Springer Proceedings in Mathematics & Statistics

Martin Takáč
Tamás Terlaky *Editors*

Modeling and Optimization: Theory and Applications

MOPTA, Bethlehem, PA, USA, August 2016 Selected Contributions

Springer

Springer Proceedings in Mathematics & Statistics

Volume 213

Springer Proceedings in Mathematics & Statistics

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

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

Martin Takáč • Tamás Terlaky
Editors

Modeling and Optimization: Theory and Applications

MOPTA, Bethlehem, PA, USA, August 2016
Selected Contributions

Springer

Editors
Martin Takáč
Industrial and Systems
Engineering Department
Lehigh University
Bethlehem, PA, USA

Tamás Terlaky
Industrial and Systems
Engineering Department
Lehigh University
Bethlehem, PA, USA

ISSN 2194-1009 ISSN 2194-1017 (electronic)
Springer Proceedings in Mathematics & Statistics
ISBN 978-3-319-66615-0 ISBN 978-3-319-66616-7 (eBook)
DOI 10.1007/978-3-319-66616-7

Library of Congress Control Number: 2017956711

Mathematics Subject Classification (2010): 49-06, 49Mxx, 65Kxx, 90-06, 90Bxx, 90Cxx

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, express 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.

Printed on acid-free paper

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

Preface

This volume contains a selection of papers that were presented at the Modeling and Optimization: Theory and Applications (MOPTA) Conference held at Lehigh University in Bethlehem, Pennsylvania, USA, between August 17 and August 19, 2016. MOPTA 2016 aimed to bring together a diverse group of researchers and practitioners, working on both theoretical and practical aspects of continuous or discrete optimization. The goal was to host presentations on the exciting developments in different areas of optimization and at the same time provide a setting for close interaction among the participants.

The topics covered at MOPTA 2016 varied from algorithms for solving convex, combinatorial, nonlinear, and global optimization problems and addressed the application of optimization techniques in finance, electricity systems, healthcare, machine learning, and other leading fields. The nine papers contained in this volume represent a sample of these topics and applications and illustrate the broad diversity of ideas discussed at the conference. The first part of the name MOPTA highlights the role that modeling plays in the solution of an optimization problem, and indeed, some of the papers in this volume illustrate the benefits of effective modeling techniques tied with theoretical guarantees.

The paper by Befekadu et al. considers a stochastic decision problem, with dynamic risk measures, in which multiple risk-averse agents make their decisions to minimize their individual accumulated risk costs over a finite time horizon. The paper by Kampas et al. focuses on the problem of packing of general (nonidentical) ellipses in a circle with a minimum radius. Their approach is based on using embedded Lagrange multipliers. The paper by Chopra et al. considers the convex recoloring problem, i.e., to recolor the nodes of a colored graph using the smallest number of color changes, such that each color induces a connected subgraph. They considered a convex recoloring problem on a tree and proposed a column generation framework that efficiently solves the large-scale convex recoloring problem. The paper by Jadamba and Raciti proposed a variational inequality formulation of a migration model with random data. They assume a simple model of population distribution based on utility function theory. In contrast to recent work, they refined the previous model by allowing random fluctuations in the data of the problem.

The paper by Cho et al. develops a fast and reliable computational framework for the inverse problem of identifying variable parameters in general mixed variational problems. One of the main contributions of their work is a thorough derivation of efficient computation schemes for the evaluation of the gradient and the Hessian of the output least-squares functional both for a discrete and continuous case. The paper by Smirnov and Dmitrieva considers a minimization of the ℓ_p-norm of Dirichlet-type boundary controls for the 1D wave equation.

The paper by Liu and Takáč proposed a projected mini-batch semi-stochastic gradient descent method. This work improved both the theoretical complexity and practical performance of the general stochastic gradient descent method. They proved a linear convergence under weak strong convexity assumption for minimizing the sum of smooth convex functions subject to a compact polyhedral set, which remains popular across the machine learning community. The paper by Adams and Anjos considered the projection polytope constraints used during optimization of relaxed semidefinite problems. They proposed a bilevel second-order cone optimization approach to find the maximally violated projection polytope constraint according to a particular depth measure and reformulate the bilevel problem as a single-level mixed binary second-order cone optimization problem. The paper by Papp deals with polynomial optimization problems; especially, it deals with an approximation of the cone of nonnegative polynomials with the cone of sum-of-squares polynomials. This approximation is polynomial-time solvable for many NP-hard optimization problems using semidefinite optimization. The paper focuses on the numerical issue of such an approximation scheme.

We thank the sponsors of MOPTA 2016, namely, AIMMS, SAS, Gurobi, and SIAM. We also thank the host, Lehigh University, as well as the rest of the organizing committee: Frank Curtis, Luis Zuluaga, Larry Snyder, Ted Ralphs, Katya Scheinberg, Robert Storer, Aurélie Thiele, Boris Defourny, Alexander Stolyar, and Eugene Perevalov.

Bethlehem, PA, USA Martin Takáč
Bethlehem, PA, USA Tamás Terlaky
May 2017

Contents

Stochastic Decision Problems with Multiple Risk-Averse Agents

Getachew K. Befekadu, Alexander Veremyev, Vladimir Boginski, and Eduardo L. Pasiliao

Abstract We consider a stochastic decision problem, with dynamic risk measures, in which multiple risk-averse agents make their decisions to minimize their individual accumulated risk-costs over a finite-time horizon. Specifically, we introduce multi-structure dynamic risk measures induced from conditional g-expectations, where the latter are associated with the generator functionals of certain BSDEs that implicitly take into account the risk-cost functionals of the risk-averse agents. Here, we also assume that the solutions for such BSDEs *almost surely* satisfy a stochastic viability property w.r.t. a certain given closed convex set. Using a result similar to that of the Arrow–Barankin–Blackwell theorem, we establish the existence of consistent optimal decisions for the risk-averse agents, when the set of all Pareto optimal solutions, in the sense of viscosity solutions, for the associated dynamic programming equations is dense in the given closed convex set. Finally,

A preliminary version of this paper was presented at the Modeling and Optimization: Theory and Applications Conference, August 17–19, 2016, Bethlehem, PA, USA.

G.K. Befekadu (✉)
NRC, Air Force Research Laboratory & Department of Industrial System Engineering,
University of Florida - REEF, 1350 N. Poquito Rd, Shalimar, FL 32579, USA
e-mail: gbefekadu@ufl.edu

A. Veremyev
Department of Industrial System Engineering, University of Florida - REEF, 1350
N. Poquito Rd, Shalimar, FL 32579, USA
e-mail: averemyev@ufl.edu

V. Boginski
Industrial Engineering & Management Systems, University of Central Florida, 12800 Pegasus
Dr., P.O. Box 162993, Orlando, FL 32816-2993, USA
e-mail: Vladimir.Boginski@ucf.edu

E.L. Pasiliao
Munitions Directorate, Air Force Research Laboratory, 101 West Eglin Blvd, Eglin AFB,
FL 32542, USA
e-mail: pasiliao@eglin.af.mil

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_1

we comment on the characteristics of acceptable risks w.r.t. some uncertain future outcomes or costs, where results from the dynamic risk analysis are part of the information used in the risk-averse decision criteria.

Keywords Dynamic programming equation • Forward-backward SDEs • Multiple risk-averse agents • Pareto optimality • Risk-averse decisions • Value functions • Viscosity solutions

1 Introduction

Let $\bigl(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P}\bigr)$ be a probability space, and let $\{B_t\}_{t\geq 0}$ be a d-dimensional standard Brownian motion, whose natural filtration, augmented by all $\mathbb{P}$-null sets, is denoted by $\{\mathcal{F}_t\}_{t\geq 0}$, so that it satisfies the *usual hypotheses* (e.g., see [21]). We consider the following controlled-diffusion process over a given finite-time horizon $T > 0$

$$dX_t^{u_\cdot} = m\bigl(t, X_t^{u_\cdot}, (u_t^1, u_t^2, \cdots, u_t^n)\bigr)dt + \sigma\bigl(t, X_t^{u_\cdot}, (u_t^1, u_t^2, \cdots, u_t^n)\bigr)dB_t,$$
$$X_0^{u_\cdot} = x, \quad 0 \leq t \leq T, \tag{1}$$

where

- $X_\cdot^{u_\cdot}$ is an $\mathbb{R}^d$-valued controlled-diffusion process,
- $u_\cdot^j$ is a U^j-valued measurable decision process, which corresponds to the jth risk-averse agent (where U^j is an open compact set in $\mathbb{R}^{m_j}$, with $j = 1, 2, \ldots, n$); and, furthermore, $u_\cdot \triangleq (u_\cdot^1, u_\cdot^2, \cdots, u_\cdot^n)$ is an n-tuple of $\prod_{i=1}^n U^i$-valued measurable decision processes such that for all $t > s$, $(B_t - B_s)$ is independent of u_r for $r \leq s$ (nonanticipativity condition) and

$$\mathbb{E}\int_s^t |u_\tau|^2 d\tau < \infty \quad \forall t \geq s,$$

- m: $[0, T] \times \mathbb{R}^d \times \prod_{i=1}^n U^i \to \mathbb{R}^d$ is uniformly Lipschitz, with bounded first derivative, and
- $\sigma : [0, T] \times \mathbb{R}^d \times \prod_{i=1}^n U^i \to \mathbb{R}^{d\times d}$ is Lipschitz with the least eigenvalue of $\sigma\,\sigma^T$ uniformly bounded away from zero for all $(x, u) \in \mathbb{R}^d \times \prod_{i=1}^n U^i$ and $t \in [0, T]$, i.e.,

$$\sigma(t, x, u)\,\sigma^T(t, x, u) \succeq \lambda I_{d\times d}, \quad \forall (x, u) \in \mathbb{R}^d \times \prod\nolimits_{i=1}^n U^i,$$
$$\forall t \in [0, T],$$

for some $\lambda > 0$.

In this paper, we specifically consider a risk-averse decision problem for the above controlled-diffusion process, in which the decision makers (i.e., the *risk-averse agents* with differing risk-averse related responsibilities and/or information) choose their decisions from progressively measurable decision sets. That is, the jth-agent's decision $u^j_\cdot$ is a U^j-valued measurable control process from

$$\mathcal{U}^j_{[0,T]} \triangleq \Big\{ u^j \colon [0,T]\times\Omega \to U^j \,\Big|\, u^j \text{ is an } \{\mathcal{F}_t\}_{t\geq 0}\text{- adapted} \\ \text{and } \mathbb{E}\int_0^T |u^j_t|^2 dt < \infty \Big\}, \quad j = 1,2,\ldots,n. \tag{2}$$

Here, we also suppose that the risk-averse agents are "rational" (in the sense of making consistent decisions that minimize their individual accumulated risk-costs) with a certain n-tuple of measurable decision processes $\hat{u} = (\hat{u}^1_\cdot, \hat{u}^2_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod_{i=1}^n \mathcal{U}^i_{[0,T]}$. Moreover, we consider the following cost functionals that provide information about the accumulated risk-costs on the time interval $[0,T]$ w.r.t. each of the risk-averse agents, i.e.,

$$\xi^j_{0,T}(u^{\neg j}) = \int_0^T c_j\big(t, X_t^{u^{\neg j}_\cdot}, u^j_t\big) dt + \Psi_j(X_T^{u^{\neg j}_\cdot}), \quad j = 1,2,\ldots,n, \tag{3}$$

where

$$u^{\neg j}_\cdot \triangleq (\hat{u}^1_\cdot, \cdots, \hat{u}^{j-1}_\cdot, u^j_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod\nolimits_{i=1}^n \mathcal{U}^i_{[0,T]},$$

with $c_j \colon [0,T] \times \mathbb{R}^d \times U^j \to \mathbb{R}$ and $\Psi_j \colon \mathbb{R}^d \to \mathbb{R}$ are measurable functions. Note that the corresponding solution $X_t^{u^{\neg j}_\cdot}$, for $j \in \{1,2,\ldots,n\}$, in Eq. (1) depends on the n-tuple admissible risk-averse decisions $u^{\neg j}_\cdot \in \prod_{i=1}^n \mathcal{U}^i_{[0,T]}$ and it also depends on the initial condition $X_0^{u^{\neg j}_\cdot} = x$. As a result of this, for any time-interval $[t,T]$, with $t \in [0,T]$, the accumulated risk-costs $\xi^j_{t,T}$, for $j = 1,2,\ldots,n$, depend on the risk-averse decisions $u^{\neg j}_\cdot \in \prod_{i=1}^n \mathcal{U}^i_{[t,T]}$.[1] Moreover, we also assume that f, σ, c_j and Ψ_j, for $p \geq 1$, satisfy the following growth conditions

$$\big|m\big(t,x,u\big)\big| + \big|\sigma\big(t,x,u\big)\big| + \big|c_j\big(t,x,u\big)\big| + \big|\Psi_j\big(x\big)\big| \\ \leq C\big(1 + \big|x\big|^p + \big|u\big|\big), \quad \forall j \in \{1,2,\ldots,n\}, \tag{4}$$

for all $\big(t,x,u\big) \in [0,T] \times \mathbb{R}^d \times \prod_{i=1}^n U^i$ and for some constant $C > 0$.

[1] Here, we use the notation $u^{\neg j}$ to emphasize the dependence on $u^j_\cdot \in \mathcal{U}^j_{[t,T]}$, where $\mathcal{U}^j_{[t,T]}$, for any $t \in [0,T]$, denotes the sets of U^j-valued $\{\mathcal{F}^t_s\}_{s\geq t}$-adapted processes (see Definition 2).

Next, we introduce the following spaces that will be useful later in the paper.

- $L^2\big(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^d\big)$ is the set of $\mathbb{R}^d$-valued $\mathcal{F}_t$-measurable random variables ξ such that $\|\xi\|^2 = \mathbb{E}\{|\xi|^2\} < \infty$;
- $L^\infty\big(\Omega, \mathcal{F}_t, \mathbb{P}\big)$ is the set of $\mathbb{R}$-valued $\mathcal{F}_t$-measurable random variables ξ such that $\|\xi\| = \operatorname{ess\,inf} |\xi| < \infty$;
- $\mathcal{S}^2\big(t, T; \mathbb{R}^d\big)$ is the set of $\mathbb{R}^d$-valued adapted processes $\big(\varphi_s\big)_{t \le s \le T}$ on $\Omega \times [t, T]$ such that $\|\varphi\|_{[t,T]}^2 = \mathbb{E}\big\{\sup_{t \le s \le T} |\varphi_s|^2\big\} < \infty$;
- $\mathcal{H}^2\big(t, T; \mathbb{R}^d\big)$ is the set of $\mathbb{R}^d$-valued progressively measurable processes $\big(\varphi_s\big)_{t \le s \le T}$ such that $\|\varphi\|_{[t,T]}^2 = \mathbb{E}\big\{\int_t^T |\varphi_s|^2 ds\big\} < \infty$.

On the same probability space $\big(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \ge 0}, \mathbb{P}\big)$, we consider the following backward stochastic differential equation (BSDE)

$$-dY_t = g\big(t, Y_t, Z_t\big)dt - Z_t dB_t, \quad Y_T = \xi, \tag{5}$$

where the terminal value $Y_T = \xi$ belongs to $L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$ and the generator functional $g\colon \Omega \times [0, T] \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, with property that $\big(g\big(t, y, z\big)\big)_{0 \le t \le T}$ is progressively measurable for each $(y, z) \in \mathbb{R} \times \mathbb{R}^d$. We also assume that g satisfies the following assumption.

Assumption 1

(i) *g is Lipschitz in (y, z), i.e., there exists a constant $C > 0$ such that, $\mathbb{P}$-a.s., for any $t \in [0, T]$, $y_1, y_2 \in \mathbb{R}$ and $z_1, z_2 \in \mathbb{R}^d$*

$$\big|g\big(t, y_1, z_1\big) - g\big(t, y_2, z_2\big)\big| \le C\big(|y_1 - y_2| + \|z_1 - z_2\|\big).$$

(ii) $g\big(t, 0, 0\big) \in \mathcal{H}^2\big(t, T; \mathbb{R}\big)$.
(iii) *$\mathbb{P}$-a.s., for all $t \in [0, T]$ and $y \in \mathbb{R}$, $g\big(t, y, 0\big) = 0$.*

Then, we state the following lemma, which is used to establish the existence of a unique adapted solution (e.g., see [16] or [10] for additional discussions).

Lemma 2 *Suppose that Assumption 1 holds true. Then, for any $\xi \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$, the BSDE in (5), with terminal condition $Y_T = \xi$, i.e.,*

$$Y_t = \xi + \int_t^T g\big(s, Y_s, Z_s\big)ds - \int_t^T Z_s dB_s, \quad 0 \le t \le T \tag{6}$$

has a unique adapted solution

$$\big(Y_t^{T,g,\xi}, Z_t^{T,g,\xi}\big)_{0 \le t \le T} \in \mathcal{S}^2\big(0, T; \mathbb{R}\big) \times \mathcal{H}^2\big(0, T; \mathbb{R}^d\big). \tag{7}$$

In the following, we give the definition for a dynamic risk measure that is associated with the generator of BSDE in (5).

Definition 1 For any $\xi \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$, let $\big(Y_t^{T,g,\xi}, Z_t^{T,g,\xi}\big)_{0 \le t \le T} \in \mathcal{S}^2\big(0, T; \mathbb{R}\big) \times \mathcal{H}^2\big(0, T; \mathbb{R}^d\big)$ be the unique solution for the BSDE in (5) with terminal condition $Y_T = \xi$. Then, we define the dynamic risk measure $\rho_{t,T}^g$ of ξ by[2]

$$\rho_{t,T}^g\big[\xi\big] \triangleq Y_t^{T,g,\xi}. \tag{8}$$

Remark 1 Note that such a risk measure is widely used for evaluating the risk of stochastic processes or uncertain outcomes, and assists with stipulating minimum interventions required by financial institutions for risk management (e.g., see [4, 9, 10, 12, 20] or [7] for related discussions). In the following section, we introduce multi-structure dynamic risk measures induced from conditional g-expectations, where the latter are associated with the generator functionals of certain BSDEs that implicitly take into account the risk-cost functionals of the risk-averse agents.

Next, let us recall the following comparison result, which is restricted to one-dimensional BSDEs (e.g., see [17]).

Lemma 3 *Given two generators g_1 and g_2 satisfying Assumption 1 and two terminal conditions $\xi_1, \xi_2 \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$. Let $\big(Y_t^1, Z_t^1\big)$ and $\big(Y_t^2, Z_t^2\big)$ be the solution pairs corresponding to $\big(\xi_1, g_1\big)$ and $\big(\xi_2, g_2\big)$, respectively. Then, we have*

(i) *Monotonicity: If $\xi_1 > \xi_2$ and $g_1 > g_2$, $\mathbb{P}$-a.s., then $Y_t^1 > Y_t^2$, $\mathbb{P}$-a.s., for all $t \in [0, T]$;*
(ii) *Strictly monotonicity: In addition to (i) above, if we assume that $\mathbb{P}\big(\xi_1 > \xi_2\big) > 0$, then $\mathbb{P}\big(Y_t^1 > Y_t^2\big) > 0$, for all $t \in [0, T]$.*

Moreover, if the generator functional g satisfies Assumption 1, then a family of time-consistent dynamic risk measures $\big\{\rho_{t,T}^g\big\}_{t \in [0,T]}$ has the following properties (see [20] for additional discussions).

Property 1

(i) *Convexity*: If g is convex for every fixed $(t, \omega) \in [0, T] \times \Omega$, then for all $\xi_1, \xi_2 \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$ and for all $\pi \in L^\infty\big(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}\big)$ such that $0 \le \pi \le 1$

$$\rho_{t,T}^g\big[\pi \xi_1 + (1 - \pi)\xi_2\big] \le \pi \rho_{t,T}^g\big[\xi_1\big] + (1 - \pi)\rho_{t,T}^g\big[\xi_1\big];$$

(ii) *Monotonicity*: For $\xi_1, \xi_2 \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$ such that $\xi_1 > \xi_2$ $\mathbb{P}$-a.s., then

$$\rho_{t,T}^g\big[\xi_1\big] > \rho_{t,T}^g\big[\xi_2\big], \quad \mathbb{P}\text{-}a.s.;$$

[2] Here, we remark that, for any $t \in [0, T]$, the conditional g-expectation (denoted by $\mathcal{E}_g\big[\xi | \mathcal{F}_t\big]$) is also defined by

$$\mathcal{E}_g\big[\xi | \mathcal{F}_t\big] \triangleq Y_t^{T,g,\xi}.$$

(iii) *Trans-invariance*: For all $\xi \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$ and $\nu \in L^2\big(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}\big)$

$$\rho_{t,T}^{g}\big[\xi + \nu\big] = \rho_{t,T}^{g}\big[\xi\big] + \nu;$$

(iv) *Positive-homogeneity*: For all $\xi \in L^2\big(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}\big)$ and for all $\pi \in L^\infty\big(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}\big)$ such that $\pi > 0$

$$\rho_{t,T}^{g}\big[\pi\xi\big] = \pi\rho_{t,T}^{g}\big[\xi\big];$$

(v) *Normalization*: $\rho_{t,T}^{g}\big[0\big] = 0$ for $t \in [0, T]$.

Remark 2 Note that, since the seminal work of Artzner et al. [4], there have been studies on axiomatic dynamic risk measures, coherency and consistency in the literature (e.g., see [9, 12, 20, 22] or [7]). Particularly relevant for us is a family of time-consistent dynamic risk measures induced from conditional g-expectations that satisfies the above properties (i)–(v).

Here, it is worth mentioning that some interesting studies on the dynamic risk measures, based on the conditional g-expectations, have been reported in the literature (e.g., see [7, 20] and [22] for establishing connection between the risk measures and the generator of BSDE; and see also [24] for characterizing the generator of BSDE according to different risk measures). Moreover, such risk measures are widely used for evaluating the risk of uncertain future outcomes or costs, and also assisting with stipulating minimum interventions for risk management (e.g., see [4, 9, 10, 12, 20] or [7] for related discussions). Recently, the authors in [23] and [5] have provided interesting results on the risk-averse decision problem for Markov decision processes, in discrete-time setting, and, respectively, a hierarchical risk-averse framework for systems governed by controlled-diffusion processes. Note that the rationale behind our framework follows in some sense the settings of these papers. However, to our knowledge, the problem of risk-aversion for systems governed by controlled-diffusion processes has not been addressed in the context of multiple risk-averse agents argument, and it is important because it provides a mathematical framework that shows how a such framework can be systematically used to obtain consistently optimal risk-averse decisions.

The remainder of this paper is organized as follows. In Sect. 2, using the basic remarks made in Sect. 1, we state our risk-averse decision problem systems governed by controlled-diffusion processes with multiple risk-averse agents. In Sect. 3, we present our main results—where we introduce a framework that requires a "rational" cooperation among the risk-averse agents so as to achieve an overall optimal risk-averseness. Moreover, we establish the existence of optimal risk-averse solutions for the associated risk-averse dynamic programming equations. Finally, Sect. 4 provides further remarks. For the sake of readability, all proofs are presented in the Appendix section.

2 Problem Formulation

In order to make our problem formulation more precise, for any $(t, x) \in [0, T] \times \mathbb{R}^d$, we consider the following forward-SDE with an initial condition $X_t^{t,x;u_\cdot^{\neg j}} = x$, for $j \in \{1, 2, \dots, n\}$,

$$\begin{aligned} dX_s^{t,x;u_\cdot^{\neg j}} &= m\big(s, X_s^{t,x;u_\cdot^{\neg j}}, (\hat{u}_s^1, \cdots, \hat{u}_s^{j-1}, u_s^j, \hat{u}_s^{j+1}, \cdots, \hat{u}_s^n)\big)dt \\ &\qquad + \sigma\big(s, X_s^{t,x;u_\cdot^{\neg j}}, (\hat{u}_s^1, \cdots, \hat{u}_s^{j-1}, u_s^j, \hat{u}_s^{j+1}, \cdots, \hat{u}_s^n)\big)dB_t, \\ X_t^{t,x;u_\cdot^{\neg j}} &= x, \quad t \le s \le T, \end{aligned} \tag{9}$$

where $u_\cdot^{\neg j} = (\hat{u}_s^1, \cdots, \hat{u}_s^{j-1}, u_s^j, \hat{u}_s^{j+1}, \cdots, \hat{u}_s^n)$ is an n-tuple of $\prod_{i=1}^n U^j$-valued measurable decision processes.

Let $\big\{\xi_j^{Target}\big\}_{j=1}^n$ be a set of real-valued random variables from $L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$ and we further suppose that the data ξ_j^{Target} take the following forms:

$$\xi_j^{Target} = \Psi_j(X_T^{t,x;u_\cdot^{\neg j}}), \quad j = 1, 2, \dots, n, \quad \mathbb{P} - a.s. \tag{10}$$

Moreover, we introduce the following risk-value functions

$$V_j^{u^j}\big(t, x\big) = \rho_{t,T}^{g_j}\big[\xi_{t,T}^j\big(u^{\neg j}\big)\big], \quad j = 1, 2, \dots, n, \tag{11}$$

where

$$\xi_{t,T}^j\big(u^{\neg j}\big) = \int_t^T c_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^j\big)ds + \Psi_j(X_T^{t,x;u_\cdot^{\neg j}}). \tag{12}$$

Then, taking into account Eq. (10) (and with the Markovian framework), we can express the above risk-value functions using standard-BSDEs as follows:

$$\begin{aligned} V_j^{u^j}\big(t, x\big) &\triangleq Y_s^{j,t,x;u_\cdot^{\neg j}} \\ &= \Psi_j(X_T^{t,x;u_\cdot^{\neg j}}) + \int_t^T g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, Y_s^{j,t,x;u_\cdot^{\neg j}}, Z_s^{j,t,x;u_\cdot^{\neg j}}\big)ds \\ &\qquad - \int_t^T Z_s^{j,t,x;u_\cdot^{\neg j}} dB_s, \quad j = 1, 2, \dots, n, \end{aligned} \tag{13}$$

where

$$\begin{aligned} &g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, Y_s^{j,t,x;u_\cdot^{\neg j}}, Z_s^{j,t,x;w}\big) \\ &\qquad = c_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^j\big) + g\big(s, Y_s^{j,t,x;u_\cdot^{\neg j}}, Z_s^{j,t,x;u_\cdot^{\neg j}}\big). \end{aligned}$$

and further noting the conditions in (4), then the pairs $\bigl(Y_s^{j,t,x;u_{\cdot}^{\neg j}}, Z_s^{j,t,x;u_{\cdot}^{\neg j}}\bigr)_{t \le s \le T}$ are adapted solutions on $[t,T] \times \Omega$ and belong to $\mathcal{S}^2\bigl(t,T;\mathbb{R}\bigr) \times \mathcal{H}^2\bigl(t,T;\mathbb{R}^d\bigr)$. Equivalently, we can also rewrite (13) as a family of BSDEs on the probability space $\bigl(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{t\ge 0}\bigr)$, i.e., for $s \in [t,T]$,

$$\begin{aligned} -dY_s^{j,t,x;u_{\cdot}^{\neg j}} &= g_j\bigl(s, X_s^{t,x;u_{\cdot}^{\neg j}}, Y_s^{j,t,x;u_{\cdot}^{\neg j}}, Z_s^{j,t,x;u_{\cdot}^{\neg j}}\bigr)ds - Z_s^{j,t,x;u_{\cdot}^{\neg j}} dB_s, \\ Y_T^{j,t,x;u_{\cdot}^{\neg j}} &= \Psi_j(X_T^{t,x;u_{\cdot}^{\neg j}}), \quad j = 1,2,\ldots,n. \end{aligned} \tag{14}$$

In the following, we denote the solutions $\bigl(Y_s^{1,t,x;u_{\cdot}^{\neg 1}}, Y_s^{2,t,x;u_{\cdot}^{\neg 2}}, \cdots, Y_s^{n,t,x;u_{\cdot}^{\neg n}} \bigr) \in \prod_{i=1}^n \mathcal{S}^2\bigl(t,T;\mathbb{R}\bigr)$ and $\bigl(Z_s^{1,t,x;u_{\cdot}^{\neg 1}}, Z_s^{2,t,x;u_{\cdot}^{\neg 2}}, \cdots, Z_s^{n,t,x;u_{\cdot}^{\neg n}} \bigr) \in \prod_{i=1}^n \mathcal{H}^2\bigl(t,T;\mathbb{R}^d\bigr)$ by bold letters $\mathbf{Y}_s^{t,x;u_{\cdot}}$ and $\mathbf{Z}_s^{t,x;u_{\cdot}}$, respectively, for any $t \in [0,T]$ and for $s \in [t,T]$. Similarly, the family of BSDEs in (14) can be rewritten as a multi-dimensional BSDE as follows:

$$\begin{aligned} -d\mathbf{Y}_s^{t,x;u_{\cdot}} &= \mathbf{G}\bigl(s, X_s^{t,x;u_{\cdot}}, \mathbf{Y}_s^{t,x;u_{\cdot}}, \mathbf{Z}_s^{t,x;u_{\cdot}}\bigr)ds - \mathbf{Z}_s^{t,x;u} dB_s, \ \ s \in [t,T], \\ \mathbf{Y}_T^{t,x;u_{\cdot}} &= \boldsymbol{\Psi}(X_T^{t,x;u_{\cdot}}), \end{aligned} \tag{15}$$

where

$$\begin{aligned} &\mathbf{G}\bigl(s, X_s^{t,x;u_{\cdot}}, \mathbf{Y}_s^{t,x;u_{\cdot}}, \mathbf{Z}_s^{t,x;u_{\cdot}}\bigr) \\ &\quad = \text{block diag}\Bigl\{ g_1\bigl(s, X_s^{t,x;u_{\cdot}^{\neg 1}}, Y_s^{1,t,x;u_{\cdot}^{\neg 1}}, Z_s^{1,t,x;u_{\cdot}^{\neg 1}}\bigr), \\ &\quad\ \ g_2\bigl(s, X_s^{t,x;u_{\cdot}^{\neg 2}}, Y_s^{2,t,x;u_{\cdot}^{\neg 2}}, Z_s^{2,t,x;u_{\cdot}^{\neg 2}}\bigr), \cdots, g_n\bigl(s, X_s^{t,x;u_{\cdot}^{\neg n}}, Y_s^{n,t,x;u_{\cdot}^{\neg n}}, Z_s^{n,t,x;u_{\cdot}^{\neg n}}\bigr)\Bigr\} \end{aligned}$$

and

$$\boldsymbol{\Psi}(X_T^{t,x;u_{\cdot}}) = \bigl(\Psi_1(X_T^{t,x;u_{\cdot}^{\neg 1}}),\ \Psi_2(X_T^{t,x;u_{\cdot}^{\neg 2}}),\ \cdots,\ \Psi_n(X_T^{t,x;u_{\cdot}^{\neg n}}) \bigr).$$

Let K be a closed convex set in $\mathbb{R}^n$, then we recall the notion of viability property for the BSDE in (15) (cf. Eqs. (13) and (14)).

Definition 2 Let $\hat{u}_{\cdot} = (\hat{u}_{\cdot}^1, \hat{u}_{\cdot}^2, \cdots, \hat{u}_{\cdot}^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$ be an n-tuple of "rational" preferable decisions for the risk-averse agents. Then, for a nonempty closed convex set $K \subset \mathbb{R}^n$ and for $u_{\cdot}^j \in \mathcal{U}_{[0,T]}^j$, with $j = 1,2,\ldots,n$

(a) A stochastic process $\bigl\{\mathbf{Y}_t^{0,x;u_{\cdot}},\ t \in [0,T]\bigr\}$ is viable in K if and only if for $\mathbb{P}$-*almost* $\omega \in \Omega$

$$\mathbf{Y}_t^{0,x;u_{\cdot}}(\omega) \in K, \quad \forall t \in [0,T]. \tag{16}$$

(b) The closed convex set K enjoys the Backward Stochastic Viability Property (BSVP) for the equation in (15) if and only if for all $\tau \in [0, T]$, with Eq. (10), i.e.,

$$\forall\, \boldsymbol{\Xi}^{Target} = \bigl(\xi_1^{Target}, \xi_2^{Target}, \cdots, \xi_n^{Target}\bigr) \in L^2\bigl(\Omega, \mathcal{F}_\tau, \mathbb{P}; \mathbb{R}^n\bigr), \tag{17}$$

there exists a solution pair $\bigl(\mathbf{Y}_\cdot^{0,x;u_\cdot}, \mathbf{Z}_\cdot^{0,x;u_\cdot}\bigr)$ to the BSDE in (15) over the time interval $[0, \tau]$,

$$\mathbf{Y}_s^{0,x;u_\cdot} = \boldsymbol{\Xi}^{Target} + \int_s^\tau \mathbf{G}\bigl(r, X_r^{0,x;u_\cdot}, \mathbf{Y}_r^{0,x;u_\cdot}, \mathbf{Z}_r^{0,x;u_\cdot}\bigr) dr - \int_s^\tau \mathbf{Z}_r^{0,x;u} dB_r, \tag{18}$$

with

$$\bigl(\mathbf{Y}_\cdot^{0,x;u_\cdot}, \mathbf{Z}_\cdot^{0,x;u_\cdot}\bigr) \in \prod\nolimits_{i=1}^n \mathcal{S}^2\bigl(0, \tau; \mathbb{R}\bigr) \times \prod\nolimits_{i=1}^n \mathcal{H}^2\bigl(0, \tau; \mathbb{R}^d\bigr),$$

such that $\bigl\{\mathbf{Y}_s^{0,x;u_\cdot},\ s \in [0, \tau]\bigr\}$ is viable in K.

For the above given closed convex set K, let us define the projection of a point a onto K as follows:

$$\Pi_K(a) = \Bigl\{b \in K \,\Big|\, |a - b| = \min_{c \in K} |a - c| = d_K(a)\Bigr\}. \tag{19}$$

Notice that, since K is convex, from the Motzkin's theorem, Π_K is single-valued. Further, we recall that $d_K^2(\cdot)$ is convex; and thus, due to Alexandrov's theorem [1], $d_K^2(\cdot)$ is almost everywhere twice differentiable.

Assume that there exists an n-tuple of "rational" decisions $\hat{u} = (\hat{u}_\cdot^1, \hat{u}_\cdot^2, \cdots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$ which is preferable by all risk-averse decision-making agents. Moreover, on the space $C_b^{1,2}([t, T] \times \mathbb{R}^d; \mathbb{R}^n)$, for any $(t, x) \in [0, T] \times \mathbb{R}^d$, we consider the following system of semilinear parabolic partial differential equations (PDEs)

$$\left.\begin{aligned} \frac{\partial \varphi_j(t,x)}{\partial t} + \inf\nolimits_{u^j \in U^j} \Bigl\{&\mathcal{L}_t^{u^{\neg j}} \varphi_j(t,x) \\ &+ g_j\bigl(t, \varphi(t,x), D_x\varphi_j(t,x) \cdot \sigma(t, x, u^{\neg j})\bigr)\Bigr\} = 0 \\ &\qquad\qquad\qquad j = 1, 2, \ldots, n \end{aligned}\right\} \tag{20}$$

with the following boundary condition

$$\begin{aligned} \varphi(T, x) &= \boldsymbol{\Psi}(x), \\ &\equiv \bigl(\Psi_1(x),\ \Psi_2(x),\ \cdots,\ \Psi_n(x)\bigr), \quad x \in \mathbb{R}^d, \end{aligned} \tag{21}$$

where, for any $\phi(x) \in C_0^\infty(\mathbb{R}^d)$, the second-order linear operators $\mathcal{L}_t^{u^{\neg j}}$ are given by

$$\mathcal{L}_t^{u^{\neg j}} \phi(x) = \frac{1}{2} \operatorname{tr}\left\{a(t, x, u^{\neg j}) D_x^2 \phi(x)\right\} + m(t, x, u^{\neg j}) D_x \phi(x),$$
$$t \in [0, T], \quad j = 1, 2, \ldots n, \tag{22}$$

with $a(t, x, u^{\neg j}) = \sigma(t, x, u^{\neg j}) \sigma^T(t, x, u^{\neg j})$, D_x and D_x^2, (with $D_x^2 = \left(\partial^2 / \partial x_k \partial x_l\right)$) are the gradient and the Hessian (w.r.t. the variable x), respectively.

Remark 3 Here, we remark that the above system of equations in (20) together with (21) is associated with the decision problem for the risk-averse agents, restricted to $\Sigma_{[t,T]}$. Moreover, such a system of equations represents a generalized family of HJB equation with additional terms g_j for $j = 1, 2, \ldots, n$. Note that the problem of FBSDEs (cf. Eqs. (9) and (15) or (14)) and the solvability of the related system of semilinear parabolic PDEs have been well studied in literature (e.g., see [2, 13, 15, 17, 18], and [19]).

Next, we recall the definition of viscosity solutions for (20) along with (21) (e.g., see [8, 11] or [14] for additional discussions on the notion of viscosity solutions).

Definition 3 The function $\varphi \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}^n$ is a viscosity solution for (20) together with the boundary condition in (21), if the following conditions hold

(i) for every $\psi \in C_b^{1,2}([0, T], \times \mathbb{R}^d; \mathbb{R}^n)$ such that $\psi \geq \varphi$ on $[0, T] \times \mathbb{R}^d$,

$$\sup_{(t,x)} \left\{\varphi(t, x) - \psi(t, x)\right\} = 0, \tag{23}$$

and for $(t_0, x_0) \in [0, T] \times \mathbb{R}^d$ such that $\psi(t_0, x_0) = \varphi(t_0, x_0)$ (i.e., a local maximum at (t_0, x_0)), then we have

$$\frac{\partial \psi_j(t_0, x_0)}{\partial t} + \inf_{u^j \in U^j} \left\{\mathcal{L}_t^{u^{\neg j}} \psi_j(t_0, x_0) \right.$$
$$\left. + g_j\big(t_0, x_0, \psi(t_0, x_0), D_x \psi_j(t_0, x_0) \cdot \sigma(t_0, x_0, u^{\neg j})\big)\right\} \geq 0, \tag{24}$$

(ii) for every $\psi \in C_b^{1,2}([0, T], \times \mathbb{R}^d; \mathbb{R}^n)$ such that $\psi \leq \varphi$ on $[0, T] \times \mathbb{R}^d$,

$$\inf_{(t,x)} \left\{\varphi(t, x) - \psi(t, x)\right\} = 0, \tag{25}$$

and for $(t_0, x_0) \in [0, T] \times \mathbb{R}^d$ such that $\psi(t_0, x_0) = \varphi(t_0, x_0)$ (i.e., a local minimum at (t_0, x_0)), then we have

$$\frac{\partial \psi_j(t_0, x_0)}{\partial t} + \inf_{u^j \in U^j} \left\{\mathcal{L}_t^{u^{\neg j}} \psi_j(t_0, x_0) \right.$$
$$\left. + g_j\big(t_0, x_0, \psi(t_0, x_0), D_x \psi_j(t_0, x_0) \cdot \sigma(t_0, x_0, u^{\neg j})\big)\right\} \leq 0, \tag{26}$$

for $j = 1, 2, \ldots, n$.

Next, let us define the viability property for the system of semilinear parabolic PDEs in (20) as follows.

Definition 4 The system of semilinear parabolic PDEs in (20) enjoys the viability property w.r.t. the closed convex set K if and only if, for any $\boldsymbol{\Psi} \in C_p(\mathbb{R}^d; \mathbb{R}^n)$ taking values in K, the viscosity solution to (20) satisfies

$$\forall (t, x) \in [0, T] \times \mathbb{R}^d, \quad \varphi(t, x) \in K. \tag{27}$$

Later in Sect. 3, assuming the Markovian framework, we provide additional results that establish a connection between the viability property of the BSDE in (15), w.r.t. the closed convex set K, and the solutions, in the sense viscosity, for the system of semilinear parabolic PDEs in (20).

In what follows, we introduce a framework that requires a "rational" cooperation among the risk-averse agents so as to achieve an overall risk-averseness (in the sense of Pareto optimality). For example, for any $t \in [0, T]$, let us assume that

$$\hat{u}_{\cdot} = (\hat{u}_{\cdot}^1, \hat{u}_{\cdot}^2, \cdots, \hat{u}_{\cdot}^n) \in \prod\nolimits_{i=1}^{n} \mathcal{U}_{[t,T]}^i$$

is an n-tuple of "rational" preferable decisions for the risk-averse agents, then the problem of finding an optimal risk-averse decision for the jth-agent, where $j \in \{1, 2, \ldots, n\}$, that minimizes the jth-accumulated risk-cost functional, is equivalent to finding an optimal solution for

$$\inf_{u_{\cdot}^j \in \mathcal{U}_{[t,T]}^j} J_j\big[\big(u^{\neg j}\big)\big], \tag{28}$$

where

$$J_j\big[\big(u^{\neg j}\big)\big] = \rho_{t,T}^{g_j}\big[\xi_{t,T}^j\big(u^{\neg j}\big)\big], \tag{29}$$

with $u_{\cdot}^{\neg j} = (\hat{u}_{\cdot}^1, \ldots, \hat{u}_{\cdot}^{j-1}, u_{\cdot}^j, \hat{u}_{\cdot}^{j+1}, \cdots, \hat{u}_{\cdot}^n) \in \prod_{i=1}^{n} \mathcal{U}_{[t,T]}^i$.

Remark 4 Here, we remark that the generator functionals g_j, for $j = 1, 2, \ldots, n$, contain a common term g that acts on different processes (see Eqs. (13) and (14)). Moreover, due to differing risk-cost functionals w.r.t. each of the agents, we also observe that $\big\{\rho_{t,T}^{g_j}\big[\cdot\big]\big\}_{j=1}^{n}$, for $t \in [0, T]$, in Eq. (29) provide multi-structure, time-consistent, dynamic risk measures vis-á-vis some uncertain future outcomes specified by a set of random variables from $L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$.

Note that, for any given $u_{\cdot}^j \in \mathcal{U}_{[t,T]}^j$, if the forward–backward stochastic differential equations (FBSDEs) in (9) and (15) (cf. Eqs. (13) and (14)) admit unique solutions and, further, $\mathbf{Y}_s^{t,x;u}(\omega) \in K$, for $\mathbb{P}$- *almost* $\omega \in \Omega$ and for all $s \in [t, T]$ and for $t \in [0, T]$. Then, any "rational" preferable decisions for the jth-agent satisfy the following

$$\hat{u}^j_\cdot \in \Big\{\tilde{u}_\cdot \in \mathcal{U}^j_{[t,T]} \,\Big|\, \rho^{g_j}_{t,T}\big[\xi^j_{t,T}\big(\tilde{u}^{\neg j}_\cdot\big)\big] \le \rho^{g_j}_{t,T}\big[\xi^j_{t,T}\big(u^{\neg j}_\cdot\big)\big],$$
$$\forall\,(\hat{u}^1_\cdot, \ldots, \hat{u}^{j-1}_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod\nolimits_{i\neq j} \mathcal{U}^i_{[t,T]},$$
$$\forall j \in \{1, 2, \ldots, n\}, \quad \mathbb{P}-a.s.\Big\}, \tag{30}$$

where $\tilde{u}^{\neg j}_\cdot = (\hat{u}^1_\cdot, \ldots, \hat{u}^{j-1}_\cdot, \tilde{u}^j_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod_{i=1}^n \mathcal{U}^i_{[t,T]}$.[3]

Next, we introduce the following definition for an admissible risk-averse decision system $\Sigma_{[t,T]}$, for $t \in [0, T]$, with multi-structure dynamic risk measures, which provides a logical construct for our main results (e.g., see also [15]).

Definition 5 For a given finite-time horizon $T > 0$, we call $\Sigma_{[t,T]}$, with $t \in [0, T]$, an admissible risk-averse decision system, if it satisfies the following conditions:

- $\big(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\ge 0}, \mathbb{P}\big)$ is a complete probability space;
- $\big\{B_s\big\}_{s\ge t}$ is a d-dimensional standard Brownian motion defined on $\big(\Omega, \mathcal{F}, \mathbb{P}\big)$ over $[t, T]$ and $\mathcal{F}^t \triangleq \big\{\mathcal{F}^t_s\big\}_{s\in[t,T]}$, where $\mathcal{F}^t_s = \sigma\big\{\big(B_s;\, t \le s \le T\big)\big\}$ is augmented by all $\mathbb{P}$-null sets in $\mathcal{F}$;
- $u^j_\cdot \colon \Omega \times [s, T] \to U^j$, for $j = 1, 2, \ldots, n$, are $\big\{\mathcal{F}^t_s\big\}_{s\ge t}$-adapted processes on $\big(\Omega, \mathcal{F}, \mathbb{P}\big)$ with
$$\mathbb{E}\int_s^T |u^j_\tau|^2 d\tau < \infty, \quad s \in [t, T];$$
- For any $x \in \mathbb{R}^d$, the FBSDEs in (9) and (15) admit a unique solution set
$$\big\{X^{s,x;u^{\neg j}_\cdot}_\cdot, Y^{j,s,x;u^{\neg j}_\cdot}_\cdot, Z^{j,s,x;u^{\neg j}_\cdot}_\cdot\big\}_{j=1}^n \quad \text{on} \quad \big(\Omega, \mathcal{F}, \mathcal{F}^t, \mathbb{P}\big)$$
and
$$\mathbf{Y}^{s,x;u}_\cdot(\omega) = \big(Y^{1,s,x;u^{\neg 1}_\cdot}_\cdot(\omega),\, Y^{2,s,x;u^{\neg 2}_\cdot}_\cdot(\omega), \cdots, Y^{n,s,x;u^{\neg n}_\cdot}_\cdot(\omega)\big) \in K,$$
$$\mathbb{P}-almost\ \omega \in \Omega, \quad \forall s \in [t, T].$$

Then, with restriction to the above admissible system, we can state the risk-averse decision problem as follows.

Problem Find an n-tuple of optimal preferable decisions for the risk-averse agents, i.e., $\hat{u}_\cdot = (\hat{u}^1_\cdot, \hat{u}^2_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod_{i=1}^n \mathcal{U}^i_{[t,T]}$, with $\xi^{Target}_j \in L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$, for $j \in \{1, 2, \ldots, n\}$, such that

[3] In the paper, we assume that the set on the right-hand side of (30) is nonempty.

$$\hat{u}^j_\cdot \in \Bigl\{ \arg\inf J_j\bigl[\bigl(u^{\neg j}\bigr)\bigr] \Bigm| \hat{u}_\cdot \text{ satisfies Eq. (30) and}$$
$$u^{\neg j}_\cdot = \bigl(\hat{u}^1_\cdot, \ldots, \hat{u}^{j-1}_\cdot, u^j_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot\bigr) \in \prod\nolimits_{i=1}^n \mathcal{U}^i_{[t,T]},$$
$$\text{with restriction to } \Sigma_{[0,T]} \Bigr\}. \tag{31}$$

Furthermore, the accumulated risk-costs J_j, for $j = 1, 2, \ldots, n$, over the time-interval $[0, T]$ are given

$$J_j\bigl[\bigl(u^{\neg j}\bigr)\bigr] = \int_0^T c_j\bigl(s, X_s^{0,x;u_\cdot^{\neg j}}, u_s^j\bigr) ds + \Psi_j\bigl(X_T^{0,x;u_\cdot^{\neg j}}\bigr),$$
$$X_0^{0,x;u_\cdot^{\neg j}} = x, \quad \text{and} \quad \Psi_j\bigl(X_T^{0,x;u_\cdot^{\neg j}}\bigr) = \xi_j^{Target}. \tag{32}$$

In the following section, we establish the existence of optimal risk-averse solutions, in the sense of viscosity, for the risk-averse decision problem in (31) with restriction to $\Sigma_{[0,T]}$.

3 Main Results

In this section, we present our main results, where we introduce a framework that requires a "rational" cooperation among the risk-averse agents so as to achieve an overall optimal risk-averseness (in the sense of Pareto optimality). Moreover, such a framework allows us to establish the existence of optimal risk-averse solutions, in the sense of viscosity solutions, to the associated risk-averse dynamic programming equations.

Proposition 1 *Suppose that the generator functional g satisfies Assumption 1. Further, let the statements in (4) along with (10) hold true. Then, for any $(t, x) \in [0, T] \times \mathbb{R}^d$ and for every $u^{\neg j}_\cdot = (\hat{u}^1_\cdot, \ldots, \hat{u}^{j-1}_\cdot, u^j_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod_{i=1}^n \mathcal{U}^i_{[t,T]}$ and $j \in \{1, 2, \ldots, n\}$, restricted to $\Sigma_{[t,T]}$, the FBSDEs in (9) and (15) admit unique adapted solutions*

$$\left.\begin{array}{r} X_\cdot^{t,x;u_\cdot^{\neg j}} \in \mathcal{S}^2\bigl(t, T; \mathbb{R}^d\bigr) \\ \bigl(Y_\cdot^{j,t,x;u_\cdot^{\neg j}}, Z_\cdot^{j,t,x;u_\cdot^{\neg j}}\bigr) \in \mathcal{S}^2\bigl(t, T; \mathbb{R}\bigr) \times \mathcal{H}^2\bigl(t, T; \mathbb{R}^d\bigr),\ j = 1, 2, \ldots, n \end{array}\right\}. \tag{33}$$

Moreover, the risk-values $V_j^{u^j}\bigl(t, x\bigr)$, for $j = 1, 2, \ldots, n$, are deterministic.

Proposition 2 *Let $(t, x) \in [0, T] \times \mathbb{R}^d$ and $u^{\neg j}_\cdot = (\hat{u}^1_\cdot, \ldots, \hat{u}^{j-1}_\cdot, u^j_\cdot, \hat{u}^{j+1}_\cdot, \cdots, \hat{u}^n_\cdot) \in \prod_{i=1}^n \mathcal{U}^i_{[t,T]}$, for $j \in \{1, 2, \ldots, n\}$, be restricted to $\Sigma_{[t,T]}$. Then, for any $r \in [t, T]$ and $\mathbb{R}^d$-valued $\mathcal{F}_r^t$-measurable random variable η, we have*

$$\begin{aligned} V_j^{u^j_\cdot}\big(r,\eta\big) &= Y_r^{j,t,x;u_\cdot^{\neg j}} \\ &\triangleq \rho_{r,T}^{g_j}\Big[\int_r^T c_j\big(s, X_s^{r,\eta;u_\cdot^{\neg j}}, u_s^j\big)ds + \Psi_j(X_T^{r,\eta;u_\cdot^{\neg j}})\Big], \quad \mathbb{P}\text{-}a.s. \end{aligned} \tag{34}$$

Proposition 3 *Let* $u_\cdot^{\neg j} = (\hat{u}_\cdot^1, \ldots, \hat{u}_\cdot^{j-1}, u_\cdot^j, \hat{u}_\cdot^{j+1}, \cdots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$, *for* $j \in \{1, 2, \ldots, n\}$, *be restricted to* $\Sigma_{[t,T]}$. *Suppose that the system of semilinear parabolic PDEs in* (20) *enjoys the viability property w.r.t. the closed convex set K. Then, there exists a constant* $C > 0$ *such that* $d_K^2(\cdot)$ *is twice differentiable at y and*

$$\begin{aligned} &\big\langle y - \Pi_K(y),\ \mathbf{G}(t,x,y,z\sigma(t,x,u^{\neg j}))\big\rangle \\ &\qquad \le \frac{1}{4}\big\langle D^2(d_K^2(y))z \cdot \mathbf{\Sigma}(t,x,u),\ z \cdot \mathbf{\Sigma}(t,x,u)\big\rangle + Cd_K^2(y), \\ &\qquad\qquad \forall (t,x,y,z) \in [0,T] \times \mathbb{R}^d \times \mathbb{R}^n \times \mathcal{L}(\mathbb{R}^d;\mathbb{R}^n), \end{aligned} \tag{35}$$

where

$$\mathbf{\Sigma}(t,x,u) = \big(\sigma(t,x,u^{\neg 1}), \sigma(t,x,u^{\neg 2}), \cdots, \sigma(t,x,u^{\neg n})\big).$$

Remark 5 The proof for the above proposition (which is an adaptation of [6] and it will appear elsewhere) involves a standard approximation procedure for the BSDE in Eq. (15), with

$$\begin{aligned} &\mathbf{Z}_s^{t,x;u_\cdot} \in \operatorname{span}\Big\{z \cdot \mathbf{\Sigma}(t, \mathbf{X}_s^{t,x;u_\cdot}, u) \,\big|\, z \in \mathcal{L}(\mathbb{R}^d;\mathbb{R}^n)\Big\}, \\ &\qquad\qquad ds \otimes d\mathbb{P} - a.e. \text{ on } [t,T],\ 0 \le t \le T, \end{aligned}$$

and a further requirement for the closed convex set K to enjoy the BSVP for the equation in (15) (i.e., the adapted solution $\big\{\mathbf{Y}_s^{t,x;u},\ s \in [0,T]\big\}$ to be viable in K). Here, we also require that the set on the right-hand side of Eq. (30) is nonempty.

In the following, suppose that Proposition 3 holds true, i.e., the system of semilinear parabolic PDEs in (20) enjoys viability property w.r.t. the closed convex set K. Moreover, for $t \in [0,T]$ and $\hat{u}_\cdot = (\hat{u}_\cdot^1, \hat{u}_\cdot^2, \cdots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[t,T]}^i$, with restriction to $\Sigma_{[t,T]}$, let $\tilde{u}_\cdot^{\neg j}$ and $u_\cdot^{\neg j}, j \in \{1, 2, \ldots, n\}$, be two n-tuple decisions from $\prod_{i=1}^n \mathcal{U}_{[t,T]}^i$, i.e.,

$$\left.\begin{aligned} \tilde{u}_\cdot^{\neg j} &= (\hat{u}_\cdot^1, \ldots, \hat{u}_\cdot^{j-1}, \tilde{u}_\cdot^j, \hat{u}_\cdot^{j+1}, \cdots, \hat{u}_\cdot^n) \in \textstyle\prod_{i=1}^n \mathcal{U}_{[t,T]}^i \\ u_\cdot^{\neg j} &= (\hat{u}_\cdot^1, \ldots, \hat{u}_\cdot^{j-1}, u_\cdot^j, \hat{u}_\cdot^{j+1}, \cdots, \hat{u}_\cdot^n) \in \textstyle\prod_{i=1}^n \mathcal{U}_{[t,T]}^i \\ &\qquad\qquad\qquad\qquad j \in \{1, 2, \ldots, n\} \end{aligned}\right\}.$$

Then, for any $(t,x) \in [0,T] \times \mathbb{R}^d$, with restriction to $\Sigma_{[t,T]}$, we can define the following partial ordering on K by

$$\begin{aligned}&\bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(\tilde{u}^{\neg 1}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(\tilde{u}^{\neg 2}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(\tilde{u}^{\neg n}\bigr)\bigr]\bigr)\\ &\qquad \prec \bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(u^{\neg 1}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(u^{\neg 2}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(u^{\neg n}\bigr)\bigr]\bigr), \qquad (36)\end{aligned}$$

if $\rho_{t,T}^{g_j}\bigl[\xi_{t,T}^j\bigl(\tilde{u}^{\neg j}\bigr)\bigr] \le \rho_{t,T}^{g_j}\bigl[\xi_{t,T}^j\bigl(u^{\neg j}\bigr)\bigr]$ for all $j = 1, 2, \ldots, n$, with strict inequality for at least one $j \in \{1, 2, \ldots, n\}$. Furthermore, we say that

$$\bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(\hat{u}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(\hat{u}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(\hat{u}\bigr)\bigr]\bigr) \in K \qquad (37)$$

is a Pareto equilibrium, in the sense of viscosity solutions, if there is no

$$\bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(u^{\neg 1}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(u^{\neg 2}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(u^{\neg n}\bigr)\bigr]\bigr) \in K \qquad (38)$$

for which

$$\begin{aligned}&\bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(\hat{u}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(\hat{u}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(\hat{u}\bigr)\bigr]\bigr)\\ &\qquad \prec \bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(u^{\neg 1}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(u^{\neg 2}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(u^{\neg n}\bigr)\bigr]\bigr). \qquad (39)\end{aligned}$$

Hence, with restriction to $\Sigma_{[t,T]}$, we can characterize the optimal decisions for the risk-averse agents as follows.

Proposition 4 *Suppose that Proposition 3 holds true and let* $\varphi \in C_b^{1,2}([0,T] \times \mathbb{R}^d; \mathbb{R}^n)$ *satisfy* (20) *with* $\varphi\bigl(T,x\bigr) = \boldsymbol{\Psi}(x)$ *for* $x \in \mathbb{R}^d$. *Then,* $\varphi_j\bigl(t,x\bigr) \le V_j^{u_\cdot^j}\bigl(t,x\bigr)$ *for* $u_\cdot^{\neg j} = (\hat{u}_\cdot^1, \ldots, \hat{u}_\cdot^{j-1}, u_\cdot^j, \hat{u}_\cdot^{j+1}, \cdots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$, *for* $j \in \{1, 2, \ldots, n\}$, *with restriction to* $\Sigma_{[t,T]}$, *and for all* $(t,x) \in [0,T] \times \mathbb{R}^d$. *Further, if an admissible optimal decision process* $\hat{u}_\cdot^j \in \mathcal{U}_{[t,T]}^j$ *exists, for almost all* $(s,\Omega) \in [t,T] \times \Omega$, *together with the corresponding solution* $X_s^{t,x;\hat{u}_\cdot}$, *and satisfies*

$$\begin{aligned}\hat{u}_s^j \in \operatorname*{arg\,inf}_{u_\cdot^j \in \mathcal{U}_{[t,T]}^j \big| \Sigma_{[t,T]}} &\Bigl\{\mathcal{L}_s^{u^{\neg j}} \varphi_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr)\\ &+ g_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}, \varphi\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr), D_x\varphi_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr) \cdot \sigma\bigl(s, X_s^{t,x;w}, u_s^{\neg j}\bigr)\bigr)\Bigr\}. \qquad (40)\end{aligned}$$

Then, $\varphi_j\bigl(t,x\bigr) = V_j^{\hat{u}_\cdot^j}\bigl(t,x\bigr)$ *for* $j \in \{1, 2, \ldots, n\}$ *and for all* $(t,x) \in [0,T] \times \mathbb{R}^d$. *Moreover, corresponding to the n-tuple of optimal risk-averse decisions* $\hat{u}_\cdot \in \prod_{i=1}^n \mathcal{U}_{[t,T]}^i$, *with restriction to* $\Sigma_{[t,T]}$, *there exists a Pareto equilibrium*

$$\bigl(\rho_{t,T}^{g_1}\bigl[\xi_{t,T}^1\bigl(\hat{u}\bigr)\bigr],\, \rho_{t,T}^{g_2}\bigl[\xi_{t,T}^2\bigl(\hat{u}\bigr)\bigr], \cdots, \rho_{t,T}^{g_n}\bigl[\xi_{t,T}^n\bigl(\hat{u}\bigr)\bigr]\bigr) \in K \qquad (41)$$

such that

$$\begin{aligned}&\big(\rho_{t,T}^{g_1}\big[\xi_{t,T}^{1}(\hat{u})\big],\, \rho_{t,T}^{g_2}\big[\xi_{t,T}^{2}(\hat{u})\big],\, \cdots,\, \rho_{t,T}^{g_n}\big[\xi_{t,T}^{n}(\hat{u})\big]\big)\\ &\qquad \prec \big(\rho_{t,T}^{g_1}\big[\xi_{t,T}^{1}(u^{\neg 1})\big],\, \rho_{t,T}^{g_2}\big[\xi_{t,T}^{2}(u^{\neg 2})\big],\, \cdots,\, \rho_{t,T}^{g_n}\big[\xi_{t,T}^{n}(u^{\neg n})\big]\big) \text{ on } K, \qquad (42)\end{aligned}$$

for all $t \in [0, T]$ *and for* $\{\xi_j^{Target}\}_{j=1}^n$ *from* $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$.

4 Further Remarks

In this section, we briefly comment on our problem formulation (i.e., the risk-averse decision problem of Sect. 3)—in which results from the dynamic risk analysis implicitly constitute part of the information used in the context of the risk-averse criteria—that requires each of the risk-averse agents to respond optimally, in the sense of *best-response correspondence*, to the decisions of the other risk-averse agents.

Notice that the notion of Pareto equilibria (w.r.t. the optimal risk sharing) in Eqs. (37)–(39) and that of optimal preference decisions in Eq. (40) are all well-defined concepts in the context of risk-aversion of (31), with accumulated risk-costs of (3). Here, we remark that, for every $\{\xi_j^{Target}\}_{j=1}^n$ from $L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$ and for all $t \in [0, T]$, if there exists an n-tuple of optimal risk-averse decisions, i.e., $\hat{u}_\cdot \in \prod_{i=1}^n \mathcal{U}_{[t,T]}^i$, such that, for any $(t, x) \in [0, T] \times \mathbb{R}^d$, the FBSDEs in (9) and (15) (cf. Eqs. (13) and (14)) admit a unique solution set

$$\{X_\cdot^{t,x;u_\cdot^{\neg j}}, Y_\cdot^{j,t,x;u_\cdot^{\neg j}}, Z_\cdot^{j,t,x;u_\cdot^{\neg j}}\}_{j=1}^n \quad \text{on} \quad (\Omega, \mathcal{F}, \mathcal{F}^t, \mathbb{P})$$

and

$$\begin{aligned}\mathbf{Y}_\cdot^{s,x;u_\cdot}(\omega) = \big(Y_\cdot^{1,s,x;u_\cdot^{\neg 1}}(\omega),\, Y_\cdot^{2,s,x;u_\cdot^{\neg 2}}(\omega),\, \cdots,\, Y_\cdot^{n,s,x;u_\cdot^{\neg n}}(\omega)\big) \in K,\\ \mathbb{P}- \textit{ almost } \omega \in \Omega, \quad \forall s \in [t, T].\end{aligned}$$

Then, verifying the above two conditions amounted to solving the stochastic target problem, which characterizes the set of all acceptable risk-exposures, when $t = 0$, vis-á-vis some uncertain future costs or outcomes specified by a set of random variables $\{\xi_j^{Target}\}_{j=1}^n$ from $L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$.

On other hand, assume that the exact information about $\xi_j^{Target} \in L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$, for $j = 1, 2, \ldots, n$, are not known, but we know that such information can be obtained from the following allocation

$$R = \sum\nolimits_{j=1}^n \alpha_j \xi_j^{Target},$$

where $R \in L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R})$ is assumed to be a-priorly known; and $\sum_{j=1}^n \alpha_j = 1$, for some $\alpha_j \geq 0, j = 1, 2, \ldots, n$. Furthermore, if there exists an n-tuple of optimal decisions, i.e., $\hat{u}_\cdot \in \prod_{i=1}^n \mathcal{U}_{[t,T]}^i$, for the risk-averse agents, then we can introduce the set of optimally allocated risk-exposures as follows:

$$\mathcal{A}_0(R) = \Bigg\{ \big(\rho_{0,T}^{g_1}\big[\xi_{0,T}^1(\hat{u})\big],\, \rho_{0,T}^{g_2}\big[\xi_{0,T}^2(\hat{u})\big],\, \cdots,\, \rho_{0,T}^{g_n}\big[\xi_{0,T}^n(\hat{u})\big] \big) \in K \,\Bigg|\, R = \sum\nolimits_{j=1}^n \alpha_j \xi_j^{Target} \in L^2(\Omega, \mathcal{F}_T, \mathbb{P}; \mathbb{R}) \text{ and } \sum\nolimits_{j=1}^n \alpha_j = 1, \text{ with } \alpha_j \geq 0,\ j = 1, 2, \ldots, n \Bigg\},$$

which provides further useful information to characterize all Pareto equilibria (of optimal risk allocations) w.r.t. the risk-averse agents.

Appendix: Proofs

In this section, we give the proofs for the main results.

Proof of Proposition 1 Notice that m and σ are bounded and Lipschitz continuous w.r.t. $(t, x) \in [0, T] \times \mathbb{R}^d$ and uniformly for $u \in \prod_{i=1}^n U^i$. Then, for any $(t, x) \in [0, T] \times \mathbb{R}^d$ and $u_\cdot^{\neg j}$, for $j = 1, 2, \ldots, n$, are progressively measurable processes, there always exists a unique path-wise solution $X_\cdot^{t,x;u_\cdot^{\neg j}} \in \mathcal{S}^2\big(t, T; \mathbb{R}^d\big)$ for the forward SDE in (9). On the other hand, consider the following BSDEs,

$$-d\hat{Y}_s^{j,t,x;u_\cdot^{\neg j}} = g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, \hat{Y}_s^{j,t,x;u_\cdot^{\neg j}}, Z_s^{j,t,x;u_\cdot^{\neg j}}\big)ds - Z_s^{j,t,x;u_\cdot^{\neg j}} dB_s, \quad j = 1, 2, \ldots, n, \tag{43}$$

where

$$\hat{Y}_T^{j;t,x;u_\cdot^{\neg j}} = \int_t^T c_j\big(\tau, X_\tau^{t,x;u_\cdot^{\neg j}}, u_\tau^j\big)d\tau + \Psi_j(X_T^{t,x;u_\cdot^{\neg j}}).$$

From Lemma 2, Eq. (43) admits unique solutions $\big(\bar{Y}_\cdot^{j,t,x;u_\cdot^{\neg j}}, Z_\cdot^{j,t,x;u_\cdot^{\neg j}}\big)$, for $j = 1, 2, \ldots, n$, in $\mathcal{S}^2\big(t, T; \mathbb{R}\big) \times \mathcal{H}^2\big(t, T; \mathbb{R}^d\big)$. Furthermore, if we introduce the following

$$Y_s^{j,t,x;u_\cdot^{\neg j}} = \hat{Y}_s^{j,t,x;u_\cdot^{\neg j}} - \int_t^s c_j\big(\tau, X_\tau^{t,x;u_\cdot^{\neg j}}, u_\tau^j\big)d\tau, \quad s \in [t, T].$$

Then, the family of forward of the BSDEs in (14) holds with $\bigl(Y_\cdot^{j,t,x;u_\cdot^{\neg j}}, Z_\cdot^{j,t,x;u_\cdot^{\neg j}}\bigr)$, for $j = 1, 2, \ldots, n$. Moreover, we also observe that $Y_t^{j,t,x;u_\cdot^{\neg j}}$, for $j = 1, 2, \ldots, n$, are deterministic. This completes the proof of Proposition 1. □

Proof of Proposition 2 For any $r \in [t, T]$, with $t \in [0, T]$, we consider the following probability space $\bigl(\Omega, \mathcal{F}, \mathbb{P}\bigl(\cdot\,|\mathcal{F}_r^t\bigr), \{\mathcal{F}^t\}\bigr)$ and notice that η is deterministic under this probability space. Then, for any $s \ge r$, there exist progressively measurable process ψ such that

$$\begin{aligned} u_s^j(\Omega) &= \psi(\Omega, B_{\cdot\wedge s}(\Omega)),\\ &= \psi(s, \bar{B}_{\cdot\wedge s}(\Omega) + B_r(\Omega)), \end{aligned} \tag{44}$$

where $\bar{B}_s = B_s - B_r$ is a standard d-dimensional Brownian motion. Note that n tuple $u_\cdot^{\neg j}$, for $j = 1, 2, \ldots, n$, are $\mathcal{F}_r^t$-adapted processes, then we have the following restriction w.r.t. $\Sigma_{[t,T]}$

$$\bigl(\Omega, \mathcal{F}, \{\mathcal{F}^t\}, \mathbb{P}\bigl(\cdot\,|\mathcal{F}_r^t\bigr)(\omega'), B_\cdot, u_\cdot^{\neg j}\bigr) \in \Sigma_{[t,T]}, \quad j = 1, 2, \ldots, n, \tag{45}$$

where $\omega' \in \Omega'$ such that $\Omega' \in \mathcal{F}$, with $\mathbb{P}(\Omega') = 1$. Furthermore, noting Lemma 2, if we work under the probability space $\bigl(\Omega', \mathcal{F}, \mathbb{P}\bigl(\cdot\,|\mathcal{F}_r^t\bigr)\bigr)$, then the statement in (34) holds $\mathbb{P}$- *almost surely*. This completes the proof of Proposition 2. □

Proof of Proposition 4 Suppose there exists an n-tuple of optimal risk-averse decisions $(\hat{u}_\cdot^1, \hat{u}_\cdot^2, \ldots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$ satisfying the statements in Definition 3. Assume that $(t, x) \in [0, T] \times \mathbb{R}^d$ is fixed. For any $u_\cdot^j \in \mathcal{U}_{[t,T]}^j$, restricted to $\Sigma_{[t,T]}$, for $j \in \{1, 2, \ldots, n\}$, we consider an $\mathbb{R}^n$-valued process $\varphi\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr)$, with

$$u_\cdot^{\neg j} = (\,\hat{u}_\cdot^1,\, \ldots,\, \hat{u}_\cdot^{j-1},\, u_\cdot^j,\, \hat{u}_\cdot^{j+1},\, \cdots,\, \hat{u}_\cdot^n\,) \in \prod\nolimits_{i=1}^n \mathcal{U}_{[t,T]}^i,$$

which is restricted to $\Sigma_{[t,T]}$. Then, using Itô integral formula, we can evaluate the difference between $\varphi_j\bigl(T, X_T^{t,x;u_\cdot^{\neg j}}\bigr)$ and $\varphi_j\bigl(t, x\bigr)$, for $j = 1, 2, \ldots, n$, as follows:[4]

$$\begin{aligned} \varphi_j\bigl(T, X_T^{t,x;u_\cdot^{\neg j}}\bigr) - \varphi_j\bigl(t, x\bigr) &= \int_t^T \Bigl[\frac{\partial}{\partial t}\varphi_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr) + \mathcal{L}_t^{u_\cdot^{\neg j}} \varphi_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr)\Bigr] ds\\ &\quad + \int_t^T D_x\varphi_j\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}\bigr) \cdot \sigma\bigl(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^{\neg j}\bigr) dB_s. \end{aligned} \tag{46}$$

Using (20), we further obtain the following

[4] Notice that $\varphi\bigl(t, x\bigr) \in C_b^{1,2}([0, T] \times \mathbb{R}^d; \mathbb{R}^n)$.

$$\begin{aligned}&\frac{\partial}{\partial t}\varphi_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big) + \mathcal{L}_t^{u^{\neg j}}\varphi_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big)\\ &\quad + g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, \varphi\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big), D_x\varphi_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big)\cdot\sigma\big(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^{\neg j}\big)\big) \ge 0,\\ &\qquad\qquad j = 1, 2, \ldots, n.\end{aligned} \tag{47}$$

Furthermore, if we combine (46) and (47), then we obtain

$$\begin{aligned}\varphi_j\big(t,x\big) &\le \Psi_j\big(X_T^{t,x;u_\cdot^{\neg j}}\big)\\ &\quad + \int_t^T g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, \varphi(s, X_s^{t,x;u_\cdot^{\neg j}}), D_x\varphi_j(s, X_s^{t,x;u_\cdot^{\neg j}})\cdot\sigma(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^{\neg j})\big)ds\\ &\quad - \int_t^T D_x\varphi_j(s, X_s^{t,x;u_\cdot^{\neg j}})\cdot\sigma(s, X_s^{t,x;u_\cdot^{\neg j}}, u_s^{\neg j})dB_s.\end{aligned} \tag{48}$$

Define $Z_s^{j,t,x;u_\cdot^{\neg j}} = D_x\varphi_j(s, X_s^{t,x;u_\cdot^{\neg j}})\cdot\sigma(s, X_s^{t,x;u_\cdot^{\neg j}}, (\hat{u}_s, v_s))$, for $s \in [t,T]$ and for $j = 1, 2, \ldots, n$, then $\varphi_j\big(t,x\big) \le Y_t^{j,t,x;u_\cdot^{\neg j}}$ follows, where $(Y_\cdot^{j,t,x;u_\cdot^{\neg j}}, Z_\cdot^{j,t,x;u_\cdot^{\neg j}})$ is a solution to BSDE in (14) (cf. Eq. (13)). As a result of this, we have

$$\varphi_j\big(t,x\big) \le V_j^{u^j}\big(t,x\big), \quad j = 1, 2, \ldots, n.$$

Moreover, if there exists at least one $\hat{u}^j$ satisfying (40), i.e., if $\hat{u}^j$ is a measurable selector of

$$\begin{aligned}&\arg\max\Big\{\mathcal{L}_s^{u^{\neg j}}\varphi_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big)\\ &\quad + g_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}, \varphi\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big), D_x\varphi_j\big(s, X_s^{t,x;u_\cdot^{\neg j}}\big)\cdot\sigma\big(s, X_s^{t,x;w}, u_s^{\neg j}\big)\big)\Big\},\\ &\qquad\qquad j \in \{1, 2, \ldots, n\}.\end{aligned}$$

Then, for $u_\cdot^j = \hat{u}_\cdot^j$, for $j \in \{1, 2, \ldots, n\}$, the inequality in (48) becomes an equality, i.e.,

$$\begin{aligned}\varphi_j(t,x) &= V_j^{\hat{u}^j}\big(t,x\big)\\ &\triangleq Y_T^{j,t,x;\hat{u}_\cdot^{\neg j}}, \; j \in \{1, 2, \ldots, n\}, \; \text{(cf. Eqs. (11) and (13))}.\end{aligned}$$

Note that the corresponding path-wise solution $X_s^{t,x;\hat{u}_\cdot}$ is progressively measurable, since $\hat{u}_\cdot \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$ is also restricted to $\Sigma_{[t,T]}$.

On the other hand, noting the relations in (11) and (13), for any $(t,x) \in [0,T] \times \mathbb{R}^n$, with restriction to $\Sigma_{[t,T]}$, define the following utility function over the closed convex set K

$$K \ni \left(\rho_{t,T}^{g_1}\left[\xi_{t,T}^1\left(u^{\neg 1}\right)\right],\, \rho_{t,T}^{g_2}\left[\xi_{t,T}^2\left(u^{\neg 2}\right)\right],\, \cdots,\, \rho_{t,T}^{g_n}\left[\xi_{t,T}^n\left(u^{\neg n}\right)\right]\right) \\ \to \mathcal{J}(u) = \sum\nolimits_{i=1}^{n} \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(u^{\neg i}\right)\right],$$

where $\pi^i > 0$, for $i = 1, 2, \ldots, n$.

Note that the utility function $\mathcal{J}(u) = \sum_i^n \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(u^{\neg i}\right)\right]$ satisfies the following property

$$\sum\nolimits_{i=1}^{n} \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(\tilde{u}^{\neg i}\right)\right] < \sum\nolimits_{i=1}^{n} \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(u^{\neg i}\right)\right],$$

i.e., $\mathcal{J}(\tilde{u}) < \mathcal{J}(u)$, whenever

$$\left(\rho_{t,T}^{g_1}\left[\xi_{t,T}^1\left(\tilde{u}^{\neg 1}\right)\right],\, \rho_{t,T}^{g_2}\left[\xi_{t,T}^2\left(\tilde{u}^{\neg 2}\right)\right],\, \cdots,\, \rho_{t,T}^{g_n}\left[\xi_{t,T}^n\left(\tilde{u}^{\neg n}\right)\right]\right) \\ \prec \left(\rho_{t,T}^{g_1}\left[\xi_{t,T}^1\left(u^{\neg 1}\right)\right],\, \rho_{t,T}^{g_2}\left[\xi_{t,T}^2\left(u^{\neg 2}\right)\right],\, \cdots,\, \rho_{t,T}^{g_n}\left[\xi_{t,T}^n\left(u^{\neg n}\right)\right]\right),$$

w.r.t. the class of admissible control processes $\prod_{i=1}^n \mathcal{U}_{[t,T]}^i$ (cf. Eqs. (36)–(39)). Then, from the Arrow–Barankin–Blackwell theorem (e.g., see [3]), for all $t \in [0, T]$, one can see that the set in

$$\Bigg\{\left(\rho_{t,T}^{g_1}\left[\xi_{t,T}^1\left(u^{\neg 1}\right)\right],\, \rho_{t,T}^{g_2}\left[\xi_{t,T}^2\left(u^{\neg 2}\right)\right],\, \cdots,\, \rho_{t,T}^{g_n}\left[\xi_{t,T}^n\left(u^{\neg n}\right)\right]\right) \in K \,\Bigg|\, \\ \exists\, \pi^i > 0,\; i = 1, 2, \ldots, n,\; \min \sum\nolimits_{i=1}^{n} \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(u^{\neg i}\right)\right] = \sum\nolimits_{i=1}^{n} \pi^i \rho_{t,T}^{g_i}\left[\xi_{t,T}^i\left(\hat{u}^{\neg i}\right)\right]\Bigg\}$$

is dense in the set of all Pareto equilibria. This further implies that, for any choice of $\pi^i > 0$, $i = 1, 2, \ldots, n$, the minimizer $\mathcal{J}(\hat{u}) = \sum_i^n \pi^i \rho_{0,T}^{g_i}\left[\xi_{0,T}^i\left(\hat{u}^{\neg i}\right)\right]$ over K satisfies the Pareto equilibrium condition w.r.t. some n-tuple of optimal risk-averse decisions $(\hat{u}_\cdot^1, \hat{u}_\cdot^2, \ldots, \hat{u}_\cdot^n) \in \prod_{i=1}^n \mathcal{U}_{[0,T]}^i$. This completes the proof of Proposition 4. □

Acknowledgements This research was supported in part by the Air Force Research Laboratory (AFRL).

References

1. Alexandrov, A.D.: The existence almost everywhere of the second differential of a convex function and some associated properties of convex surfaces. Ucenye Zapiski Leningrad. Gos. Univ. Ser. Math. **37**, 3–35 (1939, in Russian)
2. Antonelli, F.: Backward-forward stochastic differential equation. Ann. Appl. Probab. **3**, 777–793 (1993)

3. Arrow, K.J., Barankin, E.W., Blackwell, D.: Admissible points of convex sets. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games, vol. II, pp. 87–91. Princeton, NJ (1953)
4. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Math. Finance **9**, 203–228 (1999)
5. Befekadu, G.K., Veremyev, A., Pasiliao, E.L.: On the hierarchical risk-averse control problems for diffusion processes. Preprint arXiv:1603.03359 [math.OC], 20 pages, March 2016
6. Buckdahn, R., Quincampoix, M., Rascanu, A.: Viability property for a backward stochastic differential equation and applications to partial differential equations. Probab. Theory Relat. Fields **116**, 485–504 (2000)
7. Coquet, F. Hu, Y., Mémin, J., Peng, S.: Filtration-consistent nonlinear expectations and related g-expectations. Probab. Theory Relat. Fields **123**, 1–27 (2002)
8. Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Am. Math. Soc. **27**, 1–67 (1992)
9. Detlefsen, K., Scandolo, G.: Conditional and dynamic convex risk measures. Finance Stochast. **9**, 539–561 (2005)
10. El-Karoui, N., Peng, S., Quenez, M.C.: Backward stochastic differential equations in finance. Math. Finance **7**, 1–71 (1997)
11. Fleming, W.H., Soner, H.M.: Controlled Markov Processes and Viscosity Solutions. Springer, New York (2006)
12. Föllmer, H., Schied, A.: Convex measures of risk and trading constraints. Finance Stochast. **6**, 429–447 (2002)
13. Hu, Y., Peng, S.: Solutions of forward-backward stochastic differential equations. Probab. Theory Relat. Fields **103**, 273–283 (1995)
14. Krylov, N.V.: Controlled Diffusion Process. Springer, Berlin (2008)
15. Li, J., Wei, Q.: Optimal control problems of fully coupled FBSDEs and viscosity solutions of Hamilton-Jacobi-Bellman equations. SIAM J. Control Optim. **52**, 1622–1662 (2014)
16. Pardoux, E., Peng, S.: Adapted solutions of backward stochastic differential equation. Syst. Control Lett. **14**, 55–61 (1990)
17. Pardoux, E., Tang, S.J.: Forward-backward stochastic differential equations and quasilinear parabolic PDEs. Probab. Theory Relat. Fields **114**, 123–150 (1999)
18. Peng, S.: Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. Stoch. Stoch. Rep. **37**, 61–67 (1991)
19. Peng, S.: A generalized dynamic programming principle and Hamilton-Jacobi-Bellman equation. Stoch. Stoch. Rep. **38**, 119–134 (1992)
20. Peng, S.: Nonlinear Expectations, Nonlinear Evaluations and Risk Measures. Lecture Notes in Mathematics. Springer, Berlin (2004)
21. Protter, P.: Stochastic Integration and Stochastic Differential Equations: A New Approach. Springer, Berlin, Germany (1990)
22. Rosazza Gianin, E.: Risk measures via g-expectations. Insur. Math. Econ. **39**, 19–34 (2006)
23. Ruszczyński, A. : Risk-averse dynamic programming for Markov decision process. Math. Program. **125**, 235–261 (2010)
24. Stadje, M.: Extending dynamic convex risk measures from discrete time to continuous time: a convergence approach. Insur. Math. Econ. **47**, 391–404 (2010)

Optimal Packing of General Ellipses in a Circle

Frank J. Kampas, János D. Pintér, and Ignacio Castillo

Abstract Our objective is to find the optimal non-overlapping packing of a collection of general (non-identical) ellipses, with respect to a container circle that has minimal radius. Following the review of selected topical literature, we introduce a model development approach based on using embedded Lagrange multipliers. Our optimization model has been implemented using the computing system *Mathematica.* We present illustrative numerical results using the LGO nonlinear (global and local) optimization software package linked to *Mathematica*. Our study shows that the Lagrangian modeling approach combined with nonlinear optimization tools can effectively handle challenging ellipse packing problems with hundreds of decision variables and non-convex constraints.

Keywords General ellipse packings in circles • Model development using embedded lagrange multipliers • Global-Local optimization • LGO solver suite • Numerical results

1 Introduction and Review

1.1 Circle Packings

A general circle packing is an optimized non-overlapping arrangement of *n* arbitrary size circles inside a container such as a circle, square, or a general rectangle. The quality of the packing can be measured by the area of the container. The circle

F.J. Kampas
Physicist at Large Consulting LLC, Bryn Mawr, PA, USA

J.D. Pintér
Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Pintér Consulting Services Inc., Halifax, NS, Canada

I. Castillo (✉)
Lazaridis School of Business and Economics, Wilfrid Laurier University, Waterloo, ON, Canada
e-mail: icastillo@wlu.ca

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_2

packing problem—in particular, the case of packing identical circles—has received considerable attention. Due to the very special, inherently symmetric structure of this problem-type, studies dealing with identical circle packings often aim to prove the optimality of the configurations found, either theoretically or with the help of rigorous computational approaches: consult, e.g., Szabó et al. [1–3], Markót [4] with numerous related references.

The packing of general circle collections is a significant generalization of the uniform case, since now each packed circle can have a different (in principle, arbitrary) radius. Generally speaking, provably optimal configurations can be found only for very small model instances ($n \leq 4$). Therefore, studies dealing with general circle packings typically introduce and apply (generic or specifically tailored) global scope solution strategies, without the proven optimality of the results obtained. For details, cf., e.g., Riskin et al. [5], Castillo and Sim [6], Pintér and Kampas [7, 8], Kampas and Pintér [9], Addis et al. [10], Castillo et al. [11], Grosso et al. [12].

For reviews of uniform and general circle packing problems and some applications, we refer to Castillo et al. [11], Hifi and M'Hallah [13]. For more general object packing problems and a range of important real-world applications consult, e.g., [14, 15].

1.2 Ellipse Packings

Ellipse packing problems have received relatively little attention in the literature so far. Finding a high quality packing of ellipses that can have arbitrary size and orientation is a difficult computational problem. The key challenge is the modeling and enforcement of the no-overlap constraints, since the overlap between two ellipses depends on the orientation of the ellipses, in addition to the location of their centers.

Here we briefly review some related literature. Although not all works cited are aimed at handling the exact same problem-type addressed by our present study, they serve to illustrate the significant difficulty of ellipse (related) packing problems.

First, we mention an exact result that deals with the densest packing of just $n = 2$ non-overlapping congruent ellipses in a square. In this case, for all real numbers r in $[0, 1]$, Gensane and Honvault [16] analytically describe the densest packing of two ellipses with aspect ratio r.

Birgin et al. [17] study the problem of packing sets of identical circles within an optimized ellipse. The basic challenge here is the closed formula based calculation to compute the distance of an arbitrary point to the boundary of the containing ellipse. The authors note that—even when considering only identical size circles—the resulting models are hard nonlinear programming problems. In order to seek for globally optimized solutions, Birgin et al. propose stochastic multi-start and lattice-based search strategies.

Litvinchev et al. [18] find optimized packings of “circular-like” objects in a rectangular container. They propose a binary linear programming (BLP) model formulation based on a grid that approximates the container, and then consider the nodes of the grid as potential positions for assigning centers of the packed objects. The resulting BLP problem is solved using the commercial software package CPLEX. Numerical results related to packing circles, ellipses, rhombuses, and octagons are presented. Let us point out that, given the grid approximation of the container, this approach can only handle the packing of uniform sized and orthogonally oriented ellipses inside a container.

Galiev and Lisafina [19] study the problem of packing identical, orthogonally oriented ellipses inside a rectangular container. Similarly to Litvinchev et al. [18], BLP model formulations are proposed using a grid that approximates the container. Two special cases regarding the orientation of the ellipses are considered: (a) the major axes of all ellipses are parallel to the x or y axis, and (b) the major axes of some of the ellipses are parallel to the x axis and others to the y axis. A heuristic algorithm based on the BLP model is proposed, with illustrative numerical results.

Kallrath and Rebennack [20] address the problem of packing ellipses of arbitrary size and orientation into an optimized rectangle of minimal area. The packing model formulation is introduced as a cutting problem. The key idea of this work is to use separating lines to ensure that the ellipses do not overlap with each other. For problem-instances with $n \leq 14$ ellipses, the authors present feasible solutions that are globally optimal subject to the finite arithmetic precision of the global solvers at hand. However, these authors also report that for $n > 14$ ellipses none of the local or global nonlinear optimization solver engines available in conjunction with the GAMS modeling environment could compute even a feasible solution. Therefore, they propose heuristic approaches: the ellipses are added sequentially to an optimized rectangular container. This approach allows computing visually plausible, high-quality solutions for up to 100 ellipses.

Uhler and Wright [21] study the problem of packing arbitrary sized ellipsoids into an ellipsoidal container so as to minimize a measure of overlap between ellipsoids. A model formulation and two local scope solution approaches are discussed: one approach for the general case, and a simpler approach for the special case in which all ellipsoids are in fact spheres. The authors describe and illustrate their computational experience using chromosome organization in the human cell nucleus as the motivating application.

Based also on the references cited, we argue that ellipse packings have a number of important practical applications, with a view also towards the future use of such models.

Here we study the optimized non-overlapping packing of a set of ellipses with arbitrary size and orientation parameters inside a circular container. Hence, our objective is to minimize the radius of the container circle.

Packing ellipses into a circle requires (a) determination of the maximal distance from the center of the circular container to each ellipse boundary, and (b) finding the minimal distance between all pairs of the ellipses. The first requirement is necessary to determine the radius of the circumscribing circle that will be minimized. The second requirement is necessary to prevent the ellipses from overlapping. Explicit (closed form) analytical formulas for these quantities are expected to be cumbersome to use. Therefore, following Kampas et al. [22], our approach involves the determination of these quantities by embedding optimization calculations (using Lagrange multipliers) into the overall optimization strategy. In this Lagrangian setting, the optimization strategy to find the radius of the circumscribing circle and to prevent ellipse overlap proceeds simultaneously towards meeting both requirements. Our approach supports the numerical solution of the packing problem by a single call to a suitable global optimization procedure.

While—analogously to the significantly easier case of general circle packings studied by us and many other researchers—we cannot guarantee the theoretical (provable) optimality of the ellipse configurations found, our solution strategy leads to plausible and visibly high quality packings.

2 Model Formulation

The input information to the general ellipse packing problem considered here are the semi-major and semi-minor axes of the ellipses. The decision variables are the radius of the circumscribing circle, and the center position and orientation of each of the packed ellipses. Secondary (induced) variables are the positions of the points on the ellipses most distant from the center of the container circle, and the positions of the points on one of each pair of ellipses that minimizes the value of the equation describing the other ellipse. Other secondary variables are the embedded Lagrange multipliers used to determine those points. All secondary variables are all implicitly determined by the primary decision variables.

The model constraints belong to two groups. The first group uses the secondary variables to represent the constraints that keep the ellipses inside the circumscribing circle and prevent them from overlapping. The second group represents the equations generated by the embedded Lagrange multiplier conditions. In our global optimization strategy, the calculations to prevent ellipse overlaps proceed simultaneously with the minimization of the radius of the circumscribing circle, rather than performed to completion at each major iteration step towards the minimization of the radius.

In order to present our optimization model, we introduce its components. Equation $e(a, b, xc, yc, \theta; x, y)$ describes an ellipse with semi-major and semi-minor axes a and b, centered at $\{xc, yc\}$, and rotated counterclockwise by angle θ. To be more specific, function $e(a, b, xc, yc, \theta; x, y)$ is negative for all points (x, y) inside the ellipse, zero for all points on the ellipse boundary, and positive for all points outside the ellipse. Recall that a and b are given input parameters, while xc , yc and θ are decision variables for each ellipse i: the corresponding variables will be denoted by xc_i , yc_i and θ_i for $i = 1 , \ldots , n$. Equation $e(a, b, xc, yc, \theta; x, y)$ can be obtained by transforming the equation of a circle with radius 1, centered at (0, 0), as follows.

$$e\left(a, b, xc, yc, \theta; x, y\right) = \left(\frac{\cos\left(\theta\right)\ \left(x - xc\right)}{a} + \frac{\sin\left(\theta\right)\ \left(y - yc\right)}{a}\right)^2 + \left(\frac{\cos\left(\theta\right)\ \left(y - yc\right)}{b} - \frac{\sin\left(\theta\right)\ \left(x - xc\right)}{b}\right)^2 - 1 = 0 \quad (1)$$

Note that in Eq. (1) the coordinate system is rotated by an angle of –θ, which is equivalent to rotating the given ellipse by an angle θ around its center (xc_i, yc_i).

We assume that the container circle is centered at the origin: hence, its radius must be at least the maximum value of $\sqrt{x^2 + y^2}$ that can be obtained by considering all points (x, y) of the packed ellipses. The point on an ellipse with the maximal value of $x^2 + y^2$ is clearly the same as the point that maximizes $\sqrt{x^2 + y^2}$ for the ellipse. In order to determine this point, we introduce the notation $f(x, y) = e(a, b, xc, yc, \theta; x, y)$. Then the point in question can be determined using the Lagrange multiplier method by differentiating $x^2 + y^2 = \lambda \cdot f(x, y)$ with respect to x, y, and λ. Applying this method, we obtain the equations

$$\left\{\begin{array}{c} 2x = \lambda \cdot f_x\left(x, y\right) \\ 2y = \lambda \cdot f_y\left(x, y\right) \\ f\left(x, y\right) = 0 \end{array}\right\}. \quad (2)$$

In (2), $f_x(x, y)$ is the derivative of $f(x, y)$ with respect to x and $f_y(x, y)$ is the derivative of $f(x, y)$ with respect to y. The next equation follows simply from the requirement that the point sought lies on the ellipse boundary. Note that λ can be eliminated from the first two equations: hence, we obtain

$$y \cdot f_x\left(x, y\right) = x \cdot f_y\left(x, y\right). \quad (3)$$

Based on Eq. (3), the slope of the ellipse boundary at point (x, y) is given by

$$\frac{f_x\left(x, y\right)}{f_y\left(x, y\right)} = \frac{x}{y}. \quad (4)$$

Since the slope of the line from $(0, 0)$ to (x, y) is y/x, (4) gives the inverse of the slope of the ellipse at the point most distant from the origin. In other words—as one would expect—the line from the origin to the most distant point on the ellipse is orthogonal to the tangent of the ellipse at that point. It will be useful to set up and evaluate equations for the derivatives of the ellipse equation $e(x, y)$ with respect to x and y. These derivatives are given as follows:

$$\frac{de\,(x,y)}{dx}=\frac{d\,[e\,(a,b,xc,yc,\theta;x,y)]}{dx}$$
$$=\frac{2}{a^2b^2}\left(b^2\,(x-xc)\cos(\theta)^2-\left(a^2-b^2\right)(y-yc)\cos(\theta)\sin(\theta)+a^2\,(x-xc)\sin(\theta)^2\right); \tag{5}$$

$$\frac{de\,(x,y)}{dy}=\frac{d\,[e\,(a,b,xc,yc,\theta;x,y)]}{dy}$$
$$=\frac{2}{a^2b^2}\left(a^2\,(y-yc)\cos(\theta)^2-\left(a^2-b^2\right)(x-xc)\cos(\theta)\sin(\theta)+b^2\,(y-yc)\sin(\theta)^2\right). \tag{6}$$

The next equations shown below can be used to find the points that are closest to and most distant from the origin. To obtain the most distant point, λ must be positive, since increasing the size of the ellipse increases the value of the maximum distance only, assuming that the center of the circle $(0,0)$ does not lie inside the ellipse. If this is the case, then other ellipses outside that particular ellipse will determine the radius of the container circle.

$$\left\{\begin{array}{c} 2x=\lambda\cdot\frac{de(x,y)}{dx}\,(a,b,xc,yc,\theta;x,y) \\ 2y=\lambda\cdot\frac{de(x,y)}{dy}\,(a,b,xc,yc,\theta;x,y) \\ e\,(a,b,xc,yc,\theta;x,y)=0 \end{array}\right\}. \tag{7}$$

To illustrate these formulas by an example, consider the ellipse defined by equation $e(1.25,0.75,1,2,\pi/3;x,y)$. The point (x,y) on the boundary of this ellipse that is most distant from the origin can be found by solving the system of Eq. (7): the numerical solution is $(x,y)=(1.608,3.092)$, with the corresponding distance value $\sqrt{x^2+y^2}=3.485$.

In the optimization framework, the requirement regarding the positive sign of the multiplier λ will be enforced by setting search region bounds, rather than specifying a constraint. Moreover, the value of the maximum distance from the origin is obtained by evaluating $\sqrt{x^2+y^2}$ at the solution of Eq. (7) as done above. Note also that the system of Eq. (7) may fail, if the ellipse in question contains the origin. To handle this potential issue, constraints are added to the optimization strategy in order to keep the maximum distance point further from the origin than the minimum of the semi-major or semi-minor axis of the ellipse in question. To illustrate a partial configuration, a packed ellipse, a possible container circle, and the unique point of their intersection are displayed in Fig. 1. (Of course, the intersection will be empty for ellipses inside the container circle.)

Proceeding now to express the prevention of ellipse overlaps, all pairs of packed ellipses are prevented from overlapping by requiring that the minimum value of

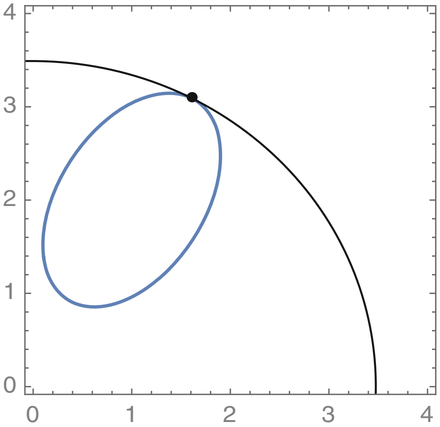


Fig. 1 A packed ellipse, the container circle, and their intersection point

the ellipse equation for the first ellipse (ellipse i), for any point on the second ellipse (ellipse j), is greater than a judiciously set (sufficiently small) $\varepsilon \geq 0$. This requirement will also be met applying the embedded Lagrange multiplier method: the complete details are presented in Kampas et al. [22]. The equations shown below determine the point on ellipse j that maximizes or minimizes the value of the function describing ellipse i. In the case considered here, λ must be negative to obtain the minimum. As indicated before, in the optimization strategy the requirement on the sign of λ will be enforced by setting its search region bounds rather than specifying an additional constraint.

$$\left\{\begin{array}{c} \frac{de(x,y)}{dx}\left(a_i, b_i, xc_i, yc_i, \theta_i; x, y\right) = \lambda \cdot \frac{de(x,y)}{dx}\left(a_j, b_j, xc_j, yc_j, \theta_j; x, y\right) \\ \frac{de(x,y)}{dy}\left(a_i, b_i, xc_i, yc_i, \theta_i; x, y\right) = \lambda \cdot \frac{de(x,y)}{dy}\left(a_j, b_j, xc_j, yc_j, \theta_j; x, y\right) \\ e\left(a_j, b_j, xc_j, yc_j, \theta_j; x, y\right) = 0 \end{array}\right\}. \tag{8}$$

In the optimization strategy, λ_i are the Lagrange multipliers in the equation for finding the point (xm_i, ym_i) on ellipse i that is most distant from the origin. The calculation is restricted to maximization by restricting the sign of λ_i to be positive. The square of the radius of the container circle obviously has to satisfy the relation $rc^2 \geq xm_i^2 + ym_i^2$ for all i. In addition, $\lambda_{i,j}$ are the Lagrange multipliers in the equations for finding the point $(x_{j,i}, y_{j,i})$ on ellipse j that minimizes the value of the equation describing ellipse i: this calculation is restricted to minimization by requiring the value of $\lambda_{i,j}$ to be negative. To summarize the model development steps described above, we obtain the following optimization model for the case of n ellipses.

$$
\begin{aligned}
\text{minimize} \quad & rc \\
\text{subject to} \quad & rc^2 \geq xm_i^2 + ym_i^2 \quad \text{for} \quad i = 1, \ldots, n \\
& xm_i^2 + ym_i^2 \geq \min\left(a_i, b_i\right)^2 \quad \text{for} \quad i = 1, \ldots, n \\
& 2 \cdot xm_i = \lambda_i \cdot \tfrac{de(x,y)}{dx}\left(a_i, b_i, xc_i, yc_i, \theta_i; xm_i, ym_i\right) \quad \text{for} \quad i = 1, \ldots, n \\
& 2 \cdot ym_i = \lambda_i \cdot \tfrac{de(x,y)}{dy}\left(a_i, b_i, xc_i, yc_i, \theta_i; xm_i, ym_i\right) \quad \text{for} \quad i = 1, \ldots, n \\
& e\left(a_i, b_i, xc_i, yc_i, \theta_i; xm_i, ym_i\right) = 0 \quad \text{for} \quad i = 1, \ldots, n \\
& \tfrac{de(x,y)}{dx}\left(a_i, b_i, xc_i, yc_i, \theta_i; x_{j,i}, y_{j,i}\right) = \lambda_{j,i} \cdot \tfrac{de(x,y)}{dx}\left(a_j, b_j, xc_j, yc_j, \theta_j; x_{j,i}, y_{j,i}\right) \\
& \qquad \text{for} \quad i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& \tfrac{de(x,y)}{dy}\left(a_i, b_i, xc_i, yc_i, \theta_i; x_{j,i}, y_{j,i}\right) = \lambda_{j,i} \cdot \tfrac{de(x,y)}{dy}\left(a_j, b_j, xc_j, yc_j, \theta_j; x_{j,i}, y_{j,i}\right) \\
& \qquad \text{for} \quad i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& e\left(a_j, b_j, xc_j, yc_j, \theta_j; x_{j,i}, y_{j,i}\right) = 0 \ \text{ for } \ i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& e\left(a_i, b_i, xc_i, yc_i, \theta_i; x_{j,i}, y_{j,i}\right) \geq \varepsilon \ \text{ for } \ i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& lxc_i \leq xc_i \leq uxc_i \quad \text{for} \quad i = 1, \ldots, n \\
& lyc_i \leq yc_i \leq uyc_i \quad \text{for} \quad i = 1, \ldots, n \\
& -\pi \leq \theta_i \leq \pi \quad \text{for} \quad i = 1, \ldots, n \\
& lxm_i \leq xm_i \leq uxm_i \quad \text{for} \quad i = 1, \ldots, n \\
& lym_i \leq ym_i \leq uym_i \quad \text{for} \quad i = 1, \ldots, n \\
& lx_{j,i} \leq x_{j,i} \leq ux_{j,i} \quad \text{for} \quad i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& ly_{j,i} \leq y_{j,i} \leq uy_{j,i} \quad \text{for} \quad i = 1, \ldots, n-1, j = i+1, \ldots, n \\
& 0 \leq \lambda_i \leq 2 \cdot u\lambda_i \quad \text{for} \quad i = 1, \ldots, n \\
& 2 \cdot l\lambda_{j,i} \leq \lambda_{j,i} \leq 0 \quad \text{for} \quad i = 1, \ldots, n-1, j = i+1, \ldots, n
\end{aligned}
\tag{9}
$$

In the model (9) the symbols $l.$ and $u.$ denote lower and upper bounds for the variable sandwiched between the corresponding pair of inequalities. Note that these bounds are defined for each ellipse packing instance, in order to facilitate achieving feasible solutions.

The optimization model (9) has $1 + 6n + 3(n-1)^2$ decision variables and, in addition to the bound constraints that are imposed on all decision variables, $5n + 4(n-1)^2$ nonlinear constraints: the latter constraints are all non-convex. Therefore model (9) represents a highly nonlinear optimization problem-class in which both the number of variables and constraints increase quadratically as a function of n. As an example, the packing problem-instance with $n = 10$ ellipses leads to a model with 304 decision variables, corresponding bound constraints, and 374 non-convex constraints. Based on these observations, we conjecture that the computational difficulty of model (9) will rapidly increase as a function of the number of packed ellipses.

3 Numerical Global Optimization for Ellipse Packings

3.1 Global Optimization: A Review of Basic Concepts

The objective of global optimization (GO) is to find the "absolutely best" solution of multi-extremal optimization problems. Most object packing problems are provably multi-modal, often possessing a large number of local optima. Without going into technical details, a simple inspection of the relations leading to the problem statement (9) leads to the conclusion that general ellipse packings lead to a difficult GO model-class.

As we already noted, one cannot expect to find analytical solutions to general object packing problems—even when considering far less complicated models than the one presented here. Therefore, we have been applying numerical optimization to handle various object configuration problems. For detailed discussions with numerical examples, cf., e.g., Pintér [23], Stortelder et al. [24], Pintér and Kampas [7, 8], Kampas and Pintér [9], Castillo et al. [11], Pintér and Kampas [25].

In this study, we apply the Lipschitz Global Optimizer (LGO) solver system for global and local (nonlinear) optimization, in its implementation linked to the computing system *Mathematica*.

3.2 The LGO Solver System for Global-Local Optimization

The LGO software package is aimed at finding the numerical global optimum of model instances from a very general class of continuous global optimization problems. The core LGO solver system with implementations for various modeling platforms has been described by Pintér [26–31], Pintér et al. [32]. For more recent development work including benchmarking studies and some applications, consult, e.g., Çaĝlayan and Pintér [33], Pintér and Horváth [34], Pintér and Kampas [25], Pintér [35]. Further technical details are discussed by the current LGO documentation [36], which also includes a rather extensive list of topical references. Therefore here we present only a brief summary of LGO features and implementation details pertinent to our study.

The core (Fortran or C/C++/C# compiler platform based) LGO solver suite integrates several derivative-free global and local optimization strategies, without requiring higher-order (gradient or Hessian) information. The strategies referred to include regularly spaced sampling, as a global pre-solver (RSS); a branch-

and-bound global search method (BB); global adaptive random search (GARS); multi-start based global random search (MS); and local search (LS). According to extensive numerical experience, in complicated GO models, MS (with added LS solver phases) often finds the best numerical solution. For this reason, MS is the recommended default LGO solver option that has been used also in our present work.

LGO is available for a number of model development platforms as a professional (commercial) solver option. Similarly to some of our circle packing studies, the ellipse packing model has been implemented in *Mathematica* (Wolfram [37]): therefore, we use here the LGO implementation linked to *Mathematica.* This implementation, with the software product name *MathOptimizer Professional*, has been used also in our more recent benchmarking studies: cf., e.g., Pintér and Kampas [25, 38].

The *MathOptimizer Professional* software package combines *Mathematica*'s optimization model development capabilities with the external LGO solver suite. To illustrate this aspect, we note that the entire ellipse packing model and the LGO call for its solution consist only of a few dozen carefully developed *Mathematica* code lines, including code to display the configurations found. Let us also note that LGO solver performance compares favorably to the corresponding optimization features of *Mathematica*: this aspect becomes increasingly important when solving difficult GO problems like the packing models discussed here. *MathOptimizer Professional* can be used to handle sizeable models, with thousands of variables and general constraints.

4 Illustrative Numerical Results

To our best knowledge, there are no previously studied model instances available for the general ellipse packing problem considered in our present study. The problem instances summarized in Table 1 are taken from Kallrath and Rebennack [20], recalling that their work was aimed at packing ellipses in optimized rectangles. This choice of test instances allows comparisons regarding the packing density of rectangular vs. circular packings—not in a competitive sense, since the configuration geometries are different.

Our calculations were (mostly) performed on a PC with a quad-core Intel i7 processor running at 3.7 GHz, with 16 GBytes of RAM, using *MathOptimizer Professional* running in conjunction with *Mathematica* version 10, and using the GCC [39] Fortran compiler to automatically generate the files for using LGO.

Table 2 summarizes the computational results, noting that CPU times are reasonable even for the last two largest problem instances. Table 2 also displays information regarding the packing fraction and maximal constraint violation of the solutions found.

Table 1 Ellipse packing instances

Test case	(a_i, b_i)	Total area to be packed
ax2a	(2.0,1.5), (1.5,1.0)	14.13717
ax2b	(2.0,1.5), (1.8,1.4)	17.34159
ax3a	ax2a + (1.0,0.8)	16.65044
ax3b	ax2b + (0.8,0.7)	19.10088
ax4a	ax3a + (0.9,0.75)	18.77102
ax4b	ax3b + (1.1,1.0)	22.55664
ax5a	ax4a + (0.8,0.6)	20.27898
ax5b	ax4b + (0.9,0.8)	24.81858
ax6	ax5a + (0.7,0.3)	20.93872
ax11	(2.0,1.5), (1.8,1.5), (1.6,1.5), (1.5,1.2), (1.3,1.0), (1.2,0.9), (1.1,0.8), (1.0,0.75), (0.9,0.6), (0.8,0.5), (0.7,0.3)	47.31239
ax14	$7 \cdot (1.0,0.75) + 7 \cdot (0.5,0.375)$	20.6167

Table 2 Ellipse packing results

Test case	Packing radius *rc*	Area of optimized container	Packing fraction	Time (s)	Max constraint violation
ax2a	2.49873	19.61501	0.72073	0.5	8E-9
ax2b	2.9	26.42079	0.65636	0.6	1E-9
ax3a	2.56257	20.63010	0.80709	1.0	5E-10
ax3b	2.9	26.42079	0.72295	1.0	5E-10
ax4a	2.74972	23.75346	0.79024	3.1	4E-9
ax4b	2.98985	28.08333	0.80320	3.0	5E-9
ax5a	2.84911	25.50165	0.79520	7.6	3E-12
ax5b	3.26085	33.40500	0.74296	7.7	9E-9
ax6	2.89647	26.35651	0.79444	20.0	4E-9
ax11	4.35292	59.52662	0.79481	31.0	5E-10
ax14	2.864	25.76890	0.80006	106.0	1E-7

In Table 3, we summarize our results for general ellipse packings in an optimized circle next to the best solutions found for packings in an optimized rectangle given by Kallrath and Rebennack [20]. As mentioned, the configuration geometries are rather different given the different optimized containers. However, we still get an overall impression regarding the range of packing densities that can be achieved for rectangular and circular containers, at least for our model instances.

Illustrative packing configurations for the two largest problem instances ax11 and ax14 are given in Figs. 2 and 3, respectively. The points shown on the ellipses and optimizer container are the points that serve to prevent ellipse overlap and to determine the radius of the container circle. Notice that to prevent the overlap of a pair of ellipses, there is only a point on one of the two ellipse boundaries, not on both.

Table 3 Packing results in a circle and in a rectangle

	Circular container		Rectangular container	
Test case	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction
ax2a	19.61501	0.72073	18.00000	0.78540
ax2b	26.42079	0.65636	22.23152	0.78005
ax3a	20.63010	0.80709	21.38577	0.77858
ax3b	26.42079	0.72295	25.22467	0.75723
ax4a	23.75346	0.79024	23.18708	0.80955
ax4b	28.08333	0.80320	28.54159	0.79031
ax5a	25.50165	0.79520	25.29557	0.80168
ax5b	33.40500	0.74296	31.28873	0.79321
ax6	26.35651	0.79444	25.51043	0.82079
ax11	59.52662	0.79481	64.59177	0.73248
ax14	25.76890	0.80006	29.65886	0.69513

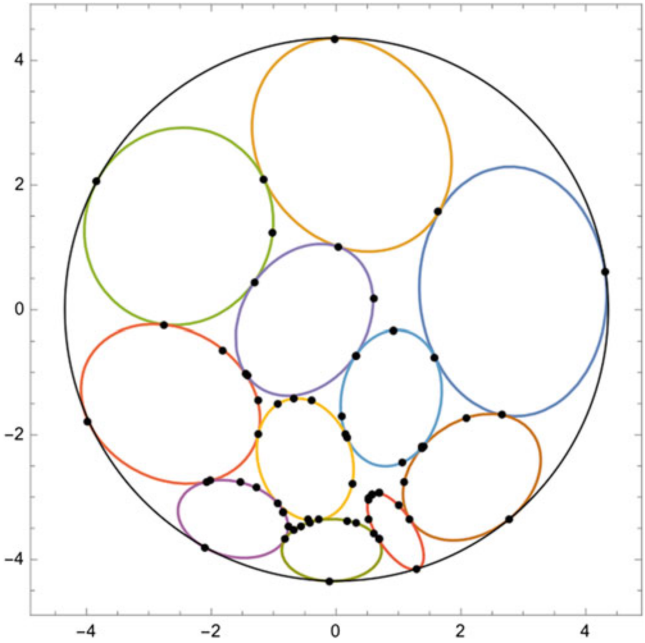


Fig. 2 Packing of the ellipses given in example ax11

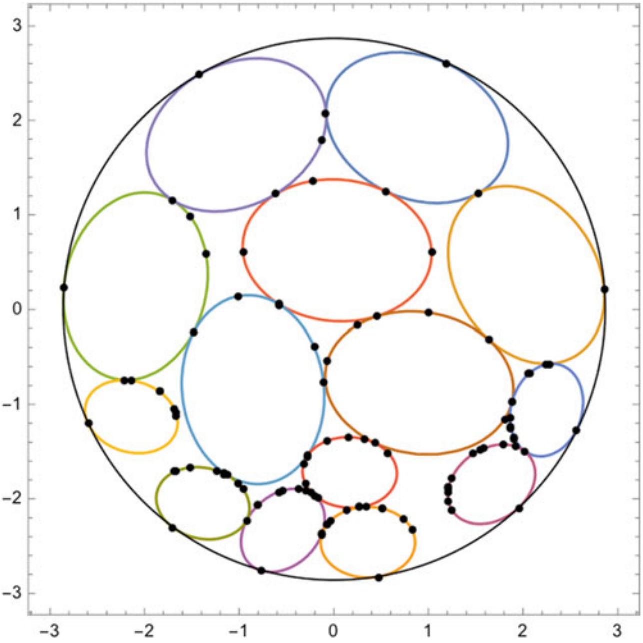


Fig. 3 Packing of the ellipses given in example ax14

5 Summary and Conclusions

In this study, we present a general ellipse packing problem with respect to a circular container with optimized radius. Our literature review illustrates the significant difficulty of similar packing problems. In the global optimization strategy, the prevention of ellipse overlaps proceeds simultaneously with the minimization of the radius of the container circle. To solve the resulting models numerically, we use the LGO solver system in its implementation linked to the computing system *Mathematica*. Our results demonstrate that the embedded Lagrangian multipliers based modeling approach combined with global optimization enables the computational solution of difficult ellipse packing problems with hundreds of variables and non-convex constraints. Preliminary research indicates that our model development approach has the potential to handle ellipse packing problems also with respect to other types of container sets.

References

1. Szabó, P.G., Csendes, T., Casado, L.G., García, I.: Equal circles packing in a square I – problem setting and bounds for optimal solutions. In: Giannessi, F., Pardalos, P.M., Rapcsák, T. (eds.) Optimization Theory: Recent Developments from Mátraháza. Kluwer, Dordrecht (2001)
2. Szabó, P.G., Markót, M.C., Csendes, T.: Global optimization in geometry – circle packing into the square. In: Audet, P., Hansen, P., Savard, G. (eds.) Essays and Surveys in Global Optimization. Kluwer, Dordrecht (2005)
3. Szabó, P.G., Markót, M.C., Csendes, T., Specht, E., Casado, L.G., García, I.: New Approaches to Circle Packing in a Square with Program Codes. Springer, New York (2007)
4. Markót, M.C.: Optimal packing of 28 equal circles in a unit square – the first reliable solution. Numerical Algorithms. **37**, 253–261 (2005)
5. Riskin, M.D., Bessette, K.C., Castillo, I.: A logarithmic barrier approach to solving the dashboard planning problem. INFOR **41**, 245–257 (2003)
6. Castillo, I., Sim, T.: A spring-embedding approach for the facility layout problem. J. Oper. Res. Soc. **55**, 73–81 (2004)
7. Pintér, J.D., Kampas, F.J.: Nonlinear optimization in *Mathematica* with *MathOptimizer Professional*. Math. Educ. Res. **10**, 1–18 (2005)
8. Pintér, J.D., Kampas, F.J.: *Mathoptimizer professional*: key features and illustrative applications. In: Liberti, L., Maculan, N. (eds.) Global Optimization: From Theory to Implementation, pp. 263–280. Springer, New York (2006)
9. Kampas, F.J., Pintér, J.D.: Configuration analysis and design by using optimization tools in *Mathematica*. The Math J. **10**, 128–154 (2006)
10. Addis, B., Locatelli, M., Schoen, F.: Efficiently packing unequal disks in a circle. Oper. Res. Lett. **36**, 37–42 (2008)
11. Castillo, I., Kampas, F.J., Pintér, J.D.: Solving circle packing problems by global optimization: numerical results and industrial applications. Eur. J. Oper. Res. **191**, 786–802 (2008)
12. Grosso, A., Jamali, A.R.M.J.U., Locatelli, M., Schoen, F.: Solving the problem of packing equal and unequal circles in a circular container. J. Glob. Optim. **47**, 63–81 (2010)
13. Hifi, M., M'Hallah, R.: A literature review on circle and sphere packing problems: models and methodologies. Adv. Oper. Res. 2009, 22, 150624 (2009). doi:https://doi.org/10.1155/2009/150624
14. Fasano, G.: Solving Non-standard Packing Problems by Global Optimization and Heuristics. Springer, Cham (2014)
15. Fasano, G., Pintér, J.D. (eds.): Optimized Packings with Applications. Springer, New York (2015)
16. Gensane, T., Honvault, P.: Optimal packings of two ellipses in a square. Forum Geom. **14**, 371–380 (2014)
17. Birgin, E.G., Bustamante, L.H., Flores Callisaya, H., Martínez, J.M.: Packing circles within ellipses. Int. Trans. Oper. Res. **20**, 365–389 (2013)
18. Litvinchev, I., Infante, L., Ozuna, L.: Packing circular-like objects in a rectangular container. J. Comput. Syst. Sci. Int. **54**, 259–267 (2015)
19. Galiev, S.I., Lisafina, M.S.: Numerical optimization methods for packing equal orthogonally oriented ellipses in a rectangular domain. Comput. Math. Math. Phys. **53**, 1748–1762 (2013)
20. Kallrath, J., Rebennack, S.: Cutting ellipses from area-minimizing rectangles. J. Glob. Optim. **59**, 405–437 (2014)
21. Uhler, C., Wright, S.J.: Packing ellipsoids with overlap. SIAM Rev. **55**, 671–706 (2013)
22. Kampas, F.J., Castillo, I. Pintér, J.D.: Optimized ellipse packings in regular polygons using embedded Lagrange multipliers (2017). Submitted for publication
23. Pintér, J.D.: Globally optimized spherical point arrangements: model variants and illustrative results. Ann. Oper. Res. **104**, 213–230 (2001)
24. Stortelder, W.J.H., de Swart, J.J.B., Pintér, J.D.: Finding elliptic Fekete point sets: two numerical solution approaches. J. Comput. Appl. Math. **130**, 205–216 (2001)

25. Pintér, J.D., Kampas, F.J.: Benchmarking nonlinear optimization software in technical computing environments. I. Global optimization in *Mathematica* with *MathOptimizer Professional*. TOP. **21**, 133–162 (2013)
26. Pintér, J.D.: Global Optimization in Action. Kluwer Academic Publishers, Dordrecht (1996). Now distributed by Springer Science + Business Media, New York
27. Pintér, J.D.: LGO − a program system for continuous and Lipschitz global optimization. In: Bomze, I., Csendes, T., Horst, R., Pardalos, P.M. (eds.) Developments in Global Optimization, pp. 183–197. Kluwer Academic Publishers, Dordrecht (1997)
28. Pintér, J.D.: Global optimization: software, test problems, and applications. In: Pardalos, P.M., Romeijn, H.E. (eds.) Handbook of Global Optimization, vol. 2, pp. 515–569. Kluwer Academic Publishers, Dordrecht (2002)
29. Pintér, J.D.: Nonlinear optimization in modeling environments: software implementations for compilers, spreadsheets, modeling languages, and integrated computing systems. In: Jeyakumar, V., Rubinov, A.M. (eds.) Continuous Optimization: Current Trends and Applications, pp. 147–173. Springer, New York (2005)
30. Pintér, J.D.: Nonlinear optimization with GAMS/LGO. J. Glob. Optim. **38**, 79–101 (2007)
31. Pintér, J.D.: Software development for global optimization. In: Pardalos, P.M., Coleman, T.F. (eds.) Global Optimization: Methods and Applications, Fields Institute Communications, vol. 55, pp. 183–204. American Mathematical Society, Providence, RI (2009)
32. Pintér, J.D., Linder, D., Chin, P.: Global optimization toolbox for maple: an introduction with illustrative applications. Optim. Methods Softw. **21**, 565–582 (2006)
33. Çaĝlayan, M.O., Pintér, J.D.: Development and calibration of a currency trading strategy using global optimization. J. Glob. Optim. **56**, 353–371 (2013)
34. Pintér, J.D., Horváth, Z.: Integrated experimental design and nonlinear optimization to handle computationally expensive models under resource constraints. J. Glob. Optim. **57**, 191–215 (2013)
35. Pintér, J.D.: How difficult is nonlinear optimization? A practical solver tuning approach, with illustrative results. Ann. Oper. Res. 1–23 (2017). https://doi.org/10.1007/s10479-017-2518-z. Preprint available at www.optimization-online.org/DB_FILE/2014/06/4409.pdf
36. Pintér, J.D.: LGO – a Model Development and Solver System for Global-Local Nonlinear Optimization, User's Guide, Current edn. Pintér Consulting Services, Inc., Halifax (2016)
37. Wolfram Research: Mathematica (Release 11). Wolfram Research, Inc., Champaign, IL (2016)
38. Pintér, J.D., Kampas, F.J.: Getting Started with *Mathoptimizer professional*. Pintér Consulting Services, Inc., Halifax (2015)
39. GCC: GCC, the GNU Compiler Collection. (2016). https://gcc.gnu.org/

Column Generation Approach to the Convex Recoloring Problem on a Tree

Sunil Chopra, Ergin Erdem, Eunseok Kim, and Sangho Shim

Abstract The convex recoloring (CR) problem is to recolor the nodes of a colored graph at minimum number of color changes such that each color induces a connected subgraph. We adjust to the convex recoloring problem the column generation framework developed by Johnson et al. (Math Program 62:133–151, 1993). For the convex recoloring problem on a tree, the subproblem to generate columns can be solved in polynomial time by a dynamic programming algorithm. The column generation framework solves the convex recoloring problem on a tree with a large number of colors extremely fast.

Keywords Convex recoloring problem • Phylogenetic tree • Bioinformatics • Clustering problem • Column generation • Linear programming • Large scale optimization • Set partition problem

1 The Convex Recoloring Problem

The convex recoloring problem on a tree was first investigated by Moran and Snir [8]. Campêlo et al. [2] studied the associated integer programming formulation and provided several classes of facet defining inequalities. Chopra et al. [3] introduced an extended integer linear programming (ILP) formulation of the CR problem on a tree and showed their formulation to be stronger than the ILP model introduced by Campêlo et al. [2]. In this paper, we present a column generation

S. Chopra
Kellogg School of Management, Northwestern University, Evanston, IL 60208, USA
e-mail: s-chopra@kellogg.northwestern.edu

E. Erdem • S. Shim (✉)
Department of Engineering, Robert Morris University, Moon Twp, PA 15090, USA
e-mail: erdem@rmu.edu; shim@rmu.edu

E. Kim
KAIST, 291 Daehak-ro, Yuseong-gu,, Daejeon 34141, Republic of Korea
e-mail: scbgm@naver.com

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_3

approach that is shown to be computationally very effective in a category of problem instances where the approach of Chopra et al. [3] is not very effective.

We now introduce the problem with the notation used by Campêlo et al. [2]. Let $\mathcal{C} = \{1, \ldots, k\}$ be a set of colors and $G = (V, E)$ be a graph with node set V and edge set E. A *partial coloring* of a graph G is a function $C : V \to \mathcal{C} \cup \{\emptyset\}$, where $\emptyset$ indicates absence of color. A node $v \in V$ is said to be *uncolored* if $C(v) = \emptyset$. The coloring C is called *total* if there is no uncolored node; i.e., $\emptyset \notin C(V)$ where $C(V)$ is the image of the function C.

A colored graph is a pair (G, C) consisting of a graph G and a coloring C of its nodes. A total coloring C is said to be *convex* if, for each $t \in \mathcal{C}$, the set of nodes with color t induces a connected subgraph of G. A *convex partial coloring* is a partial coloring that can be extended to a convex total coloring by solely assigning a color in $\mathcal{C}$ to each uncolored node. A *good coloring* is a partial coloring in which each color induces a connected subgraph. Campêlo et al. [2] point out that every good coloring of a graph $G = (V, E)$ can be extended to a convex total coloring in $O(|V| + |E|)$ time.

Given a non-convex coloring of a graph, the *recoloring distance* is defined as the minimum number of color changes at the colored nodes needed to obtain a convex partial coloring [9]. This measure can be generalized to a weighted model, where changing the color of node v costs a nonnegative weight $w(v)$ depending on v. This problem can be stated as follows:

Problem 1 (Convex Recoloring (CR)) Given a partially colored graph (G, C), an available color set $\mathcal{C}$, and a cost function $w : V \to \mathbb{Q}_{\geq 0}$, find a convex partial recoloring C' that minimizes $\sum_{v \in R_C(C')} w(v)$ where $R_C(C') = \{v \in V : C(v) \neq \emptyset \text{ and } C(v) \neq C'(v)\}$ is the set of nodes recolored by C'.

We note that, corresponding to any convex partial coloring that is not good there is a good coloring with the same weight. Thus, we will consider that in the CR problem we are interested only in finding good recolorings of the graph.

The CR problem has been shown to be NP-hard in many settings. Moran and Snir [9] showed the problem to be NP-hard on paths. Kanj and Kratsch [5] proved that it is NP-hard on paths even if each color appears at most twice. Moran et al. [10] showed that computing the convex recoloring cost of a 2-colored graph is NP-hard. Campêlo et al. [1] improved this result by showing that the unweighted uniform CR problem is NP-hard even on 2-colored grids. For a more detailed literature review, readers may refer to Chopra et al. [3].

The convex recoloring problem can measure the gap between phylogeny and taxonomy [6, 7]. Figure 1 illustrates three species by three colors on the phylogenetic tree where each leaf node represents a homologous protein sequence. The individuals of the tree on the left cannot be clustered into connected subtrees of one color with traditional methods. The taxonomically labeled phylogenetic tree depicted as a non-convex partial coloring is not concordant with the species taxonomic assignments. If the red color at the second end node changes into green, the phylogenetic tree can be clustered into connected subtrees depicted as a good coloring on the right. The minimum number of color changes is the recoloring

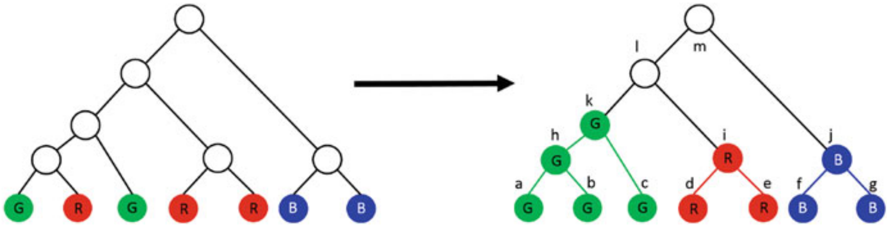


Fig. 1 Convex recoloring

distance that equals one in the example of the figure obtained by changing the color of only one colored node (the second leaf node). A short recoloring distance indicates that the phylogeny and taxonomy are close to each other.

As mentioned earlier, Chopra et al. [3] introduced an extended integer linear programming (ILP) model of the CR problem on a tree that is superior (both theoretically and computationally) to the conventional ILP model introduced by Campêlo et al. [2]. While the extended formulation of Chopra et al. [3] was computationally quite efficient when the number of colors was not large, it had some difficulty in solving problems with a very large number of colors. In particular, they failed in six large scale problem instances where the number of colors was about half the number of nodes.

In this paper, we develop a column generation scheme for the CR problem on a tree by adjusting the Johnson–Mehrotra–Nemhauser column generation framework [4], and successfully solve all the six large scale problem instances within an hour. Our column generation scheme uses the polynomial time dynamic programming algorithm introduced by Chopra et al. [3] to generate columns. From our computational experiments we observe two characteristics of the column generation approach:

- The column generation approach is extremely fast when the number of colors is large.
- The column generation approach is extremely slow when the number of colors is small.

Fortunately, the ILP model introduced by Chopra et al. [3] is extremely fast when the number of colors is small. Therefore, we naturally conclude a hybrid of the two linear programming approaches: the extended ILP model for a small number of colors and the column generation framework for a large number of colors. It is not too surprising that column generation does well when the number of colors is very large. The CR problem on a tree is equivalent to the problem of partitioning a tree into subtrees. For very large number of colors, each subtree is very small and the coefficient matrix of the restricted master problem is very sparse. This speeds up the column generation approach.

In Sects. 2 and 3, we introduce two linear programming approaches: the ILP model developed by Chopra et al. [3] and the column generation framework given by adjusting Johnson–Mehrotra–Nemhauser framework. In Sect. 3, we also review the linear time dynamic programming algorithm for the subproblem of our column generation approach. In Sect. 4, we perform computational experiments over the largest 63 problem instances provided by Chopra et al. [3] comparing the two linear programming approaches. In Sect. 5 we summarize our findings and discuss future work.

2 Integer Linear Programming Model for the CR Problem on a Tree

In this section we introduce the integer linear programming (ILP) model for the convex recoloring problem on a tree developed by Chopra et al. [3]. The convex recoloring problem aims to minimize the number of color changes at the colored nodes to obtain a good coloring. This is equivalent to maximizing the number of colored nodes that do not change their color when obtaining a good coloring. We now define the set of node variables that are used in the model. Given a set of colors $\mathcal{C} = \{1, \ldots, k\}$, a tree $T = (V, E)$, and a partial coloring C, define the *node variables* $x = (x_{ut} : u \in V, t \in \mathcal{C})$ where $x_{ut} = 1$ if node u is assigned to color $t \in \mathcal{C}$, and 0 otherwise. To express the objective function, for each $u \in V$ and $t \in \mathcal{C}$, we define a constant $w(u,t)$, which is 1 if $C(u) = t$, and is 0 otherwise. We then employ additional variables y_{et} which we call *edge variables*. An edge variable y_{et} for each edge $e \in E$ and for each color $t = 1, \ldots, k$ takes the value 1 if both end nodes of edge e are colored by t, and 0 otherwise. With edge variables, we use a basic property of a tree that the number of edges equals the number of nodes minus one.

The integer linear programming model for the CR problem on a tree developed by Chopra et al. [3] is written as follows:

$$\max \quad \sum_{t=1}^{k} \sum_{u \in V} w(u,t) x_{ut}$$

$$s.t. \quad \sum_{t=1}^{k} x_{ut} \leq 1 \text{ for } u \in V \tag{1}$$

$$\sum_{u \in V} x_{ut} - \sum_{e \in E} y_{et} \leq 1 \text{ for } t \in \mathcal{C} \tag{2}$$

$$\left.\begin{array}{l} -x_{ut} + y_{uvt} \leq 0 \\ -x_{vt} + y_{uvt} \leq 0 \end{array}\right\} \text{ for edge } uv \in E \text{ and for } t \in \mathcal{C} \tag{3}$$

$$x_{ut} \in \{0,1\} \text{ for } u \in V \text{ and } t \in \mathcal{C} \tag{4}$$

$$y_{et} \in \{0,1\} \text{ for } e \in E \text{ and } t \in \mathcal{C} \tag{5}$$

The ILP model is mathematically and computationally shown in Chopra et al. [3] to dominate the conventional ILP model developed by Campêlo et al. [2]. Inequalities (2) ensure that the nodes of each color $t \in \mathcal{C}$ induce a connected subgraph (subtree) of T. Inequalities (3) ensure that if an edge $e = (u, v)$ is assigned color t, both nodes u and v must also be assigned to color t. The coloring corresponding to a feasible solution of this formulation is shown in Chopra et al. [3] to be a good coloring.

Theorem 1 (Chopra et al. [3]) *x is a projection of an integer solution (x, y) to the formulation (1)–(5), if and only if it defines a good coloring.*

The LP-relaxation (1)–(3) along with non-negativity is shown in Chopra et al. [3] to be very strong; it ended up with integer solutions over all the 81 medium scale problem instances provided by Campêlo et al. [2]. Note that the LP-relaxation has $O(nk)$ variables and $O(nk)$ constraints where $n = |V|$ and $k = |\mathcal{C}|$.

3 Column Generation Framework

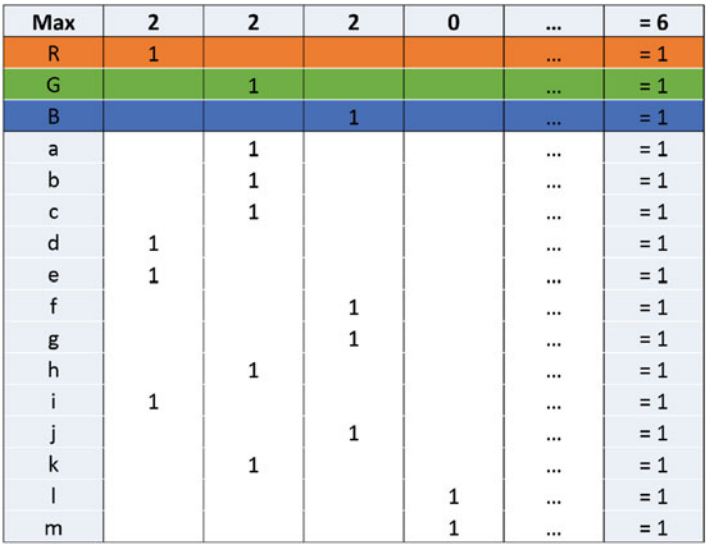
In this section, we develop a column generation scheme for the convex recoloring problem by adapting the column generation framework for min-cut clustering provided by Johnson et al. [4].

3.1 Master Problem

To define the master problem for the column generation framework, we define a binary matrix A, each of whose columns has $k + |V|$ elements and corresponds to the incidence vector of a connected subtree and an indicator for the color of the subtree. Each column $(\gamma, x)^T$ of A thus contains two binary vectors. The length of γ is k corresponding to the number of colors and the length of x corresponds to the number of nodes in V. If x is the incidence vector of the set of nodes of a subtree that have a common color t, the vector γ has $\gamma_t = 1$ corresponding to the color t, with all other components of γ being 0. As a result, $(\gamma, x)^T$ defines the nodes in a connected subtree and their corresponding color. If the vector γ has all elements 0, $(\gamma, x)^T$ defines a set of uncolored nodes. The uncolored nodes do not have to induce a connected subgraph. Given the matrix A, and a vector $\mathbf{1}$ with all components 1, we define the master problem as follows:

$$\begin{aligned} \text{Max} \quad & Wz, \\ s.t. \quad & Az = \mathbf{1}, \\ & z \geq 0. \end{aligned}$$

where all the components of the right-hand side $\mathbf{1}$ are 1 and the objective function W is defined as follows.



Max	2	2	2	0	...	= 6
R	1				...	= 1
G		1			...	= 1
B			1		...	= 1
a		1			...	= 1
b		1			...	= 1
c		1			...	= 1
d	1				...	= 1
e	1				...	= 1
f			1		...	= 1
g			1		...	= 1
h		1			...	= 1
i	1				...	= 1
j			1		...	= 1
k		1			...	= 1
l				1	...	= 1
m				1	...	= 1

Fig. 2 The coefficient matrix of the master problem which includes the columns indicating the convex recoloring in Fig. 1

The objective coefficient $W(\gamma, x)$ corresponding to a column $(\gamma, x)^T$ with $\gamma_t = 1$ is the number of the nodes v of the subtree x for which t is the original color; i.e., $w(v, t) = 1$. If t corresponds to no color, we set $w(v, t) = 0$. The objective coefficient $W(\gamma, x)$ corresponding to a column $(\gamma, x)^T$ with $\gamma_t = 1$ can be evaluated as follows:

$$W(\gamma, x) = \sum_{v \in V} w(v, t) x_v.$$

Note that if $\gamma_t = 1$, x must correspond to a connected subtree but if $\gamma = 0$, x can indicate any subset of nodes and $W(\gamma, x) = 0$. Figure 2 illustrates the coefficient matrix of the master problem which includes the columns indicating the convex recoloring in Fig. 1. For example, the first column $(\gamma, x)^T$ of the coefficient matrix indicates that three nodes d, e, and i are colored in R inducing a subtree. Since R was the initial color of two nodes d and e among the three, the objective coefficient corresponding to the column is $W(\gamma, x) = 2$.

Our column generation approach assumes that a subset of columns of A (corresponding to a set of subtrees and color assigned to each) are available at each iteration. At each iteration, a solution to the current master provides a dual variable ρ_t corresponding to each color $t \in \mathcal{C}$ and a dual variable π_v corresponding to each node $v \in V$. These dual variables are used to generate an additional column for A, which has the largest reduced cost for the current dual variables. We now detail the approach used to generate an incoming column.

3.2 The Subproblem to Generate the Optimal Subtree in the Column Generation Framework

We now use the linear time dynamic programming algorithm introduced by Chopra et al. [3] to introduce the column with the largest reduced cost. Given the dual variable vectors ρ and π, the reduced cost of a column $(\gamma, x)^T$ of A is

$$W(\gamma, x)^T - (\rho, \pi)(\gamma, x)^T = \sum_{v \in V} w(v,t)x_v - \rho\gamma - \pi x = -\rho_t + \sum_{v \in V}(w(v,t) - \pi_v)x_v,$$

where $\rho_\emptyset$ is assumed to be 0. Thus, we may generate the column $(\gamma, x)^T$ of the largest reduced cost

$$\max_{t \in \{\emptyset\} \cup \mathcal{C}} \left\{ -\rho_t + \max_x \left\{ \sum_{v \in V}(w(v,t) - \pi_v)x_v \, : x \text{ is a subtree if } t \neq \emptyset \right\} \right\}, \tag{6}$$

where x is any subset of nodes if $t = \emptyset$. If the largest reduced cost (6) is zero or negative, the current basis is optimal for the master problem.

For each color t, the subtree problem to identify the subtree colored with t of the maximum reduced cost is

$$\max_x \left\{ \sum_{v \in V}(w(v,t) - \pi_v)x_v \, : x \text{ is a subtree} \right\}. \tag{7}$$

The problem can be solved by the linear time dynamic programming algorithm introduced by Chopra et al. [3].

Figure 3 illustrates the dynamic programming algorithm introduced by Chopra et al. [3]. Given color $t \in \mathcal{C}$, let $a_i = w(i,t) - \pi_i$ denote the values of nodes $i \in V$, which are written in the circles representing the nodes in the figure. The dynamic programming algorithm solves in linear time the problem to identify the maximum subtree in the values $a_i, i \in V$.

Let us fix a root $r \in V$ arbitrarily. Such a root implicitly defines an orientation of each edge away from the root. Let S_i be the set of children of node i and p_i its parent node (undefined for the root). Let us define $T_i = (V_i, E_i)$ for each $i \in V$ as the subtree of T rooted in i, and containing all descendant nodes of i in T and corresponding edges. In Fig. 3, the root node r is the node at the top with value -3.

For each node $i \in V$, let us define $H(i)$ as the maximum value of the following problem: Find a subtree of T_i that is either empty or is rooted at i and maximizes the sum of node values in the subtree. Furthermore, let K be the value of the maximum

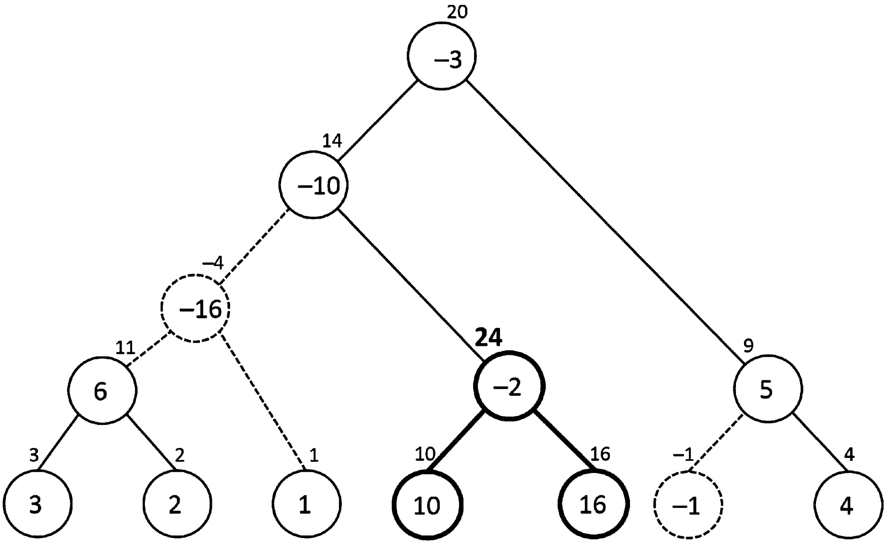


Fig. 3 Dynamic programming recursion

subtree of the tree (i.e., maximum sum of node values of a subtree of T_r). We can write the following recursions:

$$H(i) = \max\left\{0, a_i + \sum_{j\in S_i} H(j)\right\} \tag{8}$$

$$K = \max\left\{0, \max_{i\in V}\left(a_i + \sum_{j\in S_i} H(j)\right)\right\} \tag{9}$$

If $H(i) = 0$ in the recursion, we delete node i and set T_i to be empty.

In Fig. 3, we perform the recursion upward from bottom to top. The number over each node $i \in V$ is $a_i + \sum_{j\in S_i} H(j)$ in (8). We delete the nodes of values -16 and -1 where $a_i + \sum_{j\in S_i} H(j)$ are equal to -4 and -1 and T_i are empty. Observe that $K = 24$ and the maximum solution T_i is the subtree rooted at the node of value -2 and induced by the three nodes of values -2, 10, and 16.

For each color $t = 1, \ldots, k$, the dynamic programming algorithm generates a tree with the maximum reduced cost. From among these subtrees (one for each color t), a subtree of maximum reduced cost is selected as the incoming column for A. Therefore, one iteration of column generation process takes $O(nk)$ time complexity.

We start our column generation approach with the identity matrix as the initial set of columns. The identity matrix means each and every node is uncolored.

4 Computational Results on Large Scale Phylogenetic Trees

We perform computational experiments over the 63 larger problem instances generated by Chopra et al. [3] on the largest seven phylogenetic trees from TreeBASE.org which are listed in Table 1. They had used the method developed by Campêlo et al. [2] to randomly generate a total coloring according to two probability parameters: $p_c = 0.005$, 0.05 or 0.5, the probability of changing the color, and $p_n = 0.25, 0.5$ or 0.75, the probability of noise. The root of the tree is assigned to the color 1, and in a recursive manner, the child node has the same color as its immediate forefather with probability $1 - p_c$ or has the next unused color with probability p_c. Therefore, large p_c implies large number k of colors. After generating the coloring, each node keeps its color with probability $1-p_n$, or changes its color with probability p_n. If a color change occurs, a color is selected with equal probability across the available colors. Larger p_n implies bigger optimality gap of the initial coloring.

Over the 63 large scale problem instances, we compare the performance of two linear programming approaches: the integer linear programming model (Sect. 2) and the column generation framework (Sect. 3). The integer linear programming model is implemented in Python 2.7 as the language running the Python code provided by Chopra et al. [3]. The column generation approach is implemented in Java 1.8 as the language. Both linear programming approaches are implemented using Gurobi 7.0 as the solver and carried out on a machine with 32 GB of RAM and 4.0 GHz processor of CPU.

Table 2 shows that for the problem instances ($p_c = 50\%$) with the largest number of colors, the column generation (CG) approach performs much better in time and memory than the integer linear programming (ILP) model. In particular, the column generation approach solved within an hour all the six largest problem instances in which the integer linear programming model introduced by Chopra et al. [3] failed because of Out-of-Memory (OOM). For large values of k, the column generation approach performs better and better in the ratio of the computational times as the number $|V|$ of the nodes grows bigger.

However, the column generation approach does not perform well when the number of colors k is not very large. Table 3 compares the performance of the ILP model (ILP) and the column generation approach (CG) for a variety of large

Table 1 Largest data set from TreeBASE.org

TB-ID	n
Tr69195	1838
Tr60915	2025
Tr57261	2387
Tr46272	2409
Tr73427	2632
Tr47159	4586
Tr48025	5743

Table 2 The problem instances with the largest number k of colors

n	k	p_n	p_c	TimeILP (s)	TimeCG (s)	Ratio ILP/CG
1838	934	25	50	631.6	67.5	9.35
1838	945	50	50	1368.4	70.3	19.47
1838	919	75	50	2034.1	75.4	26.99
2025	985	25	50	894.2	71.4	12.53
2025	1007	50	50	1702.1	83.3	20.44
2025	1008	75	50	2194.5	89.8	24.43
2387	1221	25	50	2627.2	139.4	18.85
2387	1191	50	50	3210.7	148.6	21.60
2387	1214	75	50	4996.5	170.3	29.34
2409	1145	25	50	2015.9	123.2	16.36
2409	1231	50	50	4116.8	164.1	25.08
2409	1177	75	50	5914.4	161.3	36.68
2632	1314	25	50	3793.6	174.4	21.75
2632	1300	50	50	6861.2	188.0	36.49
2632	1309	75	50	>6 h	222.8	>96.95
4586	2275	25	50	>6 h	875.9	OOM in Chopra et al. [3]
4586	2248	50	50	>6 h	901.1	OOM in Chopra et al. [3]
4586	2273	75	50	OOM	1003.4	OOM in Chopra et al. [3]
5743	2854	25	50	OOM	1997.9	OOM in Chopra et al. [3]
5743	2895	50	50	OOM	2138.9	OOM in Chopra et al. [3]
5743	2867	75	50	OOM	2415.2	OOM in Chopra et al. [3]

problems with k increasing from 0.5% to 5% to 50%. Observe that the ILP model did much better than column generation for $k = 0.5\%$ or 5% but column generation did much better for $k = 50\%$.

To better understand why the column generation approach does well for large k but poorly for small or medium k, we focus on the results in Table 4. Table 4 records the number of column generation iterations (Iter) and the average time per iteration (RM) for solving the restricted master problem when using the column generation approach. For problems that are not solved to optimality, Table 4 contains the results for 6 h of running time. Observe that for large values of k, the time per iteration (RM) when using column generation (shown as bold in Table 4) is much shorter than for small or medium values of k. It is not so much the number of iterations but the time per iteration that seems to be much shorter for large values of k. RM does not include the computational time to generate a column. This implies that poor performance is inherent to the column generation approach for small and medium values of k.

Our results indicate that the value of k has a significant impact on the sparsity of the constraint matrix A, which in turn has a significant impact on the time per iteration. (See Figs. 4 and 5.) When k/n is large, it is reasonable to expect that each tree introduced as a column in A will have few non-zero entries, i.e., the corresponding column of A will be sparse. To test this we use results from all

Table 3 Time (s) of the ILP model (ILP) vs the column generation (CG) where TLE stands for Time Limit Exceeded (>6 h) and OOM stands for Out-Of-Memory

$\|V\|$	k	ILP	CG	$\|V\|$	k	ILP	CG	$\|V\|$	k	ILP	CG
$(p_n, p_c) = (25\%, 0.5\%)$				$(p_n, p_c) = (25\%, 5\%)$				$(p_n, p_c) = (25\%, 50\%)$			
1838	11	1.1	TLE	1838	114	56.6	TLE	1838	934	631.6	67.5
2025	16	2.3	TLE	2025	91	48.7	TLE	2025	985	894.2	71.4
2387	16	6.0	223.5	2387	129	120.7	TLE	2387	1221	2627.2	139.4
2409	7	1.8	TLE	2409	137	150.3	TLE	2409	1145	2015.9	123.2
2632	13	3.6	TLE	2632	126	131.5	TLE	2632	1314	3793.6	174.4
4586	24	41.0	TLE	4586	236	1024.3	TLE	4586	2275	TLE	875.9
5743	28	104.9	TLE	5743	312	2990.1	TLE	5743	2854	OOM	1997.9
$(p_n, p_c) = (50\%, 0.5\%)$				$(p_n, p_c) = (50\%, 5\%)$				$(p_n, p_c) = (50\%, 50\%)$			
1838	8	3.4	TLE	1838	87	72.5	TLE	1838	945	1368.4	70.3
2025	6	2.1	TLE	2025	101	108.3	TLE	2025	1007	1702.1	83.3
2387	22	35.1	TLE	2387	128	256.0	TLE	2387	1191	3210.7	148.6
2409	13	10.1	TLE	2409	112	214.0	TLE	2409	1231	4116.8	164.1
2632	18	36.4	TLE	2632	133	232.4	TLE	2632	1300	6861.2	188.0
4586	16	125.8	TLE	4586	228	12944.2	TLE	4586	2248	TLE	901.1
5743	29	2153.0	TLE	5743	316	TLE	TLE	5743	2895	OOM	2138.9
$(p_n, p_c) = (75\%, 0.5\%)$				$(p_n, p_c) = (75\%, 5\%)$				$(p_n, p_c) = (75\%, 50\%)$			
1838	8	4.3	TLE	1838	106	452.2	TLE	1838	919	2034.1	75.4
2025	12	15.0	TLE	2025	110	728.5	TLE	2025	1008	2194.5	89.8
2387	13	43.8	TLE	2387	122	1629.1	TLE	2387	1214	4996.5	170.3
2409	12	23.9	TLE	2409	115	1764.9	TLE	2409	1177	5914.4	161.3
2632	14	65.6	TLE	2632	136	4117.6	TLE	2632	1309	TLE	222.8
4586	22	814.2	TLE	4586	228	TLE	TLE	4586	2273	OOM	1003.4
5743	27	2668.4	TLE	5743	302	TLE	TLE	5743	2867	OOM	2415.2

problems solved by us and run a regression between k/n as the independent variable and the average fraction of non-zeroes in a column of A. From the data in Fig. 4, we find the relationship to be as follows:

$$\text{Average fraction of non-zeroes/column} = 0.33 - 0.675 \times (k/n)$$

The R^2 for this regression is 0.76 and the p-value for the coefficient of k/n is 2.19×10^{-20}. Both indicate that the relationship is statistically significant and the average fraction of non-zeroes in a column of A decreases as k/n increases. We then run a regression between the average fraction of non-zeroes in A as the independent variable and average time per iteration as the dependent variable. Our hypothesis is that the denser the columns of A become, the longer it takes per iteration. This is validated by our regression from the data in Fig. 5 which find the relationship to be

$$\text{Average time per iteration} = 0.17 + 3.20 \times \text{Average fraction of non-zeroes/column}$$

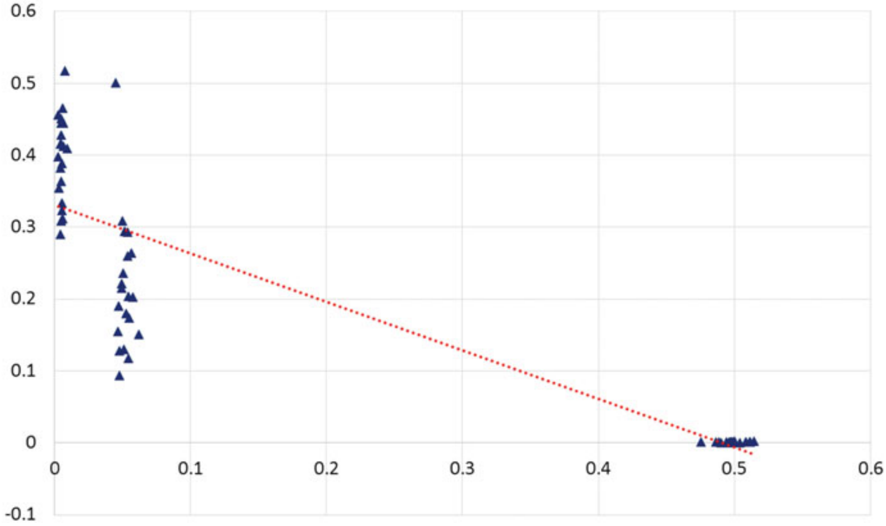


Fig. 4 (X-axis: k/n, Y-axis: Fraction of non-zeroes per column) Regression shows a significant negative association between the two. As k/n increases, the fraction of non-zeroes/column decreases

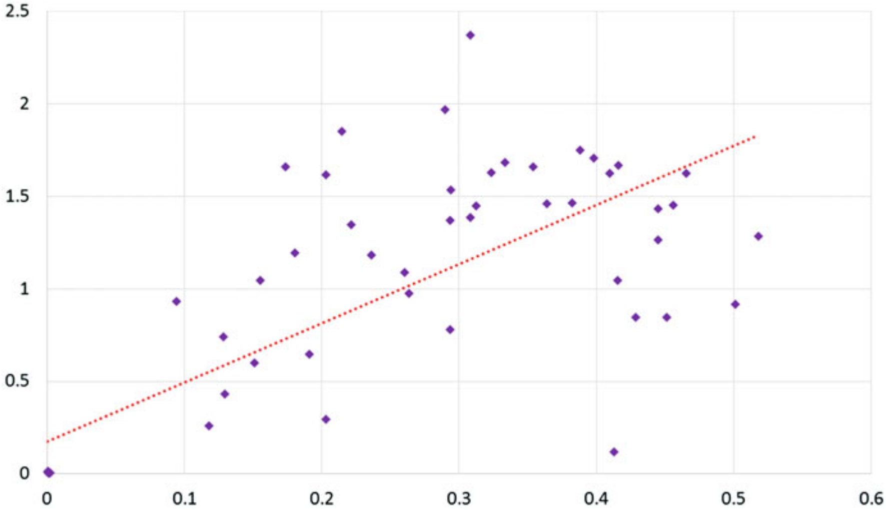


Fig. 5 (X-axis: Fraction of non-zeroes per column, Y-axis: average time per iteration) Regression shows a significant positive association between the two. As the fraction of non-zeroes/column increases, the average time per iteration also increases

Table 4 The number of iterations (Iter) and the average time of restricted master (RM) per iteration

$\|V\|$	k	Iter	RM	$\|V\|$	k	Iter	RM	$\|V\|$	k	Iter	RM
$(p_n, p_c) = (25\%, 0.5\%)$				$(p_n, p_c) = (25\%, 5\%)$				$(p_n, p_c) = (25\%, 50\%)$			
1838	11	3458	6.257	1838	114	10,241	2.107	1838	934	**2928**	**0.004**
2025	16	3497	6.182	2025	91	4953	4.362	2025	985	**2969**	**0.004**
2387	16	**1878**	**0.117**	2387	129	3083	7.009	2387	1221	**3835**	**0.005**
2409	7	3817	5.661	2409	137	3434	6.288	2409	1145	**3659**	**0.005**
2632	13	4025	5.372	2632	126	9378	2.301	2632	1314	**4092**	**0.005**
4586	24	4078	5.300	4586	236	5185	4.157	4586	2275	**6465**	**0.009**
5743	28	4743	4.554	5743	312	3365	6.407	5743	2854	**9064**	**0.012**
$(p_n, p_c) = (50\%, 0.5\%)$				$(p_n, p_c) = (50\%, 5\%)$				$(p_n, p_c) = (50\%, 50\%)$			
1838	8	2653	8.146	1838	87	11,021	1.959	1838	945	**3096**	**0.004**
2025	6	3486	6.199	2025	101	3320	6.510	2025	1007	**3223**	**0.005**
2387	22	2130	10.147	2387	128	5398	4.000	2387	1191	**4091**	**0.005**
2409	13	2972	7.288	2409	112	8513	2.535	2409	1231	**4199**	**0.005**
2632	18	2847	7.592	2632	133	6869	3.142	2632	1300	**4411**	**0.006**
4586	16	3556	6.103	4586	228	4662	4.626	4586	2248	**6730**	**0.010**
5743	29	3806	5.677	5743	316	2100	10.287	5743	2895	**9691**	**0.013**
$(p_n, p_c) = (75\%, 0.5\%)$				$(p_n, p_c) = (75\%, 5\%)$				$(p_n, p_c) = (75\%, 50\%)$			
1838	8	3443	6.276	1838	106	17,361	1.243	1838	919	**3464**	**0.004**
2025	12	4032	5.357	2025	110	15,174	1.422	2025	1008	**3460**	**0.005**
2387	13	2554	8.459	2387	122	7880	2.740	2387	1214	**4579**	**0.006**
2409	12	3448	6.268	2409	115	13,035	1.655	2409	1177	**4529**	**0.005**
2632	14	2858	7.561	2632	136	4656	4.636	2632	1309	**4933**	**0.006**
4586	22	2867	7.535	4586	228	5446	3.957	4586	2273	**7101**	**0.010**
5743	27	2685	8.056	5743	302	2629	8.202	5743	2867	**11,120**	**0.013**

The R^2 for this regression is 0.61 and the p-value for the coefficient of Average fraction of non-zeroes/column is 3.82×10^{-14}. Both indicate that the relationship is statistically significant and the average time per iteration increases as the columns of A become denser. Thus, for large values of k (relative to n), column generation is an effective approach because the time per iteration is small. The time per iteration is small because each tree introduced through column generation has few edges resulting in sparse columns of A with few non-zero entries.

Fortunately, as our results in Table 3 indicate, the strengths of the ILP approach and the column generation approach complement each other. Thus, a combination of the two approaches can be used depending upon the number of colors k relative to the number of nodes in the tree.

5 Conclusion and Future Work

In this paper we have modified the Johnson–Mehrotra–Nemhauser [4] approach using column generation to solve the convex recoloring problem on a tree. The column generation approach does very well for large values of k but performs poorly (relative to the ILP approach of Chopra et al. [3]) for small or medium values of k. It seems reasonable to change the approach used based on the value of k. More computational experiments would allow the identification of a suitable threshold for k based on which the approach used can be changed.

We may improve the implementation of our column generation approach in the following three points:

1. Our column generation approach began with the identity matrix as the coefficient matrix of the initial restricted master problem.
2. Each restricted master problem is added by a generated column and the number of columns grows proportional to the number of iterations.
3. Each restricted master problem is solved from scratch.

To improve Point 1, we need to develop a heuristic to identify a near-optimal solution. It will allow us to speed up the column generation approach starting with the initial coefficient matrix including near optimal solution. To improve Points 2 and 3, we may perform the warm start and the revised simplex. In addition, we see that the integer solutions to the restricted master problem are highly degenerate. We may speed up by changing the rule of choosing the entering column.

References

1. Campêlo, M., Huiban, C.G., Sampaio, R.M., Wakabayashi, Y.: On the complexity of solving or approximating convex recoloring problems. In: Proceedings of the 19th International Conference on Computing and Combinatorics. Lecture Notes in Computer Science, vol. 7936, pp. 614–625 (2013)
2. Campêlo, M., Freire, A.S., Lima, K.R., Moura, P., Wakabayashi, Y.: The convex recoloring problem: polyhedra, facets and computational experiments. Math. Program. **156**, 303–330 (2016)
3. Chopra, S., Filipecki, B., Lee, K., Ryu, M., Shim, S., Van Vyve, M.: The convex recoloring problem on a tree. Math. Program. (2016). doi:10.1007/s10107-016-1050-2. Online
4. Johnson, E.L., Mehrotra, A., Nemhauser, G.L.: Min-cut clustering. Math. Program. **62** 133–151 (1993)
5. Kanj, I.A., Kratsch, D.: Convex recoloring revisited: complexity and exact algorithms. In: Proceedings of the 15th Annual International Conference on Computing and Combinatorics (COCOON 2009). Lecture Notes in Computer Science, vol. 5609, pp. 388–397 (2009)
6. Matsen, F.A., Gallagher, A.: Reconciling taxonomy and phylogenetic inference: formalism and algorithms for describing discord and inferring taxonomic roots. Algorithms Mol. Biol. **7**(8), 1–12 (2012)
7. McDonald, D., Price, M., Goodrich, J., Nawrocki, E., DeSantis, T., Probst, A., Andersen, G., Knight, R., Hugenholtz, P.: An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. ISME J. **6**, 610–618 (2012)

8. Moran, S., Snir, S.: Convex recolorings of strings and trees: definitions, hardness results and algorithms. In: Proceedings WADS 2005: 9th International Workshop on Algorithms and Data Structures, pp. 218–232 (2005)
9. Moran, S., Snir, S.: Convex recolorings of strings and trees: definitions, hardness results and algorithms. J. Comput. Syst. Sci. **74**, 850–869 (2008)
10. Moran, S., Snir, S., Sung, W.K.: Partial convex recolorings of trees and galled networks: tight upper and lower bounds. ACM Trans. Algorithms **7**, 42 (2011)

A Variational Inequality Formulation of a Migration Model with Random Data

Baasansuren Jadamba and Fabio Raciti

Abstract In this note, we consider a simple model of populations distribution based on utility functions theory. The novelty of our approach is the use of a recent theory of random variational inequalities in refining a previous model by allowing random fluctuations in the data of the problem. We first present the random equilibrium conditions and prove their equivalence to a parametric random variational inequality. Then, we provide a formulation of the problem in a Lebesgue space with a probability measure. Finally, we work out a simple example, which can be solved exactly and allows us to test an approximation procedure.

Keywords Migration modeling • Equilibrium theory • Uncertainty modeling

1 Introduction

Migration is a ubiquitous phenomenon which has characterized the history of humanity from its origins. The reader interested in having a general view of the history and development of mathematical models of migration can refer to [1].

The purpose of our work is to incorporate uncertain data in an equilibrium model of migration based on variational inequalities. The variational inequality approach focuses on the concept of *equilibrium distribution* and models the attractiveness of each location using a corresponding utility function (see, e.g., [2]). We use the tools put forward in [3, 4] and further developed in [5–7] to investigate the random version of a deterministic model treated in Nagurney [9].

B. Jadamba (✉)
School of Mathematical Sciences, Rochester Institute of Technology, 85 Lomb Memorial Drive, Rochester, NY 14623, USA
e-mail: bxjsma@rit.edu

F. Raciti
Dipartimento di Matematica e Informatica, University of Catania, V.le A. Doria, 6, Catania 95125, Italy
e-mail: fraciti@dmi.unict.it

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_4

This paper is structured as follows. In the following section, we present the parametric variational inequality model of a population distribution with random data. In Sect. 3, we generate the integral formulation of the parametric variational inequality and specify the operator structure. In the last section, we reformulate the variational inequality in order to apply an approximation procedure and work out a simple example.

2 A Simple Model of a Population Distribution

The purpose of this model is to assign to each geographic location the ideal (equilibrium) population, without describing the details of migration flows and travel costs. The locations are modeled as k nodes of a network and constitute a closed economy. The *utility function* of each node i, u_i, quantifies the attractiveness of location i and is a function of all populations. The utility function at location k is nonincreasing with respect to p_k. This means that when the population increases at a location, that location becomes less appealing. Moreover, the general economic situation at each node can fluctuate according to some random events, hence our utility functions must incorporate this randomness. Thus, denote with $p \in \mathbb{R}^k$ the population vector, with $(\Omega, \mathcal{A}, P)$ a probability space, where Ω is the sample space and $\mathcal{A}$ is a sigma-algebra on Ω. The utility function vector is $u : \Omega \times \mathbb{R}^k \to \mathbb{R}^k, u(\omega, p) = (u_1(\omega, p), \dots, u_k(\omega, p))$. We assume that u is a Carathéodory function, i.e. $u(\omega, \cdot)$ is continuous and $u(\cdot, p)$ is measurable, with respect to the σ-algebra $\mathcal{A}$ on Ω.

If we assume no births and no deaths, the population distribution must sum up to a fixed quantity D. To take into account fluctuations due to births or deaths (possibly due to epidemic diseases), we can consider D as a random variable, $D : \Omega \to \mathbb{R}$. Thus, for all $\omega \in \Omega$, we can define the set of the feasible populations as the random simplex:

$$K(\omega) = \{p \in \mathbb{R}^k : p_i \geq 0, \ \forall i, \ \sum_{i=1}^{k} p_i = D(\omega)\}.$$

If we assume that migrants are rational and that migration will continue until no individual has interest to move, an equilibrium concept can be defined as follows.

Definition 1 A random vector $p^*(\omega) \in K(\omega)$ is called an equilibrium if for all $i = 1, \dots, k$ and for P-almost every $\omega \in \Omega$:

$$u_i(\omega, p^*(\omega)) \begin{cases} = \lambda(\omega) & \text{if } \ p_i^*(\omega) > 0 \\ \leq \lambda(\omega) & \text{if } \ p_i^*(\omega) = 0. \end{cases} \tag{1}$$

A useful interpretation of this definition is given by the following lemma whose proof we omit here.

Lemma 1 *Condition* (1) *is equivalent to:* $p^*(\omega) \in K(\omega)$ *and for P-almost every* $\omega \in \Omega$ *and* $\forall r, s \in 1, \dots, k$*:*

$$u_r(\omega, p^*(\omega)) < u_s(\omega, p^*(\omega)) \Longrightarrow p_r^*(\omega) = 0. \tag{2}$$

The equilibrium conditions previously introduced can be reformulated as a variational inequality, as shown in the following theorem.

Theorem 1 *Equilibrium conditions as in Definition* (1) *are equivalent to the variational inequality problem: Find* $p^*(\omega) \in K(\omega)$ *such that P-a.s. and* $\forall p \in K(\omega)$*:*

$$-u(\omega, p^*(\omega))^\top (p - p^*(\omega)) \geq 0. \tag{3}$$

Proof Fix $\omega \in \Omega$ and assume that (1) holds true, and let $I = \{1, \dots, k\}$. Moreover, let: $A = \{i \in I : u_i(\omega, p^*(\omega)) = \lambda(\omega)\}$, $B = \{i \in I : u_i(\omega, p^*(\omega)) < \lambda(\omega)\}$, where we did not specify the dependence of A and B on ω to keep notation simple. For all $p \in K(\omega)$, we have:

$$\begin{aligned}
&-\sum_{i=1}^{k} u_i(\omega, p^*(\omega))(p_i - p_i^*(\omega)) \\
&= -\sum_{i \in A} u_i(\omega, p^*(\omega))(p_i - p_i^*(\omega)) - \sum_{i \in B} u_i(\omega, p^*(\omega))(p_i - p_i^*(\omega)) \\
&\geq -\sum_{i \in A} \lambda(\omega)(p_i - p_i^*(\omega)) - \sum_{i \in B} \lambda(\omega)(p_i - p_i^*(\omega)) = \lambda(\omega)(D(\omega) - D(\omega)) = 0.
\end{aligned}$$

Hence (1) $\Rightarrow$ (3).

Fix ω and assume now that (3) holds. If we had that $\exists r, s \in I$ with $u_r(\omega, p^*(\omega)) < u_s(\omega, p^*(\omega))$ and $p_r(\omega) > 0$, consider the following feasible population vector:

$$p_i = \begin{cases} p_i^*(\omega), & i \neq r, s \\ 0, & i = r \\ p_r^*(\omega) + p_s^*(\omega), & i = s. \end{cases}$$

We thus get:

$$\begin{aligned}
&-\sum_{i=1}^{k} u_i(\omega, p^*(\omega))(p_i - p_i^*(\omega)) \\
&= -u_r(\omega, p^*(\omega))(-p_r^*(\omega) - u_s(\omega, p^*(\omega))(p_r^*(\omega)) + p_s^*(\omega) - p_s^*(\omega)) \\
&= p_r^*(\omega)(u_r(\omega, p^*(\omega)) - u_s(\omega, p^*(\omega))) < 0,
\end{aligned}$$

which contradicts (3). □

Remark 1 The existence of solutions of the previous variational inequality (3), for each $\omega \in \Omega$, is ensured by the standard theory, since $u(\omega, \cdot)$ is continuous and $K(\omega) \subset \mathbb{R}^k$ is compact (see [8], Theorem 3.1).

3 Operator Structure and Integral Formulation

We assume the following form of the utility function:

$$-u(\omega, p) = S(\omega)G(p) + H(p) - b - R(\omega)c \tag{4}$$

where: $S \in L^\infty(\Omega, P)$, such that $0 < \underline{s} \leq S(\omega) \leq \bar{s}$, P-a.s., $R \in L^2(\Omega, P)$, $b, c \in \mathbb{R}^k$, $G, H \in \mathbb{R}^{k\times k}$. The modeling is done by allowing that randomness is embodied in the utility function as follows. We sum a completely deterministic part which is represented by H and b, and another part $S(\omega)G(p)$ and $R(\omega)c$ where the random functions S and R act as random perturbations, or “modulations.” To apply the theory developed in [3, 4], we require the following strong and uniform monotonicity assumption on $-u$:

$$\exists \alpha > 0 : (\,u(\omega, q) - u(\omega, p)\,)^\top (p - q) \geq \alpha \|p - q\|^2,\ \forall p, q \in \mathbb{R}^k,\ P - \text{a.s.} \tag{5}$$

We recall that the operator $N_u : p(\omega) \mapsto u(\omega, p(\omega))$ is called the superposition (or Nemitsky) operator and, under our assumptions, it is easy to show that it maps $L^2(\Omega, P, \mathbb{R}^k)$ in $L^2(\Omega, P, \mathbb{R}^k)$. The family of closed and convex sets $K(\omega)$, $\omega \in \Omega$, generates a subset of L^2 as follows:

$$K^P = \{\, V \in L^2(\Omega, P, \mathbb{R}^k) : V(\omega) \geq 0,\ V_1(\omega) + \ldots + V_k(\omega) = D(\omega),\ \text{P-a.s.}\,\}.$$

We can now consider the integral variational inequality: Find $V^* \in K^P$ s.t. $\forall V \in K^P$:

$$\int_\Omega [S(\omega)G(V^*(\omega)) + H(V^*(\omega)]^\top (V(\omega) - V^*(\omega))\, dP(\omega)$$
$$\geq \int_\Omega (b + R(\omega)c)^\top (V(\omega) - V^*(\omega))\, dP(\omega) \tag{6}$$

The existence of a solution of (6) is ensured by the following theorem.

Theorem 2 *Let $u(\omega, p)$ be a Carathéodory function whose structure is given by* (4) *and let the uniform monotonicity condition* (5) *be satisfied. Then, variational inequality* (6) *admits a unique solution.*

Proof Let us observe that K^P is a closed and convex subset of $L^2(\Omega, P, \mathbb{R}^k)$. The set K^P is also bounded, hence weakly compact. Moreover, the Nemitsky operator $N_u : L^2(\Omega, P, \mathbb{R}^n) \to L^2(\Omega, P, \mathbb{R}^k)$ is continuous. Finally, the strong uniform monotonicity of u implies the strong monotonicity of N_u. Therefore, we can apply the Lions-Stampacchia theorem (see, e.g., [8], Theorem 2.1) and obtain the existence and uniqueness of the solution of (6). □

4 Image Space Formulation and an Example

According to the general theory described in [3, 4], to carry out an approximation procedure we have to reformulate (6) in the image space of the random variables involved. Let $s := S(\omega)$, $r := R(\omega)$, $t := D(\omega)$ and consider the vector $(s, r, t) \in \mathbb{R}^3$ along with the probability $\mathbb{P}$ induced from P in $\mathbb{R}^3$. We assume that the random variables r, s, t are independent with densities $\varphi_S(s), \varphi_R(r), \varphi_D(t)$. Hence, $d\mathbb{P}(r, s, t) = \varphi_S(s)\varphi_R(r)\varphi_D(t) ds\, dr\, dt$. Consider now the set:

$$K_{\mathbb{P}} = \{\, v \in L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k) : \; v(s, r, t) \geq 0, \; \sum_{i=1}^{k} v_k(s, r, t) = t, \; \mathbb{P} - \text{a.s.}\}.$$

We are now in a position to formulate the variational inequality in the image space:

Find $\hat{u} \in K_{\mathbb{P}}$ such that $\forall v \in K_{\mathbb{P}}$

$$\int\limits_{\underline{s}}^{\overline{s}} \int\limits_{0}^{\infty} \int\limits_{0}^{\infty} [sG(\hat{u}(s, r, t)) + H(\hat{u}(s, r, t))]^{\top} (v(s, r, t) - \hat{u}(s, r, t)) \varphi_S(s)\varphi_R(r)\varphi_D(t) ds\, dr\, dt$$

$$\geq \int_{\underline{s}}^{\overline{s}} \int_0^{\infty} \int_0^{\infty} (b + rc)^{\top} (v(s, r, t) - \hat{u}(s, r, t)) \varphi_S(s)\varphi_R(r)\varphi_D(t) ds\, dr\, dt. \quad (7)$$

Variational inequality (7) is now ready to be approximated using the procedure described in [3, 4] which also allows computing the approximate expectations of the solution. For the reader's convenience we recall the approximation procedure.

Let us start with a discretization of the space $X := L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k)$ and introduce a sequence $\{\pi_n\}_n$ of the support $\Gamma := [0, \infty) \times [\underline{s}, \overline{s}] \times \mathbb{R}_+$ of the probability measure $\mathbb{P}$ induced by the random variables R, S, D. Thus, let $\pi_n := (\pi_n^R, \pi_n^S, \pi_n^D)$, where

$$\pi_n^R := (r_n^0, \ldots, r_n^{N_n^R}), \; \pi_n^S := (s_n^0, \ldots, s_n^{N_n^S}), \; \pi_n^D := (t_n^0, \ldots, t_n^{N_n^D})$$

$$0 = r_n^0 < r_n^1 < \ldots < r_n^{N_n^R} = n$$

$$\underline{s} = s_n^0 < s_n^1 < \ldots < s_n^{N_n^S} = \overline{s}$$

$$0 = t_n^0 < t_n^1 < \ldots < t_n^{N_n^D} = n$$

$$|\pi_n^R| := \max\{r_n^j - r_n^{j-1} : j = 1, \dots, n^{N_n^R}\} \to 0 \ (n \to \infty)$$

$$|\pi_n^S| := \max\{s_n^l - s_n^{l-1} : l = 1, \dots, n^{N_n^S}\} \to 0 \ (n \to \infty)$$

$$|\pi_n^D| := \max\{t_n^h - t_n^{h-1} : h = 1, \dots, n^{N_n^D}\} \to 0 \ (n \to \infty)\,.$$

These partitions give rise to the exhausting sequence Γ_n of subsets of Γ, where each Γ_n is given by the finite disjoint union of the intervals:

$$I_{jlh}^n := [r_n^{j-1}, r_n^j) \times [s_n^{l-1}, s_n^l) \times [t_n^{h-1}, t_n^h),$$

For each $n \in \mathbb{N}$, let us consider the space of the $\mathbb{R}^k$-valued simple functions ($k \in \mathbb{N}$) on Γ_n, extended by zero outside of Γ_n:

$$X_n^k := \{v_n : v_n(r,s,t) = \sum_j \sum_l \sum_h v_{jlh}^n \, \mathbf{1}_{I_{jlh}^n}(r,s,t), \ v_{jlh}^n \in \mathbb{R}^k\},$$

where $\mathbf{1}_I$ denotes the $\{0, 1\}$-valued characteristic function of a subset I.

To approximate an arbitrary function $w \in L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R})$ we employ the mean value-truncation operator μ_0^n associated with the partition π_n given by:

$$\mu_0^n w := \sum_j \sum_l \sum_h (\mu_{jlh}^n w)\, \mathbf{1}_{I_{jlh}^n}, \tag{8}$$

where

$$\mu_{jlh}^n w := \frac{1}{\mathbb{P}(I_{jlh}^n)} \int_{I_{jlh}^n} v(y)\, d\mathbb{P}(y) \ \text{ if } \mathbb{P}(I_{jlh}^n) > 0.$$

Analogously, for an L^2 vector function $v = (v_1, \dots, v_k)$, we define $\mu_0^n v := (\mu_0^n v_1, \dots, \mu_0^n v_k)$. The following lemma can be proven:

Lemma 2 *The linear operator* $\mu_0^n : L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k) \to L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k)$*is bounded with* $\|\mu_0^n\| = 1$ *and satisfies* $\mu_0^n \to 1$ *pointwise in* $L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k)$.
In order to construct approximation for

$$K_{\mathbb{P}} := \{v \in L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k) : v \geq 0, \ \sum_{i=1}^k v_i(r,s,t) = t, \mathbb{P} - \text{a.s.}\}$$

we introduce the orthogonal projector $q : (r,s,t) \in \mathbb{R}^3 \mapsto t \in \mathbb{R}$ and let, for each elementary quadrangle I_{jlh}^n,

$$\overline{q}_{jlh}^n = (\mu_{jlh}^n q) \in \mathbb{R}, \ \ (\mu_0^n q) = \sum_{jlh} \overline{q}_{jlh}^n \mathbf{1}_{I_{jlh}^n} \in X_n.$$

Thus, we arrive at the following sequence of convex, closed sets

$$K^n_{\mathbb{P}} := \{v \in X^k_n : v \geq 0,\ \mathcal{I} v^n_{jlh} \leq \overline{q}^n_{jlh},\ \ \forall j, l, h\}. \tag{9}$$

where $\mathcal{I}$ denotes the identity matrix of dimension k. Note that the sets $K^n_{\mathbb{P}}$ are of polyhedral type. Furthermore, in order to approximate the random variables R and S, we introduce

$$\rho_n = \sum_{j=1}^{N^R_n} r^{j-1}_n 1_{[r^{j-1}_n, r^j_n)} \in X_n,\ \ \sigma_n = \sum_{l=1}^{N^S_n} r^{l-1}_n 1_{[r^{l-1}_n, r^l_n)} \in X_n.$$

Thus, we can now consider, $\forall n \in \mathbb{N}$, the following substitute problem:

Find $\hat{u}_n \in K^n_{\mathbb{P}}$ such that, $\forall v_n \in K^n_{\mathbb{P}}$:

$$\begin{aligned}&\int_{\mathbb{R}^d} \{\sigma_n(y)[G\hat{u}_n(y)]^\top [v_n(y) - \hat{u}_n(y)] + [H\hat{u}_n(y)]^\top [v_n(y) - \hat{u}_n(y)]\} d\mathbb{P}(y) \\ &\geq \int_{\mathbb{R}^d} (b + \rho_n(y)c)^\top [v_n(y) - \hat{u}_n(y)] d\mathbb{P}(y)\,.\end{aligned} \tag{10}$$

Once the partition is fixed, the above integrals can be written as follows:

$$\begin{aligned}&\sum_{jlh} \int_{I^n_{jlh}} \{\sigma_n(y)[G\hat{u}_n(y)]^\top [v_n(y) - \hat{u}_n(y)] + [H\hat{u}_n(y)]^\top [v_n(y) - \hat{u}_n(y)]\} d\mathbb{P}(y) \\ &\geq \int_{I^n_{jlh}} (b + \rho_n(y)c)^\top [v_n(y) - \hat{u}_n(y)] d\mathbb{P}(y),\end{aligned}$$

which, according to the notation introduced previously reads as:

$$\begin{aligned}&\sum_{jlh} \{s^{l-1}_n [G\hat{u}^n_{jlh}(y)]^\top [v^n_{jlh}(y) - \hat{u}^n_{jlh}(y)] + [Hu^n_{jlh}(y)]^\top [v^n_{jlh}(y) - \hat{u}^n_{jlh}(y)]\,\}\, P(I^n_{jlh}) \\ &\geq \sum_{jlh} \{(b + r^{j-1}_n c)^\top [v^n_{jlh}(y) - \hat{u}^n_{jlh}(y)]\, P(I^n_{jlh})\end{aligned} .$$

We can choose a test function $v_n \in K^n_{\mathbb{P}}$ which is equal to $\hat{u}_n$ except for a given cell I^n_{jlh}, so that, for each n, the substitute problem (10) splits in a finite number of variational inequalities on $\mathbb{R}^k$:

$\forall n \in \mathbb{N}, \forall j, l, h,$ find $\hat{u}^n_{jlh} \in K^n_{jlh}$ such that $\forall v^n_{jlh} \in K^n_{jlh}$

$$[\tilde{G}^n_l \hat{u}^n_{jlh}]^T [v^n_{jlh} - \hat{u}^n_{jlh}] \geq [\tilde{c}^n_j][v^n_{jlh} - \hat{u}^n_{jlh}] \tag{11}$$

where

$$K^n_{jlh} := \{v^n_{jlh} \in \mathbb{R}^k_+ : \mathcal{I}\, v^n_{jlh} = \overline{q}^n_{jlh}\}.$$

$$\tilde{G}^n_l := s_n^{l-1} G + H,\ \tilde{c}^n_j = b + r_n^{j-1} c.$$

We can then reconstruct the step-function solution of (10) as:

$$\hat{u}_n = \sum_j \sum_l \sum_h \hat{u}^n_{jlh} 1_{I^n_{jlh}} \in X^k_n.$$

We can prove the following approximation result along the same lines as in [3, 4].

Theorem 3 *As $n \to \infty$, the solutions $\hat{u}_n$ of the substitute problems* (10) *converge strongly in $L^2(\mathbb{R}^3, \mathbb{P}, \mathbb{R}^k)$ to the unique solution of* (7).

The functions $\hat{u}_n$ can be used to compute approximate mean values of $\hat{u}$ as it is shown in the following example:

Example 1 We consider the case of 3 locations where both the utility functions and the total population are affected by random perturbations. More precisely, the utility functions are perturbed by the nonnegative random variable s while the total population is distributed according to the (nonnegative) random variable t:

$$-u_1(s,p) = 3sp_1 - 12,\ -u_2(s,p) = sp_1 + 2sp_2 - 6,\ -u_3(s,p) = sp_1 + sp_2 + 4sp_3 - 12$$

and

$$K(t) = \{p \in \mathbb{R}^3_+ : p_1 + p_2 + p_3 = t\}.$$

According to the notation of Sect. 3 we have

$$G = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 4 \end{bmatrix},\ -b = [-12, -6, -12]^\top,\ H = 0,\ c = [0, 0, 0]^\top.$$

The parametric variational inequality reads as: $\forall s, t$, find $p^*(s,t)$ such that, $\forall p \in K(s,t)$:

$$\begin{aligned} - u_1(s,t,p^*(s,t))(p_1 - p_1^*(s,t)) - u_2(s,t,p^*(s,t))(p_2 - p_2^*(s,t)) \\ - u_3(s,t,p^*(s,t))(p_3 - p_3^*(s,t)) \geq 0 \end{aligned} \quad (12)$$

It is possible to solve this variational inequality exactly. We use a non-iterative method (see [10]) to find the solution of (12) which is given by

$$\begin{cases} (\frac{4t}{9} + \frac{1}{s},\ \frac{4t}{9} - \frac{2}{s},\ \frac{t}{9} + \frac{1}{s}) & \text{if } s \geq \frac{9}{2t} \\ (\frac{2}{3}t, 0, \frac{1}{3}t) & \text{if } s \leq \frac{9}{2t}. \end{cases} \quad (13)$$

Table 1 Mean value of $p = (p_1, p_2, p_3)$

	$N_1 = N_2 = 100$	$N_1 = N_2 = 200$	$N_1 = N_2 = 400$	Exact
$\langle p_1 \rangle$	1.6222	1.6219	1.6218	1.62186
$\langle p_2 \rangle$	0.088679	0.089235	0.089513	0.089612
$\langle p_3 \rangle$	0.7891	0.78882	0.78868	0.788527

We can now compute its expected value for some particular distributions. Assume, for instance, that t is uniformly distributed in the interval $[2, 3]$ and s is uniformly distributed in $[\frac{3}{2}, \frac{9}{4}]$. Let $D := [2, 3] \times [\frac{3}{2}, \frac{9}{4}]$. We then compute

$$E[q_1] = \iint_D q_1(s,t) dP_s \, dP_t = 4 \log \frac{3}{2}, \; E[q_2] = 8 \ln \frac{2}{3} + \frac{10}{3}, \; E[q_3] = 4 \ln \frac{3}{2} - \frac{5}{6}.$$

We can now apply our discretization procedure and compare various approximations with the above exact expectations. We choose a discretization of the parameters domain $D = [2, 3] \times \left[\frac{3}{2}, \frac{9}{4}\right]$ using $N_1 \times N_2$ grid points and solve the problem for each pair $(t(i), s(j))$ using an extragradient method. Then, we evaluate the mean value of p by using its probability distribution functions. As we see from Table 1, the approximate mean value of $p = (p_1, p_2, p_3)$ is really close to the exact value that is computed from the analytical solution.

References

1. Aleshkovski, I., Iontsev, V.: Mathematical models of migration. In: Livchits, V.N., Tokarev, V.V. (eds.) Systems Analysis and Modeling of Integrated World System, vol. II, pp. 185–214. Eolss Publishers Co Ltd, Oxford (2009)
2. Beckman, M.: On the equilibrium distribution of population in space. Bull. Math. Biophys. **19**, 81–89 (1957)
3. Gwinner, J., Raciti, F.: On a class of random variational inequalities on random sets. Numer. Funct. Anal. Optim. **27**(5–6), 619–636 (2006)
4. Gwinner, J., Raciti, F.: Some equilibrium problems under uncertainty and random variational inequalities. Ann. Oper. Res. **200**, 299–319 (2012). doi:10.1007/s10479-012-1109-2
5. Faraci, F., Jadamba, B., Raciti, F.: On stochastic variational inequalities with mean value constraints. J. Optim. Theory Appl. **171**(2), 675–693 (2016)
6. Jadamba, B., Khan, A.A., Raciti, F.: Regularization of stochastic variational inequalities and a comparison of an L_p and a sample-path approach. Nonlinear Anal. Ser. A Theory Methods **94**, 65–83 (2014)
7. Jadamba, B., Raciti, F.: On the modelling of some environmental games with uncertain data. J. Optim. Theory Appl. **167**(3), 959–968 (2015)
8. Kinderlehrer, D., Stampacchia, G.: An Introduction to Variational Inequalities and Their Applications. Academic Press, New York (1980)
9. Nagurney, A.: Migration equilibrium and variational inequalities. Econ. Lett. **31**, 109–112 (1989)
10. Raciti, F., Falsaperla, P.: Improved non iterative algorithm for the calculation of the equilibrium in the traffic network problem. J. Optim. Theory Appl. **133**, 401–411 (2007)

Identification in Mixed Variational Problems by Adjoint Methods with Applications

M. Cho, B. Jadamba, A. A. Khan, A. A. Oberai, and M. Sama

Abstract This work develops a fast and reliable computational framework for the inverse problem of identifying variable parameters in abstract mixed variational problems. One of the main contributions of this work is a thorough derivation of efficient computation schemes for the evaluation of the gradient and the Hessian of the output least-squares (OLS) functional. The derivation of all the formulas is given in continuous as well as discrete setting. Detailed numerical results for different classes of problems are presented.

Keywords Inverse problems • Elasticity imaging • Stoke's equation • Fourth-order boundary value problem • Regularization • Output least-squares • First-order adjoint method • Second-order adjoint method

1 Introduction

The primary objective of this work is to study the inverse problem of parameter identification in mixed variational problems by employing the output least-squares (OLS) formulation. The main drawback of using an OLS-based approach for inverse problems is the evaluation of the derivatives of the OLS objective which rely on the derivatives of the parameter-to-solution map and are computationally

M. Cho • B. Jadamba • A.A. Khan (✉)
Center for Applied and Computational Mathematics, School of Mathematical Sciences,
Rochester Institute of Technology, Rochester, NY 14623, USA
e-mail: mxcsma1@rit.edu; bxjsma@rit.edu; aaksma@rit.edu

A.A. Oberai
Scientific Research Computation Center (SCOREC), Rensselaer Polytechnic Institute,
Troy, NY 12180, USA
e-mail: oberaa@rpi.edu

M. Sama
Departamento de Matemática Aplicada, Universidad Nacional de Educación a Distancia,
Calle Juan del Rosal, 12, 28040 Madrid, Spain
e-mail: msama@ind.uned.es

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_5

expensive to evaluate. In the literature, adjoint methods have been used for some inverse problems for reducing this computational cost of computing the derivatives of the OLS functional. In this work, we derive new first-order and second-order adjoint methods for the inverse problem of parameter identification in general mixed variational problems. We apply our results to three different classes of problems. To be specific, we identify variables parameters in nearly incompressible elasticity, nearly incompressible Stoke's equations, and fourth-order boundary value problems.

The contents of this paper are organized in five sections. In Sect. 2, we introduce the inverse problem in an abstract setting and give a general existence result for the corresponding OLS formulation. In Sect. 3, we present a thorough derivation of the first-order adjoint approach for the computation of the first-order derivative of the OLS functional. We also present a second-order adjoint approach and one of its analogues for the evaluation of the second-order derivative of the OLS functional. Section 4 presents a detailed discretization procedure and gives schemes for the gradient and the Hessian computation. Numerical examples are given in Sect. 5.

2 Parameter Identification

Let V and Q be Hilbert spaces and let B be a Banach space. Let A be a nonempty, closed, and convex subset of B. Let $a : B \times V \times V \to \mathbb{R}$ be a trilinear map which is symmetric with respect to the second and the third arguments. Let $b : V \times Q \to \mathbb{R}$ be a bilinear map, let $c : Q \times Q \to \mathbb{R}$ be a symmetric bilinear map, and let $m : V \to \mathbb{R}$ be a linear and continuous map. We assume that there are strictly positive constants $\kappa_1, \kappa_2, \varsigma_1, \varsigma_2$, and κ_0 such that for every $\ell \in A$, $p, q \in Q$, and $\bar{u}, \bar{v} \in V$, we have

$$a(\ell, \bar{v}, \bar{v}) \geq \kappa_1 \|\bar{v}\|^2, \tag{1a}$$

$$|a(\ell, \bar{u}, \bar{v})| \leq \kappa_2 \|\ell\| \|\bar{u}\| \|\bar{v}\|, \tag{1b}$$

$$c(q, q) \geq \varsigma_1 \|q\|^2, \tag{1c}$$

$$|c(p, q)| \leq \varsigma_2 \|p\| \|q\|, \tag{1d}$$

$$|b(\bar{v}, q)| \leq \kappa_0 \|\bar{v}\| \|q\|. \tag{1e}$$

Given $\ell \in A$, the mixed variational problem seeks $(\bar{u}, p) \in V \times Q$ such that

$$a(\ell, \bar{u}, \bar{v}) + b(\bar{v}, p) = m(\bar{v}), \quad \text{for every } \bar{v} \in V, \tag{2a}$$

$$b(\bar{u}, q) - c(p, q) = 0, \quad \text{for every } q \in Q. \tag{2b}$$

In view of the coercivity and the continuity of $a(\cdot, \cdot, \cdot)$ and $c(\cdot, \cdot)$, the Lax-Milgram lemma ensures that for every $\ell \in A$, there exists a unique $u(\ell) =$

$(\bar{u}(\ell), p(\ell)) \in W := V \times Q$ satisfying (2). That is, for every $\ell \in A$, the parameter-to-solution map $\ell \to (\bar{u}(\ell), p(\ell))$ is well-defined and single-valued.

We are interested in the inverse problem of identifying $\ell \in A$ for which a solution $(\bar{u}, p)$ of (2) is closest, in some norm, to a given measurement $(\bar{z}, \hat{z})$ of $(\bar{u}, p)$. To study this inverse problem in an optimization framework, we introduce the following output least-squares (OLS, for short) functional

$$J(\ell) := \frac{1}{2}\,\|u(\ell) - z\|_W^2 = \frac{1}{2}\,\|\bar{u}(\ell) - \bar{z}\|_V^2 + \frac{1}{2}\,\|p(\ell) - \hat{z}\|_Q^2\,, \tag{3}$$

where $z = (\bar{z}, \hat{z})$ is the measured data and $u(\ell) = (\bar{u}(\ell), p(\ell))$ solves (2).

Some other formulations for this inverse problem are given in [2, 5, 8–10].

Due to the known ill-posedness of inverse problems, we shall consider the following regularized optimization problem: Find $\ell \in A$ by solving

$$\min_{\ell \in A} J_\kappa(\ell) := \frac{1}{2}\,\|\bar{u}(\ell) - \bar{z}\|_V^2 + \frac{1}{2}\,\|p(\ell) - \hat{z}\|_Q^2 + \kappa R(\ell), \tag{4}$$

where, given a Hilbert space H, $R : H \to \mathbb{R}$ is a regularizer, $\kappa > 0$ is a regularization parameter, $u(\ell) := (\bar{u}(\ell), p(\ell))$ is the unique solution of (2) that corresponds to the coefficient ℓ, and $z = (\bar{z}, \hat{z})$ is the measured data. Throughout this work, for simplicity we assume that R is twice differentiable.

We have the following existence result:

Theorem 1 *Assume that the Hilbert space H is compactly embedded into the space B, $A \subset H$ is nonempty, closed, and convex, the map R is convex, lower-semicontinuous and there exists $\alpha > 0$ such that $R(\ell) \geq \alpha \|\ell\|_H^2$, for every $\ell \in A$. Then* (4) *has a nonempty solution set.*

Proof Since $J_\kappa(\ell) \geq 0$ for every $\ell \in A$, there exists a minimizing sequence $\{\ell_n\}$ in A such that $\lim_{n\to\infty} J_\kappa(\ell_n) = \inf\{J_\kappa(\ell) |\, \ell \in A\}$, confirming that $\{\ell_n\}$ remains bounded in H. Therefore, we can extract a subsequence converging weakly in H and due to the compact embedding of H in B, strongly converging in B. Keeping the same notation for subsequences as well, let ℓ_n converge to some $\hat{\ell} \in A$. For the corresponding $u_n = (\bar{u}_n, p_n)$, we have

$$a(\ell_n, \bar{u}_n, \bar{v}) + b(\bar{v}, p_n) = m(\bar{v}), \quad \text{for every } \bar{v} \in V,$$

$$b(\bar{u}_n, q) - c(p_n, q) = 0, \quad \text{for every } q \in Q.$$

The above mixed variational problem confirms that $\{u_n\}$ remain bounded in W and hence there is a subsequence converging weakly to some $\hat{u}$. By rearranging the terms in the above mixed variational problem, it can be shown that $\hat{u} = \hat{u}(\hat{\ell})$. Furthermore, using the imposed coercivity (see (1)), it follows that in fact $\{u_n\}$ converges to $\hat{u} = \hat{u}(\hat{\ell})$ strongly.

Finally, the continuity of the norm and lower-semicontinuity of R yield

$$\begin{aligned} J_\kappa(\hat{\ell}) &= \frac{1}{2}\|\hat{u}(\hat{\ell}) - z\|^2 + \kappa R(\hat{\ell}) \\ &\leq \lim_{n\to\infty} \frac{1}{2}\|u_n(\ell) - z\|^2 + \liminf_{n\to\infty} \kappa R(\ell_n) \\ &\leq \liminf_{n\to\infty} \left\{ \frac{1}{2}\|u_n(\ell) - z\|^2 + \kappa R(\ell_n) \right\} = \inf\{J_\kappa(\ell) \,:\, \ell \in A\}, \end{aligned}$$

confirming that $\hat{\ell}$ is a solution of (4). The proof is complete.

We conclude this section with the following derivative formula (see [10]):

Theorem 2 *For each ℓ in the interior of A, $u = (\bar{u}(\ell), p(\ell))$ is infinitely differentiable at ℓ. The first derivative $\delta u = (\delta\bar{u}, \delta p) = (D\bar{u}(\ell)\delta\ell, Dp(\ell)\delta\ell)$ is the unique solution of the mixed variational problem:*

$$a(\ell, \delta\bar{u}, \bar{v}) + b(\bar{v}, \delta p) = -a(\delta\ell, \bar{u}, \bar{v}), \quad \textit{for every } \bar{v} \in V, \tag{5a}$$

$$b(\delta\bar{u}, q) - c(\delta p, q) = 0, \quad \textit{for every } q \in Q. \tag{5b}$$

3 Derivative Formulae

We now give a first-order adjoint method for the computation of the first-order derivative of the regularized OLS, and two second-order adjoint methods for the computation of its second-order derivative. Some of the most recent developments of the first-order adjoint methods are given in [1, 7, 11–17, 19] whereas second-order adjoint methods can be found in [3, 4, 6, 18, 20].

3.1 First-order Adjoint Method

We recall that the regularized OLS functional is given by

$$J_\kappa(\ell) = \frac{1}{2}\,\|\bar{u}(\ell) - \bar{z}\|_V^2 + \frac{1}{2}\,\|p(\ell) - \hat{z}\|_Q^2 + \kappa R(\ell).$$

By the chain rule, the derivative of J_κ at $\ell \in A$ in any direction $\delta\ell$ is given by

$$DJ_\kappa(\ell)(\delta\ell) = \langle D\bar{u}(\ell)(\delta\ell), \bar{u}(\ell) - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p(\ell) - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell),$$

where $Du(\ell)(\delta\ell) = (D\bar{u}(\ell)(\delta\ell), Dp(\ell)(\delta\ell))$ is the derivative of the coefficient-to-solution map u and $DR(\ell)(\delta\ell)$ is the derivative of the regularizer R, both computed at ℓ in the direction $\delta\ell$.

For an arbitrary $v = (\bar{v}, q) \in W$, define the functional $L_\kappa : B \times W \to \mathbb{R}$ by

$$L_\kappa(\ell, v) = J_\kappa(\ell) + a(\ell, \bar{u}, \bar{v}) + b(\bar{v}, p) + b(\bar{u}, q) - c(p, q) - m(\bar{v}).$$

Since $u(\ell) = (\bar{u}(\ell), p(\ell))$ is the solution of (2), we have

$$L_\kappa(\ell, v) = J_\kappa(\ell), \text{ for every } v \in W,$$

and hence for every $v \in W$, the following identity holds for any direction $\delta\ell$:

$$\partial_\ell L_\kappa(\ell, v)\,(\delta\ell) = DJ_\kappa(\ell)\,(\delta\ell)\,. \tag{6}$$

The key idea of the adjoint method is to choose v in a way to avoid a direct computation of $\delta u = Du(\ell)(\delta\ell)$. To get an insight into such a choice for v, we use the chain rule to obtain

$$\begin{aligned}\partial_\ell L_\kappa(\ell, v)\,(\delta\ell) = {} & \langle D\bar{u}(\ell)(\delta\ell), \bar{u} - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell) \\ & + a(\delta\ell, \bar{u}, \bar{v}) + a(\ell, D\bar{u}(\ell)(\delta\ell), \bar{v}) + b(\bar{v}, Dp(\ell)(\delta\ell)) \\ & + b(D\bar{u}(\ell)(\delta\ell), q) - c(Dp(\ell)(\delta\ell), q).\end{aligned} \tag{7}$$

For $\ell \in A$, let $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ be the unique solution of the *adjoint* mixed variational problem

$$a(\ell, \bar{w}, \bar{v}) + b(\bar{v}, p_w) = \langle \bar{z} - \bar{u}, \bar{v}\rangle\,, \quad \text{for every } \bar{v} \in V, \tag{8a}$$

$$b(\bar{w}, q) - c(p_w, q) = \langle \hat{z} - p, q\rangle\,, \quad \text{for every } q \in Q, \tag{8b}$$

where $(\bar{u}, p)$ solves (2) for the given ℓ and $(\bar{z}, \hat{z})$ is the given data.

Take $v = (\bar{w}, p_w)$ in (7), and use the symmetry of a and c and the fact that w solves (8), to get

$$\begin{aligned}\partial_\ell L_\kappa(\ell, w)\,(\delta\ell) = {} & \langle D\bar{u}(\ell)(\delta\ell), \bar{u} - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell) \\ & + a(\delta\ell, \bar{u}, \bar{w}) + a(\ell, D\bar{u}(\ell)(\delta\ell), \bar{w}) + b(\bar{w}, Dp(\ell)(\delta\ell)) \\ & + b(D\bar{u}(\ell)(\delta\ell), p_w) - c(Dp(\ell)(\delta\ell), p_w) \\ & = \langle D\bar{u}(\ell)(\delta\ell), \bar{u} - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell) + a(\delta\ell, \bar{u}, \bar{w}) \\ & + a(\ell, \bar{w}, D\bar{u}(\ell)(\delta\ell)) + b(D\bar{u}(\ell)(\delta\ell), p_w) + b(\bar{w}, Dp(\ell)(\delta\ell)) - c(p_w, Dp(\ell)(\delta\ell)) \\ & = \langle D\bar{u}(\ell)(\delta\ell), \bar{u} - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell) + a(\delta\ell, \bar{u}, \bar{w}) \\ & + \langle \bar{z} - \bar{u}, D\bar{u}(\ell)(\delta\ell)\rangle + \langle \hat{z} - p, Dp(\ell)(\delta\ell)\rangle \\ & = \kappa DR(\ell)(\delta\ell) + a(\delta\ell, \bar{u}, \bar{w}),\end{aligned}$$

which, in view of (6), yields the formula for the first-order derivative of J_κ:

$$DJ_\kappa(\ell)\,(\delta\ell) = \kappa DR(\ell)(\delta\ell) + a(\delta\ell, \bar{u}, \bar{w}). \tag{9}$$

Summarizing, we obtain the following scheme to compute $DJ_\kappa(\ell)\,(\delta\ell)$:

1. Compute $u(\ell) = (\bar{u}(\ell), p(\ell))$ by solving the mixed variational problem (2).
2. Compute $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ by solving the adjoint problem (8).
3. Compute $DJ_\kappa(\ell)\,(\delta\ell)$ by using (9).

3.2 Second-Order Adjoint Method

We now give a second-order adjoint method for the computation of the second-order derivative of the OLS functional. The objective of the second-order adjoint approach is to give a formula for the second-order derivative that does not involve the second-order derivative of the parameter-to-solution map u. The key idea of the second-order method is to compute the derivative δu directly by using its characterization given through (5) while the computation of the second-order derivative $\delta^2 u$ of u is avoided by the strategy of the first-order adjoint method.

For any $v = (\bar{v}, q) \in W$, and for a fixed direction $\delta\ell_2$, we define

$$\begin{aligned}
&L_\kappa(\ell, v) := DJ_\kappa(\ell)(\delta\ell_2) + a(\ell, D\bar{u}(\ell)(\delta\ell_2), \bar{v})\\
&+ b(\bar{v}, Dp(\ell)(\delta\ell_2)) + b(D\bar{u}(\ell)(\delta\ell_2), q) - c(Dp(\ell)(\delta\ell_2), q) + a(\delta\ell_2, \bar{u}, \bar{v})\\
&= \langle D\bar{u}(\ell)(\delta\ell_2), \bar{u} - \bar{z}\rangle + \langle Dp(\ell)(\delta\ell_2), p - \hat{z}\rangle + \kappa DR(\ell)(\delta\ell_2) + a(\ell, D\bar{u}(\ell)(\delta\ell_2), \bar{v})\\
&+ b(\bar{v}, Dp(\ell)(\delta\ell_2)) + b(D\bar{u}(\ell)(\delta\ell_2), q) - c(Dp(\ell)(\delta\ell_2), q) + a(\delta\ell_2, \bar{u}, \bar{v}).
\end{aligned} \tag{10}$$

By the definition of L_κ, for every $v \in W$, and for every direction $\delta\ell_1$, we have

$$\partial_\ell L_\kappa(\ell, v)(\delta\ell_1) = D^2 J_\kappa(\ell)(\delta\ell_1, \delta\ell_2). \tag{11}$$

Computing the right-hand side of the above identity using (10), we obtain

$$\begin{aligned}
&\partial_\ell L_\kappa(\ell, v)(\delta\ell_1) = \left\langle D^2\bar{u}(\ell)(\delta\ell_1, \delta\ell_2), \bar{u} - \bar{z}\right\rangle + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle\\
&+ \left\langle D^2 p(\ell)(\delta\ell_1, \delta\ell_2), p - \hat{z}\right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2)\\
&+ a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2), \bar{v}) + a(\ell, D^2\bar{u}(\ell)(\delta\ell_1, \delta\ell_2), \bar{v}) + b(\bar{v}, D^2 p(\ell)(\delta\ell_1, \delta\ell_2))\\
&+ b(D^2\bar{u}(\ell)(\delta\ell_1, \delta\ell_2), q) - c(D^2 p(\ell)(\delta\ell_1, \delta\ell_2), q) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1), \bar{v}).
\end{aligned}$$

Let $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ be the solution of (8). By taking $v = (\bar{w}, p_w)$ in the above formula, and using the symmetry of a and c, we obtain

$$
\begin{aligned}
&\partial_\ell L_\kappa(\ell,w)(\delta\ell_1) = \left\langle D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),\bar{u}-\bar{z}\right\rangle + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle \\
&+ \left\langle D^2p(\ell)(\delta\ell_1,\delta\ell_2), p-\hat{z}\right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) \\
&+ a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2),\bar{w}) + a(\ell, D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),\bar{w}) + b(\bar{w}, D^2p(\ell)(\delta\ell_1,\delta\ell_2)) \\
&+ b(D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),p_w) - c(D^2p(\ell)(\delta\ell_1,\delta\ell_2),p_w) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}) \\
&= \left\langle D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),\bar{u}-\bar{z}\right\rangle + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle \\
&+ \left\langle D^2p(\ell)(\delta\ell_1,\delta\ell_2), p-\hat{z}\right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) \\
&+ a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2),\bar{w}) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}) + a(\ell,\bar{w}, D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2)) \\
&+ b(D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),p_w) + b(\bar{w}, D^2p(\ell)(\delta\ell_1,\delta\ell_2)) - c(D^2p(\ell)(\delta\ell_1,\delta\ell_2),p_w) \\
&= \left\langle D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2),\bar{u}-\bar{z}\right\rangle + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle \\
&+ \left\langle D^2p(\ell)(\delta\ell_1,\delta\ell_2), p-\hat{z}\right\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) \\
&+ a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2),\bar{w}) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}) + \left\langle \bar{z}-\bar{u}, D^2\bar{u}(\ell)(\delta\ell_1,\delta\ell_2)\right\rangle \\
&+ \left\langle \hat{z}-p, D^2p(\ell)(\delta\ell_1,\delta\ell_2)\right\rangle \\
&= \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle \\
&+ a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2),\bar{w}) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}).
\end{aligned}
$$

Consequently, from (11) we obtain the following formula for the second-order derivative of the regularized OLS that has no explicit involvement of the second-order derivatives of the solution map:

$$
\begin{aligned}
&D^2J_\kappa(\ell)(\delta\ell_1,\delta\ell_2) = \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle \\
&+ \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + a(\delta\ell_1, D\bar{u}(\ell)(\delta\ell_2),\bar{w}) + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}).
\end{aligned}
$$

In particular, we have

$$
D^2J_\kappa(\ell)(\delta\ell,\delta\ell) = \kappa D^2R(\ell)(\delta\ell,\delta\ell) + \langle\delta\bar{u},\delta\bar{u}\rangle + \langle\delta p,\delta p\rangle + 2a(\delta\ell,\delta\bar{u},\bar{w}). \quad (12)
$$

Summarizing, we obtain the following scheme to compute $D^2J_\kappa(\ell)(\delta\ell,\delta\ell)$:

1. Compute $u(\ell) = (\bar{u}(\ell), p(\ell))$ by solving the mixed variational problem (2).
2. Compute $\delta u = (\delta\bar{u}, \delta p)$ by solving the mixed variational problem (5).
3. Compute $w(\ell) = (\bar{w}(\ell), p_w(\ell))$ by solving the adjoint problem (8).
4. Compute $D^2J_\kappa(\ell)(\delta\ell,\delta\ell)$ by (12).

3.3 Second-Order Derivative Using the First-Order Adjoint Formula

The second-order adjoint approach, given in the previous section, relies on computing the second-order derivative of regularized OLS by using a direct computation of its first-order derivative. However, if we use the first-order derivative formula of the regularized OLS obtained using the first-order adjoint approach, we get a quite different second-order adjoint approach which is discussed below.

We begin with defining the functional $L_\kappa : B \times W \times W \to \mathbb{R}$ by

$$\begin{aligned} L_\kappa(\ell,t,s) &= DJ_\kappa(\ell)(\delta\ell_2) + a(\ell,\bar{u},\bar{t}) + b(\bar{t},p) + b(\bar{u},q_t) - c(p,q_t) - m(\bar{t}) \\ &+ a(\ell,\bar{w},\bar{s}) + b(\bar{s},p_w) + b(\bar{w},q_s) - c(p_w,q_s) - \langle \bar{z}-\bar{u},\bar{s}\rangle - \langle \hat{z}-p,q_s\rangle \\ &= \kappa DR(\ell)(\delta\ell_2) + a(\delta\ell_2,\bar{u},\bar{w}) + a(\ell,\bar{u},\bar{t}) + b(\bar{t},p) + b(\bar{u},q_t) - c(p,q_t) - m(\bar{t}) \\ &+ a(\ell,\bar{w},\bar{s}) + b(\bar{s},p_w) + b(\bar{w},q_s) - c(p_w,q_s) - \langle \bar{z}-\bar{u},\bar{s}\rangle - \langle \hat{z}-p,q_s\rangle , \end{aligned}$$

where $\delta\ell_2$ is a fixed direction, $u = (\bar{u},p)$ is the solution of (2), $w = (\bar{w},p_w)$ is the solution of (8), $t = (\bar{t},q_t)$, and $s = (\bar{s},q_s)$ are arbitrary elements in W, and for $DJ_\kappa(\ell)(\delta\ell_2)$ formula (9) was used.

By the definition of the above functional, for every $t,s \in W$, we have

$$\partial_\ell L_\kappa(\ell,t,s)(\delta\ell_1) = D^2J_\kappa(\ell)(\delta\ell_1,\delta\ell_2). \tag{13}$$

Evaluating the right-hand side of the above identity directly, we obtain

$$\begin{aligned} \partial_\ell L_\kappa(\ell,t,s)(\delta\ell_1) &= \kappa D^2R(\ell)(\delta\ell_1,\delta\ell_2) \; + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1),\bar{w}) \\ &+ a(\delta\ell_2,\bar{u},D\bar{w}(\ell)(\delta\ell_1)) + a(\delta\ell_1,\bar{u},\bar{t}) + a(\ell,D\bar{u}(\ell)(\delta\ell_1),\bar{t}) \\ &+ b(\bar{t},Dp(\ell)(\delta\ell_1)) + b(D\bar{u}(\ell)(\delta\ell_1),q_t) - c(Dp(\ell)(\delta\ell_1),q_t) + a(\delta\ell_1,\bar{w},\bar{s}) \\ &+ a(\ell,D\bar{w}(\ell)(\delta\ell_1),\bar{s}) + b(\bar{s},Dp_w(\ell)(\delta\ell_1)) + b(D\bar{w}(\ell)(\delta\ell_1),q_s) \\ &- c(Dp_w(\ell)(\delta\ell_1),q_s) + \langle D\bar{u}(\ell)(\delta\ell_1),\bar{s}\rangle + \langle Dp(\ell)(\delta\ell_1),q_s\rangle . \end{aligned} \tag{14}$$

By plugging $(\bar{v},q) = (D\bar{w}(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_1))$ in (5), and adding the two resulting expressions and using the symmetry of a and c, we get

$$\begin{aligned} a(\ell,D\bar{w}(\ell)(\delta\ell_1), D\bar{u}(\ell)(\delta\ell_2)) + b(D\bar{u}(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_1))) \\ + b(D\bar{w}(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)) - c(Dp_w(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)) \\ + a(\delta\ell_2,\bar{u},D\bar{w}(\ell)(\delta\ell_1)) = 0. \end{aligned} \tag{15}$$

Since $w(\ell) = (\bar{w}(\ell),p_w(\ell))$ solves (8), it follows that the derivative $Dw(\ell)(\delta\ell_2) = (D\bar{w}(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_2))$ of $w(\ell)$ in any direction $\delta\ell_2$ is characterized as the solution of the following mixed variational problem

$$a(\ell, D\bar{w}(\ell)(\delta\ell_2), \bar{v}) + b(\bar{v}, Dp_w(\ell)(\delta\ell_2)) = -a(\delta\ell_2, \bar{w}, \bar{v}) - \langle D\bar{u}(\ell)(\delta\ell_2), \bar{v}\rangle\,, \tag{16a}$$

$$b(D\bar{w}(\ell)(\delta\ell_2), q) - c(Dp_w(\ell)(\delta\ell_2), q) = -\langle Dp(\ell)(\delta\ell_2), q\rangle\,, \tag{16b}$$

for every $(\bar{v}, q) \in V \times Q$, We set $(\bar{v}, q) = (D\bar{u}(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_1))$ and by summing up the resulting equations and using the symmetry of a and c, obtain

$$\begin{aligned} &a(\ell, D\bar{u}(\ell)(\delta\ell_1), D\bar{w}(\ell)(\delta\ell_2) + b(D\bar{w}(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1))) \\ &\qquad + b(D\bar{u}(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_2)) - c(Dp(\ell)(\delta\ell_1), Dp_w(\ell)(\delta\ell_2))) \\ &\qquad + a(\delta\ell_2, \bar{w}, D\bar{u}(\ell)(\delta\ell_1)) + \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle \\ &\qquad + \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle = 0. \end{aligned} \tag{17}$$

Set $(\bar{s}, q_s) = (D\bar{u}(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_2))$ and $(\bar{t}, q_t) = (D\bar{w}(\ell)(\delta\ell_2), Dp_w(\ell)(\delta\ell_2))$ in (14) and combine the resulting expressions with (15) and (17), to get

$$\begin{aligned} &\partial_\ell L_\kappa(\ell, t, s)(\delta\ell_1) = \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) \ + a(\delta\ell_2, D\bar{u}(\ell)(\delta\ell_1), \bar{w}) \\ &+ a(\delta\ell_2, \bar{u}, D\bar{w}(\ell)(\delta\ell_1)) + a(\delta\ell_1, \bar{u}, D\bar{w}(\ell)(\delta\ell_2)) - a(\delta\ell_2, \bar{w}, D\bar{u}(\ell)(\delta\ell_1)) \\ &- \langle D\bar{u}(\ell)(\delta\ell_2), D\bar{u}(\ell)(\delta\ell_1)\rangle - \langle Dp(\ell)(\delta\ell_2), Dp(\ell)(\delta\ell_1)\rangle + a(\delta\ell_1, \bar{w}, D\bar{u}(\ell)(\delta\ell_2)) \\ &- a(\delta\ell_2, \bar{u}, D\bar{w}(\ell)(\delta\ell_1)) + \langle D\bar{u}(\ell)(\delta\ell_1), D\bar{u}(\ell)(\delta\ell_2)\rangle + \langle Dp(\ell)(\delta\ell_1), Dp(\ell)(\delta\ell_2)\rangle \\ &= \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) + a(\delta\ell_1, \bar{u}, Dw(\ell)(\delta\ell_2)) + a(\delta\ell_1, \bar{w}, D\bar{u}(\ell)(\delta\ell_2)). \end{aligned}$$

Therefore, from (13) we obtain the following formula for the second-order derivative of the regularized OLS that has no explicit involvement of the second-order derivatives of the solution map:

$$\begin{aligned} D^2 J_\kappa(\ell)(\delta\ell_1, \delta\ell_2) = \kappa D^2 R(\ell)(\delta\ell_1, \delta\ell_2) + a(\delta\ell_1, \bar{u}, Dw(\ell)(\delta\ell_2)) \\ + a(\delta\ell_1, \bar{w}, D\bar{u}(\ell)(\delta\ell_2)). \end{aligned}$$

In particular

$$\begin{aligned} D^2 J_\kappa(\ell)(\delta\ell, \delta\ell) = \kappa D^2 R(\ell)(\delta\ell, \delta\ell) + a(\delta\ell, \bar{u}, Dw(\ell)(\delta\ell)) \\ + a(\delta\ell, \bar{w}, D\bar{u}(\ell)(\delta\ell)). \end{aligned} \tag{18}$$

Summarizing, we have the following scheme to compute $D^2 J_\kappa(\ell)(\delta\ell, \delta\ell)$:

1. Compute $u = (\bar{u}, p)$ by solving the mixed variational problem (2).
2. Compute $\delta u = (\delta\bar{u}, \delta p)$ by solving (5).
3. Compute $w = (\bar{w}, p_w)$ by solving the adjoint mixed variational problem (8).
4. Compute $\delta w = (\delta\bar{w}, \delta p_w)$ by solving (16).
5. Compute $D^2 J_\kappa(\ell)(\delta\ell, \delta\ell)$ by (18).

It should be noted that the above approach requires additional computation of (16) which is a derivative characterization of the adjoint variables.

4 Computational Framework

We now give algorithms for computing the gradient and the Hessian of the regularized OLS functional. Let T_h be a triangulation on Ω. Let L_h be the space of all piecewise continuous polynomials of degree d_ℓ relative to T_h, let V_h be the space of all piecewise continuous polynomials of degree $d_{\bar{u}}$ relative to T_h, and let Q_h be the space of all piecewise continuous polynomials of degree d_q relative to T_h. We represent bases for L_h, V_h, and Q_h by $\{\varphi_1, \varphi_2, \dots, \varphi_m\}$, $\{\psi_1, \psi_2, \dots, \psi_n\}$, and $\{\chi_1, \chi_2, \dots, \chi_k\}$, respectively. The space L_h is then isomorphic to $\mathbb{R}^m$ and for any $\ell \in L_h$, we define $L \in \mathbb{R}^m$ by $L_i = \ell(x_i)$, for $i = 1, 2, \dots, m$, where the nodal basis $\{\varphi_1, \varphi_2, \dots, \varphi_m\}$ corresponds to the nodes $\{x_1, x_2, \dots, x_m\}$. Conversely, each $L \in \mathbb{R}^m$ corresponds to $\ell \in L_h$ defined by $\ell = \sum_{i=1}^m L_i \varphi_i$. Analogously, $\bar{u} \in V_h$ will correspond to $\bar{U} \in \mathbb{R}^n$, where $\bar{U}_i = \bar{u}(y_i)$, $i = 1, 2, \dots, n$, and $\bar{u} = \sum_{i=1}^n \bar{U}_i \psi_i$, where $y_1, y_2, \dots, y_n$ are the nodes of the mesh defining U_h. Finally, $p \in Q_h$ will correspond to $P \in \mathbb{R}^k$, where $P_i = p(z_i)$, $i = 1, 2, \dots, k$, and $p = \sum_{i=1}^k P_i \chi_i$, where $z_1, z_2, \dots, z_k$ are the nodes of the mesh defining Q_h.

The discrete mixed variational problem seeks, for each ℓ_h, the unique $(\bar{u}_h, p_h) \in V_h \times Q_h$ such that

$$a(\ell_h, \bar{u}_h, \bar{v}) + b(\bar{v}, p_h) = m(\bar{v}), \quad \text{for every } \bar{v} \in V_h, \tag{19a}$$

$$b(\bar{u}_h, q) - c(p_h, q) = 0, \quad \text{for every } q \in Q_h. \tag{19b}$$

We define $S : \mathbb{R}^m \to \mathbb{R}^{n+k}$ to be the finite element solution map that assigns to each $\ell_h \in L_h$, the unique approximate solution $u_h = (\bar{u}_h, p_h) \in V_h \times Q_h$. Then $S(L) = U = (\bar{U}, P)$, where U is given by

$$\begin{pmatrix} \widehat{K}_{n\times n}(L) & B^{\mathrm{T}}_{n\times k} \\ B_{k\times n} & -C_{k\times k} \end{pmatrix} \begin{pmatrix} \bar{U} \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}. \tag{20}$$

with

$$\begin{aligned} \widehat{K}(L)_{i,j} &= a(\ell, \psi_j, \psi_i), \quad i,j = 1, 2, \dots, n, \\ B_{i,j} &= b(\psi_j, \chi_i), \quad i = 1, 2, \dots, k, \quad j = 1, 2, \dots, n, \\ C_{i,j} &= c(\chi_j, \chi_i), \quad i,j = 1, 2, \dots, k, \\ F_i &= m(\psi_i), \quad i = 1, 2, \dots, n. \end{aligned}$$

4.1 Direct Gradient Computation

We recall that the regularized OLS functional is given by

$$J_\kappa(\ell) = \frac{1}{2}\,\|\bar{u}(\ell) - \bar{z}\|_V^2 + \frac{1}{2}\,\|p(\ell) - \hat{z}\|_Q^2 + \kappa R(\ell),$$

where $(\bar{z}, \hat{z})$ is the measured data and $u(\ell) = (\bar{u}(\ell), p(\ell))$ solves (2).

The discrete analogue of the above functional is given by

$$J_\kappa(L) = \frac{1}{2}(\bar{U} - \bar{Z})^T \mathbb{M}(\bar{U} - \bar{Z}) + \frac{1}{2}(P - \hat{Z})^T \widehat{\mathbb{M}}(P - \hat{Z}) + \kappa R(L),$$

where $U = (\bar{U}, P)$ solves (20), the matrix $\mathbb{M}$ is given by

$$\langle \bar{u}_1, \bar{u}_2 \rangle_V = \bar{U}_2^T \mathbb{M} \bar{U}_1,$$

for any $u_1, u_2 \in V_h$, the matrix $\widehat{\mathbb{M}}$ satisfies

$$\langle q_1, q_2 \rangle_Q = Q_2^T \widehat{\mathbb{M}} Q_1,$$

for any $q_1, q_2 \in Q_h$, and $(\bar{Z}, \hat{Z})$ is the discrete data.

Recall that the first-order derivative of the above functional reads

$$DJ_\kappa(\ell)(\delta\ell) \;= \langle \delta\bar{u}, \bar{u} - \bar{z} \rangle + \langle \delta p, p - \hat{z} \rangle + \kappa DR\,(\ell)\,(\delta\ell),$$

where $\delta u(\ell) = (\delta\bar{u}(\ell), \delta p(\ell))$ is the unique solution of the problem

$$\begin{aligned} a(\ell, \delta\bar{u}, \bar{v}) + b(\bar{v}, \delta p) &= -a(\delta\ell, \bar{u}, \bar{v}), \quad \text{for every } \bar{v} \in V \\ b(\delta\bar{u}, q) - c(\delta p, q) &= 0, \quad \text{for every } q \in Q, \end{aligned}$$

which leads to the following discrete form

$$\begin{pmatrix} \widehat{K}(L) & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \delta\bar{U} \\ \delta P \end{pmatrix} = \begin{pmatrix} -\widehat{K}(\delta L)\bar{U} \\ 0 \end{pmatrix} = \begin{pmatrix} -\mathbb{A}(\bar{U})(\delta L) \\ 0 \end{pmatrix},$$

where $\mathbb{A}$ is the so-called *adjoint stiffness matrix* defined through the condition

$$\widehat{K}(L)\bar{V} = \mathbb{A}(\bar{V})L, \quad \text{for all } L \in \mathbb{R}^m,\ \bar{V} \in \mathbb{R}^n.$$

The Jacobian $\nabla U \in \mathbb{R}^{(n+k)\times m}$ then is computed by solving m linear systems

$$\begin{pmatrix} \widehat{K}(L) & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \nabla_i\bar{U} \\ \nabla_i P \end{pmatrix} = \begin{pmatrix} -\mathbb{A}_i(\bar{U}) \\ 0 \end{pmatrix},\ i = 1, \ldots, m, \tag{21}$$

where $\{E_1, E_2, \ldots, E_m\}$ is the basis of $\mathbb{R}^m$, $\nabla \bar{U} \in \mathbb{R}^{n\times m}$, and $\nabla P \in \mathbb{R}^{k\times m}$. Here $\nabla_i \bar{U}$, $\nabla_i P$, and $\mathbb{A}_i(\bar{U}) = \mathbb{A}(\bar{U})E_i$ are the i-th columns.

A discretization gradient formula is then given by

$$\begin{aligned} DJ_\kappa(L)(\delta L) &= \delta \bar{U}^T \mathbb{M}(\bar{U} - \bar{Z}) + \delta P^T \widehat{\mathbb{M}}(P - \widehat{Z}) + \kappa \nabla R\,(L)\,\delta L \\ &= \delta L^{\mathrm{T}} \nabla \bar{U}^{\mathrm{T}} \mathbb{M}\left(\bar{U} - \bar{Z}\right) + \delta L^{\mathrm{T}} \nabla P^{\mathrm{T}} \widehat{\mathbb{M}}\left(P - \widehat{Z}\right) + \kappa \delta L^{\mathrm{T}} \nabla R\,(L)\,, \end{aligned}$$

which leads to the following expression for the gradient:

$$\nabla J_\kappa(L) = \nabla \bar{U}^{\mathrm{T}} \mathbb{M}\left(\bar{U} - \bar{Z}\right) + \nabla P^{\mathrm{T}} \widehat{\mathbb{M}}\left(P - \widehat{Z}\right) + \kappa \nabla R\,(L)\,. \tag{22}$$

Summarizing, the following scheme computes the gradient of the OLS functional:

1. Compute $U = (\bar{U}, P)$ by solving linear system (20).
2. Compute ∇U by solving m linear systems (21).
3. Compute $\nabla J_\kappa(L)$ by using formula (22).

Remark 1 The above scheme requires solving $(m + 1)$ linear systems for the gradient computation.

4.2 Gradient Computation by the First-Order Adjoint Method

We now give an algorithm for computing the gradient of the regularized OLS by the first-order adjoint approach. Recall that the formula for the computation of the first-order derivative by using the first-order adjoint approach reads

$$DJ_\kappa(\ell)(\delta\ell) \ = \kappa DR(\ell)(\delta\ell) + a(\delta\ell, \bar{u}, \bar{w}), \tag{23}$$

where $u = (\bar{u}, p)$ and $w = (\bar{w}, q)$ are the unique solutions of (2) and (8).

The discrete counterpart of these elements are $U = (\bar{U}, P)$, which solves (20), and $W = (\bar{W}, P_w)$, which solves the following linear system

$$\begin{pmatrix} \widehat{K} & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \bar{W} \\ P_w \end{pmatrix} = \begin{pmatrix} \mathbb{M}(\bar{Z} - \bar{U}) \\ \widehat{\mathbb{M}}(\widehat{Z} - P) \end{pmatrix}. \tag{24}$$

Since $a(\delta\ell, \bar{u}, \bar{w}) = \bar{U}^{\mathrm{T}} \widehat{K}(\delta L)\bar{W} = \bar{U}^{\mathrm{T}} \mathbb{A}(\bar{W})\delta L$, the discrete derivative reads

$$DJ_\kappa(L)(\delta L) = \kappa \nabla R(L)(\delta L) + \delta L^{\mathrm{T}} \mathbb{A}(\bar{W})^{\mathrm{T}} \bar{U}$$

which yields an explicit formula for the gradient

$$\nabla J_\kappa(L) = \kappa \nabla R(L) + \mathbb{A}(\bar{W})^{\mathrm{T}} \bar{U} \tag{25}$$

The following scheme computes the gradient by the first-order adjoint method:

1. Compute $U = (\bar{U}, P)$ by solving linear system (20).
2. Compute $W = (\bar{W}, P_w)$ by solving linear system (24).
3. Compute $\nabla J_\kappa(L)$ by using formula (25).

Remark 2 The above scheme, in contrast to the direct gradient computation scheme which required solving $(m + 1)$ linear systems, only requires solving two linear systems.

4.3 Hessian Computation by the Second-Order Adjoint Method

Recall the second-order derivative of the regularized OLS reads:

$$D^2 J_\kappa(\ell)(\delta\ell, \delta\ell) \; = \kappa D^2 R(\ell)(\delta\ell, \delta\ell) + \langle \delta\bar{u}, \delta\bar{u}\rangle + \langle \delta p, \delta p\rangle + 2a(\delta\ell, \delta\bar{u}, \bar{w}).$$

To discretize the above formula, we note that given the Hessian $\nabla^2 R(L) \in \mathbb{R}^{m\times m}$ of R, we have

$$D^2 R(\ell)(\delta\ell, \delta\ell) = \delta L^T \nabla^2 R(L) \delta L.$$

For the second and the third terms, we have

$$\langle \delta\bar{u}, \delta\bar{u}\rangle = \delta L^T \nabla\bar{U}^{\mathrm{T}} \mathbb{M} \nabla\bar{U} \delta L,$$
$$\langle \delta p, \delta p\rangle = \delta L^T \nabla P^T \widehat{\mathbb{M}} \nabla P \delta L.$$

For the fourth term, the use of the adjoint stiffness matrix yields the expression

$$a(\delta\ell, \delta\bar{u}, \bar{w}) = \delta L^{\mathrm{T}} \nabla\bar{U}^{\mathrm{T}} \widehat{K}(\delta L)\bar{W} = \delta L^{\mathrm{T}} \nabla\bar{U}^{\mathrm{T}} \mathbb{A}(\bar{W})\delta L.$$

By combining the above, we get the explicit expression for the Hessian:

$$\nabla^2 J_\kappa(L) = \kappa \nabla^2 R(L) + \; \nabla\bar{U}^T \mathbb{M} \nabla\bar{U} \; + \nabla P^T \widehat{\mathbb{M}} \nabla P + 2\nabla\bar{U}^T \mathbb{A}(\bar{W}). \qquad (26)$$

In view of the above formula, the following scheme computes the Hessian:

1. Compute $U = (\bar{U}, P)$ by solving linear system (20).
2. Compute $\nabla U = (\nabla\bar{U}, \nabla P)$ by solving m linear systems (21).
3. Compute $W = (\bar{W}, P_w)$ by solving linear system (24).
4. Compute $\nabla^2 J_\kappa(L)$ by using formula (26).

Remark 3 The above scheme requires solving $(m+2)$ linear systems for the Hessian computation.

4.4 Hessian Computation Using the First-Order Adjoint Formula

A direct computation of the second-order derivative using the first-order adjoint approach yields

$$D^2 J_\kappa(\ell)(\delta\ell,\delta\ell) = \kappa D^2 R(\ell)(\delta\ell,\delta\ell) + a(\delta\ell,\bar{u}, Dw(\ell)(\delta\ell)) + a(\delta\ell,\bar{w}, D\bar{u}(\ell)(\delta\ell)).$$

The discrete analogue of the following mixed variational problem

$$\begin{aligned} a(\ell, D\bar{w}(\ell)(\delta\ell),\bar{v}) + b(\bar{v}, Dp_w(\ell)(\delta\ell)) &= -a(\delta\ell,\bar{w},\bar{v}) - \langle D\bar{u}(\ell)(\delta\ell),\bar{v}\rangle\,, \forall\ \bar{v} \in V,\\ b(D\bar{w}(\ell)(\delta\ell), q) - c(Dp_w(\ell)(\delta\ell), q) &= -\langle Dp(\ell)(\delta_\ell), q\rangle, \quad \forall\ q \in Q, \end{aligned}$$

necessary for the discrete counterpart of $Dw(\ell)(\delta\ell)$ is given by the linear system

$$\begin{aligned} \begin{pmatrix} \widehat{K}(L) & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \delta\bar{W} \\ \delta P_W \end{pmatrix} &= \begin{pmatrix} -\widehat{K}(\delta L)\bar{W} - \mathbb{M}\delta\bar{U} \\ -\widehat{\mathbb{M}}\delta P \end{pmatrix} \\ &= \begin{pmatrix} -\mathbb{A}(\bar{W})(\delta L) - \mathbb{M}\delta\bar{U} \\ -\widehat{\mathbb{M}}\delta P \end{pmatrix}. \end{aligned}$$

Therefore, the Jacobian $\nabla W \in \mathbb{R}^{(k+n)\times m}$ is computed by m linear equations

$$\begin{aligned} \begin{pmatrix} \widehat{K}(L) & B^{\mathrm{T}} \\ B & -C \end{pmatrix} \begin{pmatrix} \nabla_i\bar{W} \\ \nabla_i P_{\mathrm{w}} \end{pmatrix} &= \begin{pmatrix} -\mathbb{A}(\bar{W})E_i - \mathbb{M}\nabla U E_i \\ -\widehat{\mathbb{M}}\nabla P E_i \end{pmatrix} \\ &= \begin{pmatrix} -\mathbb{A}_i(\bar{W}) - \mathbb{M}\nabla_i U \\ -\widehat{\mathbb{M}}\nabla_i P \end{pmatrix}, \end{aligned} \tag{27}$$

where $i = 1,\ldots,m$, $\{E_1, E_2, \ldots, E_m\}$ is the basis of $\mathbb{R}^m$, $\nabla\bar{W} \in \mathbb{R}^{n\times m}$ and $\nabla P_{\mathrm{w}} \in \mathbb{R}^{k\times m}$. Therefore, the Hessian of the regularized OLS by is given by the formula:

$$\nabla^2 J_\kappa(L) = \kappa\nabla^2 R(L) + \nabla\bar{W}^T\mathbb{A}(\bar{U}) + \nabla\bar{U}^T\mathbb{A}(\bar{W}). \tag{28}$$

Summarizing, the following scheme computes the Hessian:

1. Compute $U = (\bar{U}, P)$ by solving linear system (20).
2. Compute $\nabla U = (\nabla\bar{U}, \nabla P)$ by solving m linear systems (21).
3. Compute $W = (\bar{W}, P)$ by solving linear system (24).
4. Compute $\nabla W = (\nabla\bar{W}, \nabla P_w)$ by solving m linear systems (27).
5. Compute $\nabla^2 J_\kappa(L)$ by using formula (28).

Remark 4 The above scheme requires solving $(2m + 2)$ linear systems and quite expensive in comparison to the second-order adjoint approach which only requires solving $(m + 2)$ linear systems.

5 Computational Experiments

In this section we consider three examples for the identification of a parameter μ on a two-dimensional domain $\Omega = (0, 1) \times (0, 1)$ with boundary $\partial\Omega = \Gamma_1 \times \Gamma_2$. All of our experiments are purely synthetic. Therefore, the data vectors in all the experiments are computed, not measured. In all experiments, we used an adaptive mesh to obtain an accurate solution, and used this for the data z. The identification was done in a finite-dimensional space of dimension of 1140 on a mesh with 2158 triangles. The optimization was performed using the Newton method using the second-order adjoint approach. The H^1 semi-norm regularization was used. We chose the regularization parameter by trial and error.

5.1 Elasticity Imaging Inverse Problem

Given the domain Ω as a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ and $\partial\Omega = \Gamma_1 \cup \Gamma_2$ as its boundary, the following system models the response of an isotropic elastic body to the known body forces and boundary traction:

$$-\nabla \cdot \sigma = f \text{ in } \Omega, \tag{29a}$$

$$\sigma = 2\mu\epsilon(u) + \lambda \text{div} u I, \tag{29b}$$

$$u = g \text{ on } \Gamma_1, \tag{29c}$$

$$\sigma n = h \text{ on } \Gamma_2. \tag{29d}$$

In (29), the vector-valued function $u = u(x)$ is the displacement of the elastic body, f is the applied body force, n is the unit outward normal, and $\epsilon(u) = \frac{1}{2}(\nabla u + \nabla u^{\mathrm{T}})$ is the linearized strain tensor. The resulting stress tensor σ in the stress-strain law (29b) is obtained under the condition that the elastic body is isotropic and the displacement is sufficiently small so that a linear relationship remains valid. Here μ and λ are the Lamé parameters which quantify the elastic properties of the object.

The elasticity imaging inverse problem has found interesting applications in locating soft inclusions in an incompressible object, for example, cancerous tumor in the human body. From a mathematical standpoint this inverse problem seeks μ from a measurement of the displacement vector u under the assumption that the parameter λ is very large. The key idea behind the elasticity imaging inverse problem is that the stiffness of soft tissue can vary significantly based on its molecular makeup and varying macroscopic/microscopic structure (see [17]) and such changes in stiffness are related to changes in tissue health. In other words, the elastography inverse problem mathematically mimics the practice of palpation by making use of the differing elastic properties of healthy and unhealthy tissue to identify tumors. In most of the existing literature on elasticity imaging inverse problem, the human body is modelled as an incompressible elastic object. Although this assumption simplifies the identification process as there is only one parameter μ to identify, it significantly complicates the computational process as the classical finite element methods become quite ineffective due to the so-called locking effect. The mixed variational problem framework studied in this work offers a remedy to the locking effect. We define $V = \{\bar{v} \in H^1(\Omega) \times H^1(\Omega) : \ \bar{v} = 0 \text{ on } \Gamma_1\}$.

The Green's identity and the boundary conditions (29c) and (29d) yield the following weak form of the elasticity system (29): Find $\bar{u} \in V$ such that

$$\int_\Omega 2\mu\epsilon(\bar{u}) \cdot \epsilon(\bar{v}) + \int_\Omega \lambda(\operatorname{div} \bar{u})(\operatorname{div} \bar{v}) = \int_\Omega f\bar{v} + \int_{\Gamma_2} \bar{v}h, \quad \text{for every } \bar{v} \in V. \tag{30}$$

We introduce a pressure term $p \in Q = L^2(\Omega)$ by $p = \lambda(\operatorname{div} \bar{u})$, or equivalently,

$$\int_\Omega (\operatorname{div} \bar{u})q - \int_\Omega \frac{1}{\lambda} pq = 0, \quad \text{for every } q \in Q. \tag{31}$$

Using $p = \lambda(\operatorname{div} \bar{u})$, the weak form (30) reads: Find $\bar{u} \in V$ such that

$$\int_\Omega 2\mu\epsilon(\bar{u}) \cdot \epsilon(\bar{v}) + \int_\Omega p(\operatorname{div} \bar{v}) = \int_\Omega f\bar{v} + \int_{\Gamma_2} \bar{v}h, \quad \text{for every } \bar{v} \in V. \tag{32}$$

In other words, the problem of finding $\bar{u} \in V$ satisfying (30) has now been reformulated as the problem of finding $(\bar{u}, p) \in V \times Q$ satisfying the mixed variational problems (31) and (32).

For the above problem, we present a numerical example to identify a parameter μ in (29) where the left and right domain boundaries (Γ_1) are fixed with constant Dirichlet condition $g(x, y)$ and the top and bottom boundaries (Γ_2) have Neumann condition $h(x, y)$. We set $\lambda = 10^6$, and the functions defining the coefficient, load, and boundary conditions are as follows (Fig. 1):

$$\mu(x, y) = 2.5 + \frac{1}{4}\sin(2\pi x), \quad f(x, y) = \begin{bmatrix} 2.3 + \frac{1}{10}x \\ 2.3 + \frac{1}{10}y \end{bmatrix},$$

$$g(x, y) = \frac{1}{100}\begin{bmatrix} x \\ y^2 \end{bmatrix} \text{ on } \Gamma_1, \quad h(x, y) = \frac{1}{2}\begin{bmatrix} 1 + 2x^2 \\ 1 + 2y^2 \end{bmatrix} \text{ on } \Gamma_2.$$

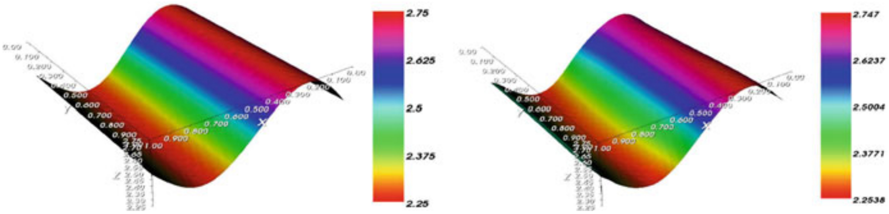


Fig. 1 Reconstruction using the Second-order Adjoint Approach (14 iterations). Exact coefficient (*left*) and estimated coefficient (*right*)

5.2 *Stokes Equations*

We now consider Stoke's equations

$$-\mu\Delta u + \nabla p = f \quad \text{in } \Omega, \tag{33a}$$

$$-\operatorname{div} u = 0 \quad \text{in } \Omega, \tag{33b}$$

where u can be considered as the velocity field of an incompressible fluid motion, and p is then the associated pressure, the constant μ is the viscosity coefficient of the fluid. Here we consider homogeneous Dirichlet boundary condition for the velocity, i.e. $u|_{\partial\Omega} = 0$. By multiplying $v \in H_0^1(\Omega)$ to (33a) and $q \in L^2(\Omega)$ to the mass equation (33b), and applying integration by part for the momentum equation, we obtain the following weak form of the Stokes equations (33): Find $u \in H_0^1(\Omega)$ and a pressure $p \in L^2(\Omega)$ such that

$$\int_\Omega \mu\nabla u\cdot\nabla v - \int_\Omega p(\operatorname{div} v) = \int_\Omega fv, \quad \text{for every } v \in H_0^1(\Omega) \tag{34}$$

$$-\int_\Omega (\operatorname{div} u)q = 0, \qquad \text{for every } q \in L^2(\Omega) \tag{35}$$

The Stokes equations (33) lead to the mixed variational form (2) by setting

$$a(\mu, u, v) = \int_\Omega \mu\nabla u\cdot\nabla v,$$

$$b(u, q) = -\int_\Omega (\operatorname{div} u)q,$$

$$c(p, q) = \int_\Omega \frac{1}{\lambda}pq,$$

$$m(v) = \int_\Omega fv,$$

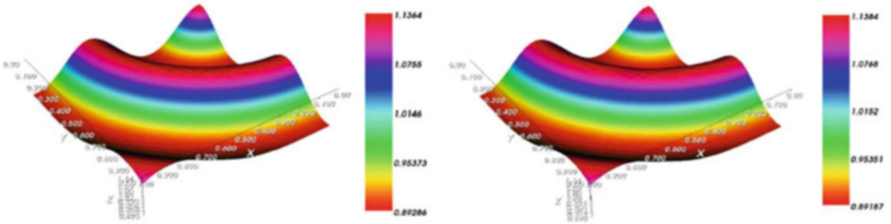


Fig. 2 Reconstruction using the Second-order Adjoint Approach (15 iterations). Exact coefficient (*left*) and estimated coefficient (*right*)

where $c(p, q)$ is the penalization that removes the zero mean restriction on pressure. Figure 2 shows the numerical results for (33) with $\lambda = 10^{13}$, and

$$\mu\,(x, y) = \left(1 - 0.12\cos(3\pi\sqrt{x^2 + y^2})\right)^{-1},\;\; f\,(x, y) = \begin{bmatrix} 1 + 0.1x^2 \\ 0.1y. \end{bmatrix}$$

5.3 Fourth-Order Elliptic Boundary-Value Problem

We consider the following fourth-order elliptic boundary value problem:

$$\Delta(\mu\Delta u) = f \quad \text{in } \Omega, \tag{36a}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{36b}$$

$$\partial_n u = 0 \quad \text{on } \partial\Omega. \tag{36c}$$

The mixed formulation for (36) reads : find $(\phi, u) \in H^1(\Omega) \times H^1_0(\Omega)$ such that

$$\int_\Omega \phi\psi + \int_\Omega \mu\nabla u\nabla\psi = 0, \quad \text{for every } \psi \in H^1(\Omega), \tag{37a}$$

$$\int_\Omega \nabla\phi\nabla v = \int_\Omega f v, \quad \text{for every } v \in H^1_0(\Omega), \tag{37b}$$

Together with a penalization matrix with $\lambda = 10^6$), we applied the second-order Adjoint method to identify μ in (36) where the solution u and the exact parameter μ are given by

$$u(x, y) = 100x^2(1 - x)^2y^2(1 - y)^2, \quad \mu(x, y) = \frac{3}{2} + \sin(\pi x)\sin(\pi y)$$

and where $f(x, y)$ is subsequently defined by (36) (Fig. 3).

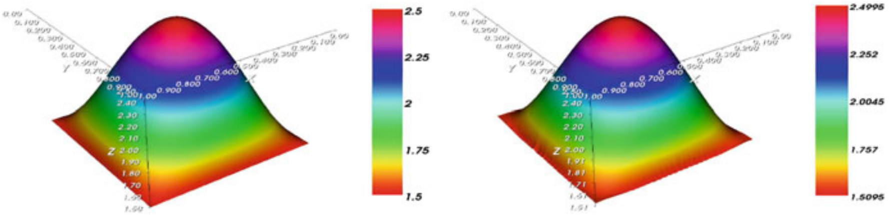


Fig. 3 Reconstruction using the Second-order Adjoint Approach (28 iterations). Exact coefficient (*left*) and estimated coefficient (*right*)

6 Concluding Remarks

We performed parameter identification in an abstract mixed variational problem by using the OLS formulation and give a rigorous derivation of the first-order and the second-order adjoint methods. We test the feasibility of the proposed adjoint approach for three different applications. The effect of the contaminated data on the identification has yet be seen and will be done in a future work.

Acknowledgements We are grateful to the reviewer for the careful reading. Our student Oladayo Eluyeffa implemented some of the results/formulas in his MS thesis entitled Heavy Ball with Friction Method for the Elastography Inverse Problem, September, 2015. Baasansuren Jadamba's work is supported by RIT's COS FEAD grant for 2016–2017. Akhtar Khan is supported by a grant from the Simons Foundation (#210443) and RIT's COS FEAD grant for 2016–2017. Miguel Sama's work is partially supported by Ministerio de Economìa y Competitividad (Spain), project MTM2015-68103-P.

References

1. Alekseev, A.K., Navon, I.M., Steward, J.L.: Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems. Optim. Methods Softw. **24**, 63–87 (2009)
2. Cahill, N., Jadamba, B., Khan, A.A., Sama, M., Winkler, B.: A first-order adjoint and a second-order hybrid method for an energy output least squares elastography inverse problem of identifying tumor location. Bound. Value Probl. **263**, 1–14 (2013)
3. Cioaca, A., Alexe, M., Sandu, A.: Second-order adjoints for solving PDE-constrained optimization problems. Optim. Methods Softw. **27**(4–5), 625–653 (2012)
4. Cioaca, A., Sandu, A.: An optimization framework to improve 4D-Var data assimilation system performance. J. Comput. Phys. **275**, 377–389 (2014)
5. Crossen, E., Gockenbach, M.S., Jadamba, B., Khan, A.A., Winkler, B.: An equation error approach for the elasticity imaging inverse problem for predicting tumor location. Comput. Math. Appl. **67**(1), 122–135 (2014)
6. Daescu, D.N., Navon, I.M.: Efficiency of a POD-based reduced second-order adjoint model in 4D-var data assimilation. Int. J. Numer. Methods Fluids **53**(6), 985–1004 (2007)
7. Dominguez, N., Gibiat, V., Esquerre, Y.: Time domain topological gradient and time reversal analogy: an inverse method for ultrasonic target detection. Wave Motion **42**(1), 31–52 (2005)

8. Doyley, M.M., Jadamba, B., Khan, A.A., Sama, M., Winkler, B.: A new energy inversion for parameter identification in saddle point problems with an application to the elasticity imaging inverse problem of predicting tumor location. Numer. Funct. Anal. Optim. **35**(7–9), 984–1017 (2014)
9. Gockenbach, M., Jadamba, B., Khan, A.A., Tammer, C., Winkler, B.: Proximal method for the elastography inverse problem of tumor identification using an equation error approach. In: Advances in Variational and Hemivariational Inequalities, pp. 169–192. Springer (2014)
10. Jadamba, B., Khan, A.A., Rus, G., Sama, M., Winkler, B.: A new convex inversion framework for parameter identification in saddle point problems with an application to the elasticity imaging inverse problem of predicting tumor location. SIAM J. Appl. Math. **74**(5), 1486–1510 (2014)
11. Kennedy, G.J., Hansen, J.S.: The hybrid-adjoint method: a semi-analytic gradient evaluation technique applied to composite cure cycle optimization. Optim. Eng. **11**(1), 23–43 (2010)
12. Knopoff, D.A., Fernández, D.R., Torres, G.A., Turner, C.V.: Adjoint method for a tumor growth PDE-constrained optimization problem. Comput. Math. Appl. **66**(6), 1104–1119 (2013)
13. Kourounis, D., Durlofsky, L.J., Jansen, J.D., Aziz, K.: Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. Comput. Geosci. **18**(2), 117–137 (2014)
14. Liu, G., Geier, M., Liu, Z., Krafczyk, M., Chen, T.: Discrete adjoint sensitivity analysis for fluid flow topology optimization based on the generalized lattice Boltzmann method. Comput. Math. Appl. **68**(10), 1374–1392 (2014)
15. Lozano, C.: Discrete surprises in the computation of sensitivities from boundary integrals in the continuous adjoint approach to inviscid aerodynamic shape optimization. Comput. Fluids **56**, 118–127 (2012)
16. Namdar Zanganeh, M., Kraaijevanger, J.F.B.M., Buurman, H.W., Jansen, J.D., Rossen, W.R.: Challenges in adjoint-based optimization of a foam EOR process. Comput. Geosci. **18**(3–4), 563–577 (2014)
17. Oberai, A.A., Gokhale, N.H., Feijóo, G.R.: Solution of inverse problems in elasticity imaging using the adjoint method. Inverse Problems **19**(2), 297–313 (2003)
18. Papadimitriou, D.I., Giannakoglou, K.C.: Aerodynamic shape optimization using first and second order adjoint and direct approaches. Arch. Comput. Methods Eng. **15**(4), 447–488 (2008)
19. Pingen, G., Evgrafov, A., Maute, K.: Adjoint parameter sensitivity analysis for the hydrodynamic lattice Boltzmann method with applications to design optimization. Comput. Fluids **38**(4), 910–923 (2009)
20. Ye, X., Li, P., Liu, F.Y.: Exact time-domain second-order adjoint-sensitivity computation for linear circuit analysis and optimization. IEEE Trans. Circuits Syst. I. Regul. Pap. **57**(1), 236–248 (2010)

Minimization of the L_p-Norm, $p \geq 1$ of Dirichlet-Type Boundary Controls for the 1D Wave Equation

Ilya Smirnov and Anastasia Dmitrieva

Abstract This paper provides a modified method that allows one to solve the problem of the optimal boundary controls $\mu(t)$ and $\nu(t)$ of displacements at two ends of a string for a large time interval $T = 2ln$, where $n = 1, 2, 3, \ldots$ It should be noted that the minimization was made in the space L_p with $p \geq 1$. Besides, it was found that the derivatives of above-mentioned functions are $2l$-periodic functions.

Keywords Optimal control • Wave equation • Boundary control • L_p-Norm

This paper develops the theme considered in [4] and [5] and provides a method allowing to calculate and present in an explicit analytical form the optimal boundary displacement controls at two ends of a string $x = 0$ and $x = l$ transferring the process of its oscillations described by the generalized solution of the wave equation

$$u_{tt}(x, t) - u_{xx}(x, t) = 0, \tag{1}$$

from an arbitrary given initial state

$$\{u(x, 0) = \phi(x), \quad u_t(x, 0) = \psi(x)\} \tag{2}$$

to an arbitrary given terminal state

$$\{u(x, T) = \widehat{\phi}(x), \quad u_t(x, T) = \widehat{\psi}(x)\} \tag{3}$$

for a quite large time interval T.

I. Smirnov (✉) • A. Dmitrieva
Lomonosov Moscow State University, Moscow, Russia
e-mail: ismirnov@cs.msu.ru

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_6

Analogously to [4], on the rectangle

$$Q_T = [0 \le x \le l] \times [0 \le t \le T], \tag{4}$$

we will consider the class of functions $\widehat{W_p^1}(Q_T)$.

Definition 1 Class of functions $\widehat{W_p^1}(Q_T)$ is a set of functions of two variables $u(x,t)$ that are continuous in the closed rectangle $\overline{Q_T}$ and have both generalized partial derivatives $u_x(x,t)$ and $u_t(x,t)$, with each of them belonging to the class $L_p(Q_T)$ as well as to the class $L_p[0 \le x \le l]$ for $\forall t \in [0,T]$, and to the class $L_p[0 \le t \le T]$ for $\forall x \in [0,l]$.

The treatment of the problem is carried out in terms of the generalized solution $u(x,t)$ from the class $\widehat{W_p^1}(Q_T)$ to the initial-boundary value problem for the wave equation (1) with initial conditions (2) and boundary conditions

$$\{u(0,t) = \mu(t), \quad u(l,t) = \nu(t)\} \tag{5}$$

In addition, it is necessary to impose the requirement of smoothness on the functions of initial, terminal and boundary conditions, since the generalized solution $u(x,t)$ belongs to the class $\widehat{W_p^1}(Q_T)$:

$$\phi(x) \in W_p^1[0,l], \quad \psi(x) \in L_p[0,l], \quad \widehat{\phi}(x) \in W_p^1[0,l], \quad \widehat{\psi}(x) \in L_p[0,l],$$

$$\mu(t) \in W_p^1[0,T], \quad \nu(t) \in W_p^1[0,T]$$

Definition 2 A function $u(x,t)$ from the class $\widehat{W_p^1}(Q_T)$ is said to be the generalized $\widehat{W_p^1}(Q_T)$ class solution to the initial-value problem for the wave equation (1) with initial conditions (2) and boundary conditions (5)which obey to the smoothness requirements (6), if $u(x,t)$ satisfies the integral identity:

$$\int_0^l \int_0^T u(x,t)[\Phi_{tt}(x,t) - \Phi_{xx}(x,t)]dxdt + \int_0^l \phi(x)\Phi_t(x,0)dx -$$

$$-\int_0^l \psi(x)\Phi(x,0)dx - \int_0^T \mu(t)\Phi_x(0,t)dt + \int_0^T \nu(t)\Phi_x(l,t)dx = 0$$

for every function $\Phi(x,t)$ from the class $C^{(2)}(\overline{Q_T})$, obeying to requirements
$\Phi(x,T) \equiv 0$, $\Phi_t(x,T) \equiv 0$ if $0 \le x \le l$, $\Phi(0,t) \equiv 0$, $\Phi(l,t) \equiv 0$ if $0 \le t \le T$.

Definition 3 Functions $\mu(t)$ and $\nu(t)$ from the class $W_p^1[0,T]$ are said to be the solutions to the boundary control problem if the generalized $\widehat{W_p^1}(Q_T)$ class solution $u(x,t)$ satisfies the terminal conditions (3) understood in the sense of equality of elements $W_p^1[0,l]$ and $L_p[0,l]$, respectively.

Since the boundary control problem under investigation has infinitely many solutions for $T > l$, which was established, for example, in [1] and [3], there arises the optimal boundary control problem for $T > l$. It consists in the selection among all the functions $\mu(t)$ and $\nu(t)$, representing boundary controls, the ones that minimize the boundary energy integral

$$\int_0^T \{[\mu'(t)]^p + [\nu'(t)]^p\}dt \tag{6}$$

with constraints imposed by given initial conditions (2), terminal conditions (3), and fitting conditions:

$$\int_0^T [\mu'(t) + \nu'(t)]dt = \widehat{\phi}(0) - \phi(0) + \widehat{\phi}(l) - \phi(l) \tag{7}$$

Let us introduce the function $\widetilde{u}(x,t)$ which was used also in papers [5]:

$$\widetilde{u}(x,t) = \begin{cases} \frac{1}{2}\Big[\phi(x+t) + \phi(x-t) + \int_{x-t}^{x+t} \psi(\xi)d\xi\Big] & \triangle_1, \\ \frac{1}{2}\Big[\phi(x+t) + \phi(0) + \int_0^{x+t} \psi(\xi)d\xi\Big] & \triangle_2, \\ \frac{1}{2}\Big[\phi(l) + \phi(x-t) + \int_{x-t}^{l} \psi(\xi)d\xi\Big] & \triangle_3, \\ \frac{1}{2}\Big[\phi(0) + \phi(l) + \int_0^l \psi(\xi)d\xi\Big] = C_0 = const & \triangle_4, \end{cases} \tag{8}$$

where

$\triangle_1$ is a triangle limited by line segments $x - t = 0$, $x + t - l = 0$, and $t = 0$;
$\triangle_2$ is a triangle limited by line segments $x - t = 0$, $x + t - l = 0$, and $x = 0$;
$\triangle_3$ is a triangle limited by line segments $x - t = 0$, $x + t - l = 0$, and $x - l = 0$;
$\triangle_4$ is a quadrangle limited by line segments $x - t = 0$, $x + t - l = 0$, $x - l = 0$, and $t - T = 0$.

Proposition 1 *For every $T > l$, the function described by (8) is a (unique, according to [2]) generalized $\widehat{W_p^1}(Q_T)$ class solution to the initial-boundary value problem $\widetilde{u}_{tt}(x,t) - \widetilde{u}_{xx}(x,t) = 0$ with initial conditions $\widetilde{u}(x,0) = \phi(x)$, $\widetilde{u}_t(x,0) = \psi(x)$ and with boundary conditions $\widetilde{u}(0,t) = \widetilde{\mu}(t)$, $\widetilde{u}(l,t) = \widetilde{\nu}(t)$, where*

$$\widetilde{\mu}(t) = \begin{cases} \frac{1}{2}\Big[\phi(t) + \phi(0) + \int_0^t \psi(\xi)d\xi\Big] & \text{for} \quad 0 \le t \le l, \\ C_0 & \text{for} \quad l \le t \le T, \end{cases} \tag{9}$$

$$\widetilde{\nu}(t) = \begin{cases} \frac{1}{2}\Big[\phi(l-t) + \phi(l) + \int_{l-t}^l \psi(\xi)d\xi\Big] & \text{for} \quad 0 \le t \le l, \\ C_0 & \text{for} \quad l \le t \le T. \end{cases} \tag{10}$$

The detailed proof of this proposition is similarly as Assertion 1, provided in [5].

We will consider the case $T = 2ln$, where $n = 1, 2, 3, \ldots$

Let $u(x, t)$ be the generalized $\widehat{W_p^1}(Q_T)$ class solution of the initial-boundary problem (1), (2), (5), and let $\widetilde{u}(x, t)$ be a function determined by relation (8). Then, the function $\widehat{u}(x, t) = u(x, t) - \widetilde{u}(x, t)$ is a unique generalized $\widehat{W_p^1}(Q_T)$ class solution to the initial-boundary value problem for the wave equation $\widehat{u}_{tt}(x, t) - \widehat{u}_{xx}(x, t) = 0$ with zero initial conditions $\widehat{u}(x, 0) = 0$, $\widehat{u}_t(x, 0) = 0$ and with boundary conditions $\widehat{u}(0, t) = \widehat{\mu}(t) = \mu(t) - \widetilde{\mu}(t)$, $\widehat{u}(l, t) = \widehat{\nu}(t) = \nu(t) - \widetilde{\nu}(t)$. It should be noted that the equations $\mu(0) = \phi(0)$, $\widetilde{\mu}(0) = \phi(0)$, $\nu(0) = \phi(l)$, $\widetilde{\nu}(0) = \phi(l)$ imply $\widehat{\mu}(0) = 0$, $\widehat{\nu}(0) = 0$.

Proposition 2 *For every T satisfying the inequality $T \leq 2l(n+1)$, where $n = 1, 2, \ldots$, the function $\widehat{u}(x, t)$ is determined by the equation:*

$$\widehat{u}(x,t) = \sum_{k=0}^{n} \underline{\widehat{\mu}}(t - x - 2kl) - \sum_{k=1}^{n+1} \underline{\widehat{\mu}}(t + x - 2kl) + \\ + \sum_{k=0}^{n} \underline{\widehat{\nu}}(t + x - 2kl - l) - \sum_{k=1}^{n-1} \underline{\widehat{\nu}}(t - x - 2kl) \quad (11)$$

where the symbols $\underline{\widehat{\mu}(t)}$ and $\underline{\widehat{\nu}(t)}$ stand for the functions coinciding with $\widehat{\mu}(t)$ and $\widehat{\nu}(t)$, respectively, for $t \geq 0$ and vanishing for $t < 0$.

The proof of this proposition is similarly as in [5]. Next, it is necessary to establish the conditions following from the given initial conditions (2) and terminal conditions (3). We will apply the method described in [4]. Taking the semi-sum and semi-difference of the relations obtained by differentiation of (11) with respect to x and t, then, substituting $T = 2ln$ into the obtained relations, and using the following from (3) conditions for the function $\widehat{u}(x, t)$,

$$\widehat{u}(x, T) = \widehat{\phi} - C_0, \quad \widehat{u}_t(x, T) = \widehat{\psi}(x),$$

we come to the following conditions:

$$-\sum_{k=1}^{n+1} \underline{\widehat{\mu}}'[2l(n-k) + x] + \sum_{k=0}^{n} \underline{\widehat{\nu}}'[2l(n-k) + x - l] = \frac{1}{2}\big[\widehat{\phi}'(x) + \widehat{\psi}(x)\big], \quad (12)$$

$$-\sum_{k=0}^{n} \underline{\widehat{\mu}}'[2l(n-k) + x] + \sum_{k=1}^{n+1} \underline{\widehat{\nu}}'[2l(n-k) - x + l] = \frac{1}{2}\big[\widehat{\phi}'(x) - \widehat{\psi}(x)\big], \quad (13)$$

Taking into account that functions $\underline{\widehat{\mu}}'(t)$ and $\underline{\widehat{\nu}}'(t)$ are equal to zero if $t < 0$ and coincide with $\widehat{\mu}'(t)$ and $\widehat{\nu}'(t)$, respectively, if $t \geq 0$, we can modify the conditions as follows:

$$-\sum_{k=1}^{n}\widehat{\mu}'[2l(n-k)+x]+\sum_{k=0}^{n-1}\widehat{\nu}'[2l(n-k)+x-l]=\frac{1}{2}\big[\widehat{\phi}'(x)+\widehat{\psi}(x)\big], \tag{14}$$

$$-\sum_{k=0}^{n-1}\widehat{\mu}'[2l(n-k)+x]+\sum_{k=1}^{n}\widehat{\nu}'[2l(n-k)-x+l]=\frac{1}{2}\big[\widehat{\phi}'(x)-\widehat{\psi}(x)\big], \tag{15}$$

where $x \in [0, l]$.

Then, substituting $t = x, m = n - k$ into the first sum and $m = n - 1k$ into the second sum of equation (14) as well as $t = l-x, m = n-1-k$ into the first sum and $m = n-k$ into the second sum of equation (15), we obtain the following equations:

$$-\sum_{m=0}^{n-1}\widehat{\mu}'[2lm+t]+\sum_{m=0}^{n-1}\widehat{\nu}'[l(2m+1)+t]=\frac{1}{2}\big[\widehat{\phi}'(t)+\widehat{\psi}(t)\big], \tag{16}$$

$$-\sum_{m=0}^{n-1}\widehat{\mu}'[l(2m+1)+t]+\sum_{m=0}^{n-1}\widehat{\nu}'[2lm+t]=\frac{1}{2}\big[\widehat{\phi}'(l-t)-\widehat{\psi}(l-t)\big], \tag{17}$$

where $t \in [0, l]$.

In order to express the conditions (16)–(17) in terms of $\mu'(t)$ and $\nu'(t)$, we will use relations $\widehat{\mu}'(t) = \mu'(t) - \widetilde{\mu}'(t)$, $\widehat{\nu}'(t) = \nu'(t) - \widetilde{\nu}'(t)$ and the explicit forms (9)–(10) of the functions $\widetilde{\mu}(t)$ and $\widetilde{\nu}(t)$. So, finally we get the following conditions:

$$-\sum_{m=0}^{n-1}\mu'[2lm+t]+\sum_{m=0}^{n-1}\nu'[l(2m+1)+t]=A(x),$$

$$-\sum_{m=0}^{n-1}\mu'[l(2m+1)+t]+\sum_{m=0}^{n-1}\nu'[2lm+t]=B(x), \tag{18}$$

where $t \in [0, l]$ and

$$A(x)=\frac{1}{2}\big[\widehat{\phi}'(t)-\phi'(t)+\widehat{\psi}(t)-\psi(t)\big]$$

$$B(x)=\frac{1}{2}\big[\widehat{\phi}'(l-t)-\phi'(l-t)-\widehat{\psi}(l-t)+\psi(l-t)\big] \tag{19}$$

Let us find the optimal boundary controls $\mu(t)$ and $\nu(t)$. The optimization problem consists in minimizing the boundary energy integral (6) that can be represented in the form:

$$\int_0^T\{[\mu'(t)]^p+[\nu'(t)]^p\}dt=\int_0^l\sum_0^{2n}\{[\mu'(lm+t)]^p+[\nu'(lm+t)]^p\}dt \tag{20}$$

with the conditions (18) and with the fitting condition (7) which takes the form:

$$\int_0^T \big[\mu'(t)+\nu'(t)\big]dt = \int_0^l \sum_{m=0}^{2n}[\mu'(lm+t)+\nu'(lm+t)]dt =$$

$$= \widehat{\phi}(0)-\phi(0)+\widehat{\phi}(l)-\phi(l) \quad (21)$$

Now, we will formulate a statement that will play an important role in further investigations:

Lemma 1 *We will say that for an arbitrary natural N, the four sets of functions*

$$a_0(x), a_1(x), a_2(x), \ldots, a_N(x) \quad (22)$$

$$b_0(x), b_1(x), b_2(x), \ldots, b_{N-1}(x) \quad (23)$$

$$c_0(x), c_1(x), c_2(x), \ldots, c_N(x) \quad (24)$$

$$d_0(x), d_1(x), d_2(x), \ldots, d_{N-1}(x) \quad (25)$$

belong to the class Ω_p with a fixed $p \geq 1$ if the following conditions are satisfied:

1. *each of the above-mentioned functions belongs to the class $L_p[0,l]$*
2. *for arbitrary real numbers $\alpha_0, \alpha_1, \alpha_2 \ldots \alpha_N$, $\beta_0, \beta_1, \beta_2 \ldots \beta_{N-1}$, $\gamma_0, \gamma_1, \gamma_2 \ldots \gamma_N$, $\delta_0, \delta_1, \delta_2 \ldots \delta_{N-1}$ and arbitrary given functions $A(x), B(x) \in L_p[0,l]$, the following equations are valid:*

$$-\sum_{m=0}^{N}\alpha_m a_m(x)+\sum_{m=0}^{N-1}\delta_m d_m(x)=A(x), \quad -\sum_{m=0}^{N}\beta_m b_m(x)+\sum_{m=0}^{N-1}\gamma_m c_m(x)=B(x) \quad (26)$$

Then, if I denotes the infimum in the class Ω_p of the sum of integrals

$$I = inf_{\Omega_p}\left\{\int_0^l \sum_{m=0}^{N}(|a_m(x)|^p+|c_m(x)|^p)\,dx + \int_0^l \sum_{m=0}^{N-1}(|b_m(x)|^p+|d_m(x)|^p)\,dx\right\} \quad (27)$$

and $I(x)$ denotes the pointwise infimum at each point $x \in [0,2l]$ of the subintegral sums

$$I(x) = \inf_{\text{at point } x} \sum_{m=0}^{N} \left(|a_m(x)|^p + |c_m(x)|^p\right) + \sum_{m=0}^{N-1} \left(|b_m(x)|^p + |d_m(x)|^p\right), \qquad (28)$$

taken for all sets of functions (22)–(25) from the class Ω_p, then, provided that the pointwise infimum (28) is attained over a set of functions from the class Ω_p, the following equation is valid:

$$I = \int_0^l I(x)dx \qquad (29)$$

Proof Let the pointwise infimum (28) be attained over four sets of functions $\widehat{a}_0(x), \widehat{a}_1(x), \widehat{a}_2(x), \ldots, \widehat{a}_N(x)$, $\widehat{b}_0(x), \widehat{b}_1(x), \widehat{b}_2(x), \ldots, \widehat{b}_{N-1}(x), \widehat{c}_0(x), \widehat{c}_1(x), \widehat{c}_2(x), \ldots, \widehat{c}_N(x), \widehat{d}_0(x), \widehat{d}_1(x), \widehat{d}_2(x), \ldots, \widehat{d}_{N-1}(x)$ belonging to the class Ω_p. Then

$$I \leq \int_0^l \left[\sum_{m=0}^{N} \left(|\widehat{a}_m(x)|^p + |\widehat{c}_m(x)|^p\right) + \sum_{m=0}^{N-1} \left(|\widehat{b}_m(x)|^p + |\widehat{d}_m(x)|^p\right) \right] dx = \int_0^l I(x)dx \qquad (30)$$

On the other side, according to the definition of the infimum (27), for any $\epsilon > 0$ there are such sets of functions $\widetilde{a}_0(x), \widetilde{a}_1(x), \widetilde{a}_2(x), \ldots, \widetilde{a}_N(x)$, $\widetilde{b}_0(x), \widetilde{b}_1(x), \widetilde{b}_2(x), \ldots, \widetilde{b}_{N-1}(x)$ $,\widetilde{c}_0(x), \widetilde{c}_1(x), \widetilde{c}_2(x), \ldots, \widetilde{c}_N(x)$, $\widetilde{d}_0(x), \widetilde{d}_1(x), \widetilde{d}_2(x), \ldots, \widetilde{d}_{N-1}(x)$ from the class Ω_p that

$$\int_0^l \left[\sum_{m=0}^{N} \left(|\widetilde{a}_m(x)|^p + |\widetilde{c}_m(x)|^p\right) + \sum_{m=0}^{N-1} \left(|\widetilde{b}_m(x)|^p + |\widetilde{d}_m(x)|^p\right) \right] dx < I + \epsilon \qquad (31)$$

Besides, for almost all the points $x \in [0, l]$ the pointwise infimum $I(x)$ satisfies the inequality:

$$I(x) \leq \left\{ \sum_{m=0}^{N} \left(|\widetilde{a}_m(x)|^p + |\widetilde{c}_m(x)|^p\right) + \sum_{m=0}^{N-1} \left(|\widetilde{b}_m(x)|^p + |\widetilde{d}_m(x)|^p\right) \right\} \qquad (32)$$

Therefore,

$$\int_0^l I(x)dx \leq \int_0^l \left[\sum_{m=0}^{N} \left(|\widetilde{a}_m(x)|^p + |\widetilde{c}_m(x)|^p\right) + \sum_{m=0}^{N-1} \left(|\widetilde{b}_m(x)|^p + |\widetilde{d}_m(x)|^p\right) \right] dx \qquad (33)$$

It follows from relations (30)–(33) that for any $\epsilon > 0$ the following inequalities are valid:

$$I \leq \int_0^l I(x)dx < I + \epsilon, \qquad (34)$$

This completes the proof of the lemma by virtue of the arbitrariness of $\epsilon > 0$.

Due to the proved lemma, the minimizing problem for the integral (20) with conditions (18) reduces itself to finding the pointwise infimum of the sum:

$$\sum_{k=0}^{2n} \left\{ |\mu'(lk+x)|^p + |\nu'(lk+x)|^p \right\} \tag{35}$$

with the same conditions (18).

We will solve this problem by using the Lagrange method. Let us fix a randomly chosen point $x \in [0, l]$ and construct the Lagrange function for this point:

$$\begin{aligned} &\sum_{k=0}^{2n} \left\{ |\mu'(lk+x)|^p + |\nu'(lk+x)|^p \right\} + \\ &\qquad + \lambda_1 \left[-\sum_{k=0}^{n} \mu'(2lk+x) + \sum_{k=0}^{n-1} \nu'[l(2k+1)+x] - A(x) \right] + \\ &\qquad\qquad + \lambda_2 \left[-\sum_{k=0}^{n-1} \mu'[l(2k+1)+x] + \sum_{k=0}^{n} \nu'(2lk+x) - B(x) \right] \end{aligned} \tag{36}$$

Equating the derivative of the function (36) with respect to $\mu'(2lk+x)$ to zero, we obtain:

$$p \times |\mu'(2lk+x)|^{p-1} \times sgn(\mu'(2lk+x)) - \lambda_2 = 0 \tag{37}$$

Then, we differentiate (36) with respect to $\mu'(l(2k+1)) + x)$:

$$p \times |\mu'(l(2k+1)+x)|^{p-1} \times sgn(\mu'(l(2k+1)+x)) - \lambda_1 = 0 \tag{38}$$

Calculating the derivative of (36) with respect to $\nu'(2lk+x)$, we have:

$$p \times |\nu'(2lk+x)|^{p-1} \times sgn(\mu'(2lk+x)) + \lambda_1 = 0 \tag{39}$$

Finally, we differentiate (36) with respect to $\nu'(l(2k+1)+x)$:

$$p \times |\nu'(l(2k+1)+x)|^{p-1} \times sgn(\mu'(l(2k+1)+x)) + \lambda_2 = 0 \tag{40}$$

Having equated (37) to (40) and (38) to (39), respectively, we get:

$$\mu'(2lk+x) = -\nu'(l(2k+1)+x), \quad \mu'(l(2k+1)) + x) = -\nu'(2lk+x) \tag{41}$$

It implies that the optimal boundary control derivatives are $2l$-periodic functions equal to:

$$\mu'(2lk + x) = -\nu'(l(2k + 1) + x) = \frac{A(x)}{2n + 1} = F_1(x), \tag{42}$$

$$\mu'(l(2k + 1)) + x) = -\nu'(2lk + x) = \frac{B(x)}{2n + 1} = F_2(x) \tag{43}$$

In the end, we will find an analytic form for the optimal boundary controls $\mu(t)$ and $\nu(t)$ themselves by using a property of their continuity on the interval $[0; T] = [0; 2ln]$, the equalities $\mu(0) = \phi(0)$, $\nu(0) = \phi(l)$, and the above found derivatives. Thus, we have:

$$\mu(t) = L_1(t) + \alpha_1(t), \quad \nu(t) = -L_2(t) - \alpha_2(t), \quad 0 \le t \le l, \tag{44}$$

$$\mu(t) = L_2(t) + \alpha_2(t), \quad \nu(t) = -L_1(t) - \alpha_1(t), \quad l \le t \le 2l, \tag{45}$$

where for each $s = 1, 2$, the symbol $L_s(t)$ denotes a linear function on the interval $[0; T] = [0; 2ln]$ that has a form

$$L_s(t) = a_s + \frac{t}{2l}\int_0^{2l} F_s(\xi)d\xi \tag{46}$$

with $a_s = \phi(0)$ for $s = 1$ and $a_s = \phi(l)$ for $s = 2$; the symbol $\alpha_s(t)$ denotes an extra term which is a periodic function with a period equal to $2l$ on the interval $[0; T] = [0; 2ln]$ and for any $m = 0, 1, \ldots, n-1, 0 \le t \le 2l$ as well as for $m = n, 0 \le t \le 2l$ has the form:

$$\alpha_s(2lm + t) = \int_0^t F_s(\xi)d\xi - \frac{t}{2l}\int_0^{2l} F_s(\xi)d\xi \tag{47}$$

Conclusion

Thus, this paper provides a modified method that allows one to solve the problem of the optimal boundary controls $\mu(t)$ and $\nu(t)$ of displacements at two ends of a string for a large time interval $T = 2ln$, where $n = 1, 2, 3, \ldots$ It should be noted that the minimization was made in the space L_p with $p \ge 1$. Besides, it was found that the derivatives of the above-mentioned functions are $2l$-periodic functions.

Acknowledgements The work was supported by the grant $MK - 3400.2017.1$.

References

1. Gugat, M., Leugerin, G.: Solutions of Lp-norm-minimal control problems for the wave equation. Comput. Appl. Math. **21**, 227–244 (2002)
2. Il'in, V.A.: On solvability of the mixed problems for the hyperbolic and parabolic equations. Usp. Mat. Nauk **15**(2), 97–154 (1960)
3. Il'in, V.A.: Two-endpoint boundary control of vibrations described by a finite-energy generalized solution of the wave equation. Differ. Equ. **36**(11), 1659–1675 (2000)
4. Il'in, V.A., Moiseev, E.I.: Minimization of the L_p-norm with arbitrary $p \geq 1$ of the derivative of a boundary displacement control on an arbitrary sufficiently large time interval T. Differ. Equ. **42**(11), 1633–1644 (2006)
5. Il'in, V.A., Moiseev E.I. : Optimization of boundary controls by displacements at two ends of a string for an arbitrary sufficiently large time interval. Differ. Equ. **43**(11), 1569–1584 (2007)

Projected Semi-Stochastic Gradient Descent Method with Mini-Batch Scheme Under Weak Strong Convexity Assumption

Jie Liu and Martin Takáč

Abstract We propose a projected semi-stochastic gradient descent method with mini-batch for improving both the theoretical complexity and practical performance of the general stochastic gradient descent method (SGD). We are able to prove linear convergence under weak strong convexity assumption. This requires no strong convexity assumption for minimizing the sum of smooth convex functions subject to a compact polyhedral set, which remains popular across machine learning community. Our PS2GD preserves the low-cost per iteration and high optimization accuracy via stochastic gradient variance-reduced technique, and admits a simple parallel implementation with mini-batches. Moreover, PS2GD is also applicable to dual problem of SVM with hinge loss.

Keywords Stochastic gradient • Variance reduction • Support vector machine (SVM) • Linear convergence • Weak strong convexity

1 Introduction

The problem we are interested in is to minimize a constrained convex problem,

$$\min_{w \in \mathcal{W}} \left\{ F(w) := g(Aw) + q^T w \right\}. \tag{1}$$

where $w \in \mathcal{W} \subseteq \mathbb{R}^d, A \in \mathbb{R}^{n \times d}$, and assume that F can be further written as

$$F(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w). \tag{2}$$

This type of problem is prevalent through machine learning community. Specifically, applications which benefit from efficiently solving this kind of problems

J. Liu (✉) • M. Takáč (✉)
Lehigh University, Bethlehem, PA 18015, USA
e-mail: jie.liu.2018@gmail.com; takac.mt@gmail.com

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_7

include face detection, fingerprint detection, fraud detection for banking systems, image processing, medical image recognition, and self-driving cars, etc. To exploit the problem, we further make the following assumptions:

Assumption 1 *The functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are convex, differentiable and have Lipschitz continuous gradients with constant $L > 0$. That is,*

$$\|\nabla f_i(w_1) - \nabla f_i(w_2)\| \leq L\|w_1 - w_2\|,$$

for all $w_1, w_2 \in \mathbb{R}^d$, where $\| \cdot \|$ is L2 norm.

Assumption 2 *The function $g : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and strongly convex with parameter $\mu > 0$ on its effective domain that is assumed to be open and non-empty, i.e., $\forall z_1, z_2 \in \text{dom}(g) \subseteq \mathbb{R}^n$,*

$$g(z_1) \geq g(z_2) + \nabla g(z_2)^T(z_1 - z_2) + \frac{\mu}{2}\|z_1 - z_2\|^2. \tag{3}$$

Assumption 3 *The constraint set is a compact polyhedral set, i.e.,*

$$\mathcal{W} = \{w \in \mathbb{R}^d : Cw \leq c\}, \text{ where } C \in \mathbb{R}^{m\times d}, c \in \mathbb{R}^m. \tag{4}$$

Remark 1 Problem (1) usually appears in machine learning problems, where A is usually constructed by a sequence of training examples $\{a_i\}_{i=1}^n \subseteq \mathbb{R}^d$. Note that n is the number of data points and d is the number of features. Problem (2) arises as a special form of problem (1) which is a general form in a finite sum structure, which covers empirical risk minimization problems. As indicated in the problem setting, there are two formulations of the problem with different pairs of A and $\mathcal{W}$ given a sequence of labeled training examples $\{(a_i, b_i)\}_{i=1}^n$ where $a_i \in \mathbb{R}^d, b_i \in \mathbb{R}$. Define the set $[m] \overset{\text{def}}{=} \{1, 2, \ldots, m\}$ for any positive integer m.

Type I Primal Setting A commonly recognized structure for this type of problem is to apply (1) to primal problem of finite sum structured problems and to represent g as $g(Aw) = \frac{1}{n}\sum_{i=1}^n g_i(a_i^T w)$ where g_i are $\mathbb{R} \to \mathbb{R}$. In this way, f_i in (1) can be defined as $f_i(w) \overset{\text{def}}{=} g_i(a_i^T w) + q^T w$. We need g_i to have Lipschitz continuous gradients with constants $L/\|a_i\|^2$ to fulfill Assumption 1, i.e.,

$$\begin{aligned}
\|\nabla f_i(w_1) - \nabla f_i(w_2)\| \\
&= \|(a_i \nabla g_i(a_i^T w_1) + q) - (a_i \nabla g_i(a_i^T w_2) + q)\| \\
&= \|a_i\| \|\nabla g_i(a_i^T w_1) - \nabla g_i(a_i^T w_2)\| \leq \|a_i\|(L/\|a_i\|^2)\|a_i^T w_1 - a_i^T w_2\| \\
&= (L/\|a_i\|)\|a_i^T(w_1 - w_2)\| \\
&\leq (L/\|a_i\|)\|a_i\|\|w_1 - w_2\| = L\|w_1 - w_2\|,
\end{aligned}$$

where the last inequality follows from Cauchy Schwartz inequality.

Popular problems in this type from machine learning community are logistic regression and least-squares problems by letting $q = 0$, i.e., $f_i(w) = g_i(a_i^T w) = \log(1 + \exp(-b_i a_i^T w))$ and $f_i(w) = g_i(a_i^T w) = \frac{1}{2}(a_i^T w - b_i)^2$, respectively. These problems are widely used in both regression and classification problems. Our results and analyses are also valid for any convex loss function with Lipschitz continuous gradient.

To deal with overfitting and enforce sparsity to the weights w in real problems, a widely used technique is to either add a regularized term to the minimization problem or enforce constraints to w, for instance,

$$\min_{w \in \mathbb{R}^d} \{f(x) + g(x)\},$$

where $g(x) = \frac{1}{2}\lambda\|x\|^2$ is called a regularizer with regularization parameter λ. A well-known fact is that regularized optimization problem can be equivalent to some constrained optimization problem under proper conditions [11], where the ℓ_2 constrained optimization problem can be denoted as

$$\min_{w \in \mathcal{W}} f(x), \text{ with } \mathcal{W} = \{w \in \mathbb{R}^d : \|w\|^2 \le \lambda\}.$$

The problem of our interest is formulated to solve constrained optimization problem. Under Assumption 3, several popular choices of polyhedral constraints exist, such as $\mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_1 \le \zeta\}$ and $\mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_\infty \le \zeta\}$.

Type II Dual Setting We can also apply (1) to dual form of some special SVM problems. With the same sequence of labeled training examples $\{(a_i, b_i)\}_{i=1}^n$, let us denote $a_i \overset{\text{def}}{=} (a_{i1}, \dots, a_{id})^T \in \mathbb{R}^d$, then an example is the dual problem of SVM with hinge loss, which has the objective function:

$$g(A\alpha) = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n b_i b_j a_i^T a_j \alpha_i \alpha_j = \frac{1}{2}\alpha^T A^T A\alpha = \frac{1}{2}\|A\alpha\|^2 \tag{5}$$

where the ith column of A is $b_i a_i$ so that $[A^T A]_{ij} = (b_i a_i)^T (b_j a_j) = b_i b_j a_i^T a_j$ and we should also know that $A \in \mathbb{R}^{d \times n}$.

By defining $a_s^{(c)} \overset{\text{def}}{=} (b_1 a_{1s}, b_2 a_{2s}, \dots, b_n a_{ns})^T \in \mathbb{R}^n, \forall s \in [d]$, then $a_s^{(c)}$ is the sth row vector of A which is also called the feature vector. By deleting unnecessary $a_s^{(c)}$ corresponds to feature s, we can guarantee that $\|a_s^{(c)}\| \ne 0, \forall s \in [d]$ and easily scale $a_s^{(c)}$; so similar to Type I, Type II problem can also satisfy Assumption 1. Under this type, $\forall i \in [d], f_i$ can be written as

$$f_i(\alpha) = g_i\left((a_s^{(c)})^T \alpha\right) + q^T \alpha$$

with

$$g_i\left((a_s^{(c)})^T\alpha\right) = \frac{d}{2}\|(a_s^{(c)})^T\alpha\|^2 \text{ and } q = (-1,\dots,-1)^T \in \mathbb{R}^n,$$

and $F(\alpha) = \frac{1}{d}\sum_{i=1}^d f_i(\alpha)$.

The dual formulation of SVM with hinge loss is

$$\min_{\alpha\in\mathcal{W}} g(A\alpha) + q^T\alpha$$

with g defined in (5), $A \in \mathbb{R}^{d\times n}, q = (-1,\dots,-1)^T \in \mathbb{R}^n$ and $\mathcal{W} = \{\alpha : \alpha_i \in [0, \lambda n], \forall i \in [n]\} \subset \mathbb{R}^n$, where λ is regularization parameter [32]. This problem satisfies Assumptions 1–3, which is within our problem setting.

Remark 2 Assumption 2 covers a wide range of problems. Note that this is not a strong convexity assumption for the original problem $F(w)$ since the convexity of F is dependent on the data A; nevertheless, the choice of g is independent of A. Popular choices for $g(z)$ have been mentioned in Remark 1, i.e., $\frac{1}{2}\|z - b\|^2$, $\frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-b_i z_i))$ in Type I and $\frac{1}{2}\|z\|^2$ in Type II.

Related Work A great number of methods have been delivered to solve problem (1) during the past years. One of the most efficient algorithms that have been extensively used is FISTA [1]. However, this is considered a full gradient algorithm, and is impractical in large-scale settings with big n since n gradient evaluations are needed per iteration. Two frameworks are imposed to reduce the cost per iteration—stochastic gradient algorithms [8, 20, 30–32, 35] and randomized coordinate descent methods [4, 5, 15–18, 22, 25–27, 29]. However, even under strong convexity assumption, the convergence rates in expectation are only sublinear, while full gradient methods can achieve linear convergence rates [23, 34]. It has been widely accepted that the slow convergence in standard stochastic gradient algorithms arises from its unstable variance of the stochastic gradient estimates. To deal with this issue, various variance-reduced techniques have been applied to stochastic gradient algorithms [3, 9, 12–14, 24, 28, 34]. These algorithms are proved to achieve linear convergence rate under strong convexity condition, and remain low-cost in gradient evaluations per iteration. As a prior work on the related topic, Zhang et al. [37] is the first analysis of stochastic variance reduced gradient method with constraints, although their convergence rate is worse than our work.

The topic whether an algorithm can achieve linear convergence without strong convexity assumptions remains desired in machine learning community. Recently, the concept of *weak strong convexity property* has been proposed and developed based on *Hoffman bound* [6, 7, 15, 33, 36]. In particular, Ji and Wright [15] *first* proposed the concept as *optimally strong convexity* in March 2014.[1] Necoara [19] established a general framework for weak non-degeneracy assumptions which cover the weak strong convexity. Karimi et al. [10] summarize the relaxed conditions of strongly convexity and analyses their differences and connections; meanwhile, they provide proximal versions of global error bound and weak strong convexity conditions, as well as the linear convergence of proximal gradient descent under these conditions. Hui [36] also provides a complete of summary on weak strong convexity, including their connections. This kind of methodology could help to improve the theoretical analyses for series of fast convergent algorithms and to apply those algorithms to a broader class of problems.

Our contributions In this paper, we combine the stochastic gradient variance-reduced technique and weak strong convexity property based on Hoffman bound to derive a projected semi-stochastic gradient descent method (PS2GD). This algorithm enjoys three benefits. First, PS2GD promotes the best convergence rate for solving (1) without strong convexity assumption from sub-linear convergence to linear convergence in theory. Second, stochastic gradient variance-reduced technique in PS2GD helps to maintain the low-cost per iteration of the standard stochastic gradient method. Last, PS2GD comes with a mini-batch scheme, which admits a parallel implementation, suggesting probably speedup in clocktime in an HPC environment.

Moreover, we have shown in Remark 1 that our framework covers the dual form of SVM problem with hinge loss. Instead of applying SDCA [28, 29], we can also apply PS2GD as a stochastic dual gradient method.

2 Projected Algorithms and PS2GD

A common approach to solve (1) is to use *gradient projection methods* [2, 6, 33] by forming a sequence $\{y_k\}$ via

$$y_{k+1} = \arg\min_{w\in\mathcal{W}} \left[U_k(w) \overset{\text{def}}{=} F(y_k) + \nabla F(y_k)^T(w - y_k) + \frac{1}{2h}\|w - y_k\|^2 \right],$$

where U_k is an upper bound on F if $h > 0$ is a stepsize parameter satisfying $h \le \frac{1}{L}$. This procedure can be equivalently written using the *projection operator* as follows:

$$y_{k+1} = \text{proj}_{\mathcal{W}}(y_k - h\nabla F(y_k)),$$

[1] Even though the concept was first proposed by Liu and Wright in [15] as *optimally strong convexity*, to emphasize it as an extended version of strong convexity, we use the term *weak strong convexity* as in [6] throughout our paper.

where

$$\operatorname{proj}_{\mathcal{W}}(z) \overset{\text{def}}{=} \arg\min_{w \in \mathcal{W}} \{\tfrac{1}{2}\|w - z\|^2\}.$$

In large-scale setting, instead of updating the gradient by evaluating n component gradients, it is more efficient to consider the *projected stochastic gradient descent* approach, in which the proximal operator is applied to a stochastic gradient step:

$$y_{k+1} = \operatorname{proj}_{\mathcal{W}}(y_k - hG_k), \tag{6}$$

where G_k is a stochastic estimate of the gradient $\nabla F(y_k)$. Of particular relevance to our work are the SVRG [9, 34] and S2GD [13] methods where the stochastic estimate of $\nabla F(y_k)$ is of the form

$$G_k = \nabla F(w) + (\nabla f_i(y_k) - \nabla f_i(w)), \tag{7}$$

where w is an "old" reference point for which the gradient $\nabla F(w)$ was already computed in the past, and $i \in [n]$ is picked uniformly at random. A mini-batch version of similar form is introduced as mS2GD [12] with

$$G_k = \nabla F(w) + \frac{1}{b}\sum_{i \in A_k}(\nabla f_i(y_k) - \nabla f_i(w)), \tag{8}$$

where the mini-batch $A_k \subset [n]$ of size b is chosen uniformly at random. Note that the gradient estimate (7) is a special case of (8) with $b = 1$. Notice that G_k is an unbiased estimate of the gradient:

$$\mathbf{E}_i[G_k] \overset{(8)}{=} \nabla F(w) + \frac{1}{b} \cdot \frac{b}{n}\sum_{i=1}^{n}(\nabla f_i(y_k) - \nabla f_i(w)) \overset{(2)}{=} \nabla F(y_k).$$

Methods such as SVRG [9, 34], S2GD [13] and mS2GD [12] update the points y_k in an inner loop, and the reference point x in an outer loop. This ensures that G_k has low variance, which ultimately leads to extremely fast convergence.

We now describe the PS2GD method in mini-batch scheme (Algorithm 1).

The algorithm includes both outer loops indexed by epoch counter k and inner loops indexed by t. To begin with, the algorithm runs each epoch by evaluating v_k, which is the full gradient of F at w_k, then it proceeds to produce t_k — the number of iterations in an inner loop, where $t_k = t \in \{1, 2, \ldots, M\}$ is chosen uniformly at random.

Algorithm 1 PS2GD

```
 1: Input: $M$ (max # of stochastic steps per epoch); $h > 0$ (stepsize); $w_0 \in \mathbb{R}^d$ (starting point);
    linear coefficients $q \in \mathbb{R}^d$; mini-batch size $b \in [n]$
 2: for $k = 0, 1, 2, \dots$ do
 3:     Compute and store $v_k \leftarrow \nabla F(w_k) = \frac{1}{n}\sum_i \nabla f_i(w_k) = \frac{1}{n}\sum_i a_i \nabla g_i(a_i^T w_k) + q$
 4:     Initialize the inner loop: $y_{k,0} \leftarrow w_k$
 5:     Let $t_k \leftarrow t \in \{1, 2, \dots, M\}$ uniformly at random
 6:     for $t = 0$ to $t_k - 1$ do
 7:         Choose mini-batch $A_{kt} \subset [n]$ of size $b$ uniformly at random
 8:         Compute a stochastic estimate of $\nabla F(y_{k,t})$:
 9:                 $G_{k,t} \leftarrow v_k + \frac{1}{b}\sum_{i \in A_{kt}} [\nabla g_i(a_i^T y_{k,t}) - \nabla g_i(a_i^T w_k)] a_i$
10:         $y_{k,t+1} \leftarrow \text{proj}_{\mathcal{W}}(y_{k,t} - hG_{k,t})$
11:     end for
12:     Set $w_{k+1} \leftarrow y_{k,t_k}$
13: end for
```

Subsequently, we run t_k iterations in the inner loop — the main part of our method (Steps 8–10). Each new iterate is given by the projected update (6); however, with the stochastic estimate of the gradient $G_{k,t}$ in (8), which is formed by using a *mini-batch* $A_{kt} \subset [n]$ of size $|A_{kt}| = b$. Each inner iteration takes *2b component gradient evaluations*.[2]

3 Complexity Result

In this section, we state our main complexity results and comment on how to optimally choose the parameters of the method. Denote $\mathcal{W}^* \subseteq \mathcal{W}$ as the set of optimal solutions. Then following ideas from the proof of Theorem 1 in [12], we conclude the following theorem. In section "Proof of Theorem 1" in Appendix, we provide the complete proof.

Theorem 1 *Let Assumptions 1, 2 and 3 be satisfied and let* $w_* \in \mathcal{W}^*$ *be any optimal solution to* (1). *In addition, assume that the stepsize satisfies* $0 < h \leq \min\left\{\frac{1}{4L\alpha(b)}, \frac{1}{L}\right\}$ *and that M is sufficiently large so that*

$$\rho \overset{def}{=} \frac{\beta + 4\mu h^2 L\alpha(b)(M+1)}{\mu h\,(1 - 4hL\alpha(b))\,M} < 1, \tag{9}$$

[2] It is possible to finish each iteration with only b evaluations for component gradients, namely $\{\nabla f_i(y_{k,t})\}_{i \in A_{kt}}$, at the cost of having to store $\{\nabla f_i(x_k)\}_{i \in [n]}$, which is exactly the way that SAG [14] works. This speeds up the algorithm; nevertheless, it is impractical for big n.

where $\alpha(b) = \frac{m-b}{b(m-1)}$ *and* β *is some finite positive number dependent on the structure of* A *in* (1) *and* C *in* (4).[3] *Then PS2GD has linear convergence in expectation:*

$$\mathbf{E}(F(w_k) - F(w_*)) \leq \rho^k (F(w_0) - F(w_*)).$$

Remark 3 Consider the special case of strong convexity, when F is strongly convex with parameter μ_F,

$$F(w) - F(w_*) \geq \frac{\mu_F}{2} \|w - w_*\|^2,$$

then we have

$$\rho = \frac{1}{h\mu_F(1 - 4hL\alpha(b))M} + \frac{4hL\alpha(b)\,(M+1)}{(1 - 4hL\alpha(b))M}, \tag{10}$$

which recovers the convergence rate from [12] and it is better than [34] computationally since their algorithm requires computation of an average over M points, while we continue with the last point, which is computationally more efficient.

In the special case when $b = 1$ we get $\alpha(b) = 1$, and the rate given by (9) exactly recovers the rate achieved by VRPSG [6] (in the case when the Lipschitz constants of ∇f_i are all equal).

From Theorem 1, it is not difficult to conclude the following corollary, which aims to detect the effects of mini-batch on PS2GD. The proof of the corollary follows from the proof of Theorem 2 in [12], and thus is omitted.

Corollary 2 *Fix target decrease* $\rho_* \geq \rho$, *where* ρ *is given by* (9) *and* $\rho_* \in (0, 1)$. *If we consider the mini-batch size* b *to be fixed and define the following quantity,*

$$\tilde{h}^b \overset{def}{=} \sqrt{\beta^2 \left(\frac{1+\rho}{\rho\mu}\right)^2 + \frac{\beta}{4\mu\alpha(b)L}} - \frac{\beta(1+\rho)}{\rho\mu},$$

then the choice of stepsize h_*^b *and size of inner loops* m_*^b, *which minimizes the work done—the number of gradients evaluated—while having* $\rho \leq \rho_*$, *is given by the following statements.*

If $\tilde{h}^b \leq \frac{1}{L}$, *then* $h_*^b = \tilde{h}^b$ *and*

$$m_*^b = \frac{2\kappa}{\rho} \left\{ \left(1 + \frac{1}{\rho}\right) 4\alpha(b) + \sqrt{\frac{4\alpha(b)}{\kappa} + \left(1 + \frac{1}{\rho}\right)^2 [4\alpha(b)]^2} \right\}, \tag{11}$$

[3] We only need to prove the existence of β and do not need to evaluate its value in practice. Lemma 4 provides the existence of β.

where $\kappa \overset{def}{=} \frac{\beta L}{\mu}$ *is the condition number; otherwise,* $h_*^b = \frac{1}{L}$ *and*

$$m_*^b = \frac{\kappa + 4\alpha(b)}{\rho - 4\alpha(b)(1+\rho)}. \tag{12}$$

If $m_*^b < m_*^1/b$ for some $b > 1$, then mini-batching can help us reach the target decrease ρ_* with fewer component gradient evaluations. Equation (11) suggests that as long as the condition $\tilde{h}^b \leq \frac{1}{L}$ holds, m_*^b is decreasing at a rate roughly faster than $1/b$. Hence, we can attain the same decrease with no more work, compared to the case when $b = 1$.

4 Numerical Experiments

In this section, we deliver preliminary numerical experiments to substantiate the effectiveness and efficiency of PS2GD. We experiment mainly on constrained logistic regression problems introduced in Remark 1 (Type I), i.e.,

$$\min_{w \in \mathcal{W}} \{F(w) := \frac{1}{n}\sum_{i=1}^{n} \log[1 + \exp(-b_i a_i^T w)]\}, \text{ with } \mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_\infty \leq \zeta\}, \tag{13}$$

where $\{(a_i, b_i)\}_{i=1}^n$ is a set of training data points with $a_i \in \mathbb{R}^d$ and $b_i \in \{+1, -1\}$ for binary classification problems.

We performed experiments on three publicly available binary classification datasets, namely *rcv1, news20*[4] and *astro-ph*.[5] In a logistic regression problem, the Lipschitz constant of function f_i can be derived as $L_i = \|a_i\|^2/4$. We assume (Assumption 1) the same constant L for all functions since all data points can be scaled to have proper Lipschitz constants. We set the bound of the norm $\zeta = 0.1$ in our experiments. A summary of the three datasets is given in Table 1, including the sizes n, dimensions d, their sparsity as proportion of nonzero elements and Lipschitz constants L.

Table 1 Summary of datasets used for experiments

Dataset	n	d	Sparsity	L
rcv1	20,242	47,236	0.1568%	0.2500
news20	19,996	1,355,191	0.0336%	0.2500
astro-ph	62,369	99,757	0.0767%	0.2500

[4] *rcv1* and *news20* are available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[5] Available at http://users.cecs.anu.edu.au/~xzhang/data/.

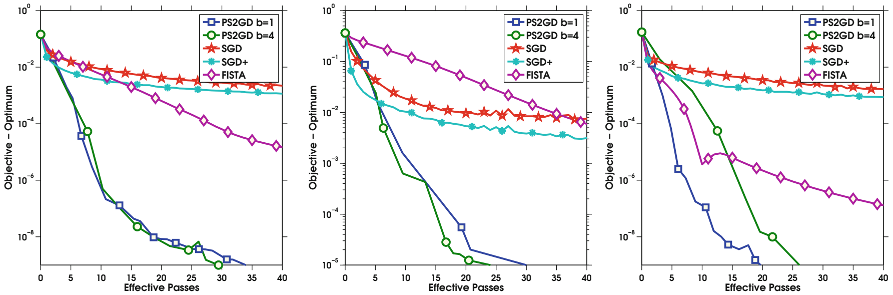


Fig. 1 Comparison of different algorithms on *rcv1* (*left*), *news20* (*middle*) and *astro-ph* (*right*)

We implemented the following prevalent algorithms. SGD, SGD+ and FISTA are only enough to demonstrate sub-linear convergence without any strong convexity assumption.

1. **PS2GD b=1**: the PS2GD algorithm without mini-batch, i.e., with mini-batch size $b = 1$. Although a safe step-size is given in our theoretical analyses in Theorem 1, we experimented with various step-sizes and used the constant step-size that gave the best performance.
2. **PS2GD b=4**: the PS2GD algorithm with mini-batch size $b = 4$. We used the constant step-size that gave the best performance.
3. **SGD**: the proximal stochastic gradient descent method with the constant step-size which gave the best performance in hindsight.
4. **SGD+**: the proximal stochastic gradient descent with adaptive step-size $h = h_0/(k + 1)$, where k is the number of effective passes and h_0 is some initial constant step-size. We used h_0 which gave the best performance in hindsight.
5. **FISTA**: fast iterative shrinkage-thresholding algorithm proposed in [1]. This is considered as the full gradient descent method in our experiments.

In Fig. 1, each effective pass is considered as n component gradient evaluations, where each f_i in (2) is named as a component function, and each full gradient evaluation counts as one effective pass. The y-axis is the distance from the current function value to the optimum, i.e., $F(w) - F(w_*)$. The nature of SGD suggests unstable positive variance for stochastic gradient estimates, which induces SGD to oscillate around some threshold after a certain number of iterations with constant step-sizes. Even with decreasing step-sizes over iterations, SGD are still not able to achieve high accuracy (shown as SGD+ in Fig. 1). However, by incorporating a variance-reduced technique for stochastic gradient estimate, PS2GD maintains a reducing variance over iterations and can achieve higher accuracy with fewer iterations. FISTA is worse than PS2GD due to large numbers of component gradient evaluations per iteration.

Meantime, increase of mini-batch size up to some threshold does not hurt the performance of PS2GD and PS2GD can be accelerated in the benefit of simple parallelism with mini-batches. Figure 2 compares the best performances of PS2GD

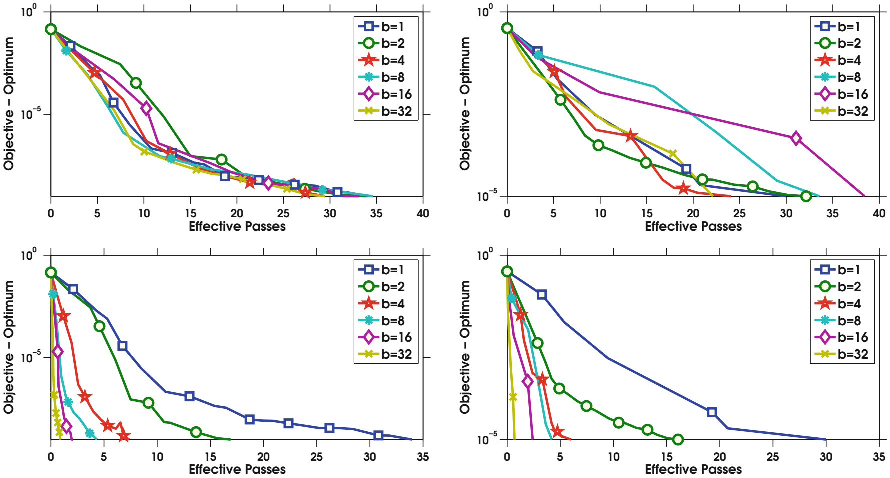


Fig. 2 Comparison of PS2GD with different mini-batch sizes on *rcv1* (*left*) and *news20* (*right*)

with different mini-batch sizes on datasets *rcv1* and *news20*. Numerical results on *rcv1* with no parallelism imply that PS2GD with $b = 2, 4, 8, 16, 32$ are comparable or sometimes even better than PS2GD without any mini-batch ($b = 1$); while on *news20*, PS2GD with $b = 4, 32$ are better than and the others are worse but comparable to PS2GD with $b = 1$. Moreover, with parallelism, the results are promising. The bottom row shows results of ideal speedup by parallelism, which would be achievable if and only if we could always evaluate the b gradients efficiently in parallel.[6]

5 Conclusion

In this paper, we have proposed a mini-batch projected semi-stochastic gradient descent method, for minimizing the sum of smooth convex functions subject to a compact polyhedral set. This kind of constrained optimization problems arise in inverse problems in signal processing and modern statistics, and is popular among the machine learning community. Our PS2GD algorithm combines the variance-reduced technique for stochastic gradient estimates and the mini-batch scheme, which ensure a high accuracy for PS2GD and speedup the algorithm. Mini-batch technique applied to PS2GD also admits a simple implementation for parallelism in HPC environment. Furthermore, in theory, PS2GD has a great improvement that

[6] In practice, it is impossible to ensure that evaluating different component gradients takes the same time; however, Fig. 2 implies the potential and advantage of applying mini-batch scheme with parallelism.

it requires no strong convexity assumption of either data or objective function but maintains linear convergence; while prevalent methods under non-strongly convex assumption only achieves sub-linear convergence. PS2GD, belonging to the gradient descent algorithms, has also been shown applicable to dual problem of SVM with hinge loss, which is usually efficiently solved by dual coordinate ascent methods. Comparisons to state-of-the-art algorithms suggest PS2GD is competitive in theory and faster in practice even without parallelism. Possible implementation in parallel and adaptiveness for sparse data imply its potential in industry.

Acknowledgements This research of Jie Liu and Martin Takáč was supported by National Science Foundation grant CCF-1618717. We would like to thank Ji Liu for his helpful suggestions on related works.

Appendix 1: Technical Results

Lemma 1 *Let set $\mathcal{W} \subseteq \mathbb{R}^d$ be nonempty, closed, and convex, then for any $x, y \in \mathbb{R}^d$,*

$$\| \operatorname{proj}_{\mathcal{W}}(x) - \operatorname{proj}_{\mathcal{W}}(y)\| \leq \|x - y\|.$$

Note that the above contractiveness of projection operator is a standard result in optimization literature. We provide proof for completeness.

Inspired by Lemma 1 in [34], we derive the following lemma for projected algorithms.

Lemma 2 (Modified Lemma 1 in [34]) *Let Assumption 1 hold and let $w_* \in \mathcal{W}^*$ be any optimal solution to Problem (1). Then for any feasible solution $w \in \mathcal{W}$, the following holds:*

$$\frac{1}{n}\sum_{i=1}^{n} \|a_i[\nabla g_i(a_i^T w) - \nabla g_i(a_i^T w_*)]\| = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(w) - \nabla f_i(w_*)\| \leq 2L[F(w) - F(w_*)]. \tag{14}$$

Lemmas 3 and 4 come from [12] and [33], respectively. Please refer to the corresponding references for complete proofs.

Lemma 3 (Lemma 4 in [12]) *Let $\{\xi_i\}_{i=1}^n$ be a collection of vectors in $\mathbb{R}^d$ and $\mu \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \xi_i \in \mathbb{R}^d$. Let $\hat{S}$ be a τ-nice sampling. Then*

$$\mathbf{E}\left[\left\|\frac{1}{\tau}\sum_{i\in\hat{S}} \xi_i - \mu\right\|^2\right] = \frac{1}{n\tau}\frac{n-\tau}{(n-1)}\sum_{i=1}^{n} \|\xi_i\|^2 . \tag{15}$$

Following from the proof of Corollary 3 in [34], by applying Lemma 3 with $\xi_i := \nabla f_i(y_{k,t-1}) - \nabla f_i(w_k) = a_i[\nabla g_i(a_i^T y_{k,t-1}) - \nabla g_i(a_i^T w_k)]$ and Lemma 2, we have the bound for variance as follows.

Theorem 3 (Bounding Variance) *Considering the definition of $G_{k,t}$ in Algorithm 1, conditioned on $y_{k,t}$, we have $\mathbf{E}[G_{k,t}] = \frac{1}{n}\sum_{i=1}^{n} \nabla g_i(y_{k,t}) + q = \nabla F(y_{k,t})$ and the variance satisfies,*

$$\mathbf{E}\left[\|G_{k,t} - \nabla F(y_{k,t})\|^2\right] \leq \underbrace{\frac{n-b}{b(n-1)}}_{\alpha(b)} 4L[F(y_{k,t}) - F(w_*) + F(w_k) - F(w_*)]. \quad (16)$$

Lemma 4 (Hoffman Bound, Lemma 15 in [33]) *Consider a non-empty polyhedron*

$$\{w_* \in \mathbb{R}^d | Cw_* \leq c, Aw_* = r\}.$$

For any w, there is a feasible point w_* such that

$$\|w - w_*\| \leq \theta(A, C) \left\| \begin{matrix} [Cw - c]^+ \\ Aw - r \end{matrix} \right\|,$$

where $\theta(A, C)$ is independent of x,

$$\theta(A, C) = \sup_{u,v} \left\{ \left\| \begin{matrix} u \\ v \end{matrix} \right\| \middle| \begin{matrix} \|C^T u + A^T v\| = 1, u \geq 0. \textit{ The corresponding rows of } C, A \\ \textit{to } u, v\textit{'s non-zero elements are linearly independent.} \end{matrix} \right\} \quad (17)$$

Lemma 5 (Weak Strong Convexity) *Let $w \in \mathcal{W} := \{w \in \mathbb{R}^d : Cw \leq c\}$ be any feasible solution (Assumption 3) and $w_* = \mathrm{proj}_{\mathcal{W}^*}(w)$ which is an optimal solution for Problem (1). Then under Assumptions 2–3, there exists a constant $\beta > 0$ such that for all $w \in \mathcal{W}$, the following holds,*

$$F(w) - F(w_*) \geq \frac{\mu}{2\beta}\|w - w_*\|^2,$$

where μ is defined in Assumption 2. β can be evaluated by $\beta = \theta^2$ where θ is defined in (17).

Appendix 2: Proofs

Proof of Lemma 1

For any $x, y \in \mathbb{R}^d$, by Projection Theorem, the following holds:

$$[y - \text{proj}_{\mathcal{W}}(y)]^T[\text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)] \leq 0, \tag{18}$$

similarly, by symmetry, we have

$$[x - \text{proj}_{\mathcal{W}}(x)]^T[\text{proj}_{\mathcal{W}}(y) - \text{proj}_{\mathcal{W}}(x)] \leq 0. \tag{19}$$

Then (18) + (19) gives

$$[(\text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)) - (x - y)]^T[\text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)] \leq 0,$$

or equivalently,

$$\| \text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)\|^2 \leq (x - y)^T[\text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)],$$

and by Cauchy-Schwarz inequality, we have

$$\| \text{proj}_{\mathcal{W}}(y) - \text{proj}_{\mathcal{W}}(x)\| \leq \|x - y\|,$$

when $\text{proj}_{\mathcal{W}}(x) = \text{proj}_{\mathcal{W}}(y)$ are distinct; in addition, when $\text{proj}_{\mathcal{W}}(x) = \text{proj}_{\mathcal{W}}(y)$, the above inequality also holds. Hence, for any $x, y \in \mathbb{R}^d$, which is the same to

$$\| \text{proj}_{\mathcal{W}}(x) - \text{proj}_{\mathcal{W}}(y)\| \leq \|x - y\|.$$

Proof of Lemma 2

For any $i \in \{1, \dots, n\}$, consider the function

$$\phi_i(w) = f_i(w) - f_i(w_*) - \nabla f_i(w_*)^T(w - w_*), \tag{20}$$

then it should be obvious that $\nabla\phi_i(w_*) = \nabla f_i(w_*) - \nabla f_i(w_*) = 0$, hence $\min_{w\in\mathbb{R}^d} \phi_i(w) = \phi_i(w_*)$ because of the convexity of f_i. By Assumption 1 and Remark 1, $\nabla\phi_i(w)$ is Lipschitz continuous with constant L, hence by Theorem 2.1.5 from [21] we have

$$\frac{1}{2L}\|\nabla\phi_i(w)\|^2 \leq \phi_i(w) - \min_{w\in\mathbb{R}^l} \phi_i(w) = \phi_i(w) - \phi_i(w_*) = \phi_i(w),$$

which, by (20), suggests that

$$\|\nabla f_i(w) - \nabla f_i(w_*)\|^2 \le 2L[f_i(w) - f_i(w_*) - \nabla f_i(w_*)^T(w - w_*)].$$

By averaging the above equation over $i = 1, \ldots, n$ and using the fact that $F(w) = \frac{1}{n}\sum_{i=1}^n f_i(w)$, we have

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(w) - \nabla f_i(w_*)\|^2 \le 2L[F(w) - F(w_*) - \nabla F(w_*)^T(w - w_*)],$$

which, together with $\nabla F(w_*)^T(w - w_*) \ge 0$ indicated by the optimality of w_* for Problem (1), completes the proof for Lemma 2.

Proof of Lemma 5

First, we will prove by contradiction that there exists a unique r such that $\mathcal{W}^* = \{w \in \mathbb{R}^d : Cw \le c, Aw = r\}$ which is non-empty. Assume that there exist distinct $w_1, w_2 \in \mathcal{W}^*$ such that $Aw_1 \ne Aw_2$. Let us define the optimal value to be F^* which suggests that $F^* = F(w_1) = F(w_2)$. Moreover, convexity of function F and feasible set $\mathcal{W}$ suggests the convexity of $\mathcal{W}^*$, then $\frac{1}{2}(w_1 + w_2) \in \mathcal{W}^*$. Therefore,

$$\begin{aligned} F^* = F\left(\frac{1}{2}(w_1 + w_2)\right) &\overset{(1)}{=} g\left(A\frac{1}{2}(w_1 + w_2)\right) + \frac{1}{2}q^T(w_1 + w_2) \\ &= g\left(\frac{1}{2}Aw_1 + \frac{1}{2}Aw_2\right) + \frac{1}{2}q^T(w_1 + w_2). \end{aligned} \tag{21}$$

Strong convexity indicated in Assumption 2 suggests that

$$\begin{aligned} F^* &= \frac{1}{2}(F(w_1) + F(w_2)) \overset{(1)}{=} \frac{1}{2}[g(Aw_1) + q^T w_1] + \frac{1}{2}[g(Aw_2) + q^T w_2] \\ &= \left(\frac{1}{2}g(Aw_1) + \frac{1}{2}g(Aw_2)\right) + \frac{1}{2}q^T(w_1 + w_2) \\ &> g\left(\frac{1}{2}Aw_1 + \frac{1}{2}Aw_2\right) + \frac{1}{2}q^T(w_1 + w_2) \overset{(21)}{=} F^*, \end{aligned}$$

which is a contradiction, so there exists a unique r such that $\mathcal{W}^*$ can be represented by $\{w \in \mathbb{R}^d : Cw \le c, Aw = r\}$.

For any $w \in \mathcal{W} = \{x \in \mathbb{R}^d : Cw \leq c\}$, $[Cw - c]^+ = 0$, then by Hoffman's bound in Lemma 4, for any $w \in \mathcal{W}$, there exists $w' \in \mathcal{W}^*$ and a constant $\theta > 0$ defined in (17), dependent on A and C, such that

$$\|w - w'\| \leq \theta \left\| \begin{matrix} [Cw - c]^+ \\ Aw - r \end{matrix} \right\| = \theta \|Aw - r\| = \theta \|Aw - Aw_*\|, \forall w_* \in \mathcal{W}^*. \quad (22)$$

Being aware of that by choosing $w_* = \text{proj}_{\mathcal{W}_*}(w)$, we have that $\|w - w_*\| \leq \|w - w'\|$, which suggests that

$$\|w - w_*\| \leq \|w - w'\| \overset{(22)}{\leq} \theta \|Aw - Aw_*\|,$$

or equivalently,

$$\|Aw - Aw_*\|^2 \geq \frac{1}{\beta} \|w - w_*\|^2, \forall w_* \in \mathcal{W}^*, \quad (23)$$

where $\beta = \theta^2 > 0$.

Optimality of w_* for Problem (1) suggests that

$$\nabla F(w_*)^T (w - w_*) \overset{(1)}{=} [A^T g(Aw_*) + q]^T (w - w_*) \geq 0, \quad (24)$$

then we can conclude the following:

$$g(Aw) \overset{(3)}{\geq} g(Aw_*) + \nabla g(Aw_*)^T (Aw - Aw_*) + \frac{\mu}{2} \|Aw - Aw_*\|^2, \quad (25)$$

which, by considering $F(w) = g(Aw) + q^T w$ in Problem (1), is equivalent to

$$\begin{aligned} F(w) - F(w_*) &\overset{(1)}{=} g(Aw) - g(Aw_*) + q^T (w - w_*) \\ &\overset{(25)}{\geq} [A^T \nabla g(Aw_*) + q]^T (w - w_*) + \frac{\mu}{2} \|Aw - Aw_*\|^2 \\ &\overset{(24)}{\geq} \frac{\mu}{2} \|Aw - Aw_*\|^2 \\ &\overset{(23)}{\geq} \frac{\mu}{2\beta} \|w - w_*\|^2. \end{aligned}$$

Proof of Theorem 1

The proof is following the steps in [12, 34]. For convenience, let us define the stochastic gradient mapping

$$d_{k,t} = \frac{1}{h}(y_{k,t} - y_{k,t+1}) = \frac{1}{h}(y_{k,t} - \text{proj}_{\mathcal{W}}(y_{k,t} - hG_{k,t})), \tag{26}$$

then the iterate update can be written as

$$y_{k,t+1} = y_{k,t} - hd_{k,t}.$$

Let us estimate the change of $\|y_{k,t+1} - w_*\|$. It holds that

$$\begin{aligned} \|y_{k,t+1} - w_*\|^2 &= \|y_{k,t} - hd_{k,t} - w_*\|^2 \\ &= \|y_{k,t} - w_*\|^2 - 2hd_{k,t}^T(y_{k,t} - w_*) + h^2\|d_{k,t}\|^2. \end{aligned} \tag{27}$$

By the optimality condition of $y_{k,t+1} = \text{proj}_{\mathcal{W}}(y_{k,t} - hG_{k,t}) = \arg\min_{w \in \mathcal{W}}\{\frac{1}{2}\|w - (y_{k,t} - hG_{k,t})\|^2\}$, we have

$$[y_{k,t+1} - (y_k - hG_{k,t})]^T(w^* - y_{k,t+1}) \geq 0,$$

then the update $y_{k,t+1} = y_{k,t} - hd_{k,t}$ suggests that

$$G_{k,t}^T(w^* - y_{k,t+1}) \geq d_{k,t}^T(w^* - y_{k,t+1}). \tag{28}$$

Moreover, Lipschitz continuity of the gradient of F implies that

$$F(y_{k,t}) \geq F(y_{k,t+1}) - \nabla F(y_{k,t})^T(y_{k,t+1} - y_{k,t}) - \frac{L}{2}\|y_{k,t+1} - y_{k,t}\|^2. \tag{29}$$

Let us define the operator $\Delta_{k,t} = G_{k,t} - \nabla F(y_{k,t})$, so

$$\nabla F(y_{k,t}) = G_{k,t} - \Delta_{k,t} \tag{30}$$

Convexity of F suggests that

$$\begin{aligned} &F(w^*) \geq F(y_{k,t}) + \nabla F(y_{k,t})^T(w^* - y_{k,t}) \\ &\overset{(29)}{\geq} F(y_{k,t+1}) - \nabla F(y_{k,t})^T(y_{k,t+1} - y_{k,t}) - \frac{L}{2}\|y_{k,t+1} - y_{k,t}\|^2 + \nabla F(y_{k,t})^T(w^* - y_{k,t}) \\ &= F(y_{k,t+1}) - \frac{L}{2}\|y_{k,t+1} - y_{k,t}\|^2 + \nabla F(y_{k,t})^T(w^* - y_{k,t+1}) \end{aligned}$$

$$
\begin{aligned}
&\overset{(26),\,(30)}{=} F(y_{k,t+1}) - \frac{Lh^2}{2}\|d_{k,t}\|^2 + (G_{k,t} - \Delta_{k,t})^T(w^* - y_{k,t+1}) \\
&\overset{(28)}{\geq} F(y_{k,t+1}) - \frac{Lh^2}{2}\|d_{k,t}\|^2 + d_{k,t}^T(w^* - y_{k,t} + y_{k,t} - y_{k,t+1}) - \Delta_{k,t}^T(w^* - y_{k,t+1}) \\
&\overset{(26)}{=} F(y_{k,t+1}) - \frac{Lh^2}{2}\|d_{k,t}\|^2 + d_{k,t}^T(w^* - y_{k,t} + hd_{k,t}) - \Delta_{k,t}^T(w^* - y_{k,t+1}) \\
&= F(y_{k,t+1}) + \frac{h}{2}(2 - Lh)\|d_{k,t}\|^2 + d_{k,t}^T(w^* - y_{k,t}) - \Delta_{k,t}^T(w^* - y_{k,t+1}) \\
&\overset{h\leq 1/L}{\geq} F(y_{k,t+1}) + \frac{h}{2}|d_{k,t}\|^2 + d_{k,t}^T(w^* - y_{k,t}) - \Delta_{k,t}^T(w^* - y_{k,t+1}),
\end{aligned}
$$

then equivalently,

$$
-d_{k,t}^T(y_{k,t} - w_*) + \frac{h}{2}\|d_{k,t}\|^2 \leq F(w_*) - F(y_{k,t+1}) - \Delta_{k,t}^T(y_{k,t+1} - w_*). \tag{31}
$$

Therefore,

$$
\begin{aligned}
\|y_{k,t+1} - w_*\|^2 &\overset{(27),(31)}{\leq} \|y_{k,t} - w_*\|^2 + 2h\left(F(w_*) - F(y_{k,t+1}) - \Delta_{k,t}^T(y_{k,t+1} - w_*)\right) \\
&= \|y_{k,t} - w_*\|^2 - 2h\Delta_{k,t}^T(y_{k,t+1} - w_*) - 2h[F(y_{k,t+1}) - F(w_*)].
\end{aligned} \tag{32}
$$

In order to bound $-\Delta_{k,t}^T(y_{k,t+1} - w_*)$, let us define the proximal full gradient update as[7]

$$
\bar{y}_{k,t+1} = \mathrm{proj}_{\mathcal{W}}(y_{k,t} - h\nabla F(y_{k,t})),
$$

with which, by using Cauchy-Schwartz inequality and Lemma 1, we can conclude that

$$
\begin{aligned}
-\Delta_{k,t}^T(y_{k,t+1} - w_*) &= -\Delta_{k,t}^T(y_{k,t+1} - \bar{y}_{k,t+1}) - \Delta_{k,t+1}^T(\bar{y}_{k,t+1} - w_*) \\
&= -\Delta_{k,t}^T\,[\mathrm{proj}_{\mathcal{W}}(y_{k,t} - hG_{k,t}) - \mathrm{proj}_{\mathcal{W}}(y_{k,t} - h\nabla F(y_{k,t}))] - \Delta_{k,t}^T(\bar{y}_{k,t+1} - w_*) \\
&\leq \|\Delta_{k,t}\|\,\|(y_{k,t} - hG_{k,t}) - (y_{k,t} - h\nabla F(y_{k,t}))\| - \Delta_{k,t}^T(\bar{y}_{k,t+1} - w_*), \\
&= h\|\Delta_{k,t}\|^2 - \Delta_{k,t}^T(\bar{y}_{k,t+1} - w_*).
\end{aligned} \tag{33}
$$

[7] Note that this quantity is never computed during the algorithm. We can use it in the analysis nevertheless.

So we have

$$\begin{aligned}&\|y_{k,t+1}-w_*\|^2\\ &\overset{(32),(33)}{\leq} \|y_{k,t}-w_*\|^2 + 2h\left(h\|\Delta_{k,t}\|^2 - \Delta_{k,t}^T(\bar{y}_{k,t+1}-w_*) - [F(y_{k,t+1})-F(w_*)]\right).\end{aligned}$$

By taking expectation, conditioned on $y_{k,t}$[8] we obtain

$$\mathbf{E}[\|y_{k,t+1}-w_*\|^2] \overset{(33),(32)}{\leq} \|y_{k,t}-w_*\|^2 + 2h\left(h\,\mathbf{E}[\|\Delta_{k,t}\|^2] - \mathbf{E}[F(y_{k,t+1})-F(w_*)]\right), \tag{34}$$

where we have used that $\mathbf{E}[\Delta_{k,t}] = \mathbf{E}[G_{k,t}] - \nabla F(y_{k,t}) = 0$ and hence $\mathbf{E}[-\Delta_{k,t}^T(\bar{y}_{k,t+1}-w_*)] = 0$.[9] Now, if we put (16) into (34) we obtain

$$\begin{aligned}\mathbf{E}[\|y_{k,t+1}-w_*\|^2] &\leq \|y_{k,t}-w_*\|^2\\ &+ 2h\left(4Lh\alpha(b)(F(y_{k,t})-F(w_*)+F(w_k)-F(w_*)) - \mathbf{E}[F(y_{k,t+1})-F(w_*)]\right),\end{aligned} \tag{35}$$

where $\alpha(b) = \frac{m-b}{b(m-1)}$.

Now, if we consider that we have just lower-bounds $\nu_F \geq 0$ of the true strong convexity parameter μ_F, then we obtain from (35) that

$$\begin{aligned}\mathbf{E}[\|y_{k,t+1}-w_*\|^2] &\leq \|y_{k,t}-w_*\|^2\\ &+ 2h\left(4Lh\alpha(b)(F(y_{k,t})-F(w_*)+F(w_k)-F(w_*)) - \mathbf{E}[F(y_{k,t+1})-F(w_*)]\right),\end{aligned}$$

which, by decreasing the index t by 1, is equivalent to

$$\begin{aligned}\mathbf{E}[\|y_{k,t}-w_*\|^2] + 2h\,\mathbf{E}[F(y_{k,t})-F(w_*)] &\leq \|y_{k,t-1}-w_*\|^2\\ &+ 8h^2L\alpha(b)(F(y_{k,t-1})-F(w_*)+F(w_k)-F(w_*)).\end{aligned} \tag{36}$$

Now, by the definition of w_k we have that

$$\mathbf{E}[F(w_{k+1})] = \frac{1}{M}\sum_{t=1}^{M}\mathbf{E}[F(y_{k,t})]. \tag{37}$$

By summing (36) multiplied by $(1-h\nu_F)^{M-t}$ for $t=1,\ldots,M$, we can obtain the left-hand side

[8] For simplicity, we omit the $\mathbf{E}[\cdot \mid y_{k,t}]$ notation in further analysis.

[9] $\bar{y}_{k,t+1}$ is constant, conditioned on $y_{k,t}$.

$$LHS = \sum_{t=1}^{M} \mathbf{E}[\|y_{k,t} - w_*\|^2] + 2h \sum_{t=1}^{M} \mathbf{E}[F(y_{k,t}) - F(w_*)] \tag{38}$$

and the right-hand side

$$\begin{aligned}
RHS = & \sum_{t=1}^{M} \mathbf{E}\, \|y_{k,t-1} - w_*\|^2 + 8h^2 L\alpha(b) \sum_{t=1}^{M} \mathbf{E}[F(y_{k,t-1}) - F(w_*) + F(w_k) - F(w_*)] \\
& = \sum_{t=0}^{M-1} \mathbf{E}\, \|y_{k,t} - w_*\|^2 + 8h^2 L\alpha(b) \left(\sum_{t=0}^{M-1} \mathbf{E}[P(y_{k,t}) - P(w_*)] \right) \\
& \qquad + 8h^2 L\alpha(b) M \, \mathbf{E}[F(w_k) - F(w_*)] \\
& \leq \sum_{t=0}^{M-1} \mathbf{E}\, \|y_{k,t} - w_*\|^2 + 8h^2 L\alpha(b) \left(\sum_{t=0}^{M} \mathbf{E}[F(y_{k,t}) - F(w_*)] \right) \\
& \qquad + 8Mh^2 L\alpha(b) \, \mathbf{E}[F(w_k) - F(w_*)].
\end{aligned} \tag{39}$$

Combining (38) and (39) and using the fact that $LHS \leq RHS$ we have

$$\begin{aligned}
& \mathbf{E}[\|y_{k,M} - w_*\|^2] + 2h \sum_{t=1}^{M} \mathbf{E}[F(y_{k,t}) - F(w_*)] \\
& \leq \mathbf{E}\, \|y_{k,0} - w_*\|^2 + 8Mh^2 L\alpha(b) \, \mathbf{E}[F(w_k) - F(w_*)] \\
& + 8h^2 L\alpha(b) \left(\sum_{t=1}^{M} \mathbf{E}[F(y_{k,t}) - F(w_*)] \right) \\
& + 8h^2 L\alpha(b) \, \mathbf{E}[F(y_{k,0}) - F(w_*)].
\end{aligned}$$

Now, using (37) we obtain

$$\begin{aligned}
& \mathbf{E}[\|y_{k,M} - w_*\|^2] + 2Mh(\mathbf{E}[F(w_{k+1})] - F(w_*)) \\
& \leq \mathbf{E}\, \|y_{k,0} - w_*\|^2 + 8Mh^2 L\alpha(b) \, \mathbf{E}[F(w_k) - F(w_*)] \\
& + 8Mh^2 L\alpha(b) \, (\mathbf{E}[F(w_{k+1})] - F(w_*)) \\
& + 8h^2 L\alpha(b) \, \mathbf{E}[F(y_{k,0}) - F(w_*)].
\end{aligned} \tag{40}$$

Note that all the above results hold for any optimal solution $w_* \in \mathcal{W}^*$; therefore, they also hold for $w'_* = \text{proj}_{\mathcal{W}^*}(w_k)$, and Lemma 5 implies that, under weak strong convexity of F, i.e., $\nu_F = 0$,

$$\|w_k - w'_*\|^2 \leq \frac{2\beta}{\mu}[F(w_k) - F(w'_*)]. \tag{41}$$

Considering $\mathbf{E}\,\|y_{k,M} - w'_*\|^2 \geq 0$, $y_{k,0} = w_k$, and using (41), the inequality (40) with w_* replaced by w'_* gives us

$$\begin{aligned}&2Mh\{1 - 4hL\alpha(b)\}[\mathbf{E}[F(w_{k+1})] - F(w'_*)]\\ &\qquad\qquad \leq \left\{\frac{2\beta}{\mu} + 8Mh^2L\alpha(b) + 8h^2L\alpha(b)\right\}[F(w_k) - F(w'_*)],\end{aligned}$$

or equivalently,

$$\mathbf{E}[F(w_{k+1}) - F(w'_*)] \leq \rho[F(w_k) - F(w'_*)],$$

when $1 - 4hL\alpha(b) > 0$ (which is equivalent to $h \leq \frac{1}{4L\alpha(b)}$), and when ρ is defined as

$$\rho = \frac{\beta/\mu + 4h^2L\alpha(b)(M+1)}{h(1 - 4hL\alpha(b))M}$$

The above statement, together with assumptions of $h \leq 1/L$, implies

$$0 < h \leq \min\left\{\frac{1}{4L\alpha(b)}, \frac{1}{L}\right\}.$$

Applying the above linear convergence relation recursively with chained expectations and realizing that $F(w'_*) = F(w_*)$ for any $w_* \in \mathcal{W}^*$ since $w_*, w'_* \in \mathcal{W}^*$, we have

$$\mathbf{E}[F(w_k) - F(w_*)] \leq \rho^k[F(w_0) - F(w_*)].$$

References

1. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
2. Calamai, P.H., Moré, J.J.: Projected gradient methods for linearly constrained problems. Math. Program. **39**, 93–116 (1987)
3. Defazio, A., Bach, F., Lacoste-Julien, S.: SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: NIPS (2014)
4. Fercoq, O., Richtárik, P.: Accelerated, parallel and proximal coordinate descent. arXiv:1312.5799 (2013)
5. Fercoq, O., Qu, Z., Richtárik, P., Takáč, M.: Fast distributed coordinate descent for non-strongly convex losses. In: IEEE Workshop on Machine Learning for Signal Processing (2014)
6. Gong, P., Ye, J.: Linear convergence of variance-reduced projected stochastic gradient without strong convexity. arXiv:1406.1102 (2014)

7. Hoffman, A.J.: On approximate solutions of systems of linear inequalities. J. Res. Natl. Bur. Stand. **49**(4), 263–265 (1952)
8. Jaggi, M., Smith, V., Takáč, M., Terhorst, J., Hofmann, T., Jordan, M.I.: Communication-efficient distributed dual coordinate ascent. In: NIPS, pp. 3068–3076 (2014)
9. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: NIPS, pp. 315–323 (2013)
10. Karimi, H., Nutini, J., Schmidt, M.: Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In: ECML PKDD, pp. 795–811 (2016)
11. Kloft, M., Brefeld, U., Laskov, P., Müller, K.-R., Zien, A., Sonnenburg, S.: Efficient and accurate lp-norm multiple kernel learning. In: NIPS, pp. 997–1005 (2009)
12. Konečný, J., Liu, J., Richtárik, P., Takáč, M.: Mini-batch semi-stochastic gradient descent in the proximal setting. IEEE J. Sel. Top. Sign. Proces. **10**, 242–255 (2016)
13. Konečný, J., Richtárik, P.: Semi-stochastic gradient descent methods. arXiv:1312.1666 (2013)
14. Le Roux, N., Schmidt, M., Bach, F.: A stochastic gradient method with an exponential convergence rate for finite training sets. In: NIPS, pp. 2672–2680 (2012)
15. Liu, J., Wright, S.J.: Asynchronous stochastic coordinate descent: parallelism and convergence properties. SIAM J. Optim. **25**(1), 351–376 (2015)
16. Mareček, J., Richtárik, P., Takáč, M.: Distributed block coordinate descent for minimizing partially separable functions. In: Numerical Analysis and Optimization 2014, Springer Proceedings in Mathematics and Statistics, pp. 261–286 (2014)
17. Necoara, I., Clipici, D.: Parallel random coordinate descent method for composite minimization: convergence analysis and error bounds. SIAM J. Optim. **26**(1), 197–226 (2016)
18. Necoara, I., Patrascu, A.: A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Comput. Optim. Appl. **57**(2), 307–337 (2014)
19. Necoara, I., Nesterov, Y., Glineur, F.: Linear convergence of first order methods for non-strongly convex optimization. arXiv:1504.06298 (2015)
20. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**(4), 1574–1609 (2009)
21. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Boston (2004)
22. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**, 341–362 (2012)
23. Nesterov, Y.: Gradient methods for minimizing composite functions. Math. Program. **140**(1), 125–161 (2013)
24. Nguyen, L.M., Liu, J., Scheinberg, K., Takáč, M.: SARAH: a novel method for machine learning problems using stochastic recursive gradient. arXiv:1703.00102 (2017)
25. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1–2), 1–38 (2014)
26. Richtárik, P., Takáč, M.: Distributed coordinate descent method for learning with big data. J. Mach. Learn. Res. **17**, 1–25 (2016)
27. Richtárik, P., Takáč, M.: Parallel coordinate descent methods for big data optimization. Math. Program. Ser. A **156**, 1–52 (2016)
28. Shalev-Shwartz, S., Zhang, T.: Accelerated mini-batch stochastic dual coordinate ascent. In: NIPS, pp. 378–385 (2013)
29. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss. J. Mach. Learn. Res. **14**(1), 567–599 (2013)
30. Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for SVM. Math. Program. Ser. A, B Spec. Issue Optim. Mach. Learn. **127**, 3–30 (2011)
31. Shamir, O., Zhang, T.: Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. In: ICML, pp. 71–79. Springer, New York (2013)
32. Takáč, M., Bijral, A.S., Richtárik, P., Srebro, N.: Mini-batch primal and dual methods for SVMs. In: ICML, pp. 537–552. Springer (2013)

33. Wang, P.-W., Lin, C.-J.: Iteration complexity of feasible descent methods for convex optimization. J. Mach. Learn. Res. **15**, 1523–1548 (2014)
34. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM J. Optim. **24**(4), 2057–2075 (2014)
35. Zhang, T.: Solving large scale linear prediction using stochastic gradient descent algorithms. In: ICML, pp. 919–926. Springer (2004)
36. Zhang, H.: The restricted strong convexity revisited: analysis of equivalence to error bound and quadratic growth. Optim. Lett. **11**(4), 817–833 (2016)
37. Zhang, L., Mahdavi, M., Jin, R.: Linear convergence with condition number independent access of full gradients. In: NIPS, pp. 980–988 (2013)

Exact Separation of *k*-Projection Polytope Constraints

Elspeth Adams and Miguel F. Anjos

Abstract A critical step of any cutting plane algorithm is to find valid inequalities, or more generally valid constraints, that improve the current relaxation of the integer-constrained problem. We consider the *k*-projection polytope constraints that are a family of constraints based on an inner description of the cut polytope of size *k* and are applied to $k \times k$ principal minors of the matrix variable of a semidefinite optimization relaxation. We propose a bilevel second order cone optimization approach to find the maximally violated *k*-projection polytope constraint according to a specific depth measure, and reformulate the bilevel problem as a single-level mixed binary second order cone optimization problem. We report computational results using the proposed approach within a cutting plane algorithm on instances of max-cut with 500 and 600 nodes.

Keywords Max-cut • cone optimization • polytope constraints • cut polytope • cutting plane algorithm

1 Introduction

Cutting planes are often used as an efficient means to tighten continuous relaxations of mixed-integer optimization problems and are a vital component of branch-and-cut algorithms. A critical step of any cutting plane method is solving the separation problem to find valid inequalities, or cuts, that are violated by the current solution but are satisfied by every feasible integer solution. The problem of finding a cut that achieves maximal violation over all possible cuts for a given solution to the relaxation is called the maximally violated valid inequality problem (MVVIP) [21].

This paper is concerned with the problem of finding the most violated *k*-projection polytope constraint (*k*PPC); we refer to this problem as the maximally violated *k*PPC problem (MV*k*PPCP). This class of constraints was introduced in [2]

E. Adams • M.F. Anjos (✉)
GERAD & Polytechnique Montreal, Montreal, QC, Canada
e-mail: elspeth.adams@polymtl.ca; anjos@stanfordalumni.org

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_8

and is defined for NP-hard combinatorial problems based on graphs for which the projection of the problem onto a subgraph shares the same structure as the original problem. Examples of such problems include the well-known max-cut and stable-set problems. While originally introduced as a means to define a new hierarchy of semidefinite optimization relaxations for this type of NP-hard problem, these constraints can be used individually to tighten any semidefinite relaxation.

This paper presents a bilevel optimization model that fits into the MVVIP framework and finds the maximally violated *k*PPC. We show how to reformulate the model as a single-level mixed integer second order cone optimization problem, and how the single-level model can be strengthened by breaking symmetry and by reformulating it using fewer binary variables. We also report preliminary computational results on large instances of max-cut. Although we focus our study on the max-cut problem, our approach can be used for other problems for which *k*PPCs can be defined [2].

This paper is organized as follows. In Sect. 2 we provide a brief literature review. Section 3 introduces first the bilevel model for finding maximally violated *k*PPCs, and second the single-level model for separating a maximally violated *k*PPC; the proof of the equivalence of the two models is given in Sect. 3.3. In Sect. 4 we show how the single-level model can be strengthened by adding symmetry-breaking constraints (Sect. 4.1) and by changing the binary variables (Sect. 4.2). Section 5 presents in some detail the cutting plane used in our computational study, and Sect. 6 reports computational results. Section 7 concludes the paper.

2 Literature Review

2.1 *Separation of Valid Inequalities and Maximally Violated Inequalities*

Separation procedures (or constraint identification problems) are defined as follows: given a point x and a family of valid constraints $\mathcal{L}$, identify one or more constraints in $\mathcal{L}$ violated by x, or prove that no such constraint exists [26]. Note that although separation procedures are typically used to identify inequalities, the framework is identical for any set of valid constraints.

Separation procedures have been studied from both the practical and theoretical perspectives and are often discussed in the context of cuts which are used to tighten relaxations. Cuts that share a special structure can be categorized into a specific family or class. Applegate et al. [4] called the paradigm of generating cuts from a given family the template paradigm. Different types of cuts include Chvátal cuts [10], Chvátal-Gomory [25], $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts [6], split cuts [11], MIR-inequalities [25] and lift-and-project cuts [5].

In practice, the separation of valid inequalities (i.e. cuts) is a key component of cutting plane algorithms. Cutting plane algorithms have been well studied and are fundamental in solving integer (and mixed integer) optimization problems. For early research see Dantzig et al. [12], Gomory [16] and Grötschel et al. [18]. For more recent advances see [22, 24], and [30].

Given a relaxation $\subseteq \mathbb{R}^n_+$ with an optimal solution $x^* \in P$ and a cut $\alpha x \leq \alpha_0$, Amaldi et al. [3] identify three types of distance measures:

- *Cut violation* is the quantity $\alpha x^* - \alpha_0$;
- *Cut depth* is $\frac{\alpha x^* - \alpha_0}{||\alpha||_2}$ where $||\alpha||_2 = \sqrt{\sum_{j=1}^n \alpha_j^2}$
- *Cut depth variant* is $\frac{\alpha x^* - \alpha_0}{\sqrt{\sum_{j=1:x_j^* \neq 0}^n \alpha_j^2 + 1}}$.

This paper will use the cut depth measure, specifically measuring the Euclidean distance between the projection of the current solution x^* and the k-cut polytope (see Sect. 3.1 below).

Different separation procedures are used for different families of cuts. Caprara and Letchford [8] examined the complexity of the separation procedure for various inequalities. They proved strong $\mathcal{NP}$-completeness for the separation of split cuts and strengthened $\mathcal{NP}$-completeness results for $\{0, \frac{1}{2}\}$-cuts (initially in [6]) and Chvátal-Gomory cuts (initially in [14]).

Optimization models have been proposed that look for maximally violated cuts. Caprara et al. [9] proposed a model that finds the mod-k cut that is maximally violated for a given point x^*. They also show that for any given k for which a prime factorization is known maximally violated mod-k cuts can be found efficiently in $O(mn \min\{m, n\})$ time.

Lodi et al. [21] propose a mixed-integer bilevel model for a general separation problem which finds the maximally violated valid inequality. They emphasize the conceptual nature of this formulation since it is challenging to explicitly write a compact description of the inner problem and there are practical issues surrounding solving bilevel problems. However for certain examples (split cuts, generalized subtour elimination constraints (GSECs) for the capacitated vehicle routing problem) the bilevel model can be converted to a single-level linear optimization problem. Two key components in MVVIPs are validity and membership. We address them in turn.

The validity verification problem determines if all points in a polyhedron satisfy the constraint. Lodi et al. [21] formalize this concept for linear inequalities. For a given polyhedron $\mathcal{P} = \{x \in \mathcal{R}^n_+ \mid Ax \geq b\}$, (α, β) defines a valid inequality if and only if there exists $u \in \mathcal{R}^m_+$ such that $\alpha \geq u^T A$ and $\beta \leq u^T b$.

The membership problem is a decision problem that asks whether a given point $\hat{X}$ is contained in a polyhedron $\mathcal{P}$ or the intersection of the polyhedron $\mathcal{P}$ and a given cut. This problem has been looked at in the context of different families of cuts, for example Chvátal-Gomory cuts [14] and $\{0, \frac{1}{2}\}$ cuts [7].

2.2 The Max-Cut Problem and its k-Projection Polytope Constraints

Our focus in this paper is on the max-cut problem. To introduce the problem, we first establish some notation.

Let $\mathcal{S}_n$ be the set of symmetric matrices of size n. Let $N := \{1, \ldots, n\}$ be the vertex set of a graph G. The max-cut problem is defined by an undirected graph G with n vertices and the weighted adjacency matrix A. It is assumed that the graph contains no loops. A 'cut' is the set of edges induced from partitioning the nodes into two sets, s and $N \setminus s$. Then the edge $[i,j]$ belongs to the cut if $i \in s$ and $j \notin s$ or $i \notin s$ and $j \in s$. Let $c_r \in \{-1, 1\}^n$ be a vector representing a partitioning of the nodes and let each of the 2^{n-1} cut matrices be denoted $C_i \in \mathcal{S}_n$ where $C_i = c_r c_r^T$. The cut polytope, CUT_n, is the convex hull of all 2^{n-1} feasible solutions C_i.

The max-cut problem is

$$z_{\text{max-cut}} = \max\{\langle L, X\rangle \, : \, X \in \text{CUT}_n\}$$

where L is the Laplacian associated with the weighted adjacency matrix A such that $L = \text{Diag}(Ae) - A$ where e is the vector of all ones. Note that Ae is the vector with the ith element equal to the degree of node i.

The set $\mathcal{C}$ of correlation matrices is

$$\mathcal{C} := \{X \in \mathcal{S}_n : \; \text{diag}(X) = e, \; X \succeq 0\}$$

and the metric polytope $\mathcal{M}$ is the set of all symmetric matrices with diagonal equal one and satisfying the triangle inequalities,

$$\mathcal{M} := \{X \in \mathcal{S}_n : \; \text{diag}(X) = e, \; x_{ij} + x_{ik} + x_{jk} \geq -1, x_{ij} - x_{ik} - x_{jk} \geq -1 \; \forall i, j, k\}$$

These two sets yield two popular semidefinite optimization relaxations of the max-cut problem:

$$z_{\mathcal{C}} := \max\{\langle L, X\rangle \, : \, X \in \mathcal{C}\}$$
$$z_{\mathcal{C}\cap\mathcal{M}} := \max\{\langle L, X\rangle \, : \, X \in \mathcal{C} \cap \mathcal{M}\}$$

Delorme and Poljak [13] introduced a relaxation for which the feasible region is defined by the set $\mathcal{C}$, and Helmberg et al. [19] solved this relaxation with an interior point method. Fischer et al. [15] presented a computationally efficient way to solve the relaxation over $\mathcal{C} \cap \mathcal{M}$, and Rendl et al. [27] proposed an exact method that begins with the relaxation over $\mathcal{C} \cap \mathcal{M}$ and uses branch-and-bound to solve the max-cut problem. A more recent application of this relaxation in a branch-and-bound setting was proposed in [20].

Our interest in this paper is in improving the above relaxations using k-projection polytope constraints. To define a kPPC, let $I \subseteq N$, $|I| = k$, and let X_I denote the principal submatrix of X indexed by I. Then we can express the kPPC $X_I \in \mathrm{CUT}_k$ corresponding to the k-subset I as:

$$X_I = \sum_{i=1}^{2^{k-1}} \lambda_i^I Q_i \text{ with } \lambda_i^I \geq 0,\ \sum_{i=1}^{2^{k-1}} \lambda_i^I = 1,$$

where $\lambda^I \in \mathbb{R}^{2^{k-1}}$ and the vectors Q_i of length $\binom{k}{2}$ represent the 2^{k-1} valid cuts for a graph on k nodes.

This paper addresses the question of checking if there is a kPPC that is not satisfied by the optimal solution of the current relaxation, and if so, how to find the most violated kPPC. Our interest here is in small values of k so that this description of CUT_k is amenable to use within a practical solution algorithm. In this paper we focus on the max-cut problem; for a more general discussion of kPPCs, see [2].

3 Finding Maximally Violated k-Projection Polytope Constraints

In this section we present two formulations of the MVkPPCP. Noting that the kPPCs are always valid because they satisfy the projection property, we see that the validity verification problem is not an issue for kPPCs. The issue of membership is considered in Sect. 3.1.

In Sect. 3.2 we state our first formulation of the MVkPPCP as a bilevel problem. While straightforward to understand, this formulation is inconvenient from a computational perspective. For this reason, we show in Sect. 3.3 that the bilevel problem can be expressed as an equivalent single-level mixed binary second order cone optimization problem.

3.1 Membership

The membership problem for a kPPC is: for a given $k \times k$ submatrix X_I of X, where $|\mathcal{I}| = k$, is $X_I \in \mathrm{CUT}_k$? If $X_I \notin \mathrm{CUT}_k$, then adding the kPPC for subset I will tighten the relaxation.

The following problem, denoted *distance-to-polytope (D2P)*, determines if a given $\hat{X}$ is in CUT_k, and quantifies the separation if $\hat{X} \notin \mathrm{CUT}_k$:

$$d^*(\hat{X}, I, k) = \min \ \left\{ ||\mathrm{triu}\left(\hat{X}_I\right) - Q\lambda|| \ : \ e^T\lambda = 1,\ \lambda \geq 0 \right\} \qquad \text{(D2P)}$$

where $\hat{X}_I$ is the principal submatrix indexed by I of $\hat{X}$, e is the vector of all ones of the appropriate size, triu(X) is the vector formed from the elements in the strictly upper triangular part of matrix X taken column-wise, and Q is a $\binom{k}{2} \times 2^{k-1}$ matrix with columns Q_i representing the 2^{k-1} valid cuts for a graph on k nodes.

The optimal value d^* equals the Euclidean distance between $\hat{X}_I$ and CUT_k, therefore:

$$\text{if } d^* = 0 \text{ then } \hat{X}_I \in \mathrm{CUT}_k$$

$$\text{if } d^* > 0 \text{ then } \hat{X}_I \notin \mathrm{CUT}_k$$

To illustrate our use of (D2P), we use the following small example.

Example 1 Consider the instance of max-cut seeking the minimum of the function

$$\sum_{1\le i<j\le 4} X_{ij} + X_{56} + X_{57} - 2\sum_{j=1}^{4} X_{j5} - X_{16} - X_{36} - X_{27} - X_{47} - X_{67}$$

over the polytope CUT_7. The optimal value of this instance is known to be 5 because Grishukhin [17] showed that

$$\sum_{1\le i<j\le 4} X_{ij} + X_{56} + X_{57} - 2\sum_{j=1}^{4} X_{j5} - X_{16} - X_{36} - X_{27} - X_{47} - X_{67} \ge 5 \qquad (1)$$

defines a facet of CUT_7. This instance of max-cut is of interest because it is an example of small dimension for which the varying behavior of relaxations can be observed.

The optimal value obtained by minimizing the left-hand side of (1) over $\mathcal{C} \cap \mathcal{M}$ is 6.0584. This is our initial bound; let X^* be the corresponding optimal solution. For $k = 5$ we can find the distance-to-polytope (d^*) of X^* for each $I \in V$ such that $|I| = 5$. We see from Table 1 that the largest value of d^* is equal to 0.1274 and is attained for two subsets I, namely [13567] and [24567]. We can add the kPPC corresponding to the first of these subsets to the initial relaxation, and again from Table 1 we see that the bound improves noticeably to 5.9800. If instead we add the kPPC corresponding to the second subset, we obtain the same bound. However, if we add both kPPCs, the bounds improve further to 5.9000.

Table 1 shows the impact of adding projection polytope constraints to the initial relaxation in this way. The third column shows the bound when the kPPC associated with the single index I is added to $\mathcal{C} \cap \mathcal{M}$. The fourth and fifth columns show the bounds when multiple kPPCs are added.

Note that for any set of indices I where the distance-to-polytope is equal to (nearly) 0 adding the corresponding PPC does not change the optimal objective function. We observe that even if we add all the PPCs corresponding to the 14 sets

Table 1 Bounds for adding different $k = 5$ PPCs to the SDP relaxation $\mathcal{C} \cap \mathcal{M}$

Distance d^* for I	Subset I	Bound after adding single kPPCs	Bound after adding multiple kPPCs	Bound with all kPPCs
0.1274	[1 3 5 6 7]	5.9800	5.9000	5.8000
	[2 4 5 6 7]	5.9800		
0.0800	[1 2 3 5 6]	6.0371	5.9412	
	[1 2 4 5 7]	6.0371		
	[1 3 4 5 6]	6.0371		
	[2 3 4 5 7]	6.0371		
0.0563	[1 2 3 4 5]	6.0485	–	
≤0.0008	14 other subsets	–	6.0584	

of indices with $d^* \leq .0008$ the optimal objective function does not change since for each of these sets of indices $X^*_{\mathcal{I}} \in \mathrm{CUT}_{|\mathcal{I}|}$ and the optimal solution X^* is still feasible. Adding all $k = 5$ projection polytope constraints improves the bound to 5.8000.

3.2 *Formulation of the MVkPPCP as a Bilevel Problem*

The following is the MVkPPCP formulation of the MVVIP for finding the maximally violated kPPC:

$$
\begin{aligned}
(\mathrm{DP}_{\mathrm{Bilevel}}) \qquad \max \quad & d \\
\text{s.t.} \quad & B^T e_n = e_k && (2)\\
& Be_k \leq e_n && (3)\\
& B \in \{0,1\}^{n \times k} && (4)\\
& d = \left\{ \min_{e^T\lambda=1,\ \lambda\geq 0} ||\mathrm{triu}\left(B^T XB\right) - Q\lambda|| \right\} && (5)
\end{aligned}
$$

The inner problem (5) solves (D2P) for fixed k, and B specifies the $k \times k$ submatrix of X. Constraints (2)–(4) ensure that B selects precisely k rows (and the corresponding columns):

$$X_I = B^T XB \ \text{ where (2)–(4) are satisfied and } \sum_{j=1}^{k} B_{ij} = \begin{cases} 1 & \text{if } i \in I \\ 0 & \text{if } i \notin I \end{cases}$$

3.3 Reformulation of the MVkPPCP as a Single-Level Problem

In this section we prove that $\mathrm{DP}_{\mathrm{Bilevel}}$ can be reformulated as a single-level mixed binary second order cone optimization problem. This reformulation is denoted $\mathrm{DP}_{\mathrm{single}}$ and it is this formulation that will be used in the rest of the paper. The reformulation is as follows:

$$
\begin{array}{llll}
(\mathrm{DP}_{\mathrm{single}}) \quad \max & d & & \\
\text{s.t. } & (2)\text{–}(4) & & (6)\\
& e^T\lambda = 1 & & (7)\\
& \mu_{jt} + Q\lambda - \sum\limits_{i=1\dots n}\ \sum\limits_{s=1\dots n\,:\,s\neq i} X_{is}\beta_{ijst} = 0 & \forall 1 \le j < t \le k & (8)\\
& \lambda \ge 0 & & (9)\\
& \begin{bmatrix} d \\ \mu \end{bmatrix} \in \mathrm{SOC}^{1+\binom{k}{2}} & & (10)\\
& ye + Q^T z \le 0 & & (11)\\
& \begin{bmatrix} 1 \\ -z \end{bmatrix} \in \mathrm{SOC}^{1+\binom{k}{2}} & & (12)\\
& d - y - \sum\limits_{ijst\in\mathcal{S}} X_{is}\gamma_{ijst} = 0 & & (13)\\
& \beta_{ijst} - b_{ij} \le 0 & \forall\ ijst \in \mathcal{S} & (14)\\
& \beta_{ijst} - b_{st} \le 0 & \forall\ ijst \in \mathcal{S} & (15)\\
& \sum\limits_{ijst\in\mathcal{S}} \beta_{ijst} = \binom{k}{2} & & (16)\\
& 0 \le \beta_{ijst} \le 1 & \forall\ ijst \in \mathcal{S} & (17)\\
& -\gamma_{ijst} - \beta_{ijst} \le 0 & \forall\ ijst \in \mathcal{S} & (18)\\
& \gamma_{ijst} - \beta_{ijst} \le 0 & \forall\ ijst \in \mathcal{S} & (19)\\
& -\gamma_{ijst} + z_{jt} + \beta_{ijst} \le 1 & \forall\ ijst \in \mathcal{S} & (20)\\
& \gamma_{ijst} - z_{jt} + \beta_{ijst} \le 1 & \forall\ ijst \in \mathcal{S} & (21)
\end{array}
$$

where $\mathcal{S} = \{ijst \mid i,s = 1,\dots,n, i \neq s, 1 \le j < t \le k\}$

In the remainder of this section, we present the steps to transform $\mathrm{DP}_{\mathrm{Bilevel}}$ into $\mathrm{DP}_{\mathrm{single}}$.

Step 1: Rewrite the Inner Problem

The first step in the reformulation is to transform the bilevel problem to a single-level problem. Recall the definition of second order cones (SOC):

$$\begin{bmatrix} x_o \\ \bar{x} \end{bmatrix} \in \mathrm{SOC}^{1+n} \Leftrightarrow x_o \geq ||\bar{x}||$$

where x_o is a scalar and $\bar{x}$ is a vector of length n.

Using this property we can reformulate the inner problem (5) to the following SOC problem:

$$\begin{aligned} (\mathrm{P_{Inner}}) \qquad & \min_{d,\lambda,\mu} && d \\ & \text{s.t.} && (7),\ (9),\ (10) \\ & && \mu_{jt} + Q\lambda - \sum_{i=1}^{n-1} \sum_{s=i+1}^{n} X_{is} b_{ij} b_{st} = 0 \qquad \forall 1 \leq j < t \leq k \end{aligned} \tag{22}$$

where constraint (22) ensures that $\mu = \operatorname{triu}\left(B^T XB\right) - Q\lambda$, and the minimization of d implies $d = ||\mu||$ at optimality. Recall that b is given (and not a variable) in this formulation. The dual of $\mathrm{P_{Inner}}$ is:

$$\begin{aligned} \max_{y,z} \quad & y + \sum_{ijst \in \mathcal{S}} X_{is} b_{ij} b_{st} z_{jt} \\ \text{s.t.} \quad & (11),\ (12) \end{aligned}$$

where $y \in \mathbb{R}$ and $z \in \mathbb{R}^{\binom{k}{2}}$ are variables.

Problem (5) is a standard quadratic problem. Therefore necessary and sufficient conditions for optimality are primal feasibility ((7), (9), (10) and (22)), dual feasibility ((11), (12)), and strong duality (23):

$$d - y - \sum_{ijst \in \mathcal{S}} X_{is} b_{ij} b_{st} z_{jt} = 0. \tag{23}$$

Step 2: Linearize

The second step of the reformulation is to linearize $b_{ij} b_{st}$ in (22) with the variable β_{ijst} to get (8) and to linearize $b_{ij} b_{st} z_{jt}$ in (23) with the variable γ_{ijst} to get (13). We consider these in turn.

Recall that b_{ij} is defined $\forall i = 1, \ldots, n\ \forall j = 1, \ldots, k$. Constraints (2)–(4) imply that exactly k of the nk variables are equal to 1 and that the remaining variables are equal to 0. These constraints imply certain $b_{ij} b_{st}$ products will always be 0. Namely,

$$(2) \Rightarrow b_{ij}b_{it} = 0 \qquad \forall i = 1,\ldots,n\ \forall j,t = 1,\ldots,k$$

$$(3) \Rightarrow b_{ij}b_{sj} = 0 \qquad \forall i,s = 1,\ldots,n\ \forall j = 1,\ldots,k$$

Therefore there is no need to linearize the terms in which $i = s$ or $j = t$. Because $b_{ij}b_{st} = b_{st}b_{ij}$, we can further limit the number of products we linearize to only those with $j < t$. Note that of the $(nk)^2$ products only $n(n-1)\binom{k}{2}$ of them need to be linearized. The indices of the terms that are linearized are denoted by $\mathcal{S}$ where

$$\mathcal{S} = \{ijst \mid i,s = 1,\ldots,n,\ i \neq s,\ 1 \leq j < t \leq t\}$$

Furthermore since exactly k elements of b are 1 then exactly $\binom{k}{2}$ of the products equal 1 with the remaining products equal to 0. Lemma 1 formalizes this idea and shows the constraints necessary to enforce it.

Lemma 1 *If constraints* (2)–(4) *are satisfied, then there exists a feasible solution to constraints* (14)–(17) *if and only if* $b_{ij}b_{st} = \beta_{ijst}\ \forall ijst \in \mathcal{S}$.

Proof Assume constraints (2)–(4) are satisfied and consider the cases in turn.

($\Rightarrow$) Let $(\hat{b},\hat{\beta})$ be any feasible solution to constraints (14)–(17). Constraint (4) implies $\hat{b}_{ij}, \hat{b}_{st} \in \{0,1\}$. Consider the cases in turn.

If $\hat{b}_{ij} = 0$ (resp. $\hat{b}_{st} = 0$), then (14) (resp. (15)) and (17) imply $\hat{\beta}_{ijst} = 0$.

If $\hat{b}_{ij} = \hat{b}_{st} = 1$, then constraints (2)–(4) imply that exactly k of the nk elements in $\hat{B}$ will be equal to 1 (with the rest equal to 0). Therefore all but $\binom{k}{2}$ of the terms in $\sum_{ijst\in\mathcal{S}} \hat{\beta}_{ijst}$ will be forced to 0 because $\hat{b}_{ij}$ or $\hat{b}_{st}$ equals 0. Since $\hat{\beta}_{ijst} \leq 1\ \forall ijst \in \mathcal{S}$ and the sum of the nonzero elements of $\hat{\beta}$ equals $\binom{k}{2}$, the remaining $\hat{\beta}$'s are forced to 1.

Therefore $\hat{\beta}_{ijst} = \hat{b}_{ij}\hat{b}_{st}\ \forall ijst \in \mathcal{S}$ as required.

($\Leftarrow$) Let $b_{ij}b_{st} = \beta_{ijst}\ \forall ijst \in \mathcal{S}$. Constraint (4) implies $b_{ij}, b_{st} \in \{0,1\}$. Therefore (17) is feasible since $\beta_{ijst} \in \{0,1\}$.

Constraint (14) implies $b_{ij}b_{st} - b_{ij} = b_{ij}(b_{st} - 1) \leq 0\ \forall b_{ij}, b_{st} \in \{0,1\}$. Constraint (15) follows similarly.

Finally, constraint (2) implies that if $b_{ij} = 1$ then $b_{it} = 0\ \forall t \neq j$ and (3) implies that there exists exactly one i for each $1 \leq j \leq k$ such that $b_{ij} = 1$ and that if $b_{ij} = 1$ then $b_{sj} = 0\ \forall s \neq i$. Therefore there exist exactly $\binom{k}{2}$ β's equal to 1 and (16) is feasible. □

Note that although β_{ijst} is binary this does not need to be included as an explicit constraint in $\mathrm{DP}_{\mathrm{single}}$.

The final step is to linearize $b_{ij}b_{st}z_{jt}$ in constraint (23) using the variable γ_{ijst}. The linearization happens over the same set $\mathcal{S}$. The linearization is formally stated in Lemma (2) and its proof follows from the application of McCormick's envelope [23].

Lemma 2 *If constraints* (2)–(4) *are satisfied, then there exists a feasible solution to* (18)–(21) *if and only if* $b_{ij}b_{st}z_{jt} = \gamma_{ijst}\ \forall ijst \in \mathcal{S}$.

4 Strengthening the Single-Level Model

4.1 Lexicographical Ordering

Symmetry exists within the exact separation problem because a subset of k indices will induce the same projection polytope regardless of the order. To eliminate this symmetry we can enforce lexicographical order on b with the following set of constraints:

$$b_{s,j-1} + \sum_{i=1}^{s} b_{ij} \leq 1 \;\; \forall s = 2, \ldots, n, j = 2, \ldots, k. \tag{24}$$

Lemma 3 *If constraints* (2)–(4) *and* (24) *are satisfied, then lexicographical order must hold.*

Proof Assume constraints (2)–(4) and (24) are satisfied. Variable b is a row selection matrix in which each column contains exactly 1 element equal to 1, with the rest equal to 0 (constraints (2)–(4)). Going through the columns in order we will show that the index of the row selected can only strictly increase.

Considering $b_{i1} \forall i$ (i.e. column 1 of b) we know there exists an i' such that $b_{i'1} = 1$ and $b_{i1} = 0 \, \forall i \in \{1, \ldots, n\} \setminus \{i'\}$. Therefore (24) implies $b_{i2} = 0 \, \forall i \leq i'$ and since each column sums to 1 (and each row sums to at most 1) then there exists $i'' > i'$ such that $b_{i''2} = 1$. By a similar argument $b_{i3} = 0 \;\; \forall i \leq i''$ and there exists $i''' > i''$ such that $b_{i'''3} = 0$. Repeating the argument k times implies that if $b_{i'1}, b_{i''2}, \ldots, b_{\bar{i},k}$ are the k elements of b equal to 1 then $i' < i'' < \cdots < \bar{i}$. □

4.2 Reformulation with Fewer Binary Variables

We reduce the number of binary variables from $2^{k-1} + nk$ to $2^{k-1} + n$ by adding constraints (25)–(29). The proof later in this section shows that the feasible region of the model does not change and moreover that we no longer need to require binarity of the variables b_{ij}.

$$\sum_{i=1}^{n} a_i = k \tag{25}$$

$$a_i - \sum_{j=1}^{k} b_{ij} = 0 \tag{26}$$

$$a_i - \sum_{i'=1}^{i=1} a_{i'} - b_{i1} \leq 0 \qquad\qquad \forall i = 1 \ldots n \tag{27}$$

$$a_i + \sum_{i'=1}^{i-1} b_{i',j-1} - \sum_{i'=1}^{i-1} b_{i',j} - b_{ij} \leq 1 \qquad \forall i = 2 \dots n, \forall j = 2 \dots k \tag{28}$$

$$a_i \in \{0, 1\} \qquad \forall i = 1 \dots n \tag{29}$$

These constraints along with the symmetry constraints (24) are included in the $\text{DP}_{\text{single}}$ model. Constraint (4) (binarity of b) is removed as it is automatically enforced by constraints (25)–(29). The model $\text{DP}_{\text{fewerBinary}}$ is defined as follows:

$$(\text{DP}_{\text{fewerBinary}}) \qquad \max \quad d$$
$$\text{s.t.} \quad (2),\ (3),\ (7)\text{–}(21),\ (24)\text{–}(29)$$

Lemma 4 certifies that the given set of constraints (including $a \in \{0, 1\}^n$) implies that $b \in \{0, 1\}^{n \times k}$.

Lemma 4 *If constraints* (2), (3), (25)–(29) *and* $0 \leq b_{ij} \leq 1\ \forall i = 1 \dots n, j = 1 \dots k$ *are satisfied, then* $b_{ij} \in \{0, 1\}$.

Proof Let (2), (3), (25)–(29) and $0 \leq b_{ij} \leq 1\ \forall i = 1 \dots n, j = 1 \dots k$ be satisfied.

Constraints (25) and (29) imply that there exist exactly k a_i's equal to 1 with the remaining $n - k$ a_i's equal to 0. Let $a_i = 1$ for $i \in \mathcal{A} := \{i_1 < i_2 < \cdots < i_k\}$

For all $i = 1 \dots n$, $\sum_{i'=1}^{i-1} a_{i'} \in \{0, 1, 2, \dots, k\}$ and $a_i \in \{0, 1\}$, therefore constraint (27) is unrestrictive (since $b_{ij} \geq 0$ is already enforced) unless $a_i = 1$ and $\sum_{i'=1}^{i-1} a_{i'} = 0$. This is the case only for $i = i_1$. Therefore $b_{i_1,1} = 1$.

For $j = 2$, $\sum_{i'=1}^{i-1} b_{i'1} = \begin{cases} 0 & \text{if } i \leq i_1 \\ 1 & \text{if } i > i_1 \end{cases}$ (since $b_{i_1,1} = 1$ and $b_{i,1} = 0\ \forall i \neq i_1$)

$a_i = \begin{cases} 1 & \text{if } i \in \mathcal{S} \\ 0 & \text{if } i \notin \mathcal{S} \end{cases}$ and $\sum_{i'=1}^{i-1} b_{i'2} = \begin{cases} 0 & \text{if } i \leq i_2 \\ 1 & \text{otherwise} \end{cases}$ (since $b_{i2} = 0\ \forall i \leq i_1$)

Combining the above in (28) we get:

$$b_{i2} \geq a_i + \sum_{i'=1}^{i-1} b_{i',j-1} - \sum_{i'=1}^{i-1} b_{i',j} - 1$$

$$= \begin{cases} 1 + 0 - 0 - 1 = 0 & \text{if } i \in \mathcal{A}, i \leq i_1 \\ 1 + 1 - \sum_{i'=1}^{i-1} b_{i',j} - 1 = 1 - \sum_{i'=1}^{i-1} b_{i',j} & \text{if } i \in \mathcal{A}, i > i_2 \\ 1 + 1 - 0 - 1 = 1 & \text{if } i \in \mathcal{A}, i > i_1, i \leq i_2 \\ 0 + 0 - 0 - 1 = -1 & \text{if } i \notin \mathcal{A}, i \leq i_1 \\ 0 + 1 - 1 - 1 = -1 & \text{if } i \notin \mathcal{A}, i > i_2 \\ 0 + 1 - 0 - 1 = 0 & \text{if } i \notin \mathcal{A}, i > i_1, i \leq i_2 \end{cases}$$

Therefore case 3 implies $b_{i_2,2} = 1$ (since $i \in \mathcal{S}, i_1 < i \leq i_2 \Rightarrow i = i_2$). If $b_{i_2,2} = 1$, then $\sum_{i'=1}^{i-1} b_{i',j} = 1 \; \forall i > i_2$ and all cases (except 3) do not restrict b_{ij}.

Repeating this process for $j = 3 \dots k$ implies $b_{i_1,1} = b_{i_2,2} = \cdots = b_{i_k,k} = 1$ and all other $b_{ij} = 0$. □

5 Cutting Plane Algorithm

This section presents the practical details of a cutting plane algorithm that uses kPPCs. The purpose of this algorithm is to show how kPPCs can tighten upper bounds of large max-cut instances after all triangle inequalities are satisfied. Specifically, triangle inequalities are first added until they are all satisfied; then the algorithm iteratively finds violated kPPCs and includes them in the relaxation.

Because our goal is to show how kPPCs can improve the bound over triangle inequalities, we aim to get as much improvement as possible from triangle inequalities so that we can then observe the effect of kPPCs in improving the bound. We note that the implementation of our cutting plane algorithm does not aim for computational efficiency, but rather to show the impact of the kPPCs on the bounds provided by the relaxation. After introducing some notation in Sect. 5.1, the triangle and kPPC cutting plane stages are given in Sects. 5.2 and 5.3, respectively.

5.1 Notation for the Cutting Plane Algorithm

For a given positive integer k, let

$$\square^k := \{I \;:\; \forall I \subseteq V, |I| = k\}$$

be the set of all induced subgraphs of size k. Therefore $|\square^k| = \binom{|V|}{k}$.

Recall that a kPPC is defined for an induced subgraph $I \in \square^k$ where $|I| = k$. Then for any set $\hat{\square}^k \subseteq \square^k$ let

$$\mathcal{PPC}(\hat{\square}^k) := \left\{ X : \; C\lambda^j = \operatorname{triu}(X_I), \; \sum_{i=1}^{2^{k-1}} \lambda_i^j = 1, \; \lambda^j \geq 0, \; \forall I \in \hat{\square}^k \right\}$$

be the solution space where X satisfies all kPPCs defined by the induced subgraphs in $\hat{\square}^k$. Let

$$\mathcal{T} := \left\{ \begin{array}{l} (I, c) : \forall\, I = (i_1, i_2, i_3) \in \square^3 \text{ and} \\ \qquad\qquad \forall\, (c_1, c_2, c_3) \in \{(-1,-1,-1), (-1,1,1), (1,-1,1), (1,1,-1)\} \end{array} \right\}$$

encode the set of all triangle inequalities. Namely, each $(I, c) \in \mathcal{T}$ defines the triangle inequality $c_1 X_{i_1,i_2} + c_2 X_{i_1,i_3} + c_3 X_{i_2,i_3} \leq 1$. Then for any $\hat{\mathcal{T}} \subseteq \mathcal{T}$ let

$$\Delta(\hat{\mathcal{T}}) := \left\{ X : \; c_1 X_{i_1,i_2} + c_2 X_{i_1,i_3} + c_3 X_{i_2,i_3} \leq 1, \; \forall \; ((i_1, i_2, i_3), (c_1, c_2, c_3)) \in \hat{\mathcal{T}} \right\}$$

be the solution space where X satisfies all triangle inequalities encoded in the set $\hat{\mathcal{T}}$.

The purpose of sets $\hat{\square}^k$ and $\hat{\mathcal{T}}$ is to encode the information needed to define the kPPCs and triangle inequalities within the relaxation. For simplicity we will refer to $\hat{\mathcal{T}}$ ($\mathcal{T}$) as a set of (all) triangle inequalities, and to $\hat{\square}^k$ ($\square^k$) as a set of (all) kPPCs (even though it is a set of induced subgraphs). Using these sets, we state the PPC-SDP relaxation:

$$\begin{aligned} \text{(PPC-SDP)} \qquad \max \quad & \langle L, X \rangle \\ \text{s.t.} \quad & X \in \{X \; : \; \operatorname{diag}(X) = e, X \succeq 0\} \\ & X \in \Delta(\hat{\mathcal{T}}) \\ & X \in \mathcal{PPC}(\hat{\square}^k) \end{aligned}$$

5.2 Triangle Inequalities Stage

Algorithm 1 describes the triangle inequalities stage. The following remarks address relevant components of this stage:

Initialization: Solve the basic SDP relaxation, i.e. the model (PPC-SDP) with $\hat{\mathcal{T}}_0 = \emptyset$ and $\hat{\square}^k_0 = \emptyset \; \forall k$. We denote the optimal solution of the relaxation by X^0 and the corresponding optimal objective value by z^0.

```
Initialize $t = 0$, $\hat{\mathcal{T}}_t = \emptyset$, and $tol$=.001
Solve (PPC-SDP) and denote $X^0$ as the optimal solution
while no stopping criterion is met do
    $t = t + 1$;
    Set $(\hat{\mathcal{T}}_t)_{\text{viol}}$
    if $|(\hat{\mathcal{T}}_t)_{viol}| = \emptyset$ then
        stop
    else
        Set $(\hat{\mathcal{T}}_t)_{hat}$
        Set $\hat{\mathcal{T}}_t = (\hat{\mathcal{T}}_t)_{\text{viol}} \cup (\hat{\mathcal{T}}_t)_{hat}$
        Solve (PPC-SDP), denote $X^t$ as the optimal solution;
        Update upper bound
    end
end
```

Algorithm 1: Triangle Inequalities Separation Stage

Stopping criteria: This stage stops when at least one of the following conditions is satisfied:

- There are no violated triangle inequalities $((\hat{\mathcal{T}}_t)_{\text{viol}} = \emptyset)$
- The SDP solver, in our case SDPT3 [28], reports a termination code not equal to 0.

Violated constraints: All $4\binom{n}{3}$ triangle inequalities are tested using the optimal solution X^{t-1} of the previous relaxation. All the inequalities that are not satisfied by a violation $\geq$ *tol* are considered, and $(\hat{\mathcal{T}}_t)_{\text{viol}}$ is defined as the 1000 triangle inequalities with the largest violations among those considered.

Active constraints: The set $(\hat{\mathcal{T}}_t)_{hat}$ denotes the triangle inequalities at iteration t that are active to within the tolerance *tol*. Active constraints at iteration t remain in the relaxation at iteration $t + 1$, and the inactive constraints are removed.

Upper bound: The upper bound is updated if the current objective value is less than the current upper bound. Because inactive triangle inequalities are removed, it is possible (though rarely the case) for the current optimal objective value to be larger than the current upper bound.

5.3 *kPPC Separation Stage*

Algorithm 2 describes the separation of *k*PPCs. It is assumed that the previous stage has stopped with final values of t, X^t, $\hat{\mathcal{T}}_t$ and $\square_t$. The following remarks address relevant components of this stage:

```
From triangle separation stage: $t$, $X^t$, $\hat{\mathcal{T}}_t$ and $\square_t$;
while kPPC stopping criteria is not met do
    $t = t + 1$;
    choose $\bar{k}$;
    set $(\hat{\square}_t^{\bar{k}})_{\text{viol}}$ using Algorithm 3;
    if $|(\hat{\square}_t^{\bar{k}})_{viol}| = \emptyset$ then
        stop
    else
        set $\Psi^k := \left\{ I \; : \; \forall I \subseteq \hat{I} \in \hat{\square}_t^{\bar{k}})_{\text{viol}} \text{ with } |I| = k, \forall \bar{k} > k \right\}$ for $k = 3$ and $k \geq 5$;
        set $\hat{\mathcal{T}}_t = \hat{\mathcal{T}}_{t-1} \setminus \Psi^3$;
        set $\forall k \geq 5,\ \hat{\square}_t^k = \begin{cases} \square_{t-1}^k \cup (\square_t^{\bar{k}})_{\text{viol}} & \text{if } k = \bar{k} \\ \square_{t-1}^k \setminus \Psi^k & \text{otherwise} \end{cases}$;
        solve (PPC-SDP) denote $X^t$ as the optimal solution;
        update upper bound
    end
end
```

Algorithm 2: *k*PPC Separation Stage

Redundant cuts: A triangle inequality or *k*PPC is a redundant constraint if it is defined on an induced subgraph that is a smaller induced subgraph of another *k*PPC. The algorithm collects these redundant triangle inequalities and *k*PPCs in the set Ψ^k and removes them.

Choosing $\bar{k}$: In general the value of $\bar{k} \geq 5$ can vary between iterations.

Stopping criteria: The PPC-cutting plane stage stops when at least one of the following conditions is satisfied

- there are no violated *k*PPCs $((\square_t)_{\text{viol}} = \emptyset)$
- The SDP solver, in our case SDPT3 [28], reports a termination code not equal to 0.
- a maximum number of iterations is reached.

5.4 Generating and Selecting Violated kPPCs

This section presents Algorithm 3. For a given $\hat{k}$, the algorithm finds a set $(\hat{\square}_t^k)_{\text{viol}}$ of violated *k*PPCs and the details of defining a set of at most *nWant* violated *k*PPCs, where *nWant* is the maximum number of *k*PPCs to be added to the relaxation.

We begin with a brief outline of the algorithm:

- The first while loop is the generation stage where induced subgraphs are constructed and tested to determine if they form a violated *k*PPC.
- The set Φ is sorted by distance (decreasing order) to give the sorted list $\hat{\Phi}$.
- The second while loop is the selection stage. The goal is to select at most *nWant* violated *k*PPCs. If fewer than *nWant* violated *k*PPCs were found in the generation stage, then all the *k*PPCs in $\hat{\Phi}$ are selected. Otherwise *k*PPCs are selected according to the 'triangle coverage' method. This is discussed in the Selection section below.

To state the algorithm, we need the following function to identify the induced subgraphs of size 3 within a given induced subgraph of size *k*. Given an induced subgraph I of size k and the set $\hat{\mathcal{T}}^{t-1}$ of triangle inequalities from the previous iteration, we define:

$$\text{triangleList} := \{\hat{I} \;:\; \hat{I} \subseteq I,\; |\hat{I}| = 3\} \cap \{\hat{I} \;:\; (\hat{I}, f) \in \hat{\mathcal{T}}^{t-1}\}$$

If no such induced subgraphs exist, the set is empty.

```
Input : $X^{t-1}$, $k$, $\hat{\mathcal{T}}_{t-1}$ and parameters $maxTime$, $tol$, $nWant$;
Output $(\hat{\square}^k_t)_{\text{viol}}$=a set of at most $nWant$ violated $k$PPCs
initialize $\Phi = \emptyset$;
set $r = \begin{cases} 2 & \text{if } k \in \{5, 6\} \\ 3 & \text{if } k \in \{7, 8, 9\} \end{cases}$;
while $runtime < maxTime$ do
    choose unique $\alpha_1, \alpha_2, \dots, \alpha_r \subseteq \hat{\mathcal{T}}_{t-1}$
    set $I = \cup_{i=1}^{r} I_{\alpha_i}$ where $\mathcal{T}_{\alpha_i} = (I_{\alpha_i}, f_{\alpha_i})$;
    if $|I| = k$ then
        solve (D2P) for $X_I^{t-1}$ to get $d^*$;
        if $d^* > tol$ then
            $\Phi = \Phi \cup (I, d^*)$;
        end
    end
end
Sort $\Phi$ so that $\hat{\Phi} := \left\{ (I, d^*) \subseteq \Phi \ : \ d_i^* \geq d_{i+1}^* \right\}$ ;
if $|\hat{\Phi}| < nWant$ then
    $\square^t = \hat{\Phi}$
else
    set $\omega = \min\{nWant, |\hat{\Phi}|\}$ and $i = 1$;
    set $(\hat{\square}^k_t)_{\text{viol}} = \{I_1 \ : (I_1, d_1) \in \hat{\Phi}\}$ and $\Gamma :=$ triangleList$(I_1)$;
    while $(|(\hat{\square}^k_t)_{viol}| < \omega)$ and $(i < |\hat{\Phi}|)$ do
        $i = i + 1$;
        if $\Gamma \cap$ triangleList$(I_i) = \emptyset$ then
            $(\hat{\square}^k_t)_{\text{viol}} = (\hat{\square}^k_t)_{\text{viol}} \cup I_i$;
        end
    end
end
```

Algorithm 3: Generation and Selection of Violated *k*PPCs

The algorithm is presented and then the details are discussed.

Initialization: X^{t-1} and $\hat{\mathcal{T}}_{t-1}$ come from the final stage of the triangle inequalities stage. The parameter *maxTime* limits the amount of time that the algorithm looks for violated *k*PPCs at each iteration, *tol*=.001, and *nWant* is the maximum number of violated *k*PPCs.

Construction: The induced subgraph I is constructed from r randomly selected triangle inequalities in $\hat{\mathcal{T}}_{t-1}$. If $|I| = k$, then the (D2P) problem is used to determine the distance (d^*) between X_I^{t-1} and CUT_k. If the distance is nonzero (i.e. $X_I^{t-1} \notin \text{CUT}_k$), the result (induced subgraph I and distance d^*) is stored in the set Φ.

Selection: This stage of the separation procedure selects which of the violated *k*PPCs in the sorted list $\hat{\Phi}$ will actually be added to the relaxation. The set $(\hat{\square}^k_t)_{\text{viol}}$ denotes the set of violated *k*PPCs that have been selected. If fewer than *nWant* violated *k*PPCs are found, we select all of them. Otherwise, we proceed as follows:

- The violated *k*PPC with the largest distance is selected.
- The induced subgraphs of size three contained in both the selected *k*PPC and $\hat{\mathcal{T}}^{t-1}$ are stored in Γ.
- We iteratively go through the list of violated *k*PPCs ($\hat{\Phi}$), and the most violated *k*PPC is selected if it does not contain an induced subgraph from $\hat{\mathcal{T}}_{t-1}$ already in Γ.
- We stop once *nWant* *k*PPCs are selected.

This selection process is motivated by the empirical observation that strictly taking the most violated *k*PPCs was not as effective as taking *k*PPCs that were generated from different triangle inequalities.

6 Computational Results

The triangle inequalities are known to be strong, and are typically able to make significant progress in tightening the relaxation [27]. For this reason, our computational study focuses on quantifying the extent to which *k*PPCs can further improve the bound for max-cut instances after the triangle inequalities have been fully used.

We focus on dense instances as these are known to be harder in general. Specifically we report results for:

gkaf5 instance: This instance is from the BiqMac Library [29] of binary quadratic optimization problems. It is fully dense and is the only such large ($n \geq 500$) instance in this library.

cmc instances: These are new randomly generated ‘c’omplete ‘m’ax-‘c’ut (cmc) instances of size 600 to 1000 generated with the graph generator rudy, created by G. Rinaldi. They are random complete graphs with integer edge weights uniformly distributed from $[-75, 75]$ with density=1 and $n = 600, 700, 800, 900$ and 1000. The calls to rudy are:

```
rudy -rnd_graph 600 100 601 -random -75 75 601 > cmc_n600
rudy -rnd_graph 700 100 701 -random -75 75 701 > cmc_n700
rudy -rnd_graph 800 100 801 -random -75 75 801 > cmc_n800
rudy -rnd_graph 900 100 901 -random -75 75 901 > cmc_n900
rudy -rnd_graph 1000 100 1001 -random -75 75 1001 > cmc_n1000
```

where the seed for the random instance is given by $601, 701, \ldots, 1001$.

In our tests we use an Acer Aspire 4752 with 6 GB of memory running Windows 7. We implemented our algorithm in MATLAB R2011b and use version 4.0 of SDPT3 [28] to solve the SDP relaxations.

Table 2 Results of *k*PPC separation stage for gkaf5 with $k = 5, 6$

	$k = 5$			$k = 6$		
Iteration	Objective value	# of *k*PPCs	# of Δ	Objective value	# of *k*PPCs	# of Δ
0	200271.941	0	6295	200271.941	0	6295
1	200248.934	50	6191	200248.611	50	6192
2	200225.794	100	6090	200227.835	100	6088
3	200206.197	150	5989	200210.781	150	5985
4	200186.827	200	5887	200190.635	200	5881
5	200168.202	250	5786	200173.727	250	5777
6	200153.306	300	5686	200155.137	300	5669
7	200134.212	350	5586	200139.970	350	5565
8	200117.241	400	5484	200123.713	400	5460
9	200104.633	450	5384	200110.155	450	5358
10	200091.290	500	5284	200096.633	500	5254

Table 3 Results of *k*PPC separation stage for gkaf5 with $k = 7, 8$

	$k = 7$			$k = 8$		
Iteration	Objective value	# of *k*PPCs	# of Δ	Objective value	# of *k*PPCs	# of Δ
0	200271.941	0	6295	200271.941	0	6295
1	200240.795	50	6131	200233.905	50	6134
2	200213.803	100	5976	200200.391	100	5966
3	200189.602	150	5819	200167.878	150	5810
4	200164.375	200	5666	200140.823	200	5650

6.1 Results for gkaf Test Instance

This section presents the results for the gkaf5 instance. The algorithm parameters were set to *maxTime*=10 min, *tol*=.001, and *nWant*=50. Tables 2 and 3 show the optimal objective value for the *k*PPC separation stage for four different choices of *k* ($k = 5, 6, 7, 8$).

The end of the triangle separation stage (i.e. once there are no more violated triangle inequalities) is denoted as iteration 0. The triangle separation terminated after 17 iterations because SDPT3 stopped with termination code −5. For each *k*, the three columns report the objective value (upper bound), number of *k*PPCs, and number of triangle inequalities still active. Note that for each set of results, the value of *k* was unchanged throughout the *k*PPC separation stage.

Tables 4 and 5 report the runtimes (in minutes) for each iteration. Solver cpu time refers to the time SDPT3 takes to solve the relaxation (PPC-SDP) at that iteration. Iteration time includes generating and selecting violated *k*PPCs, formulating and solving (PPC-SDP), updating the bound (if necessary) and checking the stopping criteria.

Table 4 Runtimes for kPPC separation stage for gkaf5 with $k = 5, 6$

	$k = 5$		$k = 6$	
Iteration	Iteration time (min)	Solver cpu time (min)	Iteration time (min)	Solver cpu time (min)
1	22.4	7.0	25.0	9.5
2	23.8	8.2	26.2	11.5
3	25.1	9.2	29.5	14.2
4	26.2	10.5	31.1	15.9
5	28.4	12.1	35.2	19.4
6	30.7	14.6	39.6	23.1
7	32.7	16.0	43.9	27.3
8	33.5	17.1	53.5	35.8
9	35.7	19.0	55.4	37.8
10	37.5	20.7	168.3	147.5

Table 5 Runtimes for kPPC separation stage for gkaf5 with $k = 5, 6$

	$k = 7$		$k = 8$	
Iteration	Iteration time (min)	Solver cpu time (min)	Iteration time (min)	Solver cpu time (min)
1	23.8	9.5	25.4	12.3
2	27.7	13.0	31.1	17.7
3	31.8	16.6	40.2	26.1
4	37.1	21.6	49.4	34.9

We observe that the kPPCs continue to improve the bound once all triangle inequalities are satisfied. Although the improvement is small the instances being tested are dense graphs, and these are known to be typically difficult for max-cut. Furthermore the time to solve the SDP relaxations increases noticeably as k increases; this is not surprising because the number of equations in the representation of a kPPC increases rapidly with k.

6.2 Results for cmc Test Instances

This section presents the results for one of the five cmc instances, namely the instance with $n = 600$. The results for the other instances are reported in [1], and the conclusions are similar.

There are two key differences between these examples and the gkaf5 example previously examined. First the triangle cutting plane stage terminated when there were no more violated triangle inequalities. Second it was much more difficult to find violated kPPCs. As a result the parameter *maxTime* was set to 20 min. *nWant* remained set to 50 but this limit was never reached as the algorithm added all the violated kPPCs found (with $tol = .001$).

Table 6 Results of kPPC separation stage for cmc_n600 with $k = 5, 6, 7$

	$k = 5$			$k = 6$			$k = 7$		
Iteration	Objective value	# of kPPC	# of Δ	Objective value	# of kPPC	# of Δ	Objective value	# of kPPC	# of Δ
0	293606.893	0	6438	293606.893	0	6438	293606.893	0	6438
1	293605.040	7	6425	293600.064	21	6395	293593.166	38	6319
2	293603.670	21	6400	293593.358	45	6344	293585.613	70	6217
3	293601.489	47	6359	293585.960	58	6317	293577.836	112	6086
4	293596.295	71	6311	293580.992	84	6264	293571.615	140	6000
5	293595.141	90	6272	293571.474	108	6214	293564.547	184	5862
6	293592.596	120	6228	293563.000	134	6161	293554.475	231	5713
7	293590.911	140	6198	293558.247	165	6097	293547.530	262	5614
8	293588.463	179	6146	293549.375	194	6039	293541.614	298	5502
9	293585.864	214	6099	293543.830	212	6003	293535.937	337	5379
10	293584.240	246	6055	293537.030	240	5946	293531.326	376	5259
11	293582.690	268	6011	293529.377	260	5905	293525.916	419	5123
12	293581.881	306	5963	293522.263	282	5858	293520.244	453	5017
13	293580.021	328	5930	293515.400	311	5798	293515.297	492	4898
14	293578.703	364	5885	293511.573	338	5744	–	–	–
15	293578.437	371	5874	293504.553	363	5690	–	–	–

Table 6 reports the optimal objective value for the kPPC cutting plane stage and Table 7 gives the computational time. These tables follow the same format as in the previous section. The 7PPC cutting plane stage was terminated after 13 iterations due to the large cpu time.

Again we see that the kPPCs continue to improve the bound after all triangle inequalities are satisfied. However we observe that for the cmc instances, the bound improves more when larger values of k are used. This is to be expected because $\mathrm{CUT}_k \subseteq \mathrm{CUT}_{k+1}$, but for the gkaf instance it is not always the case. It is unclear as to why this happens.

7 Conclusions and Future Research

We considered the kPPCs recently introduced in [2] and proposed a bilevel second order cone optimization approach to find the maximally violated k-projection polytope constraint according to a specific depth measure. We then transformed the bilevel problem into a single-level mixed binary SOC optimization problem, and improved it using lexicographical ordering and a reformulation with fewer binary variables. We implemented a cutting plane algorithm for the purpose of testing our procedure on large max-cut instances. Our computational results on instances of max-cut with 500 and 600 nodes confirm that the kPPCs can improve the bounds after all triangle inequalities are satisfied, and that the time to solve the SDP relaxations increases noticeably as k increases,

Table 7 Runtimes for kPPC separation stage for cmc_n600 with $k = 5, 6, 7$

	$k = 5$		$k = 6$		$k = 7$	
Iteration	Iteration time (min)	Solver cpu time (min)	Iteration time (min)	Solver cpu time (min)	Iteration time (min)	Solver cpu time (min)
1	27.4	6.4	48.2	7.4	49.3	8.2
2	28.1	6.7	49.1	8.1	51.6	10.2
3	29.7	7.4	49.6	8.5	55.8	13.9
4	31.5	7.4	50.7	9.3	56.8	14.6
5	32.6	8.0	51.9	10.3	61.4	18.9
6	32.3	8.8	53.6	11.6	66.7	23.6
7	31.9	9.1	55.7	13.4	70.3	26.9
8	34.8	10.6	58.3	15.7	75.7	32.2
9	35.7	11.4	59.1	16.4	82.2	38.3
10	36.7	12.2	61.2	18.2	156.2	111.1
11	39.1	13.5	62.7	19.5	308.0	262.1
12	40.1	14.8	64.8	21.4	366.1	319.6
13	40.4	15.5	67.6	23.9	526.0	480.5
14	44.7	18.8	70.0	26.1	–	–
15	41.6	17.5	72.5	28.4	–	–

There are several ways in which the performance of the kPPCs can be improved. One of them is to vary the value of k between, and perhaps within, iterations of the cutting plane algorithm. Heuristic algorithms for finding violated kPPCs could also be considered. Finally, the concept of when kPPCs as a whole are active or inactive should be explored so that kPPCs that are no longer relevant can be removed from the SDP relaxation, as is done for linear inequality constraints.

Acknowledgements We thank an anonymous reviewer for constructive comments that helped us improve this paper.

The second author acknowledges the support of this research by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant 312125.

References

1. Adams, E.: A novel approach to tightening semidefinite relaxations for certain combinatorial problems. PhD thesis, Polytechnique Montreal (2015)
2. Adams, E., Anjos, M.F., Rendl, F., Wiegele, A.: A hierarchy of subgraph projection-based semidefinite relaxations for some NP-hard graph optimization problems. INFOR Inf. Syst. Oper. Res. **53**(1), 40–48 (2015)
3. Amaldi, E., Coniglio, S., Gualandi, S.: Coordinated cutting plane generation via multi-objective separation. Math. Program. **143**(1–2), 87–110 (2014)
4. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton (2006)

5. Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0–1 programs. Math. Program. **58**(1–3), 295–324 (1993)
6. Caprara, A., Fischetti, M.: {0, 1/2}-Chvátal-Gomory cuts. Math. Program. **74**(3), 221–235 (1996)
7. Caprara, A., Fischetti, M.: Branch-and-cut algorithms. Annotated Bibliographies in Combinatorial Optimization, pp. 45–64. Wiley, Chichester (1997)
8. Caprara, A., Letchford, A.N.: On the separation of split cuts and related inequalities. Math. Program. **94**(2–3), 279–294 (2003)
9. Caprara, A., Fischetti, M., Letchford, A.N.: On the separation of maximally violated mod-*k* cuts. Math. Program. **87**(1), 37–56 (2000)
10. Chvátal, V.: Edmonds polytopes and a hierarchy of combinatorial problems. Discret. Math. **4**(4), 305–337 (1973)
11. Cook, W., Kannan, R., Schrijver, A.: Chvátal closures for mixed integer programming problems. Math. Program. **47**(1–3), 155–174 (1990)
12. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. **2**(4), 393–410 (1954)
13. Delorme, C., Poljak, S.: Laplacian eigenvalues and the maximum cut problem. Math. Program. **62**(3, Ser. A), 557–574 (1993)
14. Eisenbrand, F.: Note–on the membership problem for the elementary closure of a polyhedron. Combinatorica **19**(2), 297–300 (1999)
15. Fischer, I., Gruber, G., Rendl, F., Sotirov, R.: Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. Math. Program. **105**(2–3, Ser. B), 451–469 (2006)
16. Gomory, R.E.: An algorithm for integer solutions to linear programs. Recent Adv. Math. Program. **64**, 260–302 (1963)
17. Grishukhin, V.P.: All facets of the cut cone $\mathbf{C}_n$ for $n = 7$ are known. Eur. J. Comb. **11**(2), 115–117 (1990)
18. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)
19. Helmberg, C., Rendl, F., Vanderbei, R.J., Wolkowicz, H.: An interior-point method for semidefinite programming. SIAM J. Optim. **6**(2), 342–361 (1996)
20. Krislock, N., Malick, J., Roupin, F.: Improved semidefinite bounding procedure for solving max-cut problems to optimality. Math. Program. **143**(1–2), 61–86 (2014)
21. Lodi, A., Ralphs, T.K., Woeginger, G.J.: Bilevel programming and the separation problem. Math. Program. **146**, 437–458 (2014)
22. Marchand, H., Martin, A., Weismantel, R., Wolsey, L.: Cutting planes in integer and mixed integer programming. Discret. Appl. Math. **123**(1), 397–446 (2002)
23. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I – convex underestimating problems. Math. Program. **10**(1), 147–175 (1976)
24. Mitchell, J.E.: Branch-and-cut algorithms for combinatorial optimization problems. In: Handbook of Applied Optimization, pp. 65–77 (2002)
25. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization, vol. 18. Wiley, New York (1988)
26. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev. **33**(1), 60–100 (1991)
27. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Math. Program. **121**(2), 307–335 (2010)
28. Toh, K.-C., Todd, M.J., Tütüncü, R.H.: On the implementation and usage of SDPT3 – a MATLAB software package for semidefinite-quadratic-linear programming, version 4.0. In: Handbook on Semidefinite, Conic and Polynomial Optimization, pp. 715–754. Springer, Boston (2012)
29. Wiegele, A.: Biq mac library. http://biqmac.uni-klu.ac.at/biqmaclib
30. Zanette, A., Fischetti, M., Balas, E.: Lexicography and degeneracy: can a pure cutting plane algorithm work? Math. Program. **130**(1), 153–176 (2011)

Univariate Polynomial Optimization with Sum-of-Squares Interpolants

Dávid Papp

Abstract One of the most common tools in polynomial optimization is the approximation of the cone of nonnegative polynomials with the cone of sum-of-squares polynomials. This leads to polynomial-time solvable approximations for many NP-hard optimization problems using semidefinite programming (SDP). While theoretically satisfactory, the translation of optimization problems involving sum-of-squares polynomials to SDPs is not always practical. First, in the common SDP formulation, the dual variables are semidefinite matrices whose condition numbers grow exponentially with the degree of the polynomials involved, which is detrimental for a floating-point implementation. Second, the SDP representation of sum-of-squares polynomials roughly squares the number of optimization variables, increasing the time and memory complexity of the solution algorithms by several orders of magnitude. In this paper we focus on the first, numerical, issue. We show that a reformulation of the sum-of-squares SDP using polynomial interpolants yields a substantial improvement over the standard formulation, and problems involving sum-of-squares interpolants of hundreds of degrees can be handled without difficulty by commonly used semidefinite programming solvers. Preliminary numerical results using semi-infinite optimization problems align with the theoretical predictions. In all problems considered, available memory is the only factor limiting the degrees of polynomials.

Keywords Semidefinite programming • Polynomial optimization • Sum-of-squares • Interpolation • Design of experiments

1 Introduction

Polynomial optimization, and the closely related problem of certifying the non-negativity of nonnegative polynomials is of fundamental importance in a variety of mathematical fields such as computational algebraic geometry [8, 9], discrete

D. Papp (✉)
North Carolina State University, Raleigh, NC 27695, USA
e-mail: dpapp@ncsu.edu

M. Takáč, T. Terlaky (eds.), *Modeling and Optimization: Theory and Applications*,
Springer Proceedings in Mathematics & Statistics 213,
DOI 10.1007/978-3-319-66616-7_9

geometry [4, 5, 50], computer-assisted theorem proving [14], nonconvex global optimization [28], and have many applications in fields as diverse as radiation therapy treatment planning for cancer [49], electrical engineering [20], control theory [1, 25], signal processing [19, 34], design of experiments [15, 37], and shape-constrained statistical estimation [2, 3]. As a result, there are several algorithms, and even multiple Matlab toolboxes, available to solve polynomial optimization problems. All of these are based on similar semidefinite programming formulations, and use interior-point algorithms for semidefinite programming as their numerical optimization engine. As a result, they are not practical for large problems (they do not scale if either the degree or the number of variables increases), primarily for numerical reasons.

In the standard semidefinite programming formulations (motivated by algebraic geometry), the condition numbers of the dual feasible solutions are exponential in the degree [48]. This problem can only be somewhat mitigated using orthogonal bases, and even using orthogonal bases and state-of-the-art semidefinite programming solvers, the highest degrees that can be handled are well under 40 [24]. It is important to note that this ill-conditioning is specific to the semidefinite programming representation of sum-of-squares, and is independent of the conditioning of the original polynomial (or sum-of-squares) optimization problem at hand. This problem manifests already in the univariate setting [36], which is already relevant in the statistical, signal processing, and control applications mentioned above [2, 3, 15, 19, 25, 34, 37].

In this paper we show that the conditioning of the semidefinite programming formulations can be improved dramatically with the combination of polynomial interpolation techniques and sum-of-squares formulations.

Focusing on the univariate case, we show that problems involving very high-degree polynomials can be solved using the same semidefinite programming software that can only handle low-degree instances of the same standard formulation of the same problem. In the experiments, available memory was the only factor limiting the degrees of the polynomials. The multivariate extension is briefly discussed at the end of the paper.

2 Sum-of-Squares Interpolants

We say that a polynomial is *sum-of-squares* if it can be written as a (finite) sum of squared polynomials. Specifically, we write $p \in SOS_{2k}$ if p is a polynomial (of degree at most $2k$) that can be written as a sum of squares of polynomials of degree k. It is well-known that a univariate polynomial p of degree $2k$ is nonnegative on the entire real line if and only if $p \in SOS_{2k}$. Similarly, a polynomial p of degree n is nonnegative over $[-1, 1]$ if and only if it can be written as a weighted sum of squared polynomials [32], either in the form of

$$p(t) = (1+t)q(t) + (1-t)r(t), \quad q \in SOS_{2k-2},\ s \in SOS_{2k-2} \quad \text{if } n = 2k-1, \tag{1}$$

or in the form

$$p(t) = (1+t)(1-t)q(t) + s(t), \quad q \in SOS_{2k-2},\ s \in SOS_{2k}, \qquad \text{if } n = 2k. \tag{2}$$

Analogous sum-of-squares representations of nonnegative polynomials over other intervals (including half-lines) can be constructed via an appropriate change of variables in (1)–(2); see, e.g., [35, 42].

These sum-of-squares representations, in turn, yield semidefinite representations of the set of nonnegative polynomials of a fixed degree, using the fact that the cone of sums of squares of functions from any finite dimensional functional space is semidefinite representable [35, 39]. The precise form of this semidefinite representation depends on the bases that the polynomials being squared (q, r, and s in (1)–(2)) and the sum-of-squares polynomial (p) are represented in.

For the purposes of this paper, we need a representation that uses only the values of p and its derivatives at prescribed interpolation points. For Lagrange interpolants at the points $t_0, \dots, t_n$, this is equivalent to representing the squared polynomials in an arbitrary basis, while representing p in the Lagrange basis polynomials corresponding to the interpolation points $t_0, \dots, t_n$. The theorem below gives the explicit semidefinite representation of sum-of-squares interpolants for a fixed set of interpolation points. In the following, $\mathbf{A} \bullet \mathbf{B}$ denotes the Frobenius inner product $\sum_{i,j} \mathbf{A}_{ij}\mathbf{B}_{ij}$.

Theorem 1 ([31]) *Let $t_0, \dots, t_{2k} \in \mathbb{R}$ be distinct interpolation points and $f_0, \dots, f_{2k} \in \mathbb{R}$ be arbitrary function values prescribed at these points. Fix an arbitrary basis $p_0, \dots, p_k$ of polynomials of degree k. Then there is a nonnegative polynomial $q \in SOS_{2k}$ interpolating each (t_ℓ, f_ℓ) if and only if there exists a $(k+1) \times (k+1)$ positive semidefinite matrix $\mathbf{X}$ satisfying*

$$\mathbf{A}^{(\ell)} \bullet \mathbf{X} = f_\ell \quad \ell = 0, \dots, 2k, \qquad \textit{where} \quad \mathbf{A}^{(\ell)}_{ij} = p_i(t_\ell)p_j(t_\ell). \tag{3}$$

Proof Using the shorthand $\mathbf{p}(t)$ to denote the column vector $(p_0(t), \dots, p_k(t))^{\mathrm{T}}$, $q \in SOS_{2k}$ if and only if there exists some $(k+1) \times (k+1)$ positive semidefinite matrix $\mathbf{X}$ with which

$$q(t) = \mathbf{p}(t)^{\mathrm{T}}\mathbf{X}\mathbf{p}(t) = (\mathbf{p}(t)\mathbf{p}(t)^{\mathrm{T}}) \bullet \mathbf{X} \quad \text{for every } t \in \mathbb{R}. \tag{4}$$

Since the prescribed values and the degree determine q uniquely, Eq. (4) holds for every $t \in \mathbb{R}$ if and only if it holds for each t_ℓ, $\ell = 0, \dots, 2k$:

$$f_\ell = q(t_\ell) = \mathbf{p}(t_\ell)^{\mathrm{T}}\mathbf{X}\mathbf{p}(t_\ell) = (\mathbf{p}(t_\ell)\mathbf{p}(t_\ell)^{\mathrm{T}}) \bullet \mathbf{X} = \mathbf{A}^{(\ell)} \bullet \mathbf{X} \quad \ell = 0, \dots, 2k,$$

which is precisely Eq. (3) in our claim. □

As a side note we shall mention that Theorem 1 easily generalizes to *Hermite interpolants* as well, that is, to representations of polynomials via prescribed function and derivative values at given points:

Theorem 2 *Let $t_1, \dots, t_k \in \mathbb{R}$ be distinct interpolation points, let the fixed nonnegative integer multiplicities $m_1, \dots, m_k$ and the degree d satisfy $2d + 1 = \sum_{\ell=1}^{k}(m_\ell + 1)$, and let $f_\ell^{(m)} \in \mathbb{R}$ be arbitrary prescribed values of the mth derivative at t_ℓ for every $\ell = 1, \dots, k$ and $m = 0, \dots, m_\ell$. Also fix an arbitrary basis $p_0, \dots, p_d$ of polynomials of degree d. Then there is some nonnegative polynomial $q \in SOS_{2d}$ satisfying*

$$q^{(m)}(t_\ell) = f_\ell^{(m)} \qquad \ell = 1, \dots, k, \quad m = 0, \dots, m_\ell$$

if and only if there exists a $(d+1) \times (d+1)$ positive semidefinite matrix $\mathbf{X}$ satisfying

$$\mathbf{A}^{(\ell,m)} \bullet \mathbf{X} = f_\ell^{(m)} \quad \ell = 0, \dots, 2k, \quad \textit{where} \quad \mathbf{A}_{ij}^{(\ell,m)} = \tfrac{\mathrm{d}^m}{\mathrm{d}t^m}(p_i(t)p_j(t))\big|_{t=t_\ell}. \tag{5}$$

The key difficulty in working with polynomials of high degree is that numerical difficulties may arise if the polynomials involved are represented in an unsuitable basis, such as the monomial basis, as it is customary in the sum-of-squares literature. In the representation of Theorem 1, one can freely choose both the interpolation points and the basis p. The choice of Chebyshev points of the first kind and appropriately scaled Chebyshev polynomials works particularly well, as shown below in Lemma 1. Recall that the *Chebyshev points of the first kind* (of order n) are defined by the formula

$$t_\ell = \cos((\ell + 1/2)\pi/(n+1)) \quad \ell = 0, \dots, n, \tag{6}$$

and that the *Chebyshev polynomials of the first kind* are the sequence of polynomials of increasing degree defined by the recursion

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_i(t) = 2tT_{i-1}(t) - T_{i-2}(t) \quad i = 2, 3, \dots \tag{7}$$

The following lemma states that if we represent the polynomials to be squared in the appropriately scaled Chebyshev basis, and use Chebyshev points as the interpolation points, then the representation (3) is perfectly scaled.

Lemma 1 *Let $t_0, \dots, t_{2k}$ be the Chebyshev points given in (6) (with $n = 2k$), and define $p_0 = \sqrt{\frac{1}{2k+1}}T_0$ and $p_i = \sqrt{\frac{2}{2k+1}}T_i$ for $i = 1, \dots, k$, where the T_i are the Chebyshev polynomials given in (7). Then the vectors $(p_i(t_0), \dots, p_i(t_{2k}))$ for $i = 0, \dots, k$ form an orthonormal system.*

Proof The statement is an easy consequence of the well-known *discrete orthogonality relation* of Chebyshev polynomials [21, Eq. (3.30)],

$$\sum_{\ell=0}^{n} T_i(t_\ell)T_j(t_\ell) = K_i\delta_{ij},$$

where $K_0 = n + 1$, $K_i = (n + 1)/2$ when $i \geq 1$, and δ is the Kronecker symbol. Applying this identity, we have the following:

1. If $i = j = 0$, then

$$\sum_{\ell=0}^{2k} p_0^2(t_\ell) = \frac{1}{2k+1} \sum_{\ell=0}^{2k} T_0^2(t_\ell) = 1.$$

2. If $i > j = 0$, then

$$\sum_{\ell=0}^{2k} p_i(t_\ell)p_j(t_\ell) = \frac{\sqrt{2}}{2k+1} \sum_{\ell=0}^{2k} T_i(t_\ell)T_0(t_\ell) = 0.$$

3. If $i \geq 1$ and $j \geq 1$, then

$$\sum_{\ell=0}^{2k} p_i(t_\ell)p_j(t_\ell) = \frac{2}{2k+1} \sum_{\ell=0}^{2k} T_i(t_\ell)T_j(t_\ell) = \delta_{ij}.$$

□

We can express this relation in terms of the dual constraints as well. If y_ℓ denotes the dual variable corresponding to the linear equation $\mathbf{A}^{(\ell)} \bullet \mathbf{X} = f_\ell$ in (3), then the dual constraint corresponding to the primal variable $\mathbf{X}$ is that the matrix

$$\mathbf{Y}(\mathbf{y}) \overset{\text{def}}{=} \sum_{\ell=0}^{2d} y_\ell \mathbf{p}(t_\ell)\mathbf{p}(t_\ell)^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \operatorname{diag}(\mathbf{y})\mathbf{P} \tag{8}$$

is positive semidefinite; in the last equation $\mathbf{P} \overset{\text{def}}{=} (p_i(t_\ell))_{i,\ell}$. With the choice of $\mathbf{P}$ inspired by Lemma 1, $\mathbf{P}$ has orthonormal rows, and the dual matrix $\mathbf{Y}$ is in a conveniently factored, well-conditioned form for the solution of semidefinite programs involving $\mathbf{Y}$, even if the number of interpolation points (and the degree of the polynomials involved) is in the thousands. (See Sect. 4 for numerical examples.) An additional advantage is that algorithms that compute the values of Chebyshev polynomials at Chebyshev points to arbitrary accuracy are readily available [13, 16, 47]; also note that these values $T_i(t_\ell)$, and therefore the coefficient matrices $\mathbf{A}^{(\ell)}$ in (3) need only be computed once, offline, for every value of n.

The case of general interpolation points is in principle similarly easy [31]. For every set of points $t_0, \ldots, t_{2k}$,xt one can find a basis $p_0, \ldots, p_k$ of polynomials of degree k satisfying the discrete orthogonality relation

$$\sum_{\ell=0}^{n} p_i(t_\ell)p_j(t_\ell) = \delta_{ij},$$

by taking an arbitrary basis, and applying an orthogonalization procedure, e.g., QR factorization [26, Sect. 19]. It is important to note that the representation (3) does *not* require that the basis $p_0, \ldots, p_k$ is explicitly identified or expressed in any particular basis, only the values of the basis polynomials are needed at the interpolation points. Throughout the orthogonalization procedure one can work directly with the values of the basis polynomials at the prescribed points. For example, the initial basis can be the Chebyshev polynomial basis, as in that basis stable evaluation of the basis polynomials is easy [13, 16], and then the orthogonalization procedure applied to the vectors of function values directly computes the values of $p_i(t_\ell)$ for each interpolation point t_ℓ for the orthogonalized basis p.

The same procedure is applicable to the weighted-sum-of-squares representations (1) and (2) of polynomials that are nonnegative over an interval. In the dual constraint (8), the entry $p_i(t_\ell)$ in the coefficient matrix $\mathbf{P}$ is replaced by $w(t_\ell)^{1/2}p_i(t_\ell)$, where $w(\cdot)$ is the weight polynomial. In the case of polynomials over $[-1, 1]$, this is the polynomial $1 - t$, $1 + t$, or $1 - t^2$, depending on the parity of the degree. It is this weighted coefficient matrix that needs to be orthogonalized for a perfectly scaled representation of the weighted-sum-of-squares constraint.

3 Upsampling

If a constraint in a polynomial optimization problem involves interpolants of different degrees, it is necessary to "lift" the lower degree interpolants into the space of higher degree ones. When polynomials are represented in the monomial basis or in an orthogonal basis, this is straightforward: the coefficients of the higher degree terms simply need to be set to zero. The analogous operation for interpolants, on the surface at least, is more problematic: we must add to our formulation a constraint that the low-degree polynomial must take consistent values at the high-degree interpolants' interpolation points.

Suppose that $p(\cdot)$ is a degree-n polynomial represented by the vector of function values $\mathbf{p} \in \mathbb{R}^{n+1}$ attained at $n+1$ fixed interpolation points, and that $q(\cdot)$ is a degree-N polynomial ($N > n$) represented by the vector of function values $\mathbf{q} \in \mathbb{R}^{N+1}$ attained at $N + 1$ fixed interpolation points that may or may not contain the first $n + 1$ points. Since p and q are uniquely determined by $\mathbf{p}$ and $\mathbf{q}$, respectively, and the evaluation of a function at a given point is a linear functional, there exists a unique $(N + 1) \times (n + 1)$ matrix $\mathbf{B}$ determined by the interpolation points used to represent p and q such that $p = q$ if and only if $\mathbf{Bp} = \mathbf{q}$. In the following, we shall call this matrix the *upsampling matrix*.

The coefficient matrix $\mathbf{B}$ can be determined using interpolation formulae, such as the barycentric interpolation formula [27, Sect. 4]. The computation of $\mathbf{B}$ does not necessarily add to the computational overhead, as it can be computed offline for every pair (n, N), as long as some fixed interpolation scheme, such as Chebyshev interpolation, is used. Should $\mathbf{B}$ need to be computed from scratch, it can be obtained in $O(nN)$ time, by copying the coefficients from the definition of the barycentric interpolation formula to $\mathbf{B}$; see, for example, Eqs. (3.1)–(3.2) in [27].

It is important to note that although the previous section shows that the sum-of-squares representation of nonnegative interpolants can always be scaled, regardless of the location of the interpolation points, the problem of polynomial interpolation can be inherently ill-conditioned depending on the choice of interpolation points, meaning that small changes in the values of the degree-n polynomial can result in large changes in the upsampled values [7]. On the other hand, if the low-degree polynomial is an interpolant on the Chebyshev points (6), or any other point set with asymptotic density $(1-x^2)^{-1/2}$, then the interpolation problem is well-conditioned, and the coefficients of the upsampling matrix $\mathbf{B}$ can be computed in a numerically stable manner [27].

In this context, the choice of Chebyshev points is optimal. If both polynomials are represented as Chebyshev interpolants using the Chebyshev points of the first kind defined in (6), then the matrix $\mathbf{B}$ has condition number one:

Lemma 2 *Let $\mathbf{B}_{n,N} \in \mathbb{R}^{(N+1)\times(n+1)}$ be the upsampling matrix defined above specialized to the case when the degree-n Chebyshev interpolants are upsampled to degree-N Chebyshev interpolants, using (for both degrees) Chebyshev points of the first kind. Then $\mathbf{B}_{n,N}^{\mathrm{T}}\mathbf{B}_{n,N} = \frac{N+1}{n+1}\mathbf{I}_{n+1}$, where $\mathbf{I}_{n+1}$ is the identity matrix of order $n+1$.*

We omit the proof, as it is lengthy and not particularly insightful; it requires tedious arithmetic involving trigonometric identities.

We may conclude that the use of high-degree representations of low-degree interpolants does not introduce ill-conditioning in an otherwise well-conditioned polynomial optimization problem, provided that the interpolation points are chosen carefully. Motivated by Lemmas 1 and 2, in all the numerical examples of this paper, polynomials are represented as interpolants using Chebyshev points as interpolation points.

4 Applications and Numerical Experiments

4.1 Semi-Infinite Optimization

A linearly constrained semi-infinite convex optimization problem with infinitely many constraints indexed by an interval can be posed as:

$$\begin{aligned} \text{minimize}_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{A}(t)\mathbf{x} \leq \mathbf{b} \quad \forall\, t \in [a,b] \\ & \mathbf{x} \in X \end{aligned} \tag{9}$$

with respect to the decision variables $\mathbf{x}$, where the set $X \subseteq \mathbb{R}^n$ is convex, closed and bounded, and f is convex and continuous on X. Without restrictions on the dependence of $\mathbf{A}$ on t, (9) is a convex optimization problem, and the Weierstrass extreme value theorem guarantees that its minimum is attained.

In many applications, $\mathbf{x}$ represents a (not necessarily polynomial) function $p\colon [a,b] \to \mathbb{R}$ that is known to belong to a given finite dimensional linear space (that is, $\mathbf{x}$ is the coefficient vector of p in some fixed basis), and the infinite constraint set represents $p(t) \geq 0$ for all $t \in T$. Similar constraints on the derivatives of a differentiable function, such as $\frac{dp(t)}{dt} \geq 0$ (implying monotonicity) or $\frac{d^2p(t)}{dt^2} \geq 0$ (implying convexity) can also be represented in a similar fashion. Optimization models incorporating such constraints have been used, for example, in arrival rate estimation [3], and in semi-parametric density estimation with and without shape constraints [38], and design of experiments [36].

Suppose that the set X and the objective function f are semidefinite representable, as defined, for example, in [6, Sect. 4.2]; in other words, assume that (9) without the infinite constraint set can be equivalently written as an SDP. It follows that if the components of $\mathbf{A}(\cdot)$ are polynomials, then the infinite constraint set can be written as the constraint that the components of $\mathbf{b} - \mathbf{A}(t)\mathbf{x}$ are sums-of-squares. Therefore, in this case the semi-infinite program (9) can be formulated as an SDP.

In most applications, the components of $\mathbf{A}(\cdot)$ are not polynomials, but are continuous functions on $[a,b]$. In this case, the components $\mathbf{A}(\cdot)$ can be approximated arbitrarily closely in the uniform norm by polynomials of sufficiently high degree (Weierstrass approximation theorem; [45, Chap. 1]). This suggests the following approach to solving (9):

1. Choose a family of interpolation points. If the application does not prescribe them, use Chebyshev points defined in (6).
2. Find a componentwise polynomial approximation $\mathbf{P}(t)$ of each component of $\mathbf{A}(t)$, expressed as an interpolant, by evaluating $\mathbf{A}(\cdot)$ at each interpolation point. Tools from constructive approximation theory such as `chebfun` can be used to obtain near-machine precision approximations in an automated fashion. Otherwise, a sufficiently high degree approximation can be chosen manually considering bounds such as [47, Theorem 16.1].
3. If some constraints involve interpolants of different degree, use the upsampling constraints of Sect. 3 to ensure that the high-degree representations of low-degree polynomials are consistent.
4. Reformulate the polynomial inequalities $\mathbf{b}-\mathbf{P}(t)\mathbf{x} \geq 0$ as semidefinite constraints using Theorem 1 (if Lagrange interpolation is used) or Theorem 2 (for Hermite interpolation).
5. If the degree of the components of $\mathbf{P}(\cdot)$ is high, use the procedure in Sect. 2 to orthogonalize the semidefinite representation of the polynomial constraints. If Chebyshev points were chosen in Step 1, Lemma 1 gives the orthonormal representation in closed form, and this step can be omitted.
6. Solve the resulting SDP with a suitable solver.

Note that as long as the approximation $\mathbf{P}(t)$ for $\mathbf{A}(t)$ is sufficiently close, the original problem and the polynomial approximation are numerically equivalent. Specifically, infeasibility or unboundedness in (9) is detected in the last step.

4.2 Best One-Sided Polynomial Approximations

The example below is a semi-infinite optimization problem constructed to test the sum-of-squares Lagrange interpolants and upsampling defined in Sects. 2–3. It is numerically challenging, but can be solved in essentially closed form using a theorem of Bojanic and DeVore [10] (see Proposition 1 below), making it an ideal benchmark problem.

Example 1 Finding the best polynomial lower approximations in the L_1 norm (of different degrees $n = 1, 2, \dots$) of the function $f(t) = \exp(t^{100})$ over $[-1, 1]$.

For many smooth functions, the best one-sided approximations of in the L_1-norm are characterized as Hermite interpolants at the zeros of Legendre or Jacobi polynomials [17] of appropriate degree. In the interest of space, we shall only recall here the theorem relevant to Example 1, when n is odd:

Proposition 1 (Bojanic and DeVore [10]) *Assume that $n = 2k-1$ is odd, and that f is a continuous function on $[-1, 1]$ whose $(n+1)$-st derivative is nonnegative on $(-1, 1)$. Then the degree-n polynomial of best approximation of f from below in the L_1 norm is the unique polynomial p_n satisfying*

$$p_n(t_\ell) = f(t_\ell) \ \ \text{and} \ \ p_n'(t_j) = f'(t_\ell), \quad \ell = 1, \dots, k,$$

where $t_1, \dots, t_k$ be the zeros of the Legendre polynomial of degree k.

Proposition 1 provides a characterization of the optimal solution for $n = 2k - 1$ as an Hermite interpolant on the roots of the degree-k Legendre polynomial L_k, meaning the points of contact are the (known, and numerically precisely computable) roots of L_k, allowing us to check the accuracy of our calculations.

Following the approach outlined in Sect. 4.1, we first replace the non-polynomial function f with a close polynomial approximation. It can be shown using [47, Theorem 16.1] that the Lagrange interpolant p_{200} of f on the 200 Chebyshev points has a maximum absolute error smaller than double machine precision. Therefore, for numerical purposes, finding the best polynomial lower approximant p (of a given degree lower than 200) to f is equivalent to computing the optimal solution p to the problem

$$\begin{aligned} \text{maximize}_p \quad & \int_{-1}^{1} p(t)dt \\ \text{subject to} \quad & p(t) \le p_{200}(t) \ \ \forall\, t \in [-1, 1]. \end{aligned} \tag{10}$$

This problem is ready to be translated to a semidefinite program. Note that p has a fixed degree less than 200, therefore it has to be upsampled as discussed in Sect. 3.

As an example, we solved this problem to determine the optimal polynomial lower approximant of degree $n = 49$ (represented as an interpolant on the 50 Chebyshev points) using SeDuMi. For the highest numerically possible accuracy, we set the SeDuMi accuracy goal `eps` to zero so that the solver iterates while it can make any progress. Finally, the points of contact of the roots of optimal approximant were determined numerically using the root finding algorithm for interpolants implemented in the Matlab toolbox Chebfun v.5.0.1 [16]. The obtained roots are shown in Table 1 in the Appendix, next to the correct values with machine precision accuracy. The largest absolute error of the roots was $6.16 \cdot 10^{-7}$, the largest relative error (not defined for the root equal to zero) was $4.88 \cdot 10^{-6}$. In other words, all roots were accurate up to at least five significant digits. On the other hand, the standard SDP formulation of the same problem (following, e.g., [39] and [35]) cannot be solved with the same solver due to numerical problems.

4.3 Polynomial Envelopes of Non-smooth Functions

Our next example is a variation of the previous one involving non-smooth functions. This means that the solutions are no longer available in closed form, but on the other hand, the functions to be approximated do not have good polynomial approximations of low degree, allowing us to arbitrarily increase the degree of the polynomials involved. These examples are intended to demonstrate that the computational infrastructure presented in Sect. 2 is indeed capable of handling very high-degree polynomials without any numerical difficulties.

Consider the following problem: given degree-d polynomials $p_1, \dots, p_m$, find the greatest degree-n polynomial lower approximation of $\min(p_1, \dots, p_m)$, where the minimum is understood pointwise. Formally, we seek the optimal solution to

$$\begin{array}{ll} \text{maximize}_p & \displaystyle\int_{-1}^{1} p(t)dt \\ \text{subject to} & p(t) \leq p_i(t) \quad \forall\, t \in [-1, 1] \quad i = 1, \dots, m. \end{array} \tag{11}$$

All polynomials involved can be represented as interpolants on the same $\max(n, d) + 1$ points. The decision variables are the function values $p(t_\ell)$, $\ell = 1, \dots, \max(n, d) + 1$ at the interpolation points t_ℓ. The nonnegativity of the polynomials $p_i(t) - p(t)$ can be formulated as these polynomials being weighted sums of squares, with a representation (1) or (2) depending on the parity of the degrees. The integral in the objective can be replaced by the sum $\sum_\ell p(t_l)w_\ell$ with appropriately chosen weights w_ℓ for an explicit representation as a linear function of the decision variables. In the examples below $n \geq d$, so we do not have to use upsampling.

Random instances were generated by drawing uniformly random integer coefficients from $[-9, 9]$ for each p_i represented in the Chebyshev basis. We employed three different solvers, SeDuMi [44], SDPT3 version 4 [46], and CSDP version 6.2 [11], each running in Matlab 2014a, to confirm that the semidefinite formulations can indeed be solved with off-the-shelf SDP solvers, for different numbers of polynomials m as well as the degrees n and d.

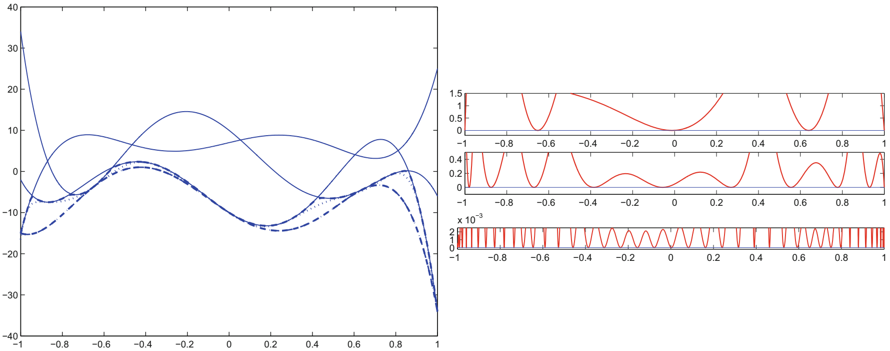


Fig. 1 *Left panel*: Three polynomials of degree 5 (*solid lines*), along with their best lower polynomial approximations of degree 5 (*dot-dashed*), 15 (*dotted*), and 75 (*dashed*). *Right panel*: Pointwise difference between the three optimal polynomials and $\min_i p_i$ from. Only the near-zero section of the plots are shown; the number of contact points can be easily read off the diagrams. The polynomials have degrees 5, 15, and 75, respectively

Example 2 Figure 1 (left panel) depicts three quintic polynomials, along with three best polynomial lower approximations of their pointwise minimum, of increasing degrees (degree 5, 15, and 75). The 75-degree lower approximation is visually nearly indistinguishable from the minimum of the three polynomials. The computations for this plot were carried out using SeDuMi.

To find the optimal polynomials with the highest numerically possible accuracy, we set the SeDuMi accuracy goal `eps` to zero so that the solver keeps iterating as long as it can make any progress. The right panel on Fig. 1 shows the plot of the difference between $\min_i p_i$ and the three polynomial lower approximations. Only the sections of the plots close to the x-axis are shown, in order to demonstrate that the resulting optimal polynomials are computed to sufficiently high accuracy that the points of contact (the points where $\min_i p_i(t) = p(t)$) can be separated, and computed to several digits of precision.

To test the limits of the approach when applied to polynomials of very high degree, similar problems were solved for higher values of n, with the three SDP solvers mentioned above (SeDuMi, SDPT3, and CSDP). As before, to obtain the highest possible accuracy, we set tolerances and accuracy goals to zero so that the solvers keep iterating as long as they can make any progress. Otherwise, default parameter settings were used with both solvers.

As the sizes of the SDPs grow quadratically with the degree of the polynomials involved, the available memory quickly becomes the firstbottleneck. Therefore, we reduced the number of constraints to $m = 2$, and then increased n as shown in the table of results (in the Appendix). Using a standard desktop computer with 32 GB RAM, the degree was increased until the solvers ran out of memory. The number of nonzeros in the constraint matrix of the semidefinite program, along with the number of iterations, the solver running time, and the final duality gap for each run of SeDuMi is shown in Table 2 in the Appendix; the same solver statistics (without repeating the problem statistics) for SDPT3 are shown in Table 3, and in Table 4 for CSDP. It is apparent from the results that the solvers are able to solve even the largest instances, involving polynomials of degree 1000, without any numerical difficulty, and the memory constraint is the only bottleneck.

4.4 Experimental Design

The goal of optimal design of experiments [12, 18, 40] is to maximize the quality of statistical inference by collecting the right data, given limited resources. In the context of linear regression, the inference is based on a data model

$$y(t) = \sum_{i=1}^{m} \beta_i f_i(t) + \epsilon(t), \tag{12}$$

where $f_1, \dots, f_m$ are known functions, and the random variable ϵ (of known probability distribution) represents measurement errors and other sources of variation unexplained by the model. In the experiment, the (noisy) values of y are observed for a number of different values of t chosen from the given *design space* $\mathcal{I}$, and the goal of the experiment is to infer the values of the unknown coefficients β_i, $i = 1, \dots, m$. By an *experimental design* we mean a set of values $\{t_1, \dots, t_s\}$ for which the response $y(t_i)$ is to be measured, along with the number of repeated measurements r_i to be taken at each t_i. The problem of deciding how many (discrete) measurements to take at what points t_i is a non-convex (combinatorial) optimization problem, which is commonly simplified to a convex problem by relaxing the integrality constraints on r_i [18, 40]. In the resulting model one can normalize the vector r by assuming $\sum_i r_i = 1$ (in addition to $r \geq 0$), so that r_i represents not the number, but the fraction of experiments to be conducted at point t_i. This way, the experiment design is mathematically a finitely supported probability distribution ξ satisfying $\xi = t_i$ with probability r_i, $i = 1, \dots, s$. It is immediate that the feasible set (the set of probability measures supported on a finite subset of a given set $\mathcal{I} \subseteq \mathbb{R}^n$) is convex.

Our goal with the experiment is to maximize our confidence in the estimated components of β. This is quantified using the Fisher information [18, 40] that the measured values carry about β. Using the notation $\mathbf{f}(t) = (f_1(t), \dots, f_m(t))^{\mathrm{T}}$, the *Fisher information matrix* of β corresponding to the design ξ is

$$\mathbf{M}(\xi) = \int_{\mathcal{I}} \mathbf{f}(t)\mathbf{f}(t)^{\mathrm{T}} \omega(t) d\xi(t); \tag{13}$$

our goal intuitively is to find the design ξ that maximizes this matrix in an appropriate sense. (The integral simplifies to a finite sum for every design.) More precisely, the optimization takes place with respect to some real-valued *optimality* *criterion* Φ that measures the quality of the Fisher information matrix. The design $\hat{\xi}$ is called Φ-*optimal* if $\Phi(\mathbf{M}(\hat{\xi}))$ is maximum. Popular choices of Φ include $\Phi(\mathbf{M}) = \det(\mathbf{M})$, $\Phi(\mathbf{M}) = \lambda_1(\mathbf{M})$ (smallest eigenvalue), $\Phi(\mathbf{M}) = -\operatorname{tr}(\mathbf{M}^{-1})$, and $\Phi(\mathbf{M}) = (\operatorname{tr}(\mathbf{M}^p))^{1/p}$ for $p \geq 1$; note that these are all semidefinite representable.

The main result of [37] (Theorem 2 of the paper) is that if the basis functions f_i in (12) are polynomials or rational functions, then the optimal design can be computed by solving two semidefinite programs. The first semidefinite program determines a polynomial whose roots are the support points of the optimal design, while the second one is used to determine the probability masses assigned to the support points once the support points are known. The first semidefinite program involves a constraint that a polynomial be nonnegative over the design space. Therefore, this is an instance of (9), with a similar requirement as in the previous examples: the optimal polynomial has to be determined with sufficiently high accuracy to allow for an accurate computation of its roots. Because of this accuracy requirement, the commonly used cutting plane methods of semi-infinite programming (including the author's own [33]) do not have a rate of convergence to be efficient or possibly even feasible, due to the large number of cuts needed to converge to a sufficiently accurate solution, while the SDP formulations can be solved accurately rather quickly using interior-point methods.

Most practical problems involve basis functions that are not polynomials or rational functions, therefore we follow the approach in Sect. 4.1, and replace each f_i by a close polynomial approximation. If any of these approximants has a high degree, then the aforementioned semidefinite program determines a high-degree polynomial that must be nonnegative over $\mathcal{I}$, and that must be computed with sufficient accuracy that allows its roots to be precisely computed.

Example 3 Consider the linear regression model (12) involving a mixture of $m = 3$ Gaussians $f_i = \exp(-3(x - \mu_i)^2)$ with $\mu_1 = -0.5$, $\mu_2 = 0$, and $\mu_3 = 0.5$, and suppose we are interested in finding the optimal design for determining the best fit with respect to the optimality criterion $\Phi(\cdot) = \lambda_1(\cdot)$, over the design space $\mathcal{I} = [-1, 1]$. Since the f_i are not polynomials, we will approximate them by high-degree polynomial interpolants.

Invoking [37, Theorem 2], we obtain that the support of the optimal design is a subset of the roots of the optimal polynomial π determined by the solution of the optimization problem

$$\begin{aligned} &\underset{y\in\mathbb{R},\ \pi\in\mathbb{R}^{d+1},\mathbf{W}\in\mathbb{S}^3_+}{\text{minimize}} && y \\ &\text{subject to} && \operatorname{tr}(\mathbf{W}) = 1 \\ &&& \pi(t) \overset{\text{def}}{=} y - \mathbf{W} \bullet \mathbf{M}_t \geq 0 \quad \forall\, t \in [-1, 1], \end{aligned}$$

where $\mathbf{M}_t = \mathbf{f}(t)\mathbf{f}(t)^T$ with $\mathbf{f}(t) = (f_1(t), f_2(t), f_3(t))^{\mathrm{T}}$. ($\mathbb{S}^3_+$ denotes the set of 3×3 positive semidefinite real symmetric matrices.)

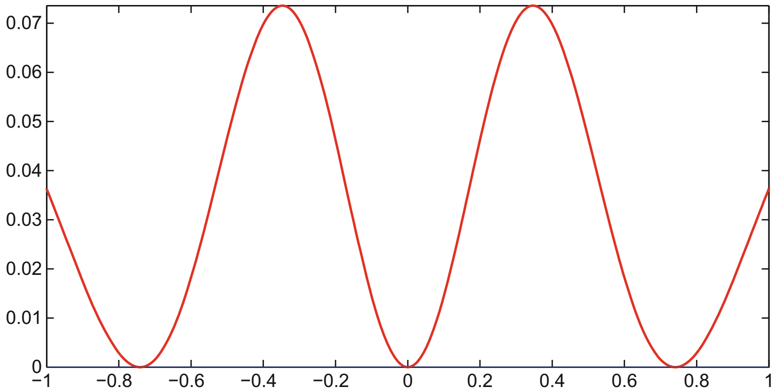


Fig. 2 The optimal polynomial π of degree 39 from the optimal design of experiments problem discussed in Example 3. The optimal design is supported on the roots of this polynomial located in $[-1, 1]$, which are approximately ± 0.7410 and 0

Using `chebfun`, we obtain that all nonpolynomial functions involved in the optimization problem (including not only f_i, but the products f_if_j) can be approximated within machine-precision uniform error over $[-1, 1]$ by polynomial interpolants of degree 39, represented by their values on 40 Chebyshev points. The nonnegativity constraint is replaced by the constraint that π is weighted-sum-of-squares with weights $1 + t$ and $1 - t$.

As in the previous examples, we solved the resulting semidefinite program, and obtained the optimal polynomial shown in Fig. 2. The polynomial has three roots in $[-1, 1]$, these are ± 0.7410 and 0.

This example was also implemented in Matlab, and solved with multiple solvers. Neither solver reported any errors or warnings during the solution, and returned the same solution (within the expected accuracy).

5 Discussion

Several questions remain open, mostly around the multivariate generalization of the methods and the efficiency of the SDPs. While the examples of Sect. 4 can be seen as toy problems, they demonstrate that sum-of-squares interpolants can handle high-degree polynomials that the standard approach cannot. This is very relevant even in the univariate setting, in particular in the models that arise from semi-infinite optimization problems involving non-polynomial functions that can only be approximated using high-degree polynomials; therefore, there is a definite need to be able to reliably optimize over cones of sum-of-squares polynomials of hundreds of degrees. As already mentioned in the introduction, all existing tools of polynomial optimization stop working at too low degrees to be practical for handling near-machine precision polynomial approximations of non-polynomial functions, while the approach presented in the paper works perfectly up to degree 1000 (and likely beyond, if not for the memory constraints). The numerical results point out an independent difficulty (not addressed in this paper) with the SDPs arising in polynomial optimization: that the SDP representation of sum-of-squares polynomials roughly squares the number of optimization variables, increasing the time and memory complexity of the solution algorithms by several orders of magnitude.

Sum-of-squares polynomials admit a semidefinite representation even in the multivariate setting, therefore it is unsurprising that Theorem 1 generalizes word-for-word for multivariate polynomials as long as a suitable set of interpolation points is used (e.g., the Lagrange interpolation problem must be well-posed). In the multivariate setting, the necessity of using high-degree polynomials is even greater, even in the optimization of low-degree polynomials over (semialgebraic) sets defined by low-degree polynomial inequalities. Recall that the standard approach for the solution of such polynomial optimization problems utilizes a sequence (or "hierarchy") of sum-of-squares relaxations that are parameterized by the degree of the sum-of-squares polynomials involved [29, 30, 39]. Each fixed level of the hierarchy provides a lower bound on the true optimal value, and these lower bounds converge asymptotically to the optimal value under conditions specified by a *Positivstellensatz* (a representation theorem of some subset of nonnegative polynomials) such as those of Putinar [41], Schmüdgen [43], Handelman [22] or Pólya [23, p. 57]. However, solving the semidefinite programs arising from these relaxations poses an increasing numerical challenge at higher levels of the hierarchy.

With the numerical problems settled, the next bottleneck (that is especially quickly reached in the multivariate setting) is the time and memory requirement of the solution of these SDPs. This, of course, is not specific to sum-of-squares interpolants, but is present even in the traditional SDP representation of sum-of-squares polynomials. Research is underway to address the huge time and memory requirements of the SDPs and the efficient solution of from sum-of-squares optimization problems in the multivariate setting.

Acknowledgements The author is grateful to Sercan Yıldız for many constructive comments on the presentation of the material.

This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Appendix

Below are the tabulated numerical results from Sect. 4 that are too large to conveniently fit in the text.

Table 1 Comparison of the numerically computed points of contact from Example 1 and the exact values (shown with double machine precision accuracy) derived from Proposition 1

Computed point of contact	Exact point of contact
−0.995556972963306	−0.995556969790498
−0.976663935477085	−0.976663921459518
−0.942974611506432	−0.942974571228974
−0.894992079192159	−0.894991997878275
−0.833442768114398	−0.833442628760834
−0.75925946688898	−0.759259263037358
−0.673566645603713	−0.673566368473468
−0.57766328506073	−0.577662930241223
−0.473003173358265	−0.473002731445715
−0.361172781372549	−0.361172305809388
−0.243867464225739	−0.243866883720988
−0.122865277764643	−0.12286469261071
$-6.15973190950935 \cdot 10^{-7}$	0
0.122864093106243	0.12286469261071
0.243866329210549	0.243866883720988
0.361171807947986	0.361172305809388
0.473002281718963	0.473002731445715
0.577662568403695	0.577662930241223
0.673566077288833	0.673566368473468
0.759259051474652	0.759259263037358
0.833442487648842	0.833442628760834
0.894991911861248	0.894991997878275
0.942974528231816	0.942974571228974
0.976663905379173	0.976663921459518
0.995556966216301	0.995556969790498

In spite of the high degree of the polynomials involved, all computed points of contact (computed as the roots of high-degree sum-of-squares interpolants) are accurate up to at least five significant digits

Table 2 Solver statistics from SeDuMi from the solution of Example 2

$n+1$	# of nonzeros	# of iterations	Solver time (s)	Primal inf.	Dual inf.	Duality gap
100	0.5 M	23	5	$5.0 \cdot 10^{-10}$	$3.0 \cdot 10^{-14}$	$3.25 \cdot 10^{-14}$
200	4.0 M	21	44	$3.5 \cdot 10^{-10}$	$1.5 \cdot 10^{-13}$	$9.04 \cdot 10^{-13}$
300	13.5 M	24	215	$1.7 \cdot 10^{-10}$	$1.4 \cdot 10^{-14}$	$6.23 \cdot 10^{-15}$
400	32.1 M	21	547	$2.3 \cdot 10^{-10}$	$1.7 \cdot 10^{-13}$	$6.01 \cdot 10^{-14}$
500	62.6 M	19	1128	$1.1 \cdot 10^{-9}$	$9.1 \cdot 10^{-13}$	$2.43 \cdot 10^{-13}$
600	108 M	20	2456	$2.6 \cdot 10^{-9}$	$2.0 \cdot 10^{-12}$	$4.56 \cdot 10^{-13}$
700	171 M	21	4847	$4.8 \cdot 10^{-10}$	$3.0 \cdot 10^{-13}$	$6.19 \cdot 10^{-14}$
800	256 M	21	8670	$7.2 \cdot 10^{-10}$	$3.2 \cdot 10^{-13}$	$5.76 \cdot 10^{-14}$
900	321 M	20	12969	$1.9 \cdot 10^{-9}$	$1.1 \cdot 10^{-12}$	$1.80 \cdot 10^{-13}$
1000	501 M	19	19875	$5 \cdot 10^{-9}$	$2.8 \cdot 10^{-12}$	$4.19 \cdot 10^{-13}$

Instances of the optimization problem (11) was solved for $m = 2$, $d = 5$, and different values of the degree n. (That is, $n + 1$ in the heading is the number of interpolation points.) M in the second column stands for millions. The last three columns show the relative and infeasibility of the optimal primal and dual solutions, and the relative duality gap. Larger problems ($n + 1 \geq 1100$) could not be solved because of memory constraints

Table 3 Solver statistics from SDPT3 from the solution of Example 2

$n+1$	# of iterations	Solver time (s)	Primal inf.	Dual inf.	Duality gap
200	25	25	$1.4 \cdot 10^{-9}$	$3.9 \cdot 10^{-12}$	$5.7 \cdot 10^{-12}$
300	29	107	$8.5 \cdot 10^{-9}$	$1.0 \cdot 10^{-12}$	$1.5 \cdot 10^{-11}$
400	26	264	$2.7 \cdot 10^{-9}$	$5.1 \cdot 10^{-12}$	$1.7 \cdot 10^{-11}$
500	29	695	$3.4 \cdot 10^{-9}$	$4.3 \cdot 10^{-13}$	$1.6 \cdot 10^{-11}$
600	30	1395	$9.7 \cdot 10^{-10}$	$1.6 \cdot 10^{-12}$	$3.1 \cdot 10^{-10}$
700	30	2527	$2.2 \cdot 10^{-9}$	$9.5 \cdot 10^{-13}$	$1.7 \cdot 10^{-10}$
800	33	4732	$3.0 \cdot 10^{-8}$	$2.3 \cdot 10^{-13}$	$5.4 \cdot 10^{-12}$
900	30	6724	$5.6 \cdot 10^{-10}$	$4.2 \cdot 10^{-12}$	$9.1 \cdot 10^{-10}$
1000	31	10505	$3.9 \cdot 10^{-10}$	$2.2 \cdot 10^{-13}$	$2.4 \cdot 10^{-11}$

Instances of the optimization problem (11) was solved for $m = 2$, $d = 5$, and different values of the degree n. Larger problems ($n + 1 \geq 1100$) could not be solved because of memory constraints

Table 4 Solver statistics from CSDP from the solution of Example 2

$n+1$	# of iterations	Solver time (s)	Primal inf.	Dual inf.	Duality gap
100	17	1	$1.89 \cdot 10^{-11}$	$6.68 \cdot 10^{-13}$	$1.74 \cdot 10^{-9}$
200	19	10	$2.57 \cdot 10^{-12}$	$1.63 \cdot 10^{-12}$	$3.71 \cdot 10^{-10}$
300	21	45	$9.13 \cdot 10^{-12}$	$2.29 \cdot 10^{-10}$	$1.70 \cdot 10^{-9}$
400	19	136	$1.83 \cdot 10^{-11}$	$7.02 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
500	21	371	$4.41 \cdot 10^{-12}$	$7.83 \cdot 10^{-10}$	$1.72 \cdot 10^{-9}$
600	23	788	$4.79 \cdot 10^{-12}$	$1.23 \cdot 10^{-10}$	$1.59 \cdot 10^{-9}$
700	22	1486	$1.54 \cdot 10^{-12}$	$3.09 \cdot 10^{-10}$	$8.39 \cdot 10^{-10}$
800	22	2474	$3.33 \cdot 10^{-12}$	$1.72 \cdot 10^{-9}$	$1.97 \cdot 10^{-9}$
900	20	4569	$7.25 \cdot 10^{-12}$	$3.57 \cdot 10^{-11}$	$7.38 \cdot 10^{-9}$
1000	22	8634	$9.00 \cdot 10^{-13}$	$1.84 \cdot 10^{-10}$	$5.68 \cdot 10^{-10}$
1100	22	15129	$9.88 \cdot 10^{-13}$	$4.46 \cdot 10^{-9}$	$7.75 \cdot 10^{-10}$

Instances of the optimization problem (11) was solved for $m = 2$, $d = 5$, and different values of the degree n. Larger problems ($n + 1 \geq 1200$) could not be solved because of memory constraints

References

1. Ahmadi, A.A., Parrilo, P.A.: Non-monotonic Lyapunov functions for stability of discrete time nonlinear and switched systems. In: 47th IEEE Conference on Decision and Control (CDC), pp. 614–621. IEEE, Piscataway (2008)
2. Alizadeh, F.: Semidefinite and second-order cone programming and their application to shape-constrained regression and density estimation. In: Proceedings of the INFORMS Annual Meeting. INFORMS, Pittsburg, PA (2006)
3. Alizadeh, F., Papp, D.: Estimating arrival rate of nonhomogeneous Poisson processes with semidefinite programming. Ann. Oper. Res. **208**(1), 291–308 (2013). doi:10.1007/s10479-011-1020-2
4. Bachoc, C., Vallentin, F.: New upper bounds for kissing numbers from semidefinite programming. J. Am. Math. Soc. **21**(3), 909–924 (2008)
5. Ballinger, B., Blekherman, G., Cohn, H., Giansiracusa, N., Kelly, E., Schürmann, A.: Experimental study of energy-minimizing point configurations on spheres. Exp. Math. **18**(3), 257–283 (2009)
6. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization. SIAM, Philadelphia, PA (2001)
7. Berrut, J.P., Trefethen, L.N.: Barycentric Lagrange interpolation. SIAM Rev. **46**(3), 501–517 (2004). doi:10.1137/S0036144502417715
8. Blekherman, G.: Nonnegative polynomials and sums of squares. J. Am. Math. Assoc. **25**(3), 617–635 (2012). doi:10.1090/S0894-0347-2012-00733-4
9. Blekherman, G., Parrilo, P.A., Thomas, R.R.: Semidefinite optimization and convex algebraic geometry. SIAM, Philadelphia (2013)
10. Bojanic, R., DeVore, R.: On polynomials of best one sided approximation. L'Enseignement Mathématique **12**(3), 139–164 (1966)
11. Borchers, B.: CSDP, a C library for semidefinite programming. Optim. Methods Softw. **11–2**(1–4), 613–623 (1999)
12. Chaloner, K.: Optimal Bayesian experimental design for linear models. Ann. Stat. **12**(1), 283–300 (1984)
13. Clenshaw, C.W.: A note on the summation of Chebyshev series. Math. Comput. **9**, 118–120 (1955). doi:10.1090/S0025-5718-1955-0071856-0
14. de Klerk, E.: Computer-assisted proofs and semidefinite programming. Optima **100**, 11–12 (2016)
15. Dette, H.: Optimal designs for a class of polynomials of odd or even degree. Ann. Stat. **20**(1), 238–259 (1992). doi:10.1214/aos/1176348520
16. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun Guide. Pafnuty Publications, Oxford, UK (2014)
17. Dunkl, C.F., Xu, Y.: Orthogonal Polynomials of Several Variables. In: Encyclopedia of Mathematics and Its Applications, vol. 81. Cambridge University Press, Cambridge, UK (2001)
18. Fedorov, V.V.: Theory of Optimal Experiments. Academic, New York, NY (1972)
19. Genin, Y., Hachez, Y., Nesterov, Y., Van Dooren, P.: Convex optimization over positive polynomials and filter design. In: Proceedings of the 2000 UKACC International Conference on Control (2000)
20. Ghaddar, B., Marecek, J., Mevissen, M.: Optimal power flow as a polynomial optimization problem. IEEE Trans. Power Syst. **31**(1), 539–546 (2016)
21. Gil, A., Segura, J., Temme, N.M.: Numerical methods for special functions. SIAM, Philadelphia, PA (2007)
22. Handelman, D.: Representing polynomials by positive linear functions on compact convex polyhedra. Pac. J. Math. **132**(1), 35–62 (1988)

23. Hardy, G.H., Littlewood, J.E., Pólya, G.: Inequalities. Cambridge University Press, Cambridge (1934)
24. Henrion, D., Lasserre, J.B., Löfberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**(4–5), 761–779 (2009)
25. Heß, R., Henrion, D., Lasserre, J.B., Pham, T.S.: Semidefinite approximations of the polynomial abscissa. SIAM J. Control Optim. **54**(3), 1633–1656 (2016)
26. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia (2002)
27. Higham, N.J.: The numerical stability of barycentric Lagrange interpolation. IMA J. Numer. Anal. **24**(4), 547–556 (2004)
28. Horst, R., Pardalos, P.M.: Handbook of Global Optimization, vol. 2. Springer, New York (2013)
29. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)
30. Lasserre, J.B., Toh, K.C., Yang, S.: A bounded degree SOS hierarchy for polynomial optimization. EURO J. Comput. Optim. **5**(1), 87–117 (2017). doi:10.1007/s13675-015-0050-y
31. Lofberg, J., Parrilo, P.A.: From coefficients to samples: a new approach to SOS optimization. In: 43rd IEEE Conference on Decision and Control, vol. 3, pp. 3154–3159. IEEE, Piscataway (2004)
32. Lukács, F.: Verschärfung der ersten Mittelwertsatzes der Integralrechnung für rationale Polynome. Math. Z. **2**, 229–305 (1918). doi:10.1007/BF01199412
33. Mehrotra, S., Papp, D.: A cutting surface algorithm for semi-infinite convex programming with an application to moment robust optimization. SIAM J. Optim. **24**(4), 1670–1697 (2014). doi:10.1137/130925013
34. Menini, L., Tornambè, A.: Exact sum of squares decomposition of univariate polynomials. In: 54th IEEE Conference on Decision and Control (CDC), pp. 1072–1077 (2015). doi:10.1109/CDC.2015.7402354
35. Nesterov, Y.: Squared functional systems and optimization problems. In: Frenk, H., Roos, K., Terlaky, T., Zhang, S. (eds.) High Performance Optimization. Applied Optimization, vol. 33, pp. 405–440. Kluwer Academic Publishers, Dordrecht, The Netherlands (2000)
36. Papp, D.: Optimization models for shape-constrained function estimation problems involving nonnegative polynomials and their restrictions. Ph.D. thesis, Rutgers University (2011)
37. Papp, D.: Optimal designs for rational function regression. J. Am. Stat. Assoc. **107**(497), 400–411 (2012). doi:10.1080/01621459.2012.656035
38. Papp, D., Alizadeh, F.: Shape constrained estimation using nonnegative splines. J. Comput. Graph. Stat. **23**(1), 211–231 (2014)
39. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**(2), 293–320 (2003). doi:10.1007/s10107-003-0387-5
40. Pukelsheim, F.: Optimal Design of Experiments. Wiley, New York (1993)
41. Putinar, M.: Positive polynomials on compact semi-algebraic sets. Indiana Univ. Math. J. **42**, 969–984 (1993)
42. Rudolf, G., Noyan, N., Papp, D., Alizadeh, F.: Bilinear optimality constraints for the cone of positive polynomials. Math. Program. **129**(1), 5–31 (2011). doi:10.1007/s10107-011-0458-y
43. Schmüdgen, K.: The K-moment problem for compact semi-algebraic sets. Math. Ann. **289**, 203–206 (1991). doi:10.1007/BF01446568
44. Sturm, J.F.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optim. Methods Softw. **11–12**(1–4), 625–653 (1999). doi:10.1080/10556789908805766. See also http://sedumi.ie.lehigh.edu/
45. Timan, A.F.: Theory of Approximation of Functions of a Real Variable. Pergamon Press, Oxford, UK (1963)
46. Toh, K.C., Todd, M.J., Tütüncü, R.H.: SDPT3—a Matlab software package for semidefinite programming, version 1.3. Optim. Methods Softw. **11–12**(1–4), 545–581 (1999). doi:10.1080/10556789908805762

47. Trefethen, L.N.: Approximation Theory and Approximation Practice. SIAM, Philadelphia, PA (2013)
48. Tyrtyshnikov, E.E.: How bad are Hankel matrices? Numer. Math. **67**(2), 261–269 (1994). doi:10.1007/s002110050027
49. Unkelbach, J., Papp, D.: The emergence of nonuniform spatiotemporal fractionation schemes within the standard BED model. Med. Phys. **42**(5), 2234–2241 (2015)
50. Vallentin, F.: Optimization in discrete geometry. Optima **100**, 1–10 (2016)