. Res. Perspect.* **6**, 100107 (2019)
7. Kuzyutin, D., Gromova, E., Smirnova, N.: On the cooperative behavior in multistage multicriteria game with chance moves. In: Kononov, A., Khachay, M., Kalyagin, V.A., Pardalos, P. (eds.) *MOTOR 2020*. LNCS, vol. 12095, pp. 184–199. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49988-4_13
8. Marin-Solano, J.: Group inefficiency in a common property resource game with asymmetric players. *Econ. Lett.* **136**, 214–217 (2015)
9. Mazalov, V.V., Rettieva, A.N.: Asymmetry in a cooperative bioresource management problem. In: *Game-Theoretic Models in Mathematical Ecology*, pp. 113–152. Nova Science Publishers (2015)
10. Parilina, E.M., Zaccour, G.: Node-consistent Shapley value for games played over event trees with random terminal time. *J. Optim. Theory Appl.* **175**(1), 236–254 (2017). <https://doi.org/10.1007/s10957-017-1148-6>
11. Petrosjan, L.A.: Stable solutions of differential games with many participants. *Viestnik Leningrad Univ.* **19**, 46–52 (1977)
12. Petrosjan, L.A., Danilov, N.N.: Stable solutions of nonantagonistic differential games with transferable utilities. *Viestnik Leningrad Univ.* **1**, 52–59 (1979)
13. Pusillo, L., Tijss, S.: E-equilibria for multicriteria games. *Ann. ISDG* **12**, 217–228 (2013)
14. Rettieva, A.N.: Equilibria in dynamic multicriteria games. *Int. Game Theory Rev.* **19**(1), 1750002 (2017)
15. Rettieva, A.N.: Dynamic multicriteria games with finite horizon. *Mathematics* **6**(9), 156 (2018)
16. Rettieva, A.N.: Dynamic multicriteria games with asymmetric players. *J. Glob. Optim.* **1**, 1–17 (2020). <https://doi.org/10.1007/s10898-020-00929-5>
17. Rettieva, A.: Rational behavior in dynamic multicriteria games. *Mathematics* **8**, 1485 (2020)
18. Rettieva, A.N.: Cooperation in dynamic multicriteria games with random horizons. *J. Global Optim.* **76**(3), 455–470 (2018). <https://doi.org/10.1007/s10898-018-0658-6>
19. Rettieva, A.: Multicriteria dynamic games with random horizon. In: Pardalos, P., Khachay, M., Kazakov, A. (eds.) *MOTOR 2021*. LNCS, vol. 12755, pp. 340–355. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77876-7_23
20. Shapley, L.S.: Equilibrium points in games with vector payoffs. *Naval Res. Log. Quart.* **6**, 57–61 (1959)
21. Sorger, G.: Recursive Nash bargaining over a productive asset. *J. Econ. Dyn. Control* **30**, 2637–2659 (2006)
22. Voorneveld, M., Grahn, S., Dufwenberg, M.: Ideal equilibria in noncooperative multicriteria games. *Math. Methods Oper. Res.* **52**, 65–77 (2000)
23. Yeung, D.W.K., Petrosyan, L.A.: Subgame consistent cooperative solutions for randomly furcating stochastic dynamic games with uncertain horizon. *Int. Game Theory Rev.* **16**(2), 1440012 (2014)
24. Zeleny, M.: *Compromising programming, multiple criteria decision making*. University of South Carolina Press, Columbia (1973)



A Novel Payoff Distribution Procedure for Sustainable Cooperation in an Extensive Game with Payoffs at All Nodes

Denis Kuzyutin^{1,2}  and Nadezhda Smirnova²  

¹ Saint Petersburg State University, Universitetskaya nab. 7/9,
199034 St. Petersburg, Russia

d.kuzyutin@spbu.ru

² HSE University, Soyuza Pechatnikov ul. 16, 190008 St. Petersburg, Russia
nvsmirnova@hse.ru

Abstract. This paper is a contribution to the problem of sustainable cooperation in an extensive-form game. We study an extension of the subgame-perfect core concept to more broad class of games (when the payoffs are defined at all nodes) which is based on the payoff distribution procedure approach. The properties of this β -S-P Core are studied and algorithm of its implementation in a 2-player game is provided. Using this algorithm we construct specific payoff distribution procedure from the β -S-P Core in an extensive-form version of the fishery-management model with asymmetric players.

Keywords: Extensive-form game · Subgame-perfect equilibrium · Payoff distribution procedure · Cooperative solution · Core · Fishery-management model

1 Introduction

The problem of a long-term sustainable cooperation designing is an important issue in the theory of dynamic games and their applications in economics (see, e.g., [4, 7, 10, 18, 20, 27, 33] and the references therein). A recent solution called subgame-perfect core (S-P Core) for games in extensive form (with only terminal payoffs) that takes into account both sustainable cooperation motivation and subgame perfection property [29] was proposed in [6]. When considering possible deviations from the cooperative agreement in a subgame along the cooperative history the S-P Core focuses only on the players which still have decision nodes in the current subgame [6, 14, 15, 27] (so-called “active” players). The S-P Core consists of such distributions of the total cooperative payoff (in the corresponding terminal nodes) that no active coalition can implement a joint profitable

The reported study was funded by RSF according to the research project 22-28-01221 (<https://rscf.ru/project/22-28-01221/>).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. Pardalos et al. (Eds.): MOTOR 2022, LNCS 13367, pp. 279–294, 2022.
https://doi.org/10.1007/978-3-031-09607-5_20

deviation from the cooperative agreement as a game unfolds along the cooperative path. An extension of the S-P Core to specific n -person discrete-time game of climate change with current payoffs was introduced in [4].

We provide an extension of the S-P Core concept to more general class of extensive-form games, when the payoffs are defined at all nodes of the game tree (such assumption is more natural when modelling different real-life situations (see, e.g., [4, 10, 13, 27])). This extension (called β -S-P Core) is based on designing a certain payoff distribution procedure (PDP) β [10, 26, 27, 33] (now the players redistribute the current total payoff at each position in the cooperative history). It is worth noting that PDP-based approach introduced for differential games in [26] has been extended to various models of dynamic games (see, e.g., [8, 10, 21, 25, 27, 33]) since it provides a powerful tool to sustain a long-term cooperative agreement. A novel solution that incorporates both the core concept and the PDP technique was introduced in [28].

We prove that β -S-P Core satisfies several advantageous properties (see Prop. 1–2, Remark 3) for class of extensive games under consideration and provides a useful mechanism for the players to sustain the cooperative agreement. To choose a specific PDP from the β -S-P Core we adopt an optimization based approach which aims to maximize the relative benefit from cooperation of the least winning player (see, e.g., [23]). Lastly, for 2-person extensive-form game we provide an algorithm to construct this specific PDP and consider how one can apply the β -S-P Core concept to the analysis of an extensive-form version of known fishery-management model with asymmetric players [2, 10, 18].

The rest of the paper is organised as follows. The specification of an extensive-form game with payoffs defined at all nodes is recalled in Sect. 2. In Sect. 3 we derive the β -S-P Core definition which is based on the payoff distribution procedure approach. The strategic and coalitional support of the β -S-P Core are examined in Sect. 4. In addition, in Sect. 4, we introduce a rule for the β -S-P Core refinement and provide an algorithm how to construct a specific PDP from the β -S-P Core. We apply the suggested mechanism of sustainable cooperation in Sect. 5 to examine a two-player asymmetric extensive-form fishery management model. Section 6 is a conclusion.

2 Extensive-Form Game with Current Payoffs at Each Node

We need to remind the basic notations and assumptions that are accepted in the theory of finite extensive-form games (see, e.g., [12, 15, 16, 27] for details). Let $N = \{1, \dots, n\}$ be the players' set; K denotes the game tree with the original node (root) x_0 and the set of all nodes (positions) P ; $S(x)$ denotes the set of direct successors of intermediate node x .

Let P_i denote the finite set of decision nodes of player i (where this player “moves”), $P_i \cap P_j = \emptyset$, for all $i, j \in N$, $i \neq j$, while $P_{n+1} = \{z^j\}_{j=1}^m$ is the set of terminal nodes, $S(z^j) = \emptyset \forall z^j \in P_{n+1}$. Denote by $\omega = (x_0, \dots, x_{t-1}, x_t, \dots, x_T)$ the trajectory in the game tree (also called the path or the history), $x_{t-1} =$

$S^{-1}(x_t)$, $1 \leq t \leq T$, $x_T = z^j \in P_{n+1}$; where subscript t in x_t indicates the number of this position in the trajectory ω . Let the payoff $h_i(x)$ of the i th player, $i \in N$ be defined at all nodes $x \in P$. Lastly, we consider the case of non-negative current payoffs, that is, $h_i(x) \geq 0 \forall i \in N, x \in P$.

In the following, we use $G^P(n)$ to denote the class of extensive-form n -player games with perfect information (see, e.g., [10,12,27] for details), where $\Gamma^{x_0} \in G^P(n)$ is a game with original node x_0 . The pure strategy $u_i(\cdot)$ of the i th player is a function that determines for every position $x \in P_i$ the subsequent position $u_i(x) \in S(x)$ the player i is going to choose at x . Denote by U_i the finite set of all possible strategies of the player i , while $U = \prod_{i \in N} U_i$ denote a corresponding set of strategy profiles. Each pure strategy profile $u = (u_1, \dots, u_n) \in U$ determines a unique history $\omega(u) = (x_0, \dots, x_t, x_{t+1}, \dots, x_T) = (x_0, x_1(u), \dots, x_t(u), x_{t+1}(u), \dots, x_T(u))$, where $x_{t+1} = u_j(x_t) \in S(x_t)$ if $x_t \in P_j$, $0 \leq t \leq T - 1$, $x_T \in P_{n+1}$, and, hence, a set of all the players' payoffs. For any strategy profile u the value of the player i 's payoff (objective) function is determined as follows:

$$H_i(u) = \tilde{h}_i(\omega(u)) = \sum_{\tau=0}^T h_i(x_\tau(u)).$$

According to [10,12,27] each intermediate position $x_t \in P \setminus P_{n+1}$ forms a subgame Γ^{x_t} with the subgame tree K^{x_t} and the subgame root x_t . Let $P_i^{x_t}$, $i \in N$, denote the restriction of P_i on K^{x_t} , while $u_i^{x_t}$, $i \in N$, is the restriction of the i th player's strategy $u_i(\cdot)$ in Γ^{x_0} on the set $P_i^{x_t}$. The strategy profile $u^{x_t} = (u_1^{x_t}, \dots, u_n^{x_t})$ generates the history $\omega^{x_t}(u^{x_t}) = (x_t, x_{t+1}, \dots, x_T) = (x_t, x_{t+1}(u^{x_t}), \dots, x_T(u^{x_t}))$ in the subgame and, hence, a set of the player's payoffs in Γ^{x_t} :

$$H_i^{x_t}(u^{x_t}) = \tilde{h}_i^{x_t}(\omega^{x_t}(u^{x_t})) = \sum_{\tau=t}^T h_i(x_\tau(u^{x_t})). \tag{1}$$

Note that (1) differs from the subgame payoff definition for extensive-form games with terminal payoffs (see [12,27] for details).

Definition 1. [24] A strategy profile $u = (u_1, u_2, \dots, u_n)$ constitutes a Nash Equilibrium (NE) in $\Gamma^{x_0} \in G^P(n)$ if $\forall i \in N$ and for any strategy $v_i \in U_i$ the following inequality holds: $H_i(v_i, u_{-i}) \leq H_i(u_i, u_{-i})$.

Definition 2. [29] A strategy profile u forms a subgame perfect equilibrium (SPE) in $\Gamma^{x_0} \in G^P(n)$ if $\forall x \in P \setminus P_{n+1}$ the restriction of u in the subgame Γ^x still constitutes a Nash equilibrium in the subgame, that is $u^x \in NE(\Gamma^x)$.

Note that every extensive-form game $\Gamma^{x_0} \in G^P(n)$ with payoffs defined at all nodes possesses pure strategy SPE (see, e.g., [27]).

3 Payoff Distribution Procedure and the Subgame-Perfect Core Concept

Let $\bar{\omega} = \bar{\omega}(\bar{u}) = (x_0 = \bar{x}_0, \dots, \bar{x}_t, \dots, \bar{x}_T)$ be a cooperative trajectory, i.e.

$$\max_{u \in U} \sum_{i \in N} H_i(u) = \sum_{i \in N} H_i(\bar{u}) = \sum_{i \in N} \sum_{\tau=0}^T h_i(\bar{x}_\tau) = \sum_{i \in N} \tilde{h}_i(\bar{\omega}). \tag{2}$$

For the sake of simplicity we focus on the case when either there exists a unique cooperative history in $\Gamma^{x_0} \in G^P(n)$ or the players employ a specific approach (for instance, the PRB algorithm introduced in [16]) to choose a unique cooperative history from all the trajectories $\bar{\omega}$ meeting (2). Note that such an approach should satisfy subgame consistency (see, e.g. [10, 16, 26, 27, 33]) that is a fragment of the cooperative path in the subgame $\Gamma^{\bar{x}_t}, \bar{x}_t \in \bar{\omega}$ has to remain cooperative history in this subgame. A vector $(p_1^{\bar{x}_t}, \dots, p_n^{\bar{x}_t})$ such that

$$\sum_{i \in N} p_i^{\bar{x}_t} = \sum_{i \in N} \sum_{\tau=t}^T h_i(\bar{x}_\tau) = \sum_{i \in N} \tilde{h}_i(\bar{\omega}^{\bar{x}_t}), \tag{3}$$

determines a possible sharing rule of the aggregated cooperative (subgame) payoff between the players and could be interpreted as a *cooperative solution* recognized in the subgame $\Gamma^{\bar{x}_t}, \bar{x}_t \in \bar{\omega}$.

Let $\beta = \{\beta_i(\bar{x}_\tau)\}, i = 1, \dots, n; \tau = 0, \dots, T; \bar{x}_\tau \in \bar{\omega}$ be the *Payoff Distribution Procedure (PDP)* which is designed for some cooperative solution $(p_1, \dots, p_n) = (p_1^{x_0}, \dots, p_n^{x_0})$ (see, e.g., [10, 13, 26, 27, 33]). The PDP approach implies that all the participants have agreed to split the aggregated cooperative payoff in Γ^{x_0} among the players according to the vector (p_1, \dots, p_n) and, moreover, to allocate every player’s cooperative payoff p_i along the cooperative history $\bar{\omega}$ in accordance with some specific payment schedule which is called PDP. Then, $\beta_i(\bar{x}_\tau)$ specifies the actual payment that the player i has to receive at position $\bar{x}_\tau \in \bar{\omega}$ instead of $h_i(\bar{x}_\tau)$ when the players use PDP β . We aim to design such a PDP β that all the coalitions $S \subset N$ will have an incentive to follow a sharing rule $(p_1^{\bar{x}_t}, \dots, p_n^{\bar{x}_t})$ in any subgame $\Gamma^{\bar{x}_t}, \bar{x}_t \in \bar{\omega}$ along the cooperative history. Denote by $\tilde{\beta}_i(\bar{\omega}^{\bar{x}_t})$ the total payment which the i -th player is expected to gain in accordance with PDP β in the subgame $\Gamma^{\bar{x}_t}$, i.e.

$$\tilde{\beta}_i(\bar{\omega}^{\bar{x}_t}) = \tilde{\beta}_i(\bar{x}_t, \bar{x}_{t+1}, \dots, \bar{x}_T) = \sum_{\tau=t}^T \beta_i(\bar{x}_\tau).$$

Consider such PDP β that satisfy the following conditions (see [13, 16, 26, 27] for details):

Definition 3. [16] *The PDP β for the payoff vector (p_1, \dots, p_n) meeting (3) satisfies the subgame efficiency property if for all $\bar{x}_t \in \bar{\omega} = (\bar{x}_0, \dots, \bar{x}_T)$ and for all $i \in N$*

$$\tilde{\beta}_i(\bar{\omega}^{\bar{x}_t}) = p_i^{\bar{x}_t}. \tag{4}$$

Condition (4) for $t = 0$ (called the efficiency in the original game [16,26,27]) means that PDP β could be reasonably considered as a time schedule for the i th player's payoff p_i allocation.

Definition 4. [13,26,27] *The PDP $\beta = \{\beta_i(\bar{x}_\tau)\}$ meets the strict balance condition if for every position $\bar{x}_\tau \in \bar{\omega}, \tau = 0, \dots, T$*

$$\sum_{i \in N} \beta_i(\bar{x}_\tau) = \sum_{i \in N} h_i(\bar{x}_\tau). \tag{5}$$

If (5) holds a rule β could be implemented by the players without any loans or credits.

Remark 1. Note that if the PDP β meets the subgame efficiency condition (4) it necessarily satisfies the strict balance constraint (5).

Lastly, we suppose that PDP β meets non-negativity condition, i.e.

$$\beta_i(\bar{x}_\tau) \geq 0 \quad \forall i \in N \quad \forall t = 0, \dots, T. \tag{6}$$

The player i is called *active* in Γ^x if he/she still has a decision position in Γ^x , i.e. $P_i \cap K^x \neq \emptyset$ (see [6,14,15,27]). Following [6], we suppose that coalition $S \subset N$ is *active at x* if each player $i \in S$ is active in Γ^x . For each coalition $S \subset N$ that is active at position x denote by $\Gamma^{x,S}$ the *induced game*, which differs from the original game Γ^x only in that coalition S becomes a new player while $h_S(y) = \sum_{i \in S} h_i(y), y \in P^x$. Lastly, we denote by $\gamma(S; x), x \in P \setminus P_{n+1}, S \subset N$ the highest possible SPE payoff of S in the induced subgame $\Gamma^{x,S}$.

Consider a coalition $S \subset N$ which follows a cooperative agreement (that is, a PDP β is implemented at each position in the cooperative history from the origin \bar{x}_0 till some intermediate position $\bar{x}_t \in \bar{\omega}, 1 \leq t \leq T - 1$), but decides to break down the cooperative agreement in the subgame $\Gamma^{\bar{x}_t}$ (S is active in $\Gamma^{\bar{x}_t}$). Then, the highest payoff a coalition S could get in the original game Γ^{x_0} is equal to $\sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau) + \gamma(S; \bar{x}_t)$. If we assume that $\sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau) + \gamma(S; \bar{x}_t) \leq \sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau) + \sum_{\tau=t}^T \beta_S(\bar{x}_\tau)$, then there is no reason for any active coalition S to deviate from the cooperative agreement at \bar{x}_t . Hence, we obtain a rather simple conditions that ensure subgame-consistency of the cooperative scenario (S is active coalition at \bar{x}_t):

$$\gamma(S; \bar{x}_t) \leq \sum_{\tau=t}^T \beta_S(\bar{x}_\tau), \quad \bar{x}_t \in \bar{\omega}, \quad t = 0, \dots, T - 1. \tag{7}$$

The concept of β -S-P Core was firstly introduced in [17], yet without providing the detailed analysis of its basic properties. The most important properties of the β -S-P Core (see Proposition 1 and 2 below) are proved in this paper.

Definition 5. [17] *The set of all payoff distribution procedures β which satisfy (4), (3), (5), (6), and (7) is called the β -subgame-perfect core (β -S-P Core) for a game $\Gamma^{x_0} \in G^P(n)$ in extensive form with payoffs defined at all nodes.*

The concept of subgame-perfect core introduced in [6] for games in extensive form with terminal payoffs implies that the players redistribute the highest total payoffs at corresponding terminal nodes to support the cooperative scenario. Note that the β -S-P Core in accordance with Def. 5 is an extension of the subgame-perfect core concept [6] to more general class of extensive games which allows the payoffs transfers (via appropriate PDP) in all nodes along the cooperative history and, hence, provides a powerful tool to sustain the cooperative agreement. Since the properties of the payoff distribution procedure β are incorporated in Def. 5 one may refer to the S-P Core for the class of extensive-form games $G^P(n)$ with payoffs defined at all nodes as the β -S-P Core or the S-P Core that is based on the payoff distribution procedure. Note that the subgame-perfect cooperative agreements suggested in [4] for specific discrete-time dynamic game of climate change also imply payoff transfers between countries in each period.

4 Properties and Implementation of the β -S-P Core

If a game $\Gamma^{x_0} \in G^P(n)$ possesses non-empty β -S-P Core then any particular PDP β from the core generates a related extensive-form game $\Gamma_\beta^{x_0}$ that differs from the original game Γ^{x_0} only in that the payoffs $h_i(\bar{x}_t)$, $i \in N$ at every position $\bar{x}_t \in \bar{\omega}$ along the cooperative history are replaced by $\beta_i(\bar{x}_t)$. Note that similar approach was used earlier, in particular, in [27] to define a “regularized game” for differential or extensive-form game with payoffs at all nodes and also in [6] to introduce a “strategic transform” of an extensive-form game with only terminal payoffs.

The following proposition is an extension of the Proposition 1 from [6] to more general class of extensive games $G^P(n)$. The difference is that now we need to adopt distinct subgame payoff definition (1) and use more general concept of the related non-cooperative game $\Gamma_\beta^{x_0}$. This proposition was firstly provided (yet without proof) in [17].

Proposition 1. *Let the β -S-P Core of a game $\Gamma^{x_0} \in G^P(n)$ in extensive form with payoffs defined at all nodes be non-empty, while $\beta = \beta_i(\bar{x}_t)$, $i \in N$, $\bar{x}_t \in \bar{\omega}$, is a PDP from the β -S-P Core. Then there exists a subgame-perfect equilibrium $\underline{u} = (\underline{u}_1, \dots, \underline{u}_n)$ in a related extensive-form game $\Gamma_\beta^{x_0}$ that generates the same (cooperative) trajectory $\bar{\omega} = (\bar{x}_0, \dots, \bar{x}_T)$ with a resulting SPE players’ payoffs*

$$\text{vector } H_i(\underline{u}) = \sum_{t=0}^T \beta_i(\bar{x}_t) = \tilde{\beta}_i(\bar{\omega}), \quad i = 1, \dots, n.$$

Proof. Let $\bar{\omega} = \bar{\omega}(\bar{u}) = (\bar{x}_0, \dots, \bar{x}_t, \bar{x}_{t+1}, \dots, \bar{x}_T)$ denote a unique cooperative history in Γ^{x_0} meeting (2) while $\bar{u}_i(\bar{x}_t) = \bar{x}_{t+1}$ denote a “correct choice” of the i -th player in her / his decision node $\bar{x}_t \in P_i$ according to the cooperative scenario $\bar{u} = (\bar{u}_1, \dots, \bar{u}_n)$.

Given payoff distribution procedure $\beta = \{\beta_i(\bar{x}_\tau)\}$, $i \in N$, $\tau = 0, \dots, T$ meeting (3)–(6) and (7) we employ a backwards induction procedure (see, for instance, [11, 27, 29]) to construct a SPE $\underline{u} = (u_1, \dots, u_n)$ in a related non-cooperative game $\Gamma_\beta^{x_0}$. Note that for any intermediate node $x \in P \setminus P^{n+1}$ apart from the cooperative history (i.e., $x \notin \bar{\omega}$) the subgame Γ_β^x and Γ^x coincide since a replacing of the payoffs in $\bar{x}_t, \bar{x}_t \in \bar{\omega}$ does not affect the payoffs in the subgame starting at x . Hence, we focus on the SPE strategies $\underline{u}_i(\cdot)$ construction only in nodes \bar{x}_t along the cooperative history $\bar{\omega}$.

Step 1. Consider subgames $\Gamma_\beta^{\bar{x}_{T-1}}$ and $\Gamma^{\bar{x}_{T-1}}$ of the length 1. Let $\bar{x}_{T-1} \in P_i$. Then $S = \{i\}$ is a coalition which is active at \bar{x}_{T-1} , and (7) takes the form:

$$\beta_i(\bar{x}_{T-1}) + \beta_i(\bar{x}_T) \geq \gamma(\{i\}, \bar{x}_{T-1}). \tag{8}$$

Since the players' payoffs in the related subgame $\Gamma_\beta^{\bar{x}_{T-1}}$ at all but one terminal node \bar{x}_T are the same as in $\Gamma^{\bar{x}_{T-1}}$, and $\gamma(\{i\}, \bar{x}_{T-1})$ denotes to the highest i -th player's SPE payoff in $\Gamma^{\bar{x}_{T-1}}$, inequality (8) implies that the choice $\underline{u}_i(\bar{x}_{T-1}) = \bar{x}_T$ corresponds to the SPE behavior of the i -th player in the related subgame $\Gamma_\beta^{\bar{x}_{T-1}}$.

Step t (t = 2, ..., T). Consider subgames $\Gamma_\beta^{\bar{x}_{T-t}}$ and $\Gamma^{\bar{x}_{T-t}}$ of the length t assuming that the SPE $\underline{u} = (u_1, \dots, u_n)$ has been already constructed in all the related subgames $\Gamma_\beta^{\bar{x}_{T-\tau}}$, $1 \leq \tau < t$. Let $\bar{x}_{T-t} \in P_j$. Then, applying (7) for $S = \{j\}$ we get:

$$\beta_j(\bar{x}_{T-t}) + \beta_j(\bar{x}_{T-t+1}) + \dots + \beta_j(\bar{x}_T) \geq \omega(\{j\}, \bar{x}_{T-t}). \tag{9}$$

Since the players' payoffs in the related subgame $\Gamma_\beta^{\bar{x}_{T-t}}$ along all but one history $\bar{\omega}(\bar{x}_{T-t}) = (\bar{x}_{T-t}, \bar{x}_{T-t+1}, \dots, \bar{x}_T)$ are the same as in $\Gamma^{\bar{x}_{T-t}}$, and $\gamma(\{j\}, \bar{x}_{T-t})$ is the highest SPE payoff the j -th player can achieve in $\Gamma^{\bar{x}_{T-t}}$, inequality (9) ensures that the choice $\underline{u}_j(\bar{x}_{T-t}) = \bar{x}_{T-t+1}$ corresponds to the SPE behavior of player j at position \bar{x}_{T-t} of the related subgame $\Gamma_\beta^{\bar{x}_{T-t}}$.

Hence, we designed (via backwards induction procedure) a strategy profile $\underline{u} = (u_1, \dots, u_n)$ which constitutes a SPE in the related game $\Gamma_\beta^{x_0}$. Note that for all $j \in N$ by the construction $\underline{u}_j(\bar{x}_t) = \bar{u}_j(\bar{x}_t)$ at each node $\bar{x}_t \in \bar{\omega} \cap P_j$. Therefore, subgame perfect equilibria (u_1, \dots, u_n) for $\Gamma_\beta^{x_0}$ generates cooperative trajectory $\bar{\omega}$ and the players' payoffs $\tilde{\beta}_i(\bar{\omega}) = \sum_{\tau=0}^T \beta_i(\bar{x}_\tau)$, $i \in N$.

An extensive-form game $\Gamma \in G^P(n)$ can be converted into a coalitional (cooperative) game by defining a worth (or a characteristic) function for each coalition $S \subset N$. We adopt in the paper the concept of γ -characteristic function introduced in [5] which implies that the guaranteed payoff of coalition S in Γ^{x_0} is the Nash equilibria payoff which this coalition is expected to get in the induced game $\Gamma^{x_0, S}$.

Suppose that the players consider some cooperative solution $(p_1^{x_0}, \dots, p_n^{x_0})$ meeting (3) for $t = 0$, that is a distribution of the total cooperative payoff

$$v(N) = v^{x_0}(N) = \sum_{i \in N} \tilde{h}_i(\bar{\omega}) \tag{10}$$

in $\Gamma^{x_0} \in G^P(n)$ while β be any payoff distribution procedure for vector $(p_1^{x_0}, \dots, p_n^{x_0})$.

For any coalition $S \subset N, S \neq N$, one can calculate the value

$$v(S) = v^{x_0}(S) = \max_{\bar{x}_t \in \bar{\omega} \setminus \{\bar{x}_T\}} \left(\sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau) + \gamma(S; \bar{x}_t) \right), \tag{11}$$

while the maximum has to be taken only over those decision positions \bar{x}_t in the cooperative history $\bar{\omega}$ at which S is active. Note that (11) determines the highest guaranteed payoff a coalition S could receive in the original game Γ^{x_0} given all possible behavior schemes with one switching from cooperative scenario to non-cooperative one (see also the paragraph above formula (7)).

Note that for any $(p_1^{x_0}, \dots, p_n^{x_0})$ meeting (3) and any PDP β function (10), (11) is well-defined for every coalition $S \subset N$ since every coalition is active at x_0 . Hereafter we will consider a coalitional game (v, N) with characteristic function (10), (11).

Definition 6. (see., e.g., [10, 27, 30]). *The core of the coalitional game (v, N) is a payoff vector $(p_1^{x_0}, \dots, p_n^{x_0})$ meeting (3) and the inequalities:*

$$\sum_{i \in S} p_i^{x_0} \geq v(S), \quad S \subset N. \tag{12}$$

The following proposition is an extension of the Proposition 2 from [6] to more broad class of extensive-form games $G^P(n)$ with payoffs at all nodes. The difference is that now we use characteristic function of special type that takes into account the dynamics of payments in an extensive game $\Gamma^{x_0} \in G^P(n)$. In addition, some properties of the PDP β (like subgame efficiency condition) are now involved in Prop. 2.

Proposition 2. *Let payoff distribution procedure β belong to the β -S-P Core of a game $\Gamma^{x_0} \in G^P(n)$ in extensive form with payoffs defined at all nodes. Then the corresponding vector $(p_1^{x_0}, \dots, p_n^{x_0})$ belongs to the core of the cooperative game (v, N) with characteristic function (10), (11).*

Conversely, let PDP β satisfy subgame efficiency condition (4) and non-negativity constraint (6) while the corresponding cooperative solution $(p_1^{x_0}, \dots, p_n^{x_0})$ is in the core of the cooperative game (v, N) with characteristic function (10), (11). Then PDP β belongs to the β -S-P Core of an extensive-form game $\Gamma^{\bar{x}_0}$.

Proof. If PDP β belongs to the β -S-P Core then according to (7) for any coalition $S \subset N$ and every node $\bar{x}_t \in \bar{\omega} \setminus \{\bar{x}_T\}$ such that S is active at \bar{x}_t the following inequality holds

$$\sum_{\tau=t}^T \beta_S(\bar{x}_\tau) \geq \gamma(S; \bar{x}_t).$$

If we add $\sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau)$ to both sides and take maximum of the right hand side over all nodes \bar{x}_t in the cooperative history $\bar{\omega}$ at which coalition S is active, we get in accordance with (11)

$$\sum_{i \in S} \tilde{\beta}_i(\bar{\omega}) \geq v(S).$$

Since $\tilde{\beta}_i(\bar{\omega}) = p_i^{x_0}$ due to (4) for $t = 0$ the vector $(p_1^{x_0}, \dots, p_n^{x_0})$ satisfies (12) and hence belongs to the core of (v, N) .

Conversely, if $(p_1^{x_0}, \dots, p_n^{x_0})$ belongs to the core of the cooperative game (v, N) then it follows from (12), (11) and (4) for $t = 0$ that for any coalition $S \subset N$ and every node $\bar{x}_t \in \bar{\omega} \setminus \{\bar{x}_T\}$ such that S is active at \bar{x}_t the following inequality holds:

$$\sum_{i \in S} \sum_{\tau=0}^T \beta_i(\bar{x}_\tau) \geq \sum_{\tau=0}^{t-1} \beta_S(\bar{x}_\tau) + \gamma(S; \bar{x}_t). \tag{13}$$

If we eliminate coinciding addends in both parts of (13) we get

$$\sum_{\tau=t}^T \beta_S(\bar{x}_\tau) \geq \gamma(S; \bar{x}_t),$$

hence, PDP β satisfies (7).

The strict balance constraint (5) holds due to Remark 1. Since all the conditions (4), (3), (5), (6), and (7) are valid the PDP β belongs to the β -S-P Core of Γ^{x_0} .

Remark 2. Proposition 2 provides a coalitional support to the β -S-P Core, namely any PDP β in the β -subgame-perfect core corresponds to the payoff distribution $(p_1^{x_0}, \dots, p_n^{x_0})$ from the core of certain coalitional game (v, N) .

Remark 3. Since the components $\beta_i(\bar{x}_t)$, $i \in N$, $t = 0, \dots, T$, of the PDP β from the β -S-P Core have to satisfy a finite number of non-strict linear inequalities (7), (6) and linear equations (3)–(5) a nonempty β -S-P Core is a convex polytope B in $R^{n \times (T+1)}$.

Obviously, the main concern of the j th player when choosing a specific PDP β from β -S-P Core is maximization of her / his cooperative payoff $p_j = \tilde{\beta}_j(\bar{\omega})$. We adopt in the paper an optimization-based approach for the refinement of the β -S-P Core, i.e. a specific rule for choosing $\tilde{\beta}_j(\bar{\omega})$, $j \in N$. This rule takes care of the relative benefit from cooperation (RBC) of the least winning player. Denote by Δ_j the range of the j -th player’s payoffs in Γ^{x_0} . The value $\frac{\tilde{\beta}_j - \gamma(\{j\}; x_0)}{\Delta_j}$ could be considered as the relative benefit of player j that is achieved due to cooperation. Note that the relative benefit of cooperation could be also interpreted as the "value of cooperation" (see, e.g. [9] for details). Hence, the above approach which

we will refer later as the *maxmin RBC rule* implies the solution of the following problem

$$\max_{\beta \in B} \min_{j \in N} \frac{\tilde{\beta}_j - \gamma(\{j\}; x_0)}{\Delta_j}. \tag{14}$$

Remark 4. In a game with only 2 players optimization problem (14) takes the form

$$\frac{\tilde{\beta}_1 - \gamma(\{1\}; x_0)}{\Delta_1} = \frac{\tilde{\beta}_2 - \gamma(\{2\}; x_0)}{\Delta_2}. \tag{15}$$

In a 2-player game $\Gamma^{x_0} \in G^P(2)$ to determine unique distribution $\{\beta_i(\bar{x}_t)\}$, $\bar{x}_t \in \omega$, of the total i -th player payoff $\tilde{\beta}_i$ (15) meeting (5), (6) and (7) one can use the following algorithm which aims to minimize the payoff transfers at each node \bar{x}_t .

Algorithm:

1. Find a cooperative trajectory $\bar{\omega} = (\bar{x}_0, \dots, \bar{x}_t, \dots, \bar{x}_T)$. If there are at least two paths meeting (2) employ PRB algorithm (see [13]) to choose a unique cooperative history.
2. Check whether the system of non-strict linear inequalities (6), (7) and linear equations (3)–(5) is compatible (one could use, for instance, any software package for solving Linear Programming problems). If no, then subgame-perfect cooperative agreement in Γ^{x_0} does not exist, i.e. the β -S-P Core is empty. If β -S-P Core is non-empty solve (15) to calculate $\tilde{\beta}_1 = \sum_{\tau=0}^T \beta_1(\bar{x}_\tau)$ and $\tilde{\beta}_2$.
3. Using (5), (6) and (7) obtain the system of double inequalities for $\tilde{\beta}_1(\bar{\omega}^{\bar{x}^1}) = \sum_{\tau=1}^T \beta_1(\bar{x}_\tau)$, $\tilde{\beta}_1(\bar{\omega}^{\bar{x}^2}) = \sum_{\tau=2}^T \beta_1(\bar{x}_\tau), \dots, \tilde{\beta}_1(\bar{\omega}^{\bar{x}^{T-1}}) = \beta_1(\bar{x}_{T-1}) + \beta_1(\bar{x}_T)$ in the form:

$$\begin{cases} c_1^1 \leq \tilde{\beta}_1(\bar{\omega}^{\bar{x}^1}) \leq C_1^1 \\ c_1^2 \leq \tilde{\beta}_1(\bar{\omega}^{\bar{x}^2}) \leq C_1^2 \\ \vdots \\ c_1^{T-1} \leq \beta_1(\bar{x}_{T-1}) + \beta_1(\bar{x}_T) \leq C_1^{T-1} \end{cases}, \tag{16}$$

where $0 \leq c_1^t \leq C_1^t$, $t = 1, 2, \dots, T - 1$.

4. Accept $\beta_1(\bar{x}_T) = h_1(\bar{x}_T)$, hence, no transfers are expected at terminal node \bar{x}_T .
5. Solve (16) backwards, implying minimal payoff transfer $|\beta_1(\bar{x}_t) - h_1(\bar{x}_t)|$ at each node \bar{x}_t , $t = T - 1, T - 2, \dots, 1$, that is sufficient to satisfy (16).
6. Calculate $\beta_1(\bar{x}_0) = \tilde{\beta}_1 - \sum_{\tau=1}^T \beta_1(\bar{x}_\tau)$.
7. Calculate $\beta_2(\bar{x}_t)$, $t = 0, \dots, T$, using strict balance condition (5).

5 β -S-P Core for Fishery-Management Extensive-Form Model

To provide an application of the β -S-P Core in the resource exploitation models in extensive form we explore a finite version of the original fishery-management model [18] that has been studied in [10]. Note that this game-theoretical fishery-management model belongs to a broad class of renewable resource extraction models (see, e.g., [1, 2, 18, 19, 21]).

The main sources of the players' asymmetry in the renewable resource extraction models as well as in dynamic environmental models are discussed in [2, 3, 22]. The players may have different costs, different discount rates, they may value the residual stock differently, e.t.c. We focus in the paper on the case when the players have the same discount factor δ , and the only source for asymmetry is that the players value differently the resource residual stock (after the fishery process ends). Namely, we assume that $K_1 > K_2$ in (18). Note that such an assumption is discussed in [3, 10, 17]. However, an algorithm for sustainable cooperation provided in the paper could be employed for extensive-form fishery-management model that takes into account other sources of the players' asymmetry.

It is worth noting that the symmetric case of extensive-form fishery-management model was studied in [17], and we will briefly compare the results obtained.

Example. (An extensive-form fishery-management with asymmetric players).

Denote by $y(t)$ a fish biomass (state variable) in year t , $t = 0, 1, \dots, T$, which evolves in accordance with the difference equation $y(t + 1) = a \cdot y(t)$, where $a > 1$ is the annual net growth rate. Suppose that only two players exploit the fishery and let $u_j(t) \geq 0$ denote the catch amount of player j in year t (strategy or control variable). Given the initial state condition $y(0) = y^0$ the system dynamics is described by the equation

$$y(t + 1) = a \cdot (y(t) - (u_1(t) + u_2(t))), \tag{17}$$

while $0 \leq u_1(t) + u_2(t) \leq y(t)$. Player j aims to maximize the objective (payoff) function of the form

$$H_j(\cdot) = \sum_{t=0}^{T-1} \delta_j^t \sqrt{u_j(t)} + K_j \cdot \delta_j^T \sqrt{y(T)}, \quad j = 1, 2, \tag{18}$$

where $\delta_j \in [0, 1)$ is a discount factor and $K_j > 0$ reflects how the j th player evaluates the worth of the fish biomass remainder (fishery's scrap or bequest) after the fishery process ends.

To specify the fishery-management model (17)–(18) in an extensive-form framework we need to make some additional assumptions. Firstly, for the sake of simplicity assume that each player can fish out at only two levels: High ($u_j^H = H_j$) and Low ($u_j^L = L_j$) and consider two-period model, i.e., $t = 0, 1, 2$. Secondly, to deal with a game of perfect information assume that the player 1 moves first at each year while the player 2 moves second, i.e., player 2 is aware of the first player's decision when choosing own harvest level.

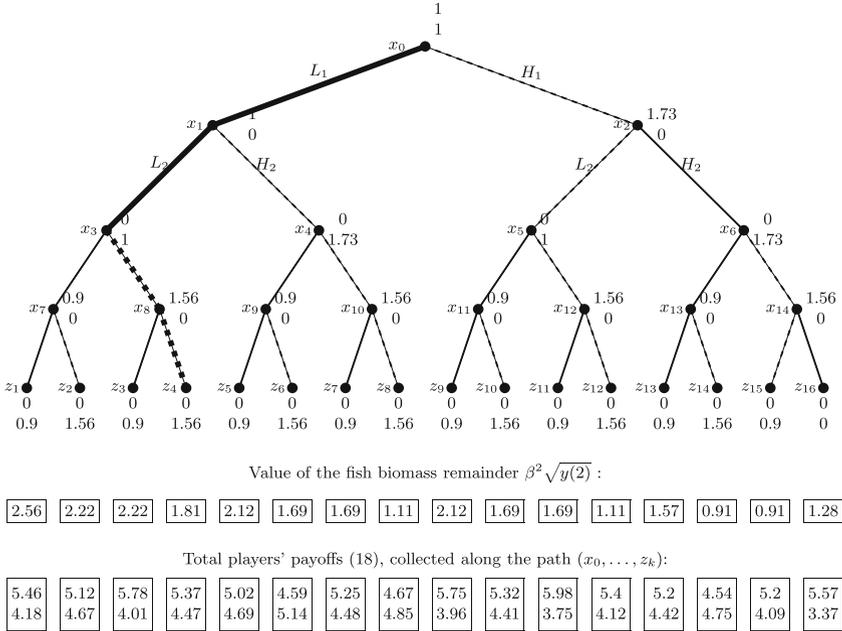


Fig. 1. Fishery-management model in extensive form

The resulting fishery-management model for given values of parameters: $y(0) = 10$; $a = 1.25$; $u_j^H = H_j = 3$, $u_j^L = L_j = 1$; $\delta_1 = \delta_2 = \delta = 0.9$; $K_1 = 1$, $K_2 = 0.5$, is presented in Fig. 1. Let $P_1 = \{x_0, x_3, x_4, x_5, x_6\}$, $P_2 = \{x_1, x_2, x_7 - x_{14}\}$, $P_{n+1} = \{z_1, \dots, z_{16}\}$, the right move at each decision position $x_k \in P_j$ corresponds to $u_j(x_k) = H_j$ (with only one exception for x_{14}) while the left alternative corresponds to L_j . According to (18) the current payoffs $\sqrt{u_j(0)}$ in $x_1 - x_6$ correspond to the first year of fishery process (hence, $\delta^0 = 1$) whereas the current payoffs $\sqrt{u_j(1)}$ in $x_7 - x_{14}$ and $z_1 - z_{16}$ correspond to the second year and have to be multiplied by δ . Note that if $u_1(0) = u_2(0) = H$ then $y(1) = 5$, and if the first player again chooses $u_1(1) = H_1$ at x_6 , then the player 2 can not choose $u_2(1) = H_2$ at x_{14} (the case of over-fishing) due to constraint $u_1(1) + u_2(1) \leq y(1)$. We take it into account assuming that $h_2(z_{16}) = 0$. Additionally, we admit some positive payoffs $h_j(x_0)$, $j = 1, 2$, at the root x_0 which could be interpreted as the players' initial assets (for instance, we let $h_j(x_0) = 1$, $j = 1, 2$, in Fig. 1).

The above fishery-management model in extensive form admits a unique SPE (the players' equilibrium strategies for each decision positions are determined via backwards induction procedure and marked as dotted lines in Fig. 1). This SPE generates history $\omega^{SPE} = (x_0, x_2, x_5, x_{12}, z_{12})$ with the payoffs (5.4; 4.12).

The cooperative history $\tilde{\omega} = (x_0, x_1, x_3, x_8, z_4)$, that is marked in bold in Fig. 1, implies maximal summary payoff $\tilde{h}_1(\tilde{\omega}) + \tilde{h}_2(\tilde{\omega}) = 5.37 + 4.47 = 9.84$.

Comparing these results with symmetric case of similar model studied in [17] one can notice that the SPE history is preserved, while the cooperative history has been changed. Namely, in asymmetric model the cooperative behavior implies more fishing efforts of both players at the second time period and consequently lower fish biomass remainder. The β -S-P Core is turned out to be non-empty for both cases. Obviously, these results rely heavily on the chosen parameters values.

Note that system (3)–(6) and (7) for the extensive-form game in Fig. 1 is compatible, and hence, the β -S-P Core is non-empty. To determine unique payment vector $(\tilde{\beta}_1(\bar{\omega}), \tilde{\beta}_2(\bar{\omega}))$ from the β -S-P Core one can employ the maxmin RBC rule. If we denote $\tilde{\beta}_1(\bar{\omega}) = 5.37 + \varepsilon$, $\tilde{\beta}_2(\bar{\omega}) = 4.47 - \varepsilon$, Eq. (15) takes the form

$$\frac{(5.37 + \varepsilon) - 5.4}{5.98 - 4.54} = \frac{(4.47 - \varepsilon) - 4.12}{5.14 - 3.37} \iff \varepsilon = 0.17.$$

Hence, $\tilde{\beta}_1(\bar{\omega}) = 5.54$, $\tilde{\beta}_2(\bar{\omega}) = 4.3$. We apply the above algorithm to calculate all the components $\beta_i(x_t)$ of this payoff distribution procedure β . The (minimal in

$\bar{\omega}$	x_0	x_1	x_3	x_8	z_4	$\tilde{\beta}_j(\bar{\omega})$
$\beta_1(x_t)$	1.55	0.62	0	1.56	1.81	5.54
$\beta_2(x_t)$	0.45	0.38	1	0	2.47	4.3
$\beta_1(x_t) - h_1(x_t)$	0.55	-0.38	0	0	0	0.17

absolute value) payoff transfers (from player 2 to player 1) at every position in the cooperative history are presented in the lowest row. Note that the cooperative history for the subgame Γ^{x_3} coincides with the SPE path, hence, no transfers in nodes x_3 , x_8 and z_4 are needed to sustain cooperative agreement.

However, inequalities (7) for $S = \{1\}$ and $S = \{2\}$ in $x_1 \in \bar{\omega}$ take the form: $\gamma(\{1\}; x_1) = 3.67 \leq \beta_1(x_1) + 3.37$; $\gamma(\{2\}; x_1) = 3.85 \leq \beta_2(x_1) + 3.47$. Obviously, $\beta_1(x_1) \geq 0.3$, $\beta_2(x_1) \geq 0.38$, and the players have to redistribute current payoffs at x_1 (a transfer at least 0.38 from player 1 to player 2 is needed to create an incentive to cooperate for player 2 at Γ^{x_1}). Similar remark is valid for the origin x_0 .

Note that if we treat this extensive-form game as the game where the payoffs are defined and could be redistributed between the players in only terminal nodes the S-P Core [6] of such a game is empty.

6 Concluding Remarks

Since the β -S-P Core definition for a game $\Gamma^{x_0} \in G^P(n)$ with payoffs defined at all nodes allows the payoffs transfers at each node of the cooperative history, it provides a powerful tool to sustain cooperative scenario. We employ the maxmin RBC rule for selecting unique payment vector $(\tilde{\beta}_1(\bar{\omega}), \tilde{\beta}_2(\bar{\omega}))$ from the β -S-P

Core in Example. One may consider a natural refinement of the maxmin RBC rule which implies that the players should initially implement the sequential elimination of strictly dominated strategies procedure (see, e.g., [10, 23, 27]) and then estimate the absolute ranges Δ_j of their payoffs. If we apply this refinement to the fishery-management model in Example we obtain another PDP from the β -S-P Core, namely $\tilde{\beta}_1(\bar{\omega}) = 5.516$; $\tilde{\beta}_2(\bar{\omega}) = 4.324$. It is surely of interest to consider other approaches for the β -S-P Core refinement as well as to study additional properties of this cooperative solution (for instance, time consistency [10, 26, 27] and irrational-behavior-proof condition [32, 33]).

Hopefully, similar approach could be applied to other discrete-time dynamic games of bio-resource management (see, e.g., [1, 21]) and other dynamic models of climate change (see, e.g., [20]). An open question is whether the β -S-P Core concept can be adapted to the analysis of the dynamic interaction between different political and religious movements (see [31] for details).

References

1. Breton, M., Dahmouni, I., Zaccour, G.: Equilibria in a two-species fishery. *Math. Biosci.* **309**, 78–91 (2019)
2. Breton, M., Keoula, M.Y.: A great fish war model with asymmetric players. *Ecol. Econ.* **97**, 209–223 (2014)
3. Cabo, F., Tidball, M.: Cooperation in a dynamic setting with asymmetric environmental valuation and responsibility. *Dyn. Games Appl.* (2021). <https://doi.org/10.1007/s13235-021-00395-y>
4. Chander, P.: Subgame-perfect cooperative agreements in a dynamic game of climate change. *J. Environ. Econ. Manag.* **84**, 173–188 (2017)
5. Chander, P., Tulkens, H.: The core of an economy with multilateral environmental externalities. *Int. J. Game Theory* **26**, 379–401 (1997)
6. Chander, P., Wooders, M.: Subgame-perfect cooperation in an extensive game. *J. Econ. Theory* **187**, 105017 (2020)
7. Crettez, B., Hayek, N., Zaccour, G.: Do charities spend more on their social programs when they cooperate than when they compete? *Eur. J. Oper. Res.* **283**, 1055–1063 (2020)
8. Gromova, E.V., Plekhanova, T.M.: On the regularization of a cooperative solution in a multistage game with random time horizon. *Discrete Appl. Math.* **255**, 40–55 (2019)
9. Chebotareva, A., Shimai, S., Tretyakova, S., Gromova, E.: On the value of the preexisting knowledge in an optimal control of pollution emissions. *Contrib. Game Theory Manag.* **14**, 49–58 (2021)
10. Haurie, A., Krawczyk, J.B., Zaccour, G.: *Games and Dynamic Games*. Scientific World, Singapore (2012)
11. Hillas, J., Kvasov, D.: Backward induction in games without perfect recall. *Games Econ. Behav.* **124**, 207–218 (2020). <https://doi.org/10.1016/j.geb.2020.08.011>
12. Kuhn, H.: Extensive games and the problem of information. *Ann. Math.* **28**, 193–216 (1953)

13. Sustainable cooperation in multicriteria multistage games: Kuzyutin, D., Gromova, E., Pankratova, Ya.: *Oper. Res. Lett.* **46**, 557–562 (2018)
14. Kuzyutin, D., Lipko, I., Pankratova, Y., Tantlevskij, I.: Cooperation enforcing in multistage multicriteria game: new algorithm and its implementation. In: Petrosyan, L., Mazalov, V., Zenkevich, N. (eds.) *Frontiers of Dynamic Games. Static & Dynamic Game Theory: Foundations & Applications*. Birkhäuser, Cham, 2020. https://doi.org/10.1007/978-3-030-51941-4_10
15. Kuzyutin, D., Romanenko, I.: On properties of equilibrium solutions for n -person games in extensive form. *Vestnik St. Petersburg Univ. Math.* **3**(15), 17–27 (1998). (in Russian)
16. Kuzyutin, D., Smirnova, N.: Subgame consistent cooperative behavior in an extensive form game with chance moves. *Mathematics* **8**(7), 1061 (2020). <https://doi.org/10.3390/math8071061>
17. Kuzyutin, D., Smirnova, N., Skorodumova, Y.: Implementation of subgame-perfect cooperative agreement in an extensive-form game. In: Petrosyan, L.A., Zenkevich, N.A. (eds.) *Contributions to Game Theory and Management*, pp. 257–272. St. Petersburg State Univ., St. Petersburg, Russia (2021). <http://hdl.handle.net/11701/33701>
18. Levhari, D., Mirman, L.J.: The great fish war: an example using a dynamic Cournot-Nash solution. *Bell J. Econ.* **11**(1), 322–334 (1980)
19. Mazalov, V., Parilina, E., Zhou, J.: Altruistic-like equilibrium in a differential game of renewable resource extraction. In: Pardalos, P., Khachay, M., Kazakov, A. (eds.) *MOTOR 2021*. LNCS, vol. 12755, pp. 326–339. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77876-7_22
20. Masoudi, N., Zaccour, G.: Adapting to climate change: is cooperation good for the environment? *Econ. Lett.* **153**, 1–5 (2017). <https://doi.org/10.1016/j.econlet.2017.01.018>
21. Mazalov, V.V., Rettiyeva, A.N.: The discrete-time bioresource sharing model. *J. Appl. Math. Mech.* **75**, 180–188 (2011)
22. Mazalov, V. V., Rettieva, A.N.: Asymmetry in a cooperative bioresource management problem. In: *Game-Theoretic Models in Mathematical Ecology*, pp. 113–152. Nova Science Publishers (2015)
23. Moulin, H.: *Axioms of Cooperative Decision Making*. Cambridge University Press, Cambridge (1988)
24. Nash, J.F.: Equilibrium points in n -person games. *Proc. Nat. Acad. Sci. USA* **36**, 48–49 (1950)
25. Parilina, E., Zaccour, G.: Node-consistent core for games played over event trees. *Automatica* **55**, 304–311 (2015)
26. Petrosyan, L.A., Danilov, N.N.: Stability of solutions in non-zero sum differential games with transferable payoffs. *Astron.* **1**, 52–59 (1979)
27. Petrosyan, L., Kuzyutin, D.: *Games in Extensive Form: Optimality and Stability*. Saint Petersburg University Press, St. Petersburg (2000). (in Russian)
28. Petrosian, O., Zakharov, V.: IDP-Core: novel cooperative solution for differential games. *Mathematics* **8**, 721 (2020)
29. Selten, R.: Reexamination of the perfectness concept for equilibrium points in extensive games. *Int. J. Game Theory* **4**, 25–55 (1975)
30. Shapley, L.S.: On balanced sets and cores. *Nav. Res. Logist.* **14**(4), 453–460 (1967)
31. Tantlevskij, I., Gromova, E., Gromov, D.: Network analysis of the interaction between different religious and philosophical movements in early Judaism. *Philosophies* **6**(1), 2 (2021). <https://www.mdpi.com/2409-9287/6/1/2>

32. Yeung, D.: An irrational-behavior-proof condition in cooperative differential games. *Int. Game Theory Rev.* **8**(4), 739–744 (2006)
33. Yeung, D., Petrosyan, L.: *Subgame Consistent Economic Optimization: An Advanced Cooperative Dynamic Game Analysis*. Springer, New York (2012).
<https://doi.org/10.1007/978-0-8176-8262-0>



The Core of Cooperative Differential Games on Networks

Anna Tur^(✉)  and Leon Petrosyan 

St. Petersburg State University, 7/9, Universitetskaya nab.,
Saint-Petersburg 199034, Russia
{a.tur,l.petrosyan}@spbu.ru

Abstract. A class of differential games on networks is considered. The construction of cooperative optimality principles using a special type of characteristic function that takes into account the network structure of the game is investigated. It is assumed that interaction on the network is possible between neighboring players and between players connected by paths whose length does not exceed a given value. It is shown that in such games the characteristic function is convex even if there are cycles in the network. The core is used as cooperative optimality principles. A necessary and sufficient condition for an imputation to belong to the core is obtained. The network differential resource extraction game is investigated as an example.

Keywords: Differential game · Cooperative game · Network game · The core · The Shapley value · The position value

1 Introduction

Many real-life multi-agents processes can be interpreted as a scheduling problem on a network. Vertices in networks can correspond to agents (region, country), and connections can represent the ability of agents to interact (transport connection, an information transfer, resource distribution). In such problems, methods of cooperative game theory turn out to be effective and practical.

Different principles for measuring the power of a player in a network are considered, for example, in [1,4–6].

If the evolution of decision making is continuous in time, the problem is usually viewed as a differential game. In such processes, it is important to take into account the ability of players to change the chosen cooperative solution at some intermediate time instant in the game. Cooperative differential games on networks were first considered in [8]. There was introduced a new type of strategies with possibility of cutting links with neighboring players during the game.

The new type of strategies has led to the possibility of construction a novel form for characteristic function [10]. It was proposed a measuring of coalition's

This research was supported by the Russian Science Foundation grant No. 22-11-00051.

worth without considering the actions of players who are not members of this coalition. Also it was considered the case when payoff of a player depends not only on his neighbors' actions but also on players' actions connected with this player by a path in the network [11]. In [15], it was shown that the convexity of the new characteristic function in such games can only be guaranteed if there are no cycles in the graph.

This paper is a continuation of the indicated research on cooperative differential games on networks. Another type of players' payoff is treated. It is assumed that players can interact with each other only if the distance between them on the network is not greater than a given value. This assumption allows to generalize the convexity of the characteristic function for the case of a graph with cycles. A necessary and sufficient condition for an imputation to belong to the core is derived. The strong time-consistency of the core is proved.

The paper is structured as follows. The definition of the cooperative differential game on a network is given in Sect. 2. In Sect. 3, the definition of the characteristic function based on cooperative strategies used by players from a coalition is given. A necessary and sufficient condition for the imputation to belong to the core is derived in Sect. 4. The strong time-consistency of the core is proved in Sect. 5. The Shapley value and the position value are discussed in Sect. 6. As an illustrative example, a differential game of resource extraction on the network is investigated in Sect. 7.

2 Problem Formulation

Consider a class of n -person differential games with prescribed duration T . Let $N = \{1; 2; \dots; n\}$ be the set of players. Players are connected in a network system. A pair (N, L) is called a network, where N is a set of nodes, and $L \subset N \times N$ is a given set of links. The players are represented by nodes. If pair $(i, j) \in L$, there is a link connecting players $i \in N$ and $j \in N$. It is supposed that all links are undirected.

Denote the set of players directly connected to player i as $K(i) = \{j : (i, j) \in L\}$.

Denote by $K^m(i)$, where $m \geq 2$, the set of players connected with player $i \in N$ by a shortest path containing exactly m edges (only paths without cycles and loops are considered), and let $K^1(i) = K(i) \cup i$, for $i \in N$.

Every player $i \in N$ at any instant of time can cut a connection with any other players from $K(i)$.

The system dynamics is given by

$$\dot{x}_i(\tau) = f_i(x_i(\tau); u_i(\tau)); \quad x_i(t_0) = x_i^0; \quad \text{for } \tau \in [t_0; T] \text{ and } i \in N. \quad (1)$$

Here $x_i(t) \in R^m$ is the state variable of player $i \in N$ at time t , and $u_i(t) \in U_i \subset R^k$ – the control variable of player $i \in N$. Function $f_i(x_i; u_i)$ are continuously differentiable in x_i and u_i .

The payoff of player i is given as

$$H_i(x^0; u_1, \dots, u_n) = \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} \int_{t_0}^T h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau)) d\tau. \quad (2)$$

The term $h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau))$ is the instantaneous gain that player i can obtain through interaction with player $j \in K^m(i)$, and $h_i^i(x_i(\tau); u_i(\tau))$ is the instantaneous gain that player i can obtain by itself. Suppose that functions $h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau))$, for $j \in K^m(i)$, $j \neq i$ are non-negative. Assume that $\delta \in (0, 1)$. The multiplier δ^{m-1} shows that the more remote network players from player i , the less their behavior affects the payoff of that player. The parameter $r : n - 1 \geq r \geq 1$ corresponds to the length of the shortest path between the player i and the connected player, from which the player i receives any benefit. Such a restriction on the interaction of players may be due to territorial features. For example, if the interaction of players is associated with transport costs, then it is assumed that the interaction is justified only if the distance between the players is not greater than a given value. We denote by $x^0 = (x_1^0; x_2^0; \dots; x_n^0)$ the vector of initial conditions.

We consider the game $\Gamma(x^0, T - t_0)$ if the network (N, L) is defined, the system dynamics (1) and the sets of feasible controls U_i , $i \in N$ are given, and the players' payoffs are determined by (2). Each player, choosing a control variable u_i from his set of feasible controls, steers his own state according to the differential equation (1) and seeks to maximize his objective functional (2).

Suppose that players can cooperate in order to achieve the maximum joint payoff

$$\sum_{i \in N} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} \int_{t_0}^T h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau)) d\tau. \quad (3)$$

subject to dynamics (1).

The optimal cooperative strategies of players $\bar{u}(t) = (\bar{u}_1(t), \dots, \bar{u}_n(t))$, for $t \in [t_0; T]$ are defined as follows

$$\bar{u}(t) = \arg \max_{u_1, \dots, u_n} \sum_{i \in N} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} \int_{t_0}^T h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau)) d\tau. \quad (4)$$

The trajectory corresponding to the optimal cooperative strategies $(\bar{u}_1(t), \dots, \bar{u}_n(t))$ is the optimal cooperative trajectory $\bar{x}(t) = (\bar{x}_1(t); \bar{x}_2(t); \dots; \bar{x}_n(t))$. The maximum joint payoff can be expressed as:

$$\begin{aligned} & \sum_{i \in N} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} \int_{t_0}^T h_i^j(\bar{x}_i(\tau); \bar{x}_j(\tau); \bar{u}_i(\tau); \bar{u}_j(\tau)) d\tau = \\ & \max_{u_1, \dots, u_n} \left\{ \sum_{i \in N} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} \int_{t_0}^T h_i^j(x_i(\tau); x_j(\tau); u_i(\tau); u_j(\tau)) d\tau \right\} \quad (5) \end{aligned}$$

subject to dynamics

$$\dot{\bar{x}}_i = f_i(\bar{x}_i(\tau); \bar{u}_i(\tau)); \quad \bar{x}_i(t_0) = x_i^0; \quad \text{for } \tau \in [t_0; T] \text{ and } i \in N. \quad (6)$$

To determine how to allocate the maximum total payoff among the players under an agreeable scheme, defining the characteristic function is necessary.

Usually, in cooperative games, the concept of a characteristic function is used to determine how to distribute the maximum payoff between players. There are different approaches for construction of characteristic functions (see [3, 16]).

We define the characteristic function in the same way as it was proposed in [10]. It was supposed there to find the value of the characteristic function of S on the cooperative trajectory when players from S use cooperative strategies under the condition that connections with players from $N \setminus S$ are cut off (since the worst thing they can do for the coalition S is to cut the connection with players from S). The characteristic function constructed in this way is easier to compute and possesses some advantageous properties. In this paper, we will apply this approach to the class of games under consideration.

Let $S \subset N$ is a subset of vertices and L_S denote the set of all edges between vertices from S in L . A pair (S, L_S) is called a subgraph induced by S . For player $i \in S$ denote by $K_S^m(i)$, where $m \geq 2$, the set of players connected with player i by the shortest path in (S, L_S) containing exactly m edges, and let $K_S^1(i) = K(i) \cap S \cup i$, for $i \in S$.

The worth of coalition S in the game is evaluated along the cooperative trajectory

$$V(S; x_0, T - t_0) = \sum_{i \in S} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K_S^m(i)_{t_0}} \int_{t_0}^T h_i^j(\bar{x}_i(\tau); \bar{x}_j(\tau); \bar{u}_i(\tau); \bar{u}_j(\tau)) d\tau, \quad (7)$$

where $\bar{x}_i(t)$ and $\bar{u}_i(t)$ are the solutions obtained in (4) and (6).

Similarly, the cooperative-trajectory characteristic function of the subgame $\Gamma(\bar{x}(t), T - t)$ starting at time $t \in [t_0; T]$ can be evaluated as

$$V(S; \bar{x}(t), T - t) = \sum_{i \in S} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K_S^m(i)_t} \int_t^T h_i^j(\bar{x}_i(\tau); \bar{x}_j(\tau); \bar{u}_i(\tau); \bar{u}_j(\tau)) d\tau. \quad (8)$$

3 Properties of the Characteristic Function

Characteristic function $V(S; x_0, T - t_0)$ is called *convex* (or supermodular) if for any coalitions $S_1, S_2 \subseteq N$ the following condition holds: $V(S_1 \cup S_2; x_0, T - t_0) + V(S_1 \cap S_2; x_0, T - t_0) \geq V(S_1; x_0, T - t_0) + V(S_2; x_0, T - t_0)$. A game is called convex if its characteristic function is convex.

In [10], the characteristic function was constructed in a similar way, but it was assumed that $h_i^j = 0$ if the players i and j are not connected by an edge. It was shown that such characteristic function is convex. In [15], for the case when

there are not restrictions on the length of the shortest path between interacting players, it was shown that such characteristic function is convex if there are no cycles in the network.

The restrictions on the interaction of players introduced in this paper allowed us to expand the class of networks for which the characteristic function is convex.

Define functions $W(S; t)$ that can be interpreted as instantaneous values of the characteristic function according to the following rule

$$W(S; t) = \sum_{i \in S} \sum_{m=1}^r \delta^{m-1} \sum_{j \in K_S^m(i)} h_i^j(\bar{x}_i(t); \bar{x}_j(t); \bar{u}_i(t); \bar{u}_j(t)). \tag{9}$$

Proposition 1. *Let $S_1 \subset N, S_2 \subset S_1$. If there are no cycles of length less than $2r + 1$ in the network (N, L) , then the following inequality holds for each $i \in N \setminus S_1$ and each $t \in [0, T]$:*

$$W(S_1 \cup \{i\}; t) - W(S_1; t) \geq W(S_2 \cup \{i\}; t) - W(S_2; t). \tag{10}$$

Proof. For simplicity, we denote

$$h_i^j(\bar{x}_i(t); \bar{x}_j(t); \bar{u}_i(t); \bar{u}_j(t)) = \bar{h}_i^j(t),$$

$$\bar{h}_i^j(t) + \bar{h}_j^i(t) = \bar{h}_{i,j}(t)$$

The absence of cycles of length less than $2r + 1$ in the network means that there can be only one path of length less than $r + 1$ between any two vertices. Let $D_S^m(i)$ be the set of pairs of vertices $\{p, q\}$ such that $p \in S, q \in S$, the distance between them equals $m \leq r$, all vertices of the path between p and q belong to S , and i lies on this path. Then

$$W(S_1 \cup \{i\}; t) - W(S_1; t) = \bar{h}_i^i(t) + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_{S_1 \cup \{i\}}^m(i)} \bar{h}_{p,q}(t), \tag{11}$$

$$W(S_2 \cup \{i\}; t) - W(S_2; t) = \bar{h}_i^i(t) + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_{S_2 \cup \{i\}}^m(i)} \bar{h}_{p,q}(t). \tag{12}$$

Since $S_2 \subset S_1$, we have $D_{S_2 \cup \{i\}}^m(i) \subset D_{S_1 \cup \{i\}}^m(i)$. Then

$$\sum_{m=1}^k \delta^{m-1} \sum_{\{p,q\} \in D_{S_1 \cup \{i\}}^m(i)} \bar{h}_{p,q}(t) \geq \sum_{m=1}^k \delta^{m-1} \sum_{\{p,q\} \in D_{S_2 \cup \{i\}}^m(i)} \bar{h}_{p,q}(t).$$

It follows that (10) is satisfied.

Remark 1. Note that the presence of a cycle of length less than $2r + 1$ in the network can lead to the violation of property (10).

Indeed, the presence of a cycle of length less than $2r + 1$ in the network allows several paths of length less than $r + 1$ between two vertices.

Assume that there are two paths of length $l \leq r$ between vertices $p^* \in S_2$ and $q^* \in S_2$. Suppose that all vertices from the first path belong to $S_2 \cup \{i\}$, and vertex i lies on this path. Assume also that the second path contains vertices from $S_1 \setminus S_2$ and vertex i does not lie on this path.

Note that there exists a path of length less than $r + 1$ between p^* and q^* in $(S_2 \cup \{i\}, L_{S_2 \cup \{i\}})$, but there is no such path between these vertices in (S_2, L_{S_2}) (since the first path goes through player i who is no longer in the coalition and the second path does not belong to (S_2, L_{S_2})). Then there is a term $\delta^{l-1} \bar{h}_{p^*, q^*}(t)$ in the right-hand side of (12).

There are two paths of length $l \leq r$ between vertices p^* and q^* in $(S_1 \cup \{i\}, L_{S_1 \cup \{i\}})$. One of them passes through vertex i and the other does not. So there exists path between p^* and q^* in (S_1, L_{S_1}) . This means that in the right-hand side of (11), there is no term corresponding to the vertices p^* and q^* .

Thus, if $\bar{h}_{p^*, q^*}(t)$ is large enough, inequality (10) will not hold.

Corollary 1. *If there are no cycles of length less than $2r + 1$ in the network (N, L) , then the characteristic function defined in (7)–(8) is convex.*

Proof. It was shown that for each $\tau \in [t_0, T]$, $S_1 \subset N$, $S_2 \subset S_1$ and each $i \in N \setminus S_1$, the following inequality holds

$$W(S_1 \cup \{i\}; t) - W(S_1; t) \geq W(S_2 \cup \{i\}; t) - W(S_2; t). \tag{13}$$

Integrating both sides of this inequality with respect to t we have for each $t \in [t_0, T]$, each $S_1 \subset N$, $S_2 \subset S_1$, and each $i \in N \setminus S_1$

$$V(S_1 \cup \{i\}; \bar{x}(t), T - t) - V(S_1; \bar{x}(t), T - t) \geq V(S_2 \cup \{i\}; \bar{x}(t), T - t) - V(S_2; \bar{x}(t), T - t).$$

This means that the characteristic function defined in (7)–(8) is convex (see [14]).

Further, we will assume that there are no cycles of length less than $2r + 1$ in the network (N, L) .

4 The Core

The set of all imputations in the game $\Gamma(x_0, T - t_0)$ is given by

$$E(x_0, T - t_0) = \{ \xi(x_0, T - t_0) = (\xi_1(x_0, T - t_0), \dots, \xi_n(x_0, T - t_0)) : \sum_{i \in N} \xi_i(x_0, T - t_0) = V(N; x_0, T - t_0), \xi_i(x_0, T - t_0) \geq V(\{i\}; x_0, T - t_0), i \in N \}. \tag{14}$$

Definition 1 ([14]). *The core $C(x_0, T - t_0)$ of the game $\Gamma(x_0, T - t_0)$ is the subset of the imputation set $E(x_0, T - t_0)$, such that*

$$C(x_0, T - t_0) = \{ \xi(x_0, T - t_0) \in E(x_0, T - t_0) : \sum_{i \in S} \xi_i(x_0, T - t_0) \geq V(S; x_0, T - t_0), S \subset N \}. \quad (15)$$

Similarly, for every $t \in [t_0, T]$ denote by $E(\bar{x}(t), T - t)$ the set of all imputations and by $C(\bar{x}(t), T - t)$ the core in the subgame $\Gamma(\bar{x}(t), T - t)$ along the cooperative trajectory.

Consider a pair of vertices $\{p, q\}$, $p, q \in N$. Let the length of the shortest path between them is $l \leq r$, and the shortest path is $\{i_1, \dots, i_{l+1}\}$ (here $p = i_1$, $q = i_{l+1}$). We denote by $\Phi^{p,q}$ the set of vectors $\phi^{p,q} = (\phi_{i_1}^{p,q}, \dots, \phi_{i_{l+1}}^{p,q})$, such that $\sum_{j=1}^{l+1} \phi_{i_j}^{p,q} = 1$, $0 \leq \phi_{i_j}^{p,q} \leq 1$, for each $j = \overline{1, l+1}$ (for each vertex i_j , $j = \overline{1, l+1}$, belonging to the path between p and q , the coefficient $\phi_{i_j}^{p,q}$ is given).

Proposition 2. *In the class of games under consideration, an imputation $\xi(x_0, T - t_0) \in E(x_0, T - t_0)$ belongs to the core if and only if it can be represented in the following form*

$$\xi_i(x_0, T - t_0) = \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^T \phi_i^{p,q} \bar{h}_{p,q}(\tau) d\tau, \quad i \in N. \quad (16)$$

Here $\phi^{p,q} \in \Phi^{p,q}$ for each pair of vertices $p \in N$, $q \in N$.

Proof. First, prove that the vector defined by Eq. (16) is an imputation.

$$\begin{aligned} \sum_{i=1}^n \bar{\xi}_i(x_0, T - t_0) &= \sum_{i=1}^n \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{i=1}^n \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^T \phi_i^{p,q} \bar{h}_{p,q}(\tau) d\tau = \\ &= \sum_{i=1}^n \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in \bigcup_{i \in N} D_N^m(i)} (\phi_{i_1}^{p,q} + \dots + \phi_{i_{m+1}}^{p,q}) \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau = \\ &= \sum_{i=1}^n \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in \bigcup_{i \in N} D_N^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau = V(N; x_0, T - t_0) \end{aligned} \quad (17)$$

Also note that $V(\{i\}; x_0, T - t_0) = \int_{t_0}^T \bar{h}_i^i(\tau) d\tau$, then $\xi_i(x_0, T - t_0) \geq V(\{i\}; x_0, T - t_0)$ and $\xi(x_0, T - t_0)$ is an imputation in the game $\Gamma(x_0, T - t_0)$ according to (14).

Now show that if an imputation can be represented in the form (16) then it belongs to the core.

$$\begin{aligned}
 \sum_{i \in S} \xi_i(x_0, T - t_0) &= \sum_{i \in S} \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^T \phi_i^{p,q} \bar{h}_{p,q}(\tau) d\tau \\
 &= \sum_{i \in S} \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_S^m(i)} \int_{t_0}^T \phi_i^{p,q} \bar{h}_{p,q}(\tau) d\tau \\
 &\quad + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i) \setminus D_S^m(i)} \phi_i^{p,q} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau \\
 &= \sum_{i \in S} \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in \bigcup_{i \in S} D_S^m(i)} (\phi_{i_1}^{p,q} + \dots + \phi_{i_{m+1}}^{p,q}) \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau \\
 &\quad + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i) \setminus D_S^m(i)} \phi_i^{k,l} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau \\
 &= V(S; x_0, T - t_0) + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i) \setminus D_S^m(i)} \phi_i^{p,q} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau.
 \end{aligned} \tag{18}$$

Then

$$\sum_{i \in S} \xi_i(x_0, T - t_0) \geq V(S; x_0, T - t_0),$$

which proves that $\xi(x_0, T - t_0)$ belongs to the core.

Last show that any imputation from the core can be represented in the form (16).

Let π represents a simple ordering of the players. $\pi(i)$ – the number of player i in π . Π – the set of all possible permutations of N . Define

$$S_{\pi,k} = \{i \in N : \pi(i) \leq k\}, \quad k = 0, 1, \dots, n.$$

It was shown by Shapley [14] that the core is the convex hull of marginal imputations $\alpha(\pi) = (\alpha_1^\pi, \alpha_2^\pi, \dots, \alpha_n^\pi)$, where

$$\alpha_i^\pi = V(S_{\pi,\pi(i)}, x_0, T - t_0) - V(S_{\pi,\pi(i)-1}, x_0, T - t_0), \quad i \in N.$$

So the vertices of the core have coordinates $\alpha(\pi) = (\alpha_1^\pi, \alpha_2^\pi, \dots, \alpha_n^\pi)$ for all possible $\pi \in \Pi$, and

$$\alpha_i^\pi = V(\{i\}) + \sum_{m=1}^r \delta^{m-1} \sum_{\{k,l\} \in D_{S_{\pi,\pi(i)}}^m(i)} \int_{t_0}^T \bar{h}_{k,l}(\tau) d\tau. \tag{19}$$

The core is a compact convex polyhedron. Every its point can be represented as a convex combination of vertices (extreme points).

If the vector $\tilde{\xi}(t)$ belongs to the core, then its components can be written as

$$\tilde{\xi}_i(x_0, T - t_0) = \sum_{\pi \in \Pi} \varepsilon_\pi \alpha_i^\pi,$$

here $\sum_{\pi \in \Pi} \varepsilon_\pi = 1, 0 \leq \varepsilon_\pi \leq 1$ for all $\pi \in \Pi$. Then

$$\begin{aligned} \tilde{\xi}_i(x_0, T - t_0) &= \\ &= \sum_{\pi \in \Pi} \varepsilon_\pi V(\{i\}; x_0, T - t_0) + \sum_{\pi \in \Pi} \varepsilon_\pi \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_{S_{\pi, \pi(i)}}^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau = \\ &= V(\{i\}; x_0, T - t_0) + \sum_{\pi \in \Pi} \varepsilon_\pi \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_{S_{\pi, \pi(i)}}^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau. \end{aligned} \tag{20}$$

Consider the pair of vertices $(k_{i_1}, k_{i_{l+1}})$ with the length of the shortest path between them $l \leq r$. Let $k_{i_1}, k_{i_2}, \dots, k_{i_{l+1}}$ – the vertices from the shortest path between k_{i_1} and $k_{i_{l+1}}$. The term $\delta^{m-1} \int_{t_0}^T \bar{h}_{k_{i_1}, k_{i_{l+1}}}(\tau) d\tau$ is in the right-hand part of (20) only if i belongs to the shortest path between k_{i_1} and $k_{i_{l+1}}$. This follows from the form of marginal contributions (19) and imputation (20).

Also note that for each permutation π the value $\delta^{m-1} \int_{t_0}^T \bar{h}_{k_{i_1}, k_{i_{l+1}}}(\tau) d\tau$ occurs as a term only in one component $\alpha_{k_{i^*}}(\pi)$ of vector $\alpha(\pi)$, where $k_{i^*} = \arg \max_{j=1, \dots, l+1} \pi(k_{i_j})$ (k_{i^*} is the last player from $k_{i_1}, k_{i_2}, \dots, k_{i_{l+1}}$ in permutation π).

This means that in $\tilde{\xi}_{k_{i_1}}(t) + \tilde{\xi}_{k_{i_2}}(t) + \dots + \tilde{\xi}_{k_{i_{l+1}}}(t)$ we get the value $\delta^{m-1} \int_{t_0}^T \bar{h}_{k_{i_1}, k_{i_{l+1}}}(\tau) d\tau$ with the coefficient $\sum_{\pi \in \Pi} \varepsilon_\pi = 1$. Hence, the value $\delta^{m-1} \int_{t_0}^T \bar{h}_{k_{i_1}, k_{i_{l+1}}}(\tau) d\tau$ should be shared only between players lying on the shortest path from k_{i_1} to $k_{i_{l+1}}$.

Since this must hold for any pair of vertices, it follows that the arbitrary imputation $\tilde{\xi}(t)$ from the core has the form (16).

This concludes the proof.

5 Strong Time-Consistency of the Core

In Proposition 2, it is shown that if players use the core as an optimality principle, to choose an imputation they need to determine the value of coefficients $\phi^{p,q}$ for every pair of vertices p, q . But at some intermediate time instant of the game, players can change their mind and choose other coefficients, thus choose another

imputation from the core. It is important that the resulting imputation of the game remains in the core. This is true only if the cooperative solution is strong time-consistent. This property of cooperation solution was introduced in [7]. Let us show that the core of games under consideration is strong time-consistent.

Definition 2 (see [9]). *A function $\beta(t) = (\beta_1(t), \dots, \beta_n(t))$, $t \in [t_0, T]$ is the imputation distribution procedure (IDP) for imputation $\alpha \in E(x_0, T - t_0)$, if*

$$\alpha_i = \int_{t_0}^T \beta_i(\tau) d\tau, \quad i \in N.$$

Definition 3 (see [7]). *An optimality principle $M(x_0, T - t_0) \subset E(x_0, T - t_0)$ is called strong time-consistent if*

1. $M(\bar{x}(t), T - t) \neq \emptyset, \forall t \in [t_0, T]$.
2. For each $\alpha \in M(x_0, T - t_0)$ there exists an IDP $\beta(\tau) = (\beta_1(\tau), \dots, \beta_n(\tau))$, $\tau \in [t_0, T]$, such that $\alpha = \int_{t_0}^T \beta_i(\tau) d\tau, i \in N$ and

$$M(x_0, T - t_0) \supset \int_{t_0}^t \beta(\tau) d\tau \oplus M(\bar{x}(t), T - t), \quad \forall t \in [t_0, T]. \quad (21)$$

For $a \in R^n, B \subset R^n$, the symbol \oplus means the following: $a \oplus B = \{a + b : b \in B\}$.

Proposition 3. *In the game under consideration the core is strong-time consistent.*

Proof. Consider an arbitrarily chosen imputation $\bar{\xi}(x_0, T - t_0)$ from the core. According to Proposition 2, there exist such coefficients $\bar{\phi}^{p,q} \in \Phi^{p,q}$ for every pair of vertices (\underline{p}, q) with the length of the shortest path between them less than $r + 1$, that $\bar{\xi}$ can be represented as

$$\bar{\xi}_i(x_0, T - t_0) = \int_{t_0}^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^T \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau, \quad i \in N. \quad (22)$$

Define the following imputation distribution procedure (IDP) for $\bar{\xi}(x_0, T - t_0)$

$$\bar{\beta}_i(t) = \bar{h}_i^i(t) + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau).$$

Consider the vector with components

$$\int_t^T \bar{\beta}_i(\tau) d\tau = \int_t^T \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_t^T \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau. \quad (23)$$

Note that this vector belongs to the core $C(\bar{x}(t), T - t)$ of the subgame $\Gamma(\bar{x}(t), T - t)$, because it can be represented in the form (16) with coefficients $\bar{\phi}^{p,q} \in \Phi^{p,q}$. Then $\sum_{i \in N} \int_t^T \bar{\beta}_i(\tau) d\tau = V(N; \bar{x}(t), T - t)$ and $\sum_{i \in N} \int_{t_0}^t \bar{\beta}_i(\tau) d\tau = V(N; x_0, T - t_0) - V(N; \bar{x}(t), T - t)$.

Let $\bar{\xi}_i(\bar{x}(t), T - t)$ is an imputation from the core $C(\bar{x}(t), T - t)$ of the subgame $\Gamma(\bar{x}(t), T - t)$. Then

$$\sum_{i \in S} \bar{\xi}_i(\bar{x}(t), T - t) \geq V(S; \bar{x}(t), T - t).$$

Also note that $\sum_{i \in N} \bar{\xi}_i(\bar{x}(t), T - t) = V(N; \bar{x}(t), T - t)$ and $\bar{\xi}_i(\bar{x}(t), T - t) \geq V(\{i\}; \bar{x}(t), T - t)$.

Let

$$\tilde{\xi}(x_0, T - t_0) = \int_{t_0}^t \bar{\beta}(\tau) d\tau + \bar{\xi}(\bar{x}(t), T - t). \tag{24}$$

First show, that $\tilde{\xi}(x_0, T - t_0)$ is an imputation in $C(x_0, T - t_0)$. Show that the group rationality holds:

$$\begin{aligned} \sum_{i \in N} \tilde{\xi}_i(x_0, T - t_0) &= \sum_{i \in N} \int_{t_0}^t \bar{\beta}_i(\tau) d\tau + \sum_{i \in N} \bar{\xi}_i(\bar{x}(t), T - t) = \\ &= V(N; x_0, T - t_0) - V(N; \bar{x}(t), T - t) + V(N; \bar{x}(t), T - t) = V(N; x_0, T - t_0). \end{aligned} \tag{25}$$

And individual rationality holds:

$$\begin{aligned} \tilde{\xi}_i(x_0, T - t_0) &= \int_{t_0}^t \bar{\beta}_i(\tau) d\tau + \bar{\xi}_i(\bar{x}(t), T - t) = \\ &= \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^t \bar{\phi}_{p,q}^i \bar{h}_{p,q}(\tau) d\tau + \bar{\xi}_i(\bar{x}(t), T - t) \geq \\ &= \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^t \bar{\phi}_{p,q}^i \bar{h}_{p,q}(\tau) d\tau + V(\{i\}; \bar{x}(t), T - t) = \\ &= \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^t \bar{\phi}_{p,q}^i \bar{h}_{p,q}(\tau) d\tau + \int_t^T \bar{h}_i^i(\tau) d\tau = \\ &= V(\{i\}; x_0, T - t_0) + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^t \bar{\phi}_{p,q}^i \bar{h}_{p,q}(\tau) d\tau \geq V(\{i\}; x_0, T - t_0). \end{aligned} \tag{26}$$

So, $\tilde{\xi}(x_0, T - t_0)$ indeed is an imputation in $\Gamma(x_0, T - t_0)$.

Now we need to prove that $\tilde{\xi}(x_0, T - t_0)$ belongs to the core $C(x_0, T - t_0)$. For this, consider the value $\sum_{i \in S} \int_{t_0}^t \bar{\beta}(\tau) d\tau$:

$$\begin{aligned} \sum_{i \in S} \int_{t_0}^t \bar{\beta}(\tau) d\tau &= \sum_{i \in S} \left(\int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i)} \int_{t_0}^t \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau \right) = \\ &= \sum_{i \in S} \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_S^m(i)} \int_{t_0}^t \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau + \\ &\quad + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i) \setminus D_S^m(i)} \int_{t_0}^t \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau = \\ &= \sum_{i \in S} \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in \bigcup_{i \in S} D_S^m(i)} \int_{t_0}^t (\bar{\phi}_{i_1}^{p,q} + \dots + \bar{\phi}_{i_{m+1}}^{p,q}) \bar{h}_{p,q}(\tau) d\tau + \\ &\quad + \sum_{i \in S} \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in D_N^m(i) \setminus D_S^m(i)} \int_{t_0}^t \bar{\phi}_i^{p,q} \bar{h}_{p,q}(\tau) d\tau \geq \\ &\geq \sum_{i \in S} \int_{t_0}^t \bar{h}_i^i(\tau) d\tau + \sum_{m=1}^{n-1} \delta^{m-1} \sum_{\{p,q\} \in \bigcup_{i \in S} D_S^m(i)} \int_{t_0}^t \bar{h}_{p,q}(\tau) d\tau = \\ &= V(S; x_0, T - t_0) - V(S; \bar{x}(t), T - t). \end{aligned} \tag{27}$$

Then

$$\begin{aligned} \sum_{i \in S} \tilde{\xi}_i(t) &= \sum_{i \in S} \int_{t_0}^t \bar{\beta}(\tau) d\tau + \sum_{i \in S} \bar{\xi}_i(\bar{x}(t), T - t) \geq \\ &\geq V(S; x_0, T - t_0) - V(S; \bar{x}(t), T - t) + V(S; \bar{x}(t), T - t) = V(S; x_0, T - t_0) \end{aligned} \tag{28}$$

According to (28), $\tilde{\xi}(x_0, T - t_0)$ belongs to the core $C(x_0, T - t_0)$. This proves the strong-time consistency of the core.

6 Some Imputations from the Core

Proposition 2 shows that an imputation belongs to the core, if and only if, the payoff from the interaction of any pair of players is divided only between the players belonging to the shortest path between these players. The choice of a particular imputation from the core must correspond to some idea of fairness among the participants in the game. Consider some of such imputations.

6.1 The Shapley Value

The Shapley value [13] is a classic cooperative solution. According to this solution, each player receives an expected marginal contribution in the game with respect to a uniform distribution over the set of all permutations on the set of players.

The Shapley value is defined by:

$$Sh_i(x_0, T - t_0) = \sum_{S: i \in S} \frac{(s - 1)!(n - s)!}{n!} (V(S; x_0, T - t_0) - V(S \setminus i; x_0, T - t_0)).$$

The Shapley value in our game has the following form

$$Sh_i(x_0, T - t_0) = V(\{i\}; x_0, T - t_0) + \sum_{m=1}^r \frac{\delta^{m-1}}{m+1} \sum_{\{p,q\} \in D^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau. \tag{29}$$

Here $\delta^{m-1} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau$ – the payoff from the interaction of two players p and q with the length of shortest path m between them. It is assumed that this value is divided equally among all the players lying on the shortest path between p and q .

In the game under consideration, players can interact only if they are connected by a path of length no more than r . This means that the game can be considered as a game with restricted cooperation. Then, taking into account the form of the characteristic function, we can conclude that the Myerson value [6] coincides with the Shapley value in this game. And this solution is the unique allocation rule that satisfies fairness (loss of players i and j from the removal of a link ij are the same) and efficiency (component efficiency requires the total value of a component to be allocated among the members of the component).

6.2 Position Value Solution

Meessen [5] introduced another value function for communication situations, called the position value. This solution assumes a dual point of view and concentrates on the role of links. The idea of the position value is as follows. The communicative strength of a link is measured by means of the Shapley value of a kind of “dual” game on the links of graph. And assuming each player has veto power of the use of any arc that he is an endpoint of, it seems reasonable to divide the worth of an arc equally between the two players who are at its endpoints [1].

Denote by $\bar{V}(S; x_0, T - t_0)$ the characteristic function in “dual” game on the links. Let S^m is the set of pairs of vertices $\{p, q\}$, such that $p \in S, q \in S$, the distance between them is $m \leq r$, all vertices from the shortest path between p and q belong to S . Then

$$\bar{V}(S; x_0, T - t_0) = \sum_{m=1}^r \delta^{m-1} \sum_{\{p,q\} \in S^m} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau. \tag{30}$$

Denote as $D^m(i, j)$ the set of pairs $\{p, q\}$, such that the distance between them is $m \leq r$, link (i, j) belongs to the shortest path between p and q .

The Shapley value in link game has the form:

$$Sh_{i,j}(x_0, T - t_0) = \int_{t_0}^T \bar{h}_{i,j}(\tau) d\tau + \sum_{m=2}^r \frac{\delta^{m-1}}{m} \sum_{\{p,q\} \in D^m(i,j)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau. \tag{31}$$

Then the position value can be defined as

$$P_i(x_0, T - t_0) = \int_{t_0}^T \bar{h}_{i,j}(\tau) d\tau + \sum_{j \in K(i)} \frac{1}{2} Sh_{i,j}(x_0, T - t_0). \tag{32}$$

Let $D_N^m(i) = E^m(i) \cup F^m(i)$. Here $E^m(i)$ is the set of pairs of vertices $\{p, q\}$ such that the distance between them equals $m \leq r$, and i coincides with p or q (i is an endpoint of the path between p and q). $F^m(i)$ is the set of pairs of vertices $\{p, q\}$ such that the distance between them equals $m \leq r$, i lies on the path between p and q , and i is not an endpoint in it. Then the position value has the following form

$$P_i(x_0, T - t_0) = V(\{i\}; x_0, T - t_0) + \sum_{m=1}^r \frac{\delta^{m-1}}{2m} \sum_{\{p,q\} \in E^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau + \sum_{m=1}^r \frac{\delta^{m-1}}{m} \sum_{\{p,q\} \in F^m(i)} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau. \tag{33}$$

According to this imputation, the payoff of interaction between any pair of vertices p and q (with the distance m between them) is divided as follows.

Players p and q receive $\frac{1}{2m} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau$ and all other players from the shortest path between them receive $\frac{1}{m} \int_{t_0}^T \bar{h}_{p,q}(\tau) d\tau$. Such distribution of the payoff can be explained by the following reasoning. Endpoint players have the ability to remove only one link on the shortest path, and all intermediate players can remove two links.

7 Example

Consider a model of non-renewable resource extraction (see, for example, [2]). Let n players (regions or countries) exploit a non-renewable natural resource.

But we assume separate extraction, which may be due to territorial features. Further, $x_i(t)$ denotes the state corresponding to the resource stock at time t available to the extraction for player i . The dynamics of the stock is given by the following differential equations with the initial conditions $x_{i0} > 0$:

$$\dot{x}_i(t) = -u_i(t), \quad x_i(t_0) = x_{i0}. \tag{34}$$

Here, $u_i(t)$ denotes the extraction effort of player i at time t .

In compliance with the physical nature of the problem we require that $u_i(t) \geq 0$ and $x_i(t) \geq 0$ for all $t \geq t_0$, and that, if $x_i(t) = 0$, then the only feasible rate of extraction is $u_i(t) = 0$ for all $i = 1, \dots, n$.

The payoff of region i is positively related to its extraction effort and the extraction efforts of regions linked in the region network.

It is assumed that the instantaneous payoff is discounted at a constant rate $\rho > 0$. The gain region i obtains through interaction with region j has the form $h_i^j(t) = e^{-\rho t} b_i^j u_j^\mu(t)$. And $h_i^i(t) = e^{-\rho t} u_i^\mu(t)$ is the instantaneous gain that player i can obtain by itself. Here $\mu \in (0, 1)$. Then the objective function of the i th player is defined as

$$J_i(x_0, u_1, \dots, u_n) = \int_{t_0}^T e^{-\rho t} \left(u_i^\mu(t) + \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} b_i^j u_j^\mu(t) \right) dt. \tag{35}$$

The players cooperate in order to achieve the maximum total payoff:

$$\sum_{i \in N} J_i(x_0, u_1, \dots, u_n) = \sum_{i \in N} \int_{t_0}^T e^{-\rho t} a_i u_i^\mu(t) dt, \tag{36}$$

where $a_i = 1 + \sum_{m=1}^r \delta^{m-1} \sum_{j \in K^m(i)} b_j^i$.

Following [12], we define the Hamiltonian function as

$$H = \sum_{i \in N} e^{-\rho t} a_i u_i^\mu(t) - \sum_{i \in N} \psi_i(t) u_i(t).$$

Taking into account that $x_i(t) \geq 0$ and $u_i(t) \geq 0$, the optimal controls \bar{u}_i are obtained from the first-order optimality conditions $\frac{\partial H}{\partial u_i} = 0$ as

$$\bar{u}_i(t) = \left(\mu a_i \frac{e^{-\rho t}}{\psi_i(t)} \right)^{\frac{1}{1-\mu}}.$$

Solving the respective canonical system with the transversality conditions $\psi_i(T)x_i(T) = 0$, we obtain

$$\psi_i = \mu a_i \left[\frac{(1 - \mu) \left(e^{\frac{-\rho t_0}{1-\mu}} - e^{\frac{-\rho T}{1-\mu}} \right)}{\rho x_0} \right]^{1-\mu}.$$

The optimal state trajectory of player i is thus

$$\bar{x}_i(t) = x_{i0} + \frac{x_{i0} \left(e^{\frac{-\rho t}{1-\mu}} - e^{\frac{-\rho t_0}{1-\mu}} \right)}{\left(e^{\frac{-\rho t_0}{1-\mu}} - e^{\frac{-\rho T}{1-\mu}} \right)}.$$

The optimal value of the total payoff then equals

$$V(N; x_0, T - t_0) = \sum_{i \in N} J_i(x_0, \bar{x}) = \left(\frac{(1 - \mu) \left(e^{\frac{-\rho t_0}{1-\mu}} - e^{\frac{-\rho T}{1-\mu}} \right)}{\rho} \right)^{1-\mu} \sum_{i \in N} x_{i0}^\mu a_i.$$

According to (7) we can find $V(S; x_0, T - t_0)$:

$$V(S; x_0, T - t_0) = \left(\frac{(1 - \mu) \left(e^{\frac{-\rho t_0}{1-\mu}} - e^{\frac{-\rho T}{1-\mu}} \right)}{\rho} \right)^{1-\mu} \sum_{i \in S} x_{i0}^\mu a_{iS}$$

where $a_{iS} = 1 + \sum_{m=1}^r \delta^{m-1} \sum_{j \in K_S^m(i)} b_j^i$.

Consider now the numeric example for the case of 6-players game. Assume $r = 2$. Figure 1 shows the structure of the network in the game.

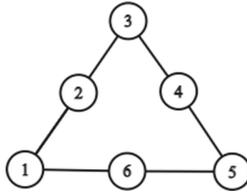


Fig. 1. The network structure of the game

According to Proposition 2 to find the core of the game we need to define the set of vectors

$$\begin{aligned} \phi^{1,2} &= (\phi_1^{1,2}, \phi_2^{1,2}), \quad \phi^{2,3} = (\phi_2^{2,3}, \phi_3^{2,3}), \quad \phi^{3,4} = (\phi_3^{3,4}, \phi_4^{3,4}) \\ \phi^{4,5} &= (\phi_4^{4,5}, \phi_5^{4,5}), \quad \phi^{5,6} = (\phi_5^{5,6}, \phi_6^{5,6}), \quad \phi^{1,6} = (\phi_1^{1,6}, \phi_6^{1,6}) \\ \phi^{1,3} &= (\phi_1^{1,3}, \phi_2^{1,3}, \phi_3^{1,3}), \quad \phi^{2,4} = (\phi_2^{2,4}, \phi_3^{2,4}, \phi_4^{2,4}) \\ \phi^{3,5} &= (\phi_3^{3,5}, \phi_4^{3,5}, \phi_5^{3,5}), \quad \phi^{4,6} = (\phi_4^{4,6}, \phi_5^{4,6}, \phi_6^{4,6}) \\ \phi^{1,5} &= (\phi_1^{1,5}, \phi_6^{1,5}, \phi_5^{1,5}), \quad \phi^{2,6} = (\phi_2^{2,6}, \phi_1^{2,6}, \phi_6^{2,6}). \end{aligned}$$

For every vector $\phi^{p,q}$ the sum of its components is equal to 1 and $0 \leq \phi_k^{p,q} \leq 1$.

Then every imputation from the core can be represented in the following form

$$\begin{aligned} \xi_1(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_1^1(\tau) + \phi_1^{1,2} (\bar{h}_2^1(\tau) + \bar{h}_1^2(\tau)) + \phi_1^{1,6} (\bar{h}_6^1(\tau) + \bar{h}_1^6(\tau)) + \\ & + \delta\phi_1^{1,3} (\bar{h}_3^1(\tau) + \bar{h}_1^3(\tau)) + \delta\phi_1^{1,5} (\bar{h}_5^1(\tau) + \bar{h}_1^5(\tau)) + \delta\phi_1^{2,6} (\bar{h}_2^6(\tau) + \bar{h}_6^2(\tau))] d\tau, \end{aligned}$$

$$\begin{aligned} \xi_2(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_2^2(\tau) + \phi_2^{1,2} (\bar{h}_2^1(\tau) + \bar{h}_1^2(\tau)) + \phi_2^{2,3} (\bar{h}_3^2(\tau) + \bar{h}_2^3(\tau)) + \\ & + \delta\phi_2^{1,3} (\bar{h}_3^1(\tau) + \bar{h}_1^3(\tau)) + \delta\phi_2^{2,4} (\bar{h}_4^2(\tau) + \bar{h}_2^4(\tau)) + \delta\phi_2^{2,6} (\bar{h}_2^6(\tau) + \bar{h}_6^2(\tau))] d\tau, \end{aligned}$$

$$\begin{aligned} \xi_3(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_3^3(\tau) + \phi_3^{2,3} (\bar{h}_3^2(\tau) + \bar{h}_2^3(\tau)) + \phi_3^{3,4} (\bar{h}_4^3(\tau) + \bar{h}_3^4(\tau)) + \\ & + \delta\phi_3^{1,3} (\bar{h}_3^1(\tau) + \bar{h}_1^3(\tau)) + \delta\phi_3^{3,5} (\bar{h}_5^3(\tau) + \bar{h}_3^5(\tau)) + \delta\phi_3^{2,4} (\bar{h}_2^4(\tau) + \bar{h}_4^2(\tau))] d\tau, \end{aligned}$$

$$\begin{aligned} \xi_4(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_4^4(\tau) + \phi_4^{4,5} (\bar{h}_5^4(\tau) + \bar{h}_4^5(\tau)) + \phi_4^{3,4} (\bar{h}_4^3(\tau) + \bar{h}_3^4(\tau)) + \\ & + \delta\phi_4^{2,4} (\bar{h}_4^2(\tau) + \bar{h}_2^4(\tau)) + \delta\phi_4^{4,6} (\bar{h}_6^4(\tau) + \bar{h}_4^6(\tau)) + \delta\phi_4^{3,5} (\bar{h}_5^3(\tau) + \bar{h}_3^5(\tau))] d\tau, \end{aligned}$$

$$\begin{aligned} \xi_5(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_5^5(\tau) + \phi_5^{4,5} (\bar{h}_5^4(\tau) + \bar{h}_4^5(\tau)) + \phi_5^{5,6} (\bar{h}_6^5(\tau) + \bar{h}_5^6(\tau)) + \\ & + \delta\phi_5^{1,5} (\bar{h}_5^1(\tau) + \bar{h}_1^5(\tau)) + \delta\phi_5^{4,6} (\bar{h}_6^4(\tau) + \bar{h}_4^6(\tau)) + \delta\phi_5^{3,5} (\bar{h}_5^3(\tau) + \bar{h}_3^5(\tau))] d\tau, \end{aligned}$$

$$\begin{aligned} \xi_6(x_0, T - t_0) = & \int_{t_0}^T [\bar{h}_6^6(\tau) + \phi_6^{5,6} (\bar{h}_6^5(\tau) + \bar{h}_5^6(\tau)) + \phi_6^{1,6} (\bar{h}_6^1(\tau) + \bar{h}_1^6(\tau)) + \\ & + \delta\phi_6^{2,6} (\bar{h}_2^6(\tau) + \bar{h}_2^6(\tau)) + \delta\phi_6^{4,6} (\bar{h}_6^4(\tau) + \bar{h}_4^6(\tau)) + \delta\phi_6^{1,5} (\bar{h}_5^1(\tau) + \bar{h}_1^5(\tau))] d\tau. \end{aligned}$$

For the Shapley value we have

$$\phi^{1,2} = \phi^{2,3} = \phi^{3,4} = \phi^{4,5} = \phi^{5,6} = \phi^{1,6} = \left(\frac{1}{2}, \frac{1}{2}\right),$$

$$\phi^{1,3} = \phi^{2,4} = \phi^{3,5} = \phi^{4,6} = \phi^{1,5} = \phi^{2,6} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

The payoff of player 1 according to the Shapley value is

$$Sh_1(x_0, T - t_0) = \int_{t_0}^T [\bar{h}_1^1(\tau) + \frac{1}{2} (\bar{h}_2^1(\tau) + \bar{h}_1^2(\tau)) + \frac{1}{2} (\bar{h}_6^1(\tau) + \bar{h}_1^6(\tau)) + \frac{1}{3} \delta (\bar{h}_3^1(\tau) + \bar{h}_1^3(\tau)) + \frac{1}{3} \delta (\bar{h}_5^1(\tau) + \bar{h}_1^5(\tau)) + \frac{1}{3} \delta (\bar{h}_2^6(\tau) + \bar{h}_2^6(\tau))] d\tau. \quad (37)$$

For the position value we use the following vectors:

$$\phi^{1,2} = \phi^{2,3} = \phi^{3,4} = \phi^{4,5} = \phi^{5,6} = \phi^{1,6} = \left(\frac{1}{2}, \frac{1}{2}\right),$$

$$\phi^{1,3} = \phi^{2,4} = \phi^{3,5} = \phi^{4,6} = \phi^{1,5} = \phi^{2,6} = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right).$$

The payoff of player 1 according to the position value is

$$P_1(x_0, T - t_0) = \int_{t_0}^T [\bar{h}_1^1(\tau) + \frac{1}{2} (\bar{h}_2^1(\tau) + \bar{h}_1^2(\tau)) + \frac{1}{2} (\bar{h}_6^1(\tau) + \bar{h}_1^6(\tau)) + \frac{1}{4} \delta (\bar{h}_3^1(\tau) + \bar{h}_1^3(\tau)) + \frac{1}{4} \delta (\bar{h}_5^1(\tau) + \bar{h}_1^5(\tau)) + \frac{1}{2} \delta (\bar{h}_2^6(\tau) + \bar{h}_2^6(\tau))] d\tau. \quad (38)$$

We assume the following values of the parameters: $\mu = 0.5$, $T = 100$, $t_0 = 0$, $\rho = 0.01$, $x_{0i} = 1000$, $i = 1, \dots, 6$, $\delta = 0.5$, $b_1^2 = b_1^3 = b_1^5 = b_2^4 = b_3^4 = b_4^6 = b_5^6 = b_6^1 = b_6^5 = b_3^5 = b_5^5 = b_4^6 = b_6^5 = 0.01$, $b_2^2 = b_4^4 = b_6^2 = b_4^3 = b_3^2 = b_4^5 = b_5^6 = 0.02$, $b_1^6 = b_2^6 = b_5^3 = b_4^5 = 0.03$. Then $V(N; x_0, T - t_0) = 4142.38$, $V(\{i\}; x_0, T - t_0) \approx 657, 52$. Figure 2 shows the optimal trajectory and optimal control of one player.

$$Sh(x_0, T - t_0) = (690.4, 688.2, 687.11, 693.68, 692.59, 690.4),$$

$$P(x_0, T - t_0) = (692.04, 687.11, 687.11, 694.51, 692.04, 689.57).$$

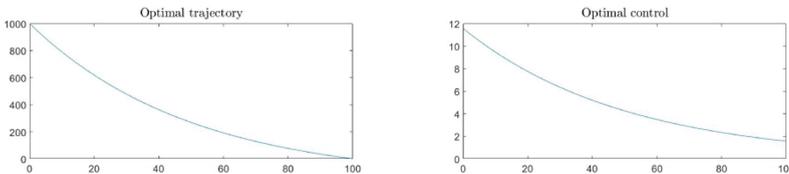


Fig. 2. Optimal trajectory and optimal control of player 1

To illustrate the strong-time consistency of the core assume players choose the Shapley value as cooperative solution, but change it on the position value at

the moment $t = 50$. Then the payoffs of players at the period $[t_0, 50]$ are (504.72, 503.12, 502.32, 507.12, 506.32, 504.72) and at the period $[50, 100]$ are (186.12, 184.79, 184.79, 186.78, 186.12, 185.46). The resulting imputation (690.84, 687.91, 687.11, 693.9, 692.44, 690.18) (see Eq. (24)) belongs to the core of the initial game.

8 Conclusion

One class of cooperative differential games on networks is investigated. It is assumed that players can interact with each other only if the distance between them on the network is not greater than a given value. A condition is derived under which the characteristic function is convex even in the presence of sufficiently large cycles. This expanded the class of problems that are described by the models discussed earlier.

A necessary and sufficient condition for the imputation to belong to the core is provided. The strong time-consistency of the core is proved. This means that players can switch at intermediate moments of the game to imputations from the core, different from the one chosen at the beginning. This procedure does not output resulting imputation from the core of the original game. As an illustrative example, a differential game of resource extraction on the network is investigated.

References

1. Borm, P.E.M., Owen, G., Tijs, S.H.: On the position value for communication situations. *SIAM J. Discrete Math.* **5**(3), 305–320 (1992)
2. Dockner, E., Jørgensen, S., Van Long, N., Sorger, G.: *Differential Games in Economics and Management Science*. Cambridge University Press, Cambridge (2000)
3. Gromova, E., Marova, E., Gromov, D.: A substitute for the classical Neumann-Morgenstern characteristic function in cooperative differential games. *J. Dyn. Games* **7**(2), 105–122 (2020). <https://doi.org/10.3934/jdg.2020007>
4. Mazalov, V.V., Trukhina, L.I.: Generating functions and the Myerson vector in communication networks. *Discrete Math. Appl.* **24**(5), 295–303 (2014). <https://doi.org/10.1515/dma-2014-0026>
5. Meessen, R.: *Communication games*. Master's thesis, Department of Mathematics, University of Nijmegen, The Netherlands (1988). (in Dutch)
6. Myerson, R.B.: Graphs and cooperation in games. *Math. Oper. Res.* **2**, 225–229 (1977)
7. Petrosyan, L.A.: Strong time-consistent differential optimality principles. *Vestn. Leningrad. Univ.* **4**, 35–40 (1993)
8. Petrosyan, L.A.: Cooperative differential games on networks. *Trudy Inst. Mat. I Mekh. UrO RAN* **16**, 143–150 (2010)
9. Petrosyan, L.A., Danilov, N.N.: Stable solutions in nonantagonistic differential games with transferable payoffs. *Vestn. Leningrad. Univ.* **1**, 52–79 (1979)
10. Petrosyan, L., Yeung, D.: Construction of dynamically stable solutions in differential network games. In: Tarasyev, A., Maksimov, V., Filippova, T. (eds.) *Stability, Control and Differential Games*. LNCISP, pp. 51–61. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42831-0_5

11. Petrosyan, L., Yeung, D., Pankratova, Y.: Dynamic cooperative games on networks. In: Strelalovsky, A., Kochetov, Y., Gruzdeva, T., Orlov, A. (eds.) MOTOR 2021. CCIS, vol. 1476, pp. 403–416. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86433-0_28
12. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E. F.: *Mathematical Theory of Optimal Processes*. CRC Press (1987)
13. Shapley, L.S.: A value for n-person games. In: Kuhn, H., Tucker, A., Eds., *Contributions to the Theory of Games II*. Princeton University Press, Princeton, pp. 307–317 (1953). <https://doi.org/10.1515/9781400881970-018>
14. Shapley, L.S.: Cores of convex games. *Int. J. Game Theory* **1**, 11–26 (1971)
15. Tur, A., Petrosyan, L.: Strong time-consistent solution for cooperative differential games with network structure. *Mathematics* **9**, 755 (2021). <https://doi.org/10.3390/math9070755>
16. Von Neumann, J., Morgenstern, O.: *Theory of Games and Economic Behavior*. Princeton University Press, Princeton (1953)

Author Index

- Albu, Alla 153
- Chebotareva, Angelina 221
- Chirkova, Julia V. 169
- Ćirković, Petar 79
- Danilova, Marina 3
- Davidović, Tatjana 79
- Đorđević, Predrag 79
- Dvinskikh, Darina 18
- Dvurechensky, Pavel 62
- Fominyh, Alexander 34
- Gasnikov, Alexander 18, 62
- Gromova, Ekaterina 221
- Gruzdeva, Tatiana V. 139
- Ivanov, Sergey V. 182
- Ivashko, Anna 194
- Khamisov, Oleg. O. 62
- Konnov, Igor 46
- Konovalchikova, Elena 194
- Kudria, Sergey 123
- Kuzyutin, Denis 235, 279
- Mamchur, Aleksandra V. 182
- Marakulin, Valeriy 210
- Matijević, Luka 94
- Mazalov, Vladimir V. 169
- Milićević, Miloš 79
- Mladenović, Nenad 108
- Pankratova, Yaroslavna 250
- Petrosyan, Leon 250, 295
- Ren, Jie 123
- Rettieva, Anna 264
- Rogozin, Alexander 62
- Rybin, Dmitry 123
- Skorodumova, Yulia 235
- Smirnova, Nadezhda 235, 279
- Su, Shimai 221
- Todosijević, Raca 108
- Tominin, Iaroslav 18
- Tominin, Vladislav 18
- Tur, Anna 295
- Urošević, Dragan 108
- Ushakov, Anton V. 139
- Vasilyev, Igor 123
- Voronina, Elizaveta 221
- Yarmoshik, Demyan 62
- Zhang, Dong 123
- Zubov, Vladimir 153