Integer and Matrix Optimization: A Nonlinear Approach

by

Ryan Cory-Wright

B.E. (Hons), University of Auckland (2017)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author .....
Sloan School of Management
May 1, 2022

Certified by .....
Dimitris J. Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Accepted by .....
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electricial Engineering and Computer Science
Co-Director, Operations Research Center

Integer and Matrix Optimization:

A Nonlinear Approach

by

Ryan Cory-Wright

Submitted to the Sloan School of Management
on May 1, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Many important problems from the operations research and statistics literatures exhibit either (a) logical relations between continuous variables x and binary variables z of the form "$x = 0$ if $z = 0$", or (b) rank constraints. Indeed, start-up costs in machine scheduling and financial transaction costs exhibit logical relations, while important problems such as reduced rank regression and matrix completion contain rank constraints. These constraints are commonly viewed as seperate entities and studied by seperate subfields—integer and global optimization respectively—who propose entirely different strategies for optimizing over them.

In this thesis, we adopt a different perspective on logical and rank constraints. We interpret both constraints as purely algebraic ones: logical constraints are nonlinear constraints of the form $x = z \circ x$ for x continuous and z binary (meaning $z^2 = z$), while rank constraints, $\text{Rank}(\boldsymbol{X}) \leq k$, are nonlinear constraints of the form $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ intersected with a linear constraint $\text{tr}(\boldsymbol{Y}) \leq k$ for an orthogonal projection matrix $\boldsymbol{Y}$ (meaning $\boldsymbol{Y}^2 = \boldsymbol{Y}$). Under this lens, we show that regularization drives the computational tractability of problems with both logical and rank constraints.

The first three chapters propose a unified framework to address a class of mixed-integer problems. In numerical experiments, we establish that a general-purpose strategy which combines cutting-plane, rounding, and local search methods, solves these problems faster and at a larger scale than state-of-the-art methods. Our approach solves network design problems with 100s of nodes and provides solutions up to 40% better than the state-of-the-art; sparse portfolio selection problems with up to $3,200$ securities; and sparse PCA problems with up to $5,000$ covariates.

The last two chapters extend this framework to model rank constraints via orthogonal projection matrices. By leveraging regularization and duality, we design outer-approximation algorithms to solve low-rank problems to certifiable optimality, compute lower bounds via their semidefinite relaxations, and provide near optimal solutions through rounding and local search techniques. By invoking matrix perspective functions, we also propose a new class of semidefinite-representable convex relaxations for low-rank problems which outperform the popular nuclear norm penalty.

Thesis Supervisor: Dimitris J. Bertsimas
Title: Boeing Professor of Operations Research

Acknowledgments

First and foremost, I would like to thank my advisor, Dimitris Bertsimas, for his constant support, encouragement, enthusiasm, and friendship. Since I first walked through his office door five years ago, Dimitris has pushed me to become a better researcher and person through his relentless enthusiasm for research and confidence in me. He also leads by example: by forming a 30-person team to tackle COVID within a week of MIT closing in March 2020, and through his impeccable taste in impactful research. I can only aspire to achieve a fraction of the impact he has. I am also grateful for the freedom he granted me during this Ph.D.: to explore projects which most advisors would have deemed dead-ends, to retreat home to a COVID-free New Zealand for five months in 2020, and to write a book with him. I would not be half the researcher I currently am without his support.

I would also like to thank the members of my thesis committee, Alexandre Jacquillat and Robert Freund, for their support and guidance while writing this thesis and their many letters of recommendation. Alex, thank you for your valuable advice as I navigated the academic job market and for the inspiration to improve my running form via your scorching-fast Strava times. Rob, thank you for the general career advice that I will keep in mind as I graduate. Finally, I would like to thank Rob and Bart van Parys for serving on my General Examination committee. Bart, thank you also for the pleasure of being a TA for you twice and for your flexibility in allowing me to TA from New Zealand at the apex of the pandemic.

Many other faculty and staff at MIT deserve thanks. Thank you to Dimitris, Georgia Perakis, and Patrick Jaillet for serving as co-directors of the ORC during my time at MIT. Thank you to Dimitris, Tamara Broderick, Dick den Hertog, Alex, Retsef Levi, Pablo Parrilo, Nikos Trichakis, and Juan Pablo Vielma for enriching classes. Thank you also to Vivek Farias, Daniel Freund, Rob, Alex, and Andy Sun for interesting and enjoyable conversations. Finally, thank you to Laura Rose and Andrew Carvalho for keeping the ORC up and running.

My thanks also go to Jean Pauphilet, who informally served as a second advisor

during much of this thesis. I am grateful for his ability to separate my more valuable ideas from esoteric or downright harebrained ones, for our recurring and delightful hour-long meetings where we always discover something new, and his rigorous testing of my French language skills. I am also grateful for his advice during the academic job market. I am glad that I ended up next door, and I look forward to continuing our collaboration. More importantly, I am grateful to count Jean as a friend.

During my time at MIT, I have also enjoyed working with many other students. I thank Nicholas Johnson for being an enthusiastic collaborator who was truly impressive in his ability to hit the ground running when he arrived at MIT, Sean Lo for his contagious excitement about optimization and his peerless work ethic, Periklis Petridis for a very exciting collaboration on network design, Vassilis Digalakis and Kailyn Byrk for being excellent collaborators on our green energy project, Arthur Delaure, Ted Papalexopoulos and Michael Li for serving as dedicated collaborators on an industry project, and Nihal Koduri, Michael, Vassilis, Léonard Boussioux and Cynthia Zeng for serving as TAs alongside me.

From when I first visited the Operations Research Center, it has been clear to me that it is a welcoming and vibrant community, in addition to a world-class research center. I am grateful to all my classmates and friends for many great conversations over lunch in the ORC kitchen, while grabbing coffee, at the Muddy Charles, at INFORMS, and at the annual retreat in Maine. Thank you to Peter Cohen (and Roya Moussapour) for many great discussions about life and politics, the half-marathon training, and being game enough to walk through some of the proofs in this thesis. Thank you to Holly Wiberg for being an excellent job market buddy and for your brilliance, kindness, and relentless positivity. Thank you also to Arthur and Brad Sturt for invaluable advice on research, writing and teaching which made this Ph.D. immeasurably better, and for some calm words of reassurance during the job market. Thank you to Holly and Elisabeth Paulson for creating the ORC brew club, a fine institution. In addition, my time at MIT has also been made immeasurably better by the following people, among others: Agni Orfanoudaki, Andreea Georgescu, Andrew Li, Arthur, Andy Zeng, Bartolomeo Stellato, Brad, Elisabeth, Emily Meigs, Holly, Ivan

Paskov, Jean, Jess Zhu, Joey Hutchette, Jonathan Amar, Lindsey Blanks, Matthew Sobiesk, Max Biggs, Michael, Nicholas Renegar, Patricio Foneca, Peter, Sam Gilmour, Sam Humphries, Shuvo das Gupta, Ted, Vassilis, Xiaoyue Gong, Yannis Spantidakis, Yeesian Ng, and Zach Blanks.

I am also grateful to my undergraduate advisors, Golbon Zakeri and Andy Philpott, for their encouragement to do a Ph.D., their excellent taste in research in renewable energy, and for making me a better researcher and writer through their feedback.

I would also like to thank my friends on the other side of the world, for being a constant source of encouragement and wisdom. In particular, I am grateful to Ryan Tonkin and Nastassia Subritzky for their support and encouragement. I am also grateful to James Tidswell, Michael Gravatt, Kevin Jia, Alex Carlton, Thomas Adams, Oscar Dowson, and Regan Baucke for stimulating and collegial discussions.

I would be remiss not to acknowledge the important role that two of my high-school teachers, Warwick Gibbs and Anna McHardy, played in making me the person I am today. While I never seemed to pick up German, Mr. Gibbs taught me to think critically about the world. Our wide-ranging discussions when I was meant to be studying French by correspondence school remain one of my fondest memories of high-school. And I remain deeply grateful to Mrs. McHardy (who recently passed away from cancer) for seeing something in me I didn't see in myself when I walked into her classroom, and helping me to become a better version of myself in the four years she was my mathematics teacher. One of my fondest memories from this adventure was sending her an email shortly after arriving at MIT with the subject line "I'm starting a Ph.D. at MIT, and it's because of you."

Finally, I would like to thank my family for their support and unconditional love. To my sister Erin, thank you for keeping me grounded. And to my parents Nigel and Robyn, thank you for your support during this adventure, and for making me who I am today. Thank you for letting your son fly halfway across the world. In the last email she sent me, Anna McHardy taught me one final lesson: the reason I started this Ph.D. wasn't (just) because of her; it's (also) because of my parents. As usual, she is right. Nigel and Robyn, I dedicate this thesis to you.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Many important problems from the operations research, machine learning, and statistics literatures exhibit either (a) logical relations between continuous variables x and binary variables z of the form "$x = 0$ if $z = 0$", or (b) rank constraints. Among others, start-up costs in machine scheduling, financial transaction costs, cardinality constraints, and fixed costs in facility location problems exhibit logical relations. Moreover, important problems such as factor analysis, optimal control, and matrix completion, which model notions of minimal complexity, low dimensionality, or orthogonality in a system, contain rank constraints. These constraints are commonly viewed as seperate entities and studied by seperate subfields of the optimization community—integer and global optimization respectively—who propose entirely different strategies for optimizing over them.

Since the work of Glover [123], logical relations have been well studied by the integer optimization community. They are typically enforced through a linear "big-M" constraint of the form $-Mz \leq x \leq Mz$ for a sufficiently large constant $M > 0$, and optimized over via branch-and-bound or branch-and-cut. Glover's work has been so influential that big-M constraints are now considered as intrinsic components of the initial problem formulations themselves, to the extent that textbooks in the field introduce facility location, network design or sparse portfolio problems with big-M constraints *by default* [see, e.g., 28], although they are actually *reformulations* of logical constraints.

On the other hand, rank constraints are commonly regarded as intractable by the global optimization and machine learning communities, since they cannot be represented using mixed-integer convex optimization [162], and there do not exist any generic codes which solve low-rank optimization problems to certifiable optimality at even moderate problem sizes. This state of affairs has led influential works on low-rank optimization such as [62, 198] to characterize low-rank optimization as intractable and advocate convex relaxations or heuristics which do not enjoy assumption-free optimality guarantees.

In this thesis, we question this state of affairs by proposing one unified approach which addresses both classes of constraints, and solves both sparsity and rank-constrained problems to certifiable optimality or near optimality faster and more accurately than via existing state of the art methods. Eventually, we propose the use of a judicious combination of cutting-plane methods, convex relaxations and greedy rounding techniques. The key insight which facilitates this is an algebraic one, which we lay out in the next section of the chapter. We hope that this approach gives rise to exciting new challenges for the optimization community to tackle, beyond the problems addressed in this thesis.

In this chapter, we outline the contributions of the thesis and provide a chapter by chapter outline. We also introduce the notation we use throughout this thesis.

1.1 Algebraic Formulation and Main Contributions

In this thesis, we adopt a different perspective on both logical and rank constraints. Namely, we interpret both types of constraints as purely algebraic constraints: logical constraints are nonlinear constraints of the form $x = z \circ x$ for x continuous and z binary, while rank constraints, $\text{Rank}(\boldsymbol{X}) \leq k$, are a nonlinear constraint of the form $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ intersected with a linear constraint $\text{tr}(\boldsymbol{Y}) \leq k$ for an orthogonal projection matrix $\boldsymbol{Y}$. Under this lens, we show that regularization drives the computational tractability of problems with logical or rank constraints, and propose an efficient algorithmic strategy which exploits regularization to solve a broad class of problems

with logical or rank constraints to certifiable optimality at scale. We also explore efficient alternatives to the big-M paradigm for both logical and rank constraints, and derive a new class of valid and often very strong convex relaxations for rank-constrained optimization problems. By doing so, we argue that even though sparsity constraints and rank constraints arise in different applications, and are addressed by different research communities using different algorithms, they are really two different aspects of the same unified story. In particular, algorithms which have been known to the mixed-integer community in one form or another for almost 50 years can, if adapted appropriately, solve both classes of problems more accurately and more efficiently than algorithms that are currently considered to be state-of-the-art.

Remark 1. *From a theoretical perspective, the framework proposed in this thesis exhibits an interesting connection to the theory of Euclidean Jordan algebras. Indeed, to model both sparsity and rank constraints, we work with idempotent elements z_i contained in a Jordan algebra such that $z_i^2 = z_i$, $z_i z_j = 0$ if $i \neq j$ and $\sum_{i=1}^n z_i = e$, where e denotes an identity of appropriate dimension; see Faraut and Koranyi [98] for an introduction to analysis over symmetric cones. Moreover, we can interpret both the cardinality of a vector $\boldsymbol{x}$ and the rank of a matrix $\boldsymbol{X}$ as a special case of the Jordan-algebraic rank, i.e., the minimum number of idempotents required to provide a spectral decomposition of $\boldsymbol{x} = \sum_{i=1}^n \lambda_i z_i$ for a binary unit vector z_i or $\boldsymbol{X} = \sum_{i=1}^n \lambda_i \boldsymbol{Y}_i$ for orthogonal projection matrices $\boldsymbol{Y}_i$.*

The thesis comes in two parts: in the first part–consisting of Chapters 2-4–we explore how our framework applies to logically constrained problems, and undertake detailed studies of its scalability for sparse portfolio selection and sparse principal component analysis problems. In the second part–consisting of Chapters 5-6–we explore its implications for low-rank problems, by proposing a technique for solving low-rank problems exactly and undertaking a detailed study of the convex relaxations of a class of low-rank problems.

The main contributions of the first part of the thesis are as follows:

- In Chapter 2, we provide four main contributions: First, we reformulate the

logical constraint "$x_i = 0$ if $z_i = 0$" in a non-linear way, by substituting $z_i x_i$ for x_i in Problem (1.1). Second, we leverage the regularization term $\Omega(\boldsymbol{x})$ to derive a tractable reformulation of (1.1). Third, by invoking strong duality, we reformulate (1.1) as a mixed-integer saddle-point problem, which is solvable via outer approximation. Finally, we demonstrate that algorithms derived from our framework can outperform state-of-the-art solvers. On network design problems with 100s of nodes and binary quadratic optimization problems with 100s of variables, we improve the objective value of the returned solution by 5 to 40% and 5 to 85% respectively, and our edge increases as the problem size increases.

- In Chapter 3, we provide two main contributions. First, we propose augmenting sparse portfolio selection problems with a ridge regularization term. This yields a more practically tractable problem for two reasons. First, the duality gap between a sparse porfolio selection problem and its second-order cone relaxation decreases as we increase the amount of regularization and becomes 0 at with a sufficiently large but finite amount of regularization. Second, as we numerically establish in computational experiments, the algorithms developed in this chapter converge more rapidly with more regularization. Our second main contribution is specializing the outer-approximation method developed in the previous chapter to sparse portfolio selection problems, and demonstrating that this allows us to solve large-scale sparse portfolio selection problems with up to $3,200$ securities to certifiable optimality in hundreds or thousands of seconds.

- In Chapter 4, we provide two main contributions. First, we reformulate sparse PCA exactly as a mixed-integer semidefinite optimization problem; a reformulation which is, to the best of our knowledge, novel. Second, we leverage this MISDO formulation to design efficient algorithms for solving non-convex mixed-integer quadratic optimization problems, such as sparse PCA, to certifiable optimality or within $1-2\%$ of optimality in practice at a larger scale than existing state-of-the-art methods.

The main contributions of the second part of the thesis are as follows:

- The key contributions of Chapter 5 are threefold. First, we propose using orthogonal projection matrices which satisfy $\boldsymbol{Y}^2 = \boldsymbol{Y}$ to model low-rank constraints via the non-linear equation $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$. Under this lens, low-rank problems admit reformulations as optimization problems where some decision variables comprise a projection matrix. We term this family of problems *Mixed-Projection Conic Optimization* (MPCO), in reference to mixed-integer optimization. Second, by leveraging regularization and strong duality we rewrite low-rank optimization problems as saddle-point problems over the space of orthogonal projection matrices that can be solved to optimality via outer-approximation, and propose an outer-approximation method to solve the saddle-point problem to certifiable optimality. Third, by analyzing the saddle-point problem, we derive new convex relaxations and rounding schemes which provide certifiably near-optimal solutions in polynomial time in theory and rapidly in practice. Using a generic spatial branch-and-bound code, we are already able to solve low-rank optimization problems exactly for matrices with 30 rows and columns, and find near-exact solutions for matrices with up to 600 rows and columns. To our knowledge, our approach is the first mathematical framework that solves low-rank optimization problems to certifiable (near-)optimality.

- The main contributions of Chapter 6 are twofold. First, we propose a general reformulation technique for obtaining high-quality relaxations of low-rank optimization problems: introducing an orthogonal projection matrix to model a low-rank constraint, and strengthening the formulation by taking the matrix perspective of an appropriate substructure of the problem. Second, by applying this technique, we obtain explicit characterizations of convex hulls of low-rank sets which frequently arise in low-rank problems.

1.2 Overview and Structure of the Thesis

We now provide a high-level overview and section-by-section summary of each chapter.

Chapter 2 In this chapter, we consider optimization problems which unfold over two stages. In the first stage, a decision-maker activates binary variables, while satisfying budget constraints and incurring activation costs. In the second stage, the decision-maker optimizes over the continuous variables. Formally, we consider:

$$\min_{\boldsymbol{z}\in\mathcal{Z},\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{1.1}$$

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\boldsymbol{c} \in \mathbb{R}^n$ is a cost vector, $g(\cdot)$ is a generic convex function which possibly models convex constraints $\boldsymbol{x} \in \mathcal{X}$ for a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ implicitly—by requiring that $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$, and $\Omega(\cdot)$ is a convex regularization function; in spirit either a big-M regularizer $\Omega(\boldsymbol{x}) = 0$ if $\|\boldsymbol{x}\|_\infty \leq M$ and $+\infty$ otherwise, or a ridge regularizer $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

This chapter is structured as follows:

- In Section 2.1, we provide some background and perform a literature review on existing methods for addressing Problem (1.1).
- In Sections 2.2-2.3, we identify a general class of mixed-integer optimization problems, which encompasses sparse regression, sparse portfolio selection, sparse principal component analysis, unit commitment, facility location, network design and binary quadratic optimization as special cases. For this class of problems, we discuss how imposing either big-M or ridge regularization accounts for non-linear relationships between continuous and binary variables in a tractable fashion. We also establish that regularization controls the convexity and smoothness of Problem (1.1)'s objective function.
- In Sections 2.4-2.7, we propose a conjunction of general-purpose numerical algorithms to solve Problem (1.1). The backbone of our approach is an outer approximation framework, enhanced with first-order methods to solve the Boolean relaxations and obtain improved lower bounds, certifiably near-optimal warm-starts via randomized rounding, and a discrete local search procedure. We also connect our approach to the perspective cut approach [110] from a theoretical and implementation standpoint.

- Finally, in Section 2.8, we demonstrate empirically that algorithms derived from our framework can outperform state-of-the-art solvers. On network design problems with 100s of nodes and binary quadratic optimization problems with 100s of variables, we improve the objective value of the returned solution by 5 to 40% and 5 to 85% respectively, and our edge increases as the problem size increases. We then analyze the benefits of the different ingredients in our numerical recipe on facility location problems, and discuss the relative merits of different regularization approaches on unit commitment instances.

The work in this chapter is based on the article [33], authored with Dimitris Bertsimas and Jean Pauphilet.

Chapter 3 Since the Nobel-prize winning work of Markowitz [173], the problem of selecting an optimal portfolio of securities has received an enormous amount of attention from practitioners and academics alike. In a universe containing n distinct securities with expected marginal returns $\boldsymbol{\mu} \in \mathbb{R}^n$ and a variance-covariance matrix $\boldsymbol{\Sigma} \in S^n_+$, the Markowitz model selects a portfolio which provides the highest expected return for a given amount of variance, by solving

$$\min_{\boldsymbol{x}\in\mathbb{R}^n_+} \ \frac{\sigma}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} - \boldsymbol{\mu}^\top\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{e}^\top\boldsymbol{x} = 1, \tag{1.2}$$

where $\sigma \geq 0$ controls the trade-off between the portfolios risk and return. To improve its realism, Bienstock [42] augmented Problem (1.2) with two sets of inequalities. The first set is a generic system of linear inequalites $\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}$ which ensures that various real-world constraints such as allocating an appropriate amount of capital to each market sector hold. The second inequality limits the number of non-zero positions held to $k \ll n$, by requiring that the portfolio is k-sparse, i.e., $\|\boldsymbol{x}\|_0 \leq k$. By introducing a ridge regularization term, this yields the following portfolio model:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n_+} \ \frac{\sigma}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} - \boldsymbol{\mu}^\top\boldsymbol{x} + \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ \|\boldsymbol{x}\|_0 \leq k, \tag{1.3}$$

where $\gamma > 0$ is a hyperparameter which controls the models robustness to noise.

This chapter is structured as follows:

- In Section 3.1, we provide some background and perform a literature review on existing methods for addressing Problem (1.3).
- In Section 3.2, we propose an efficient numerical strategy for solving Problem (1.3). By observing that Problem (1.3)'s inner dual problem supplies subgradients with respect to the positions held, we design an outer-approximation procedure which solves Problem (1.3) to provable optimality. We also discuss practical aspects of the procedure, including a computationally efficient subproblem strategy and a prepossessing technique for decreasing the bound gap at the root node. In addition, we study the problems sensitivity to γ, and establish theoretically that the support of an optimal portfolio (although not the amount allocated to each security) is stable under small changes in γ.
- In Section 3.3, we propose techniques for obtaining certifiably near-optimal solutions quickly. First, we introduce a heuristic which supplies high-quality warm-starts. Second, we observe that Problem (1.3)'s continuous relaxation supplies a near-exact second-order cone representable lower bound, and exploit this observation by deriving a sufficient condition for the bound to be exact.
- In Section 3.4, we apply the cutting-plane method to the problems described in [69], [110], and three larger scale data sets: the S&P 500, Russell 1000, and Wilshire 5000. We also explore Problem (3.4)'s sensitivity to its hyperparameters, and establish empirically that optimal support indices tend to be stable for reasonable hyperparameter choices.

The work in this chapter is based on [24], authored with Dimitris Bertsimas.

Chapter 4 In the era of big data, interpretable methods for compressing a high-dimensional dataset into a lower dimensional set which shares the same essential characteristics are imperative. Principal component analysis (PCA) is one of the most popular approaches for completing this task. Given data $\boldsymbol{A} \in \mathbb{R}^{n \times p}$ and its sample covariance matrix $\boldsymbol{\Sigma} := \frac{1}{n-1}\boldsymbol{A}\boldsymbol{A}^\top \in \mathbb{R}^{p \times p}$, PCA selects the leading eigenvectors, or

principal components, of $\boldsymbol{\Sigma}$ and subsequently projects $\boldsymbol{A}$ onto these eigenvectors, by multiplying $\boldsymbol{A}$ by the leading principal components. In principal component analysis, one commonly desired property is that the PCs are interpretable, since they are usually a linear combination of all p original features.

One common method for obtaining interpretable principal components is to stipulate that they are sparse. This leads to the following mixed-integer quadratically constrained problem:

$$\max_{\boldsymbol{x}\in\mathbb{R}^p} \quad \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x}^\top \boldsymbol{x} = 1, \quad ||\boldsymbol{x}||_0 \leq k, \tag{1.4}$$

where the constraint $||\boldsymbol{x}||_0 \leq k$ forces variance to be explained in a simple fashion.

The structure of the chapter is as follows:

- In Section 4.1, we provide some background and perform a literature review on existing methods for addressing Problem (1.4).
- In Section 4.2-4.3, we reformulate Problem (1.4) as a mixed-integer SDO under big-M and ridge regularization respectively, and demonstrate that the subproblems which arise from this formulation under Section 2's decomposition scheme are actually solvable in closed form. This is significant because sparse PCA is traditionally treated as a low-rank optimization problem, which as we show in the next chapter is likely a harder class of problems to address.
- The Gershgorin Circle theorem has been empirically successful for deriving upper-bounds on the objective value of (1.4) [20]. We theoretically analyze the quality of such bounds in Section 4.4 and show that even tighter bounds derived from Brauer's ovals of Cassini theorem can also be imposed via mixed-integer second-order cone constraints.
- In Section 4.5, we analyze the semidefinite reformulation's convex relaxation, and introduce a greedy rounding scheme which supplies high-quality solutions to Problem (1.4) in polynomial time, together with a sub-optimality gap. To further improve the quality of rounded solution and the optimality gap, we in-

troduce strengthening inequalities. While solving the strengthened formulation exactly would result in an intractable MISDO problem, solving its relaxation and rounding the solution appears as an efficient strategy to return high-quality solutions with a numerical certificate.

- In Section 4.6, we apply the cutting-plane and random rounding methods to derive optimal and near optimal sparse principal components for problems in the UCI data set. We also compare our method's performance against the method of Berk and Bertsimas [20], and find that our exact cutting-plane method performs comparably, while our relax+round approach successfully scales to problems an order of magnitude larger and often returns solutions which outperform the exact method at sizes which the exact method cannot currently scale to. A key feature of our numerical success is that we sidestep the computational difficulties in solving SDOs at scale by proposing semidefinite-free methods for solving the convex relaxations, i.e., solving second-order cone relaxations.

The work in this chapter is based on the article [37], authored with Dimitris Bertsimas and Jean Pauphilet.

Chapter 5 In this chapter, we consider the problem:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} \quad \langle \boldsymbol{C}, \boldsymbol{X}\rangle + \lambda \cdot \mathrm{Rank}(\boldsymbol{X}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B},\ \mathrm{Rank}(\boldsymbol{X}) \leq k,\ \boldsymbol{X} \in \mathcal{K}, \tag{1.5}$$

where λ (resp. k) prices (bounds) the rank of $\boldsymbol{X}$, $(\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}^{\ell\times n} \times \mathbb{R}^{\ell\times m}$ defines an affine subspace, and $\mathcal{K}$ is a proper cone, i.e., closed, convex, solid and pointed.

This chapter is structured as follows:

- In Section 5.1, we provide some background and perform a literature review on existing methods for addressing Problem (1.5).
- In Section 5.2, we show that projection matrices are a natural generalization of binary vectors to matrices. Inspired by a common tactic in cardinality constrained optimization, namely introducing binary variables to encode the support of the decision vector, we propose introducing a projection matrix to encode

the image of the decision matrix and thereby model rank. We also investigate the complexity of low-rank optimization problems and show that rank minimization is in PSPACE.

- In Section 5.3, we derive the MPCO formulations of the aforementioned rank optimization problems. By introducing a constraint on the spectral norm of $\boldsymbol{X}$ or a penalty on its Frobenius norm - the matrix analogs of big-M constraints and perspective formulations [126] respectively, we leverage strong duality, reformulate Problem (1.5) as a saddle-point problem, and prove the resulting optimization problem admits a convex objective.

- We propose numerical algorithms to solve these MPCO problem to provable (near) optimality in Section 5.4-5.6, by extending some of the most successful techniques from MICO. First, we propose an outer-approximation scheme for solving Problem (1.5) exactly. Then, we obtain valid lower-bounds from solving its convex relaxations and propose an alternating minimization algorithm to do so. In addition, we prove that a singular value decomposition (SVD) followed by greedily rounding the eigenvalues provides certifiably near-optimal solutions in polynomial time.

- In Section 5.7, we implement and numerically evaluate our proposed algorithms. On examples from matrix completion and sensor location, we demonstrate that methods proposed in this paper solve instances of Problem (1.5) to certifiable optimality in minutes for n in the tens. To our knowledge, our work is the first to demonstrate that moderately sized rank constrained problems can be solved to provable optimality in a tractable fashion. For n in the hundreds, our proposal scales and provides in minutes solutions of higher quality than existing heuristics, such as nuclear norm minimization.

The work in this chapter is based on the article [34], authored with Dimitris Bertsimas and Jean Pauphilet.

Chapter 6 In this chapter, we develop strong convex relaxations for the following low-rank optimization problem:

$$\begin{aligned} \min_{\boldsymbol{X}\in\mathcal{S}^n_+} \quad & \langle \boldsymbol{C}, \boldsymbol{X}\rangle + \Omega(\boldsymbol{X}) + \mu \cdot \text{Rank}(\boldsymbol{X}) \\ \text{s.t.} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X}\rangle = b_i \ \forall i \in [m],\ \boldsymbol{X} \in \mathcal{K},\ \text{Rank}(\boldsymbol{X}) \leq k, \end{aligned} \tag{1.6}$$

where $\boldsymbol{C}, \boldsymbol{A}_i, \dots \boldsymbol{A}_m \in \mathcal{S}^n$ are $n \times n$ symmetric matrices, $b_1, \dots b_m \in \mathbb{R}$ are scalars, $[n]$ denotes the set of running indices $\{1, ..., n\}$, and $\mu \in \mathbb{R}_+, k \in \mathbb{N}$ are parameters which controls the complexity of $\boldsymbol{X}$ by respectively penalizing and constraining its rank. The set $\mathcal{K}$ is a proper—i.e., closed, convex, solid and pointed—cone, and $\Omega(\boldsymbol{X}) = \text{tr}(f(\boldsymbol{X})$ is the trace of a matrix-convex function.

The structure of the chapter is as follows:

- In Section 6.1, we provide some background and perform a literature review on existing methods for addressing Problem (1.6).
- In Section 6.2 we supply some background on perspective functions and review their role in developing tight formulations of mixed-integer problems.
- In Section 6.3-6.4, we introduce the matrix perspective function and its properties, extend the function's definition to allow semidefinite in addition to positive definite arguments, and propose a matrix perspective reformulation technique (MPRT) which successfully obtains high-quality relaxations for low-rank problems which commonly arise in the literature.
- In Section 6.5-6.6, we connect the matrix perspective function to the convex hulls of epigraphs of simple matrix convex functions under rank constraints.
- In Section 6.7, we illustrate the utility of this connection by deriving tighter relaxations of low-rank problems than are currently available in the literature.
- Finally, in Section 6.8, we numerically verify the utility of our approach on rank regression, D-optimal design and non-negative matrix problems.

The work in this chapter is based on the preprint [35], authored with Dimitris

Bertsimas and Jean Pauphilet. The preprint has benefited from one round of major revisions at Math. Programming and is currently under a second round of review.

1.3 Notation

Throughout this thesis, ordinary lowercase letters (x, y) denote scalars, boldfaced lowercase letters $(\boldsymbol{x}, \boldsymbol{y}, \ldots)$ denote vectors, boldfaced capital letters $(\boldsymbol{X}, \boldsymbol{Y}, \ldots)$ denote matrices, and boldface Euler script letters $(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}, \ldots)$ denote higher-order tensors. Calligraphic type $(\mathcal{S}, \mathcal{U}, \ldots)$ denotes sets. The notation $[n]$ denotes the set of running indices $\{1, \ldots, n\}$.

Given a vector $\boldsymbol{x} \in \Re^p$, the set $\text{supp}(\boldsymbol{x}) \triangleq \{i : x_i \neq 0,\ i \in [p]\}$ denotes the support of $\boldsymbol{x}$. $\|\boldsymbol{x}\|_0 \triangleq |\text{supp}(\boldsymbol{x})| = \sum_{i\in[p]} \mathcal{I}\{x_i \neq 0\}$ counts the number of nonzero entries of $\boldsymbol{x}$. If $f(\boldsymbol{x})$ is a convex function then its perspective function $\varphi(\boldsymbol{x}, t)$, defined as $\varphi(\boldsymbol{x}, t) = tf(\boldsymbol{x}/t)$ if $t > 0$, $\varphi(\boldsymbol{0}, 0) = 0$, and ∞ elsewhere, is also convex [54, Chapter 3.2.6.]. We let $\text{relint}(\mathcal{X})$ denote the relative interior of a convex set $\mathcal{X}$, i.e., the set of points on the interior of the affine hull of $\mathcal{X}$ [see 54, Section 2.1.3]. Finally, we let $\mathcal{Z}_k^n$ denote the set of k-sparse binary vectors, i.e, $\mathcal{Z}_k^n := \{\boldsymbol{z} \in \{0, 1\}^n :\ \boldsymbol{e}^\top \boldsymbol{z} \leq k\}$.

Given a matrix $\boldsymbol{X}$, $\sigma_i(\boldsymbol{X})$ denotes the ith largest singular value of a matrix $\boldsymbol{X}$, $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product between two vectors or matrices of the same size, $\boldsymbol{X}^\dagger$ denote the Moore-Penrose pseuedoinverse of a matrix $\boldsymbol{X}$, $\|\cdot\|_F$ denote the Frobenius norm of a matrix, $\|\cdot\|_\sigma$ denote the spectral norm of a matrix, and $\|\cdot\|_*$ denote the nuclear norm of a matrix.

We also use a variety of convex cones. We let S^n denote the $n \times n$ cone of symmetric matrices, S_+^n denote the $n \times n$ positive semidefinite cone, $C_+^n = \{\boldsymbol{C} \in \mathbb{R}^{n\times n} : \boldsymbol{C} = \boldsymbol{D}\boldsymbol{D}^\top, \boldsymbol{D} \in \mathbb{R}_+^{n\times n}\}$ denote the $n \times n$ completely positive cone and $DNN^n = S_+^n \cap \mathbb{R}_+^{n\times n} \subseteq C_+^n$ denote the doubly non-negative cone.

Part I

Logical and Sparsity Constraints

Chapter 2

A Unified Approach to MIO With Logical Constraints

In this chapter, we consider optimization problems that unfold over two stages. In the first stage, a decision-maker activates binary variables while satisfying resource budget constraints and incurring activation costs. Subsequently, in the second stage, the decision-maker optimizes over the continuous variables. Formally, we consider:

$$\min_{\boldsymbol{z}\in\mathcal{Z},\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{2.1}$$

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\boldsymbol{c} \in \mathbb{R}^n$ is a cost vector, $g(\cdot)$ is a convex function which possibly models convex constraints $\boldsymbol{x} \in \mathcal{X}$ for a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ implicitly—by requiring that $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$, and $\Omega(\cdot)$ is a convex regularizer which drives the theoretical and practical tractability of the problem; we state its structure in Assumption 2.1.

Observe that the structure of Problem (2.1) is quite general, as the feasible set $\mathcal{Z}$ can capture known lower and upper bounds on $\boldsymbol{z}$, relationships between different z_i's, or a cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$. Moreover, constraints of the form $\boldsymbol{x} \in \mathcal{X}$, for some convex set $\mathcal{X}$, can be encoded within the domain of g, by defining $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$. As a result, Problem (2.1) encompasses a large number of problems from the Operations Research and Machine Learning literatures, such as the network design problem described in Example 2.1. These problems are typically studied separately.

However, the techniques developed for each problem are actually different facets of a single unified story, and can be applied to a much more general class of problems than is often appreciated.

In this chapter, we provide three main insights: First, we reformulate the logical constraint "$x_i = 0$ if $z_i = 0$" in a non-linear way, by substituting $z_i x_i$ for x_i in Problem (2.1). Second, we leverage the regularization term $\Omega(\boldsymbol{x})$ to derive a tractable reformulation of (2.1). Finally, by invoking strong duality, we reformulate (2.1) as a mixed-integer saddle-point problem, which is solvable via a cutting-plane approach.

2.1 Background and Literature Review

Our work falls into two areas of the mixed-integer optimization literature which are often considered in isolation: (a) modeling forcing constraints which encode whether continuous variables are active and can take non-zero values or are inactive and forced to 0, and (b) decomposition algorithms for mixed-integer optimization problems.

Formulations of forcing constraints

The most popular way to impose forcing constraints on continuous variables is to introduce auxiliary discrete variables which encode whether the variables are active, and relate the discrete and continuous variables via the big-M approach of [123]. This approach was first applied to mixed-integer optimization (MIO) in the context of portfolio selection by [42]. With the big-M approach, the original MINLO admits bounded relaxations and can therefore be solved via branch-and-bound. Moreover, because the relationship between discrete and continuous variables is enforced via linear constraints, a big-M reformulation has a theoretically low impact on the tractability of the MINLOs continuous relaxations. However, in practice, high values of M lead to numerical instability and provide low-quality bounds [see 14, Section 5].

This observation led [110] to propose a class of cutting-planes for MINLO problems with indicator variables, called perspective cuts, which often provide a tighter reformulation of the logical constraints. Their approach was subsequently extended

by [4], who, building upon the work of [17, pp. 88, item 5], proved that MINLO problems with indicator variables can often be reformulated as mixed-integer second-order cone problems (see [127] for a survey). More recently, a third approach for coupling the discrete and the continuous in MINLO was proposed independently for sparse regression by [193] and [36]: augmenting the objective with a strongly convex term of the form $\|\boldsymbol{x}\|_2^2$, called a ridge regularizer.

In the present chapter, we synthesize the aforementioned and seemingly unrelated three lines of research under the unifying lens of regularization. Notably, our framework includes big-M and ridge regularization as special cases, and provides an elementary derivation of perspective cuts.

Numerical algorithms for mixed-integer optimization

A variety of "classical" general-purpose decomposition algorithms have been proposed for general MINLOs. The first such decomposition method is known as Generalized Benders Decomposition, and was proposed by [122] as an extension of [19]. A similar method, known as outer-approximation was proposed by [92], who proved its finite termination. The outer-approximation method was subsequently generalized to account for non-linear integral variables by [107]. These techniques decompose MINLOs into a discrete master problem and a sequence of continuous separation problems, which are iteratively solved to generate valid cuts for the master problem.

Though slow in their original implementation, decomposition schemes have benefited from recent improvements in mixed-integer linear solvers in the past decades, beginning with the branch-and-cut approaches of [187, 195], which embed the cut generation process within a single branch-and-bound tree, rather than building a branch-and-bound tree before generating each cut. We refer to [105, 106] for recent successful implementations of "modern" decomposition schemes. From a high-level perspective, these recent successes require three key ingredients: First, a fast cut generation strategy. Second, as advocated by [105], a rich cut generation process at the root node. Finally, a cut selection rule for degenerate cases where multiple valid inequalities exist (e.g., the Pareto optimality criteria of [168]).

In this chapter, we connect the regularization used to reformulate logical constraints with the aforementioned key ingredients for modern decomposition schemes. Hence, instead of considering a MINLO formulation as a given and subsequently attempt to solve it at scale, our approach view big-M constraints as one of many alternatives. We argue that regularization is a modeling choice that impacts the tractability of the formulation and should be made accordingly.

2.2 Framework and Examples

In this section, we present the family of problems to which our analysis applies, discuss the role of regularization, and provide some examples from the Operations Research, machine learning, and statistics literatures.

Problem (2.1) has a two-stage structure which comprises first "turning on" some indicator variables $\boldsymbol{z}$, and second solving a continuous optimization problem over the active components of $\boldsymbol{x}$. Precisely, Problem (2.1) can be viewed as a discrete optimization problem:

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}), \tag{2.2}$$

where the inner minimization problem

$$f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{2.3}$$

yields a best choice of $\boldsymbol{x}$ given $\boldsymbol{z}$. As we illustrate in this section, a number of problems of practical interest exhibit this structure.

Example 2.1. *Network design is an important example of problems of the form* (2.1)*. Given a set of m nodes, the network design problem consists of constructing edges to minimize the construction plus flow transportation cost. Let E denote the set of all*

potential edges and let $n = |E|$. *Then, the network design problem is given by:*

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{Z},\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \qquad\qquad\qquad\qquad\qquad\qquad x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E, \end{aligned} \tag{2.4}$$

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *is the flow conservation matrix,* $\boldsymbol{b} \in \mathbb{R}^m$ *is the vector of external demands and* $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{d} \in \mathbb{R}^n$ *define the quadratic and linear costs of flow circulation. We assume that* $\boldsymbol{Q} \succeq \boldsymbol{0}$ *is a positive semidefinite matrix. Inequalities of the form* $\boldsymbol{\ell} \leq \boldsymbol{z} \leq \boldsymbol{u}$ *can be incorporated within* $\mathcal{Z}$ *to account for existing/forbidden edges in the network. Problem* (2.4) *is of the same form as Problem* (2.1) *with*

$$g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) := \begin{cases} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x}, & \text{if } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, \\ +\infty, & \text{otherwise.} \end{cases}$$

Within our two-stage structure, this gives the formulation

$$f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E.$$

Example 2.1 illustrates that the single-commodity network design problem is a special case of Problem (2.1). We now formulate the k-commodity network design problem with directed capacities as minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{x},\boldsymbol{f}^j\in\mathbb{R}^n_+} \quad & \boldsymbol{c}^\top \boldsymbol{z} + \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{f}^j = \boldsymbol{b}^j \quad \forall j \in [k], \\ & \qquad\qquad\qquad\qquad \boldsymbol{x} = \sum_{j=1}^m \boldsymbol{f}^j, \ \boldsymbol{x} \leq \boldsymbol{u}, \\ & \qquad\qquad\qquad\qquad x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E. \end{aligned} \tag{2.5}$$

Facility location

Given a set of n facilities and m customers, the facility location problem consists of constructing facilities $i \in [n]$ at cost c_i to satisfy demand at minimal cost, i.e.,

minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{X} \in \mathbb{R}_+^{n \times m}} \; & \boldsymbol{c}^\top \boldsymbol{z} + \sum_{j=1}^{m} \sum_{i=1}^{n} C_{ij} X_{ij} \\ \text{s.t.} \; & \sum_{j=1}^{m} X_{ij} \leq U_i \; \forall i \in [n], \\ & \sum_{i=1}^{n} X_{ij} = d_j \; \forall j \in [m], \\ & X_{ij} = 0 \;\; \text{if} \;\; z_i = 0 \; \forall i \in [n], j \in [m]. \end{aligned} \tag{2.6}$$

In this formulation, X_{ij} corresponds to the quantity produced in facility i and shipped to customer j at a marginal cost of C_{ij}. Each facility i has a maximum output capacity of U_i and each customer j has a demand of d_j. In the uncapacitated case where $U_i = \infty$, the inner minimization problems decouple into independent knapsack problems for each customer j.

Sparse portfolio selection

Given an expected marginal return vector $\boldsymbol{\mu} \in \mathbb{R}^n$, estimated covariance matrix $\boldsymbol{\Sigma} \in \mathcal{S}_+^n$, uncertainty budget parameter $\sigma > 0$, cardinality budget parameter $k \in \{2, \ldots, n-1\}$, linear constraint matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, and right-hand-side bounds $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^m$, investors determine an optimal allocation of capital between assets by minimizing over $\mathcal{Z} = \left\{\boldsymbol{z} \in \{0,1\}^n : \; \boldsymbol{e}^\top \boldsymbol{z} \leq k\right\}$ the function

$$\begin{aligned} f(\boldsymbol{z}) = \min_{\boldsymbol{x} \in \mathbb{R}_+^n} \; & \frac{\sigma}{2} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{x} \\ \text{s.t.} \; & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}, \; \boldsymbol{e}^\top \boldsymbol{x} = 1, \; x_i = 0 \text{ if } \; z_i = 0 \; \forall i \in [n]. \end{aligned} \tag{2.7}$$

We apply the algorithms derived in this chapter to sparse empirical risk minimization problems in Chapter 3.

Sparse empirical risk minimization

Given a matrix of covariates $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and a response vector $\boldsymbol{y} \in \mathbb{R}^n$, sparse empirical risk minimization seeks a vector $\boldsymbol{w}$ which explains the response in a compelling manner, i.e., minimizes over $\mathcal{Z} = \{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k\}$ the function:

$$f(\boldsymbol{z}) := \min_{\boldsymbol{w} \in \mathbb{R}^p} \quad \sum_{i=1}^{n} \ell\left(y_i, \boldsymbol{w}^\top \boldsymbol{x}_i\right) \quad \text{s.t.} \quad w_j = 0 \text{ if } z_j = 0 \quad \forall j \in [p], \tag{2.8}$$

where ℓ is an appropriate convex loss function; we provide examples of suitable loss functions in Table 2.1.

Table 2.1: Loss functions and Fenchel conjugates for ERM problems.

Method	Loss function	Domain	Fenchel conjugate
OLS	$\frac{1}{2}(y-u)^2$	$y \in \mathbb{R}$	$\ell^\star(y,\alpha) = \frac{1}{2}\alpha^2 + \alpha y$
SVM	$\max(1-yu, 0)$	$y \in \{\pm 1\}$	$\ell^\star(y,\alpha) = \begin{cases} \alpha y, & \text{if } \alpha y \in [-1,0], \\ \infty, & \text{otherwise.} \end{cases}$

Sparse principal component analysis (PCA)

Given a covariance matrix $\boldsymbol{\Sigma} \in S_+^p$, the sparse PCA problem is to select a vector $\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k$ which maximizes over $\{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k\}$ the function:

$$f(\boldsymbol{z}) = \max_{\boldsymbol{x} \in \mathbb{R}^p} \quad \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} \text{ s.t. } \|\boldsymbol{x}\|_2^2 = 1, x_i = 0 \text{ if } z_i = 0 \; \forall i \in [p]. \tag{2.9}$$

This function is non-concave in $\boldsymbol{z}$, because $f(\boldsymbol{z})$ is the optimal value of a non-convex quadratic optimization problem. Fortuitously however, the leading eigenvalue of a positive semidefinite matrix $\boldsymbol{\Sigma}$ can be expressed as the optimal value of the following semidefinite problem:

$$\lambda_{\max}(\boldsymbol{\Sigma}) = \max_{\boldsymbol{X} \in \mathcal{S}_+^n} \quad \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1, \tag{2.10}$$

which implies that, since $f(\boldsymbol{z})$ is simply the leading eigenvalue of a submatrix of $\boldsymbol{\Sigma}$ induced by $\boldsymbol{z}$, Problem (2.9) admits an exact mixed-integer semidefinite reformulation:

$$f(\boldsymbol{z}) = \max_{\boldsymbol{X} \in S_+^p} \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1, X_{i,j} = 0 \text{ if } z_i = 0 \ \forall i, j \in [p]. \tag{2.11}$$

We apply the algorithms derived in this chapter to sparse principal component analysis problems in Chapter 4.

Unit commitment

In the DC-load-flow unit commitment problem, each generation unit i incurs a cost given by a quadratic cost function $f^i(x) = a_i x^2 + b_i x + c_i$ for its power generation output $x \in [0, u_i]$. Let $\mathcal{T}$ denote a finite set of time periods covering a time horizon (e.g., 24 hours). At each time period $t \in \mathcal{T}$, there is an estimated demand d_t. The objective is to generate sufficient power to satisfy demand at minimum cost, while respecting minimum time on/time off constraints.

By introducing binary variables $z_{i,t}$, which denote whether generation unit i is active in time period t, requiring that $\boldsymbol{z} \in \mathcal{Z}$, i.e., $\boldsymbol{z}$ obeys physical constraints such as minimum time on/off, the unit commitment problem admits the formulation:

$$\min_{\boldsymbol{z}} \quad f(\boldsymbol{z}) + \sum_{t \in \mathcal{T}} \sum_{i=1}^{n} c_i z_{i,t} \quad \text{s.t.} \quad \boldsymbol{z} \in \mathcal{Z} \subseteq \{0,1\}^{n \times |\mathcal{T}|},$$

$$\begin{aligned} \text{where:} \quad f(\boldsymbol{z}) = \min_{\boldsymbol{x}} \quad & \sum_{t \in \mathcal{T}} \left(\sum_{i=1}^{n} \tfrac{1}{2} a_i x_{i,t}^2 + b_i x_{i,t} \right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{i,t} \geq D_t \quad \forall t \in \mathcal{T}, \\ & x_{i,t} \in [0, u_{i,t}] \quad \forall i \in [n], \forall t \in \mathcal{T}, \\ & x_{i,t} = 0 \text{ if } z_{i,t} = 0 \quad \forall i \in [n], \forall t \in \mathcal{T}. \end{aligned} \tag{2.12}$$

Binary quadratic optimization

Given a symmetric cost matrix $\boldsymbol{Q}$, the binary quadratic optimization problem consists of selecting a vector of binary variables $\boldsymbol{z}$ which minimizes over $\mathcal{Z} = \{0,1\}^n$:

$$f(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{Q} \boldsymbol{z}. \tag{2.13}$$

This formulation is non-convex and does not include continuous variables. However, introducing auxiliary continuous variables yields the equivalent formulation [109] of minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{Y} \in \mathbb{R}^{n \times n}_+} \quad \langle \boldsymbol{Q}, \boldsymbol{Y} \rangle \quad \text{s.t.} \quad & Y_{i,j} \leq 1 && \forall i,j \in [n], \\ & Y_{i,j} \geq z_i + z_j - 1 && \forall i \in [n], \forall j \in [n] \backslash \{i\}, \\ & Y_{i,i} \geq z_i && \forall i \in [n], \\ & Y_{i,j} = 0 \text{ if } z_i = 0 && \forall i,j \in [n], \\ & Y_{i,j} = 0 \text{ if } z_j = 0 && \forall i,j \in [n]. \end{aligned}$$

Union of ellipsoidal constraints

We now demonstrate that an even broader class of problems than MIOs with logical constraints can be cast within our framework. Concretely, we demonstrate that constraints $\boldsymbol{x} \in \mathcal{S} := \bigcup_{i=1}^k (Q_i \cap P_i)$, where $Q_i := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^\top \boldsymbol{Q}_i \boldsymbol{x} + \boldsymbol{h}_i^\top \boldsymbol{x} + g_i \leq 0\}$, with $\boldsymbol{Q}_i \succeq \boldsymbol{0}$, is an ellipsoid and $P_i := \{\boldsymbol{x} : \boldsymbol{A}_i \boldsymbol{x} \leq \boldsymbol{b}_i\}$ is a polytope, can be reformulated as a special case of our framework. We remark that the constraint $\boldsymbol{x} \in \mathcal{S}$ is very general. Indeed, if we were to omit the quadratic constraints then we obtain a so-called ideal union of polyhedra formulation, which essentially all mixed-binary linear feasible regions admit [see 217].

To derive a mixed-integer formulation with logical constraints of $\mathcal{S}$ that fits within our framework, we introduce $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\delta_i \in \{0,1\}^n$, such that $\boldsymbol{x}_i \in Q_i \cap P_i$ if $\delta_i = 1$, $\boldsymbol{x}_i = \boldsymbol{0}$ otherwise, and $\boldsymbol{x} = \sum_i \boldsymbol{x}_i$. We enforce $\boldsymbol{x}_i \in Q_i \cap P_i$ by introducing slack variables $\boldsymbol{\xi_i}$, ρ_i for the linear and quadratic constraints respectively, and forcing

them to be zero whenever $\delta_i = 1$. Formally, $\mathcal{S}$ admits the following formulation

$$\begin{aligned} \boldsymbol{x} = \sum_{i=1}^{k} \boldsymbol{x}_i,\ \sum_{i=1}^{k} \delta_i = 1, & \qquad (2.14) \\ \boldsymbol{A}_i \boldsymbol{x}_i \leq \boldsymbol{b}_i + \boldsymbol{\xi}_i \quad \forall i \in [k], & \\ \boldsymbol{x}_i^\top \boldsymbol{Q}_i \boldsymbol{x}_i + \boldsymbol{h}_i^\top \boldsymbol{x}_i + g_i \leq \rho_i \quad \forall i \in [k], & \\ \boldsymbol{x}_i = \boldsymbol{0} \text{ if } \delta_i = 0 \quad \forall i \in [k], & \\ \boldsymbol{\xi}_i = 0 \text{ if } (1 - \delta_i) = 0 \quad \forall i \in [k], & \\ \rho_i = 0 \text{ if } (1 - \delta_i) = 0 \quad \forall i \in [k]. & \end{aligned}$$

A Regularization Assumption

When we stated Problem (2.1), we assumed that its objective function consists of a convex function $g(\boldsymbol{x})$ plus a regularization term $\Omega(\boldsymbol{x})$. We now formalize this:

Assumption 2.1. *In Problem* (2.1)*, the regularization term* $\Omega(\boldsymbol{x})$ *is one of:*

- *a big-M penalty function,* $\Omega(\boldsymbol{x}) = 0$ *if* $\|\boldsymbol{x}\|_\infty \leq M$ *and* ∞ *otherwise,*
- *a ridge penalty,* $\Omega(\boldsymbol{x}) = \dfrac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

This decomposition often constitutes a modeling choice in itself. We now illustrate this idea via the network design example.

Example 2.2. *In the network design example* (2.4)*, given the flow conservation structure* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$*, we have that* $\boldsymbol{x} \leq M\boldsymbol{e}$*, where* $M = \sum_{i:b_i>0} b_i$*. In addition, if* $\boldsymbol{Q} \succ \boldsymbol{0}$ *then the objective function naturally contains a ridge regularization term with* $1/\gamma$ *equal to the smallest eigenvalue of* $\boldsymbol{Q}$*. Moreover, it is possible to obtain a tighter natural ridge regularization term by solving the following auxiliary semidefinite optimization problem apriori*

$$\max_{\boldsymbol{q} \geq \boldsymbol{0}} \ \boldsymbol{e}^\top \boldsymbol{q} \quad s.t. \quad \boldsymbol{Q} - \mathrm{Diag}(\boldsymbol{q}) \succeq \boldsymbol{0},$$

and using q_i *as the ridge regularizer for each index* i *[112].*

Big-M constraints are often considered to be a modeling trick. However, our framework demonstrates that imposing either big-M constraints or a ridge penalty is a regularization method, rather than a modeling trick. Interestingly, ridge regularization accounts for the relationship between the binary and continuous variables just as well as big-M regularization, without performing an algebraic reformulation of the logical constraints.

Conceptually, both regularization functions are equivalent to a soft or hard constraint on the continuous variables $\boldsymbol{x}$. However, they admit practical differences: For big-M regularization, there usually exists a finite value M_0, typically unknown a priori, such that if $M < M_0$, the regularized problem is infeasible. Alternatively, for every value of the ridge regularization parameter γ, if the original problem is feasible then the regularized problem is also feasible. Consequently, if there is no natural choice of M then imposing ridge regularization may be less restrictive than imposing big-M regularization. However, for any $\gamma > 0$, the objective of the optimization problem with ridge regularization is different from its unregularized limit as $\gamma \to \infty$, while for big-M regularization, there usually exists a finite value M_1 above which the two objective values match.

2.3 Duality to the Rescue

In this section, we derive Problem (2.3)'s dual and reformulate $f(\boldsymbol{z})$ as a maximization problem. This reformulation is significant for two reasons: First, as shown in the proof of Theorem 2.1, it leverages a non-linear reformulation of the logical constraints "$x_i = 0$ if $z_i = 0$" by introducing additional variables v_i such that $v_i = z_i x_i$. Second, it proves that the regularization term $\Omega(\boldsymbol{x})$ drives the convexity and smoothness of $f(\boldsymbol{z})$, and thereby drives the computational tractability of the problem. To derive Problem (2.3)'s dual, we require:

Assumption 2.2. *For each subproblem generated by $f(\boldsymbol{z})$, where $\boldsymbol{z} \in \mathcal{Z}$, either the optimization problem is infeasible, or strong duality holds.*

Note that all seven examples stated in this chapter satisfy Assumption 2.2, as their

inner problems are either convex quadratics with linear constraints or linear semidefinite problems which satisfy Slater's condition [54, Section 5.2.3]. More generally, the assumption may fail to hold if the inner problem is feasible but fails to satisfy a constraint qualification which guarantees strong duality (e.g., Slater's condition [54, Section 5.2.3], which requires that the problem has non-empty relative interior). A classic example of this [54, Exercise 5.21] is the optimization problem

$$\min_{x\in\mathbb{R},w\in\mathbb{R}^+} \exp(-x) \text{ s.t. } \frac{x^2}{w} \leq 0$$

with optimal objective 1, which has the dual problem $\max_{\lambda\geq 0} 0$ s.t.$\lambda \geq 0$ with optimal objective 0. Augmenting this problem with a logical constraint results in an (artificial) logically-constrained problem which violates Assumption 2.2.

Noting however that constraint qualification failures are usually artificial and indicate modeling errors rather than real phenomena [see 177, Section 8.4, for a discussion], let us suppose that Assumption 2.2 holds. Then, the following theorem reformulates Problem (2.2) as a saddle-point problem:

Theorem 2.1. *Under Assumption 2.2, Problem* (2.2) *is equivalent to:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^{n} z_i\, \Omega^\star(\alpha_i), \tag{2.15}$$

where $h(\boldsymbol{\alpha}) := \inf_{\boldsymbol{v}}\ g(\boldsymbol{v}) - \boldsymbol{v}^\top \boldsymbol{\alpha}$ *is, up to a sign, the Fenchel conjugate of* g*, and*

$$\Omega^\star(\beta) := M|\beta| \quad \textit{for the big-M penalty,}$$
$$\Omega^\star(\beta) := \tfrac{\gamma}{2}\beta^2 \qquad \textit{for the ridge penalty.}$$

Proof. Let us fix some $\boldsymbol{z} \in \{0,1\}^n$, and suppose that strong duality holds for the inner minimization problem which defines $f(\boldsymbol{z})$. Then, after introducing additional variables $\boldsymbol{v} \in \mathbb{R}^n$ such that $v_i = z_i x_i$, we have

$$f(\boldsymbol{z}) = \min_{\boldsymbol{x},\boldsymbol{v}}\ g(\boldsymbol{v}) + \Omega(\boldsymbol{x}) \quad \text{ s.t. } \boldsymbol{v} = \mathrm{Diag}(\boldsymbol{z})\boldsymbol{x}.$$

Let $\boldsymbol{\alpha}$ denote the dual variables associated with the coupling constraint $\boldsymbol{v} = \text{Diag}(\boldsymbol{z})\boldsymbol{x}$. The minimization problem is then equivalent to its dual problem, which is given by:

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha}} \; h(\boldsymbol{\alpha}) + \min_{\boldsymbol{x}} \left[\Omega(\boldsymbol{x}) + \boldsymbol{\alpha}^\top \text{Diag}(\boldsymbol{z})\boldsymbol{x}\right],$$

Since $\Omega(\cdot)$ is decomposable, i.e., $\Omega(\boldsymbol{x}) = \sum_i \Omega_i(x_i)$, we obtain:

$$\begin{aligned}\min_{\boldsymbol{x}} \left[\Omega(\boldsymbol{x}) + \boldsymbol{\alpha}^\top \text{Diag}(\boldsymbol{z})\boldsymbol{x}\right] &= \sum_{i=1}^{n} \min_{x_i} \left[\Omega_i(x_i) + z_i x_i \alpha_i\right] \\ &= \sum_{i=1}^{n} -\Omega^\star(-z_i\alpha_i) = -\sum_{i=1}^{n} z_i \Omega^\star(\alpha_i),\end{aligned}$$

where the last equality holds as $z_i > 0$ for the big-M and $z_i^2 = z_i$ for the ridge penalty.

Alternatively, if the inner minimization problem defining $f(\boldsymbol{z})$ is infeasible, then its dual problem is unbounded by weak duality[1]. □

Remark 2. *Without regularization, i.e., $\Omega(\boldsymbol{x}) = 0$, a similar proof shows that Problem* (2.2) *admits an interesting saddle-point formulation:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) \; \textit{s.t.} \; \alpha_i = 0, \; \textit{if } z_i = 1 \quad \forall i \in [n],$$

since $\Omega^\star(\alpha) = \min_x [x\alpha - \Omega(x)] = 0$ if $\alpha = 0$, and $+\infty$ otherwise. Consequently, the regularized formulation can be regarded as a relaxation of the original problem where the hard constraint $\alpha_i = 0$ if $z_i = 1$ is replaced with a soft penalty term $-z_i\Omega^\star(\alpha_i)$.

Remark 3. *The proof of Theorem 2.1 exploits three attributes of the regularizer $\Omega(\boldsymbol{x})$. Namely, (1) decomposability, i.e., $\Omega(\boldsymbol{x}) = \sum_i \Omega_i(x_i)$, for appropriate scalar functions Ω_i, (2) the convexity of $\Omega(\boldsymbol{x})$ in $\boldsymbol{x}$, and (3) the fact that $\Omega(\cdot)$ regularizes towards 0, i.e., $\mathbf{0} \in \arg\min_{\boldsymbol{x}} \Omega(\boldsymbol{x})$. However, the proof does not explicitly require that $\Omega(\boldsymbol{x})$ is either a big-M or a ridge regularizer. This suggests that our framework could be extended to other regularization functions.*

[1]Weak duality implies that the dual problem is either unfeasible or unbounded. Since the feasible set of the maximization problem does not depend on $\boldsymbol{z}$, it is always feasible, unless the original problem (2.1) is itself infeasible. Therefore, we assume without loss of generality that it is unbounded.

Example 2.3. *For the network design problem* (2.4)*, we have*

$$\begin{aligned} h(\boldsymbol{\alpha}) &= \min_{\boldsymbol{x}\geq \mathbf{0}:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{d}-\boldsymbol{\alpha})^\top \boldsymbol{x}, \\ &= \max_{\boldsymbol{\beta}_0\geq \mathbf{0},\boldsymbol{p}} \quad \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right)^\top \boldsymbol{Q}^{-1}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right). \end{aligned}$$

Introducing $\boldsymbol{\xi} = \boldsymbol{Q}^{-1/2}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right)$*, we can further write*

$$h(\boldsymbol{\alpha}) = \max_{\boldsymbol{\xi},\boldsymbol{p}} \quad \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\|\boldsymbol{\xi}\|_2^2 \;\; s.t \;\; \boldsymbol{Q}^{1/2}\boldsymbol{\xi} \geq \boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha}.$$

Hence, Problem (2.4) *is equivalent to minimizing over* $\boldsymbol{z} \in \mathcal{Z}$ *the function*

$$\begin{aligned} \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}) = \max_{\alpha,\boldsymbol{\xi},\boldsymbol{p}} \quad & \boldsymbol{c}^\top \boldsymbol{z} + \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\|\boldsymbol{\xi}\|_2^2 - \sum_{j=1}^{n} z_j\, \Omega^\star(\alpha_j) \\ s.t \quad & \boldsymbol{Q}^{1/2}\boldsymbol{\xi} \geq \boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha}. \end{aligned}$$

Theorem 2.1 reformulates $f(\boldsymbol{z})$ as an inner maximization problem, namely

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad h(\boldsymbol{\alpha}) - \sum_{i=1}^{n} z_i\, \Omega^\star(\alpha_i), \tag{2.16}$$

for any feasible binary $\boldsymbol{z} \in \mathcal{Z}$. The regularization term Ω will be instrumental in our numerical strategy for it directly controls both the convexity and smoothness of f. Note that (2.16) extends the definition of $f(\boldsymbol{z})$ to the convex set $\mathrm{Bool}(\mathcal{Z})$, obtained by relaxing the constraints $\boldsymbol{z} \in \{0,1\}^p$ to $\boldsymbol{z} \in [0,1]^p$ in the definition of $\mathcal{Z}$.

Convexity

$f(\boldsymbol{z})$ is convex in $\boldsymbol{z}$ as a point-wise maximum of linear function of $\boldsymbol{z}$. In addition, denoting $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ a solution of (2.16), we have the lower-approximation:

$$f(\tilde{\boldsymbol{z}}) \geq f(\boldsymbol{z}) + \nabla f(\boldsymbol{z})^\top (\tilde{\boldsymbol{z}} - \boldsymbol{z}) \quad \forall \tilde{\boldsymbol{z}} \in \mathcal{Z}, \tag{2.17}$$

where $[\nabla f(\boldsymbol{z})]_i := -\Omega^\star(\alpha^\star(\boldsymbol{z})_i)$ is a sub-gradient of f at $\boldsymbol{z}$.

We remark that if the maximization problem in $\boldsymbol{\alpha}$ defined by $f(\boldsymbol{z})$ admits multiple optimal solutions then the corresponding lower-approximation of f at $\boldsymbol{z}$ may not be unique. This behavior can severely hinder the convergence of cutting-plane schemes such as Benders' decomposition. Since the work of [168] on Pareto optimal cuts, many strategies have been proposed to improve the cut selection process in the presence of degeneracy [see 105, Section 4.4 for a review]. However, the use of ridge regularization ensures that the objective function in (2.15) is strongly concave in α_i such that $z_i > 0$, and therefore guarantees that there is a unique optimal choice of $\alpha_i^\star(\boldsymbol{z})$. In other words, ridge regularization naturally inhibits degeneracy.

Smoothness

$f(\boldsymbol{z})$ is smooth, in the sense of Lipschitz continuity, which is a crucial property for deriving bounds on the integrality gap of the Boolean relaxation, and designing local search heuristics in Section 2.4. The following proposition follows from Theorem 2.1:

Proposition 2.1. *For any* $\boldsymbol{z}, \boldsymbol{z}' \in \text{Bool}\,(\mathcal{Z})$,

(a) With big-M regularization, $f(\boldsymbol{z}') - f(\boldsymbol{z}) \leq M \sum_{i=1}^{n} (z_i - z_i')|\alpha^\star(\boldsymbol{z}')_i|$.

(b) With ridge regularization, $f(\boldsymbol{z}') - f(\boldsymbol{z}) \leq \frac{\gamma}{2} \sum_{i=1}^{n} (z_i - z_i')\alpha^\star(\boldsymbol{z}')_i^2$.

Proof. By Equation (2.15),

$$\begin{aligned} f(\boldsymbol{z}') - f(\boldsymbol{z}) &= \max_{\boldsymbol{\alpha}' \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}') - \sum_{i=1}^{n} z_i' \Omega^\star(\alpha_i') \right) - \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}) - \sum_{i=1}^{n} z_i \Omega^\star(\alpha_i) \right), \\ &= h(\boldsymbol{\alpha}^\star(\boldsymbol{z}')) - \sum_{i=1}^{n} z_i' \Omega^\star(\alpha^\star(\boldsymbol{z}')_i) - h(\boldsymbol{\alpha}^\star(\boldsymbol{z}')) + \sum_{i=1}^{n} z_i \Omega^\star(\alpha^\star(\boldsymbol{z}')_i), \\ &\leq \sum_{i=1}^{n} (z_i - z_i') \Omega^\star(\alpha^\star(\boldsymbol{z}')_i), \end{aligned}$$

where the inequality holds as an optimal $\boldsymbol{\alpha}'$ is a feasible choice of $\boldsymbol{\alpha}$. □

Proposition 2.1 demonstrates that, when the coordinates of $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ are uniformly

bounded[2] with respect to $\boldsymbol{z}$, $f(\boldsymbol{z})$ is Lipschitz-continuous, with a constant proportional to M (resp. γ) in the big-M (resp. ridge) case.

Theoretical Merits of Ridge, Big-M Regularization

In this section, we proposed a framework to reformulate MINLOs with logical constraints, which comprises regularizing MINLOs via either the widely used big-M modeling paradigm or the less popular ridge regularization paradigm. We summarize the advantages and disadvantages of each regularizer in Table 2.2. However, note that we have not yet established how these characteristics impact the numerical tractability and quality of the returned solution; this is the topic of the following two sections.

Table 2.2: Summary of advantages (+) /disadvantages (−) of both techniques.

Regularization	Characteristics
Big-M	(+) Linear constraints (+) Supplies same objective if $M > M_1$, for some $M_1 < \infty$ (−) Leads to infeasiblity if $M < M_0$, for some $M_0 < \infty$
Ridge	(+) Strongly convex objective (−) Leads to a different objective for any $\gamma > 0$ (+) Preserves the feasible set

2.4 An Efficient Numerical Approach

We now present an efficient numerical approach to solve Problem (2.15). The backbone is a cutting-plane strategy, embedded within a branch-and-bound procedure to solve the problem exactly. We also propose local search and rounding heuristics to find good feasible solutions, and use information from the Boolean relaxation to improve the duality gap.

[2]Such a uniform bound always exists, as $f(\boldsymbol{z})$ is only supported on a finite number of binaries.

Overall Cutting-Plane Scheme

Theorem 2.1 reformulates the function $f(\boldsymbol{z})$ as an inner maximization problem, and demonstrates that $f(\boldsymbol{z})$ is convex in $\boldsymbol{z}$, meaning a linear lower approximation provides a valid underestimator of $f(\boldsymbol{z})$, as outlined in Equation (2.17). Consequently, a valid numerical strategy for minimizing $f(\boldsymbol{z})$ is to iteratively minimize a piecewise linear lower-approximation of f and refining this approximation at each step until some approximation error ε is reached, as described in Algorithm 2.1. Note that this scheme converges in a finite, yet exponential in the worst case, number of iterations, because there are finitely many binary solutions.

Algorithm 2.1 Cutting-plane scheme

Require: Initial solution $\boldsymbol{z}^1$
  $t \leftarrow 1$
  repeat
    Compute $\boldsymbol{z}^{t+1}, \eta^{t+1}$ solution of

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \eta} \boldsymbol{c}^\top \boldsymbol{z} + \eta \quad \text{s.t. } \forall s \in \{1, \ldots, t\},\ \eta \geq f(\boldsymbol{z}^s) + \nabla f(\boldsymbol{z}^s)^\top (\boldsymbol{z} - \boldsymbol{z}^s)$$

    Compute $f(\boldsymbol{z}^{t+1})$ and $\nabla f(\boldsymbol{z}^{t+1})$
    $t \leftarrow t + 1$
  until $f(\boldsymbol{z}^{t+1}) - \eta^{t+1} \leq \varepsilon$ **return** $\boldsymbol{z}^t$

As suggested in the pseudocode, this strategy can be integrated within a single branch-and-bound procedure using lazy callbacks to avoid solving a mixed-integer linear optimization problem at each iteration. Lazy callbacks are now standard tools in commercial solvers such as Gurobi and CPLEX and provide significant speed-ups for cutting-plane algorithms. With this, the commercial solver constructs a single branch-and-bound tree and generates a new cut at a feasible solution $\boldsymbol{z}$.

We remark that the second-stage minimization problem may be infeasible at some $\boldsymbol{z}^t$. In this case, we generate a feasibility cut. In particular, the constraint $\sum_i z_i^t(1 - z_i) + \sum_i (1 - z_i^t) z_i \geq 1$ excludes the iterate $\boldsymbol{z}^t$ from the feasible set. Stronger feasibility cuts can be obtained by leveraging problem specific structure. For instance, when the feasible set satisfies $\boldsymbol{z}^t \notin \mathcal{Z} \implies \forall \boldsymbol{z} \leq \boldsymbol{z}^t,\ \boldsymbol{z} \notin \mathcal{Z}$, $\sum_i (1 - z_i^t) z_i \geq 1$ is a valid feasibility cut. Alternatively, one can invoke conic duality if $g(\boldsymbol{x})$ generates a conic

feasibility problem. Formally, assume

$$g(\boldsymbol{x}) = \begin{cases} \boldsymbol{c}^\top \boldsymbol{x}, & \text{if } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathcal{K}, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{K}$ is a closed convex cone. Assuming that $g(\boldsymbol{x})$ is of the prescribed form, we have the dual conjugate

$$h(\boldsymbol{\alpha}) = \inf_{\boldsymbol{x}}\ \boldsymbol{x}^\top \boldsymbol{\alpha} - g(\boldsymbol{x}) = \max_{\boldsymbol{\pi}}\ \boldsymbol{b}^\top \boldsymbol{\pi} + \begin{cases} 0, & \text{if } \boldsymbol{c} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\pi} \in \mathcal{K}^\star, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{K}^\star$ is the dual cone to $\mathcal{K}$. In this case, if some binary vector $\boldsymbol{z}$ gives rise to an infeasible subproblem, i.e., $f(\boldsymbol{z}) = +\infty$, then the conic duality theorem implies[3] that there is a *certificate* of infeasibility $(\boldsymbol{\alpha}, \boldsymbol{\pi})$ such that

$$\boldsymbol{c} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\pi} \in \mathcal{K}^\star, \boldsymbol{b}^\top \boldsymbol{\pi} > \sum_{i=1}^{n} z_i \Omega^\star(\alpha_i). \tag{2.18}$$

Therefore, to restore feasibility, we simply impose the following cut:

$$\boldsymbol{b}^\top \boldsymbol{\pi} \leq \sum_{i=1}^{n} z_i \Omega^\star(\alpha_i). \tag{2.19}$$

We now provide some guidelines for accelerating the convergence of Algorithm 2.1:

1. *Fast cut generation strategy:* To generate a cut, one solves the second-stage minimization problem (2.3) (or its dual) in $\boldsymbol{x}$, which contains no discrete variables and is usually orders of magnitude faster to solve than the original mixed-

[3] We should note that this statement is, strictly speaking, not true unless we impose regularization. Indeed, the full conic duality theorem [17, Theorem 2.4.1] allows for the possibility that a problem is infeasible but asymptotically feasible, i.e.,

$$\nexists \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathcal{K} \text{ but } \exists \{\boldsymbol{x}_t\}_{t=1}^{\infty} : \boldsymbol{x}_t \in \mathcal{K}\ \forall t \text{ with } \|\boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b}\| \to 0.$$

Fortunately, the regularizer $\Omega(\boldsymbol{x})$ alleviates this issue, because it is coercive (i.e., "blows up" to $+\infty$ as $\|\boldsymbol{x}\| \to \infty$) and therefore renders all unbounded solutions infeasible and ensures the compactness of the level sets of $g(\boldsymbol{x}) + \Omega(\boldsymbol{x})$.

integer problem (2.1). Moreover, the minimization problem in $\boldsymbol{x}$ needs to be solved only for the coordinates x_i such that $z_i = 1$. In practice, this approach yields a sequence of subproblems of much smaller size than the original problem, especially if $\mathcal{Z}$ contains a cardinality constraint. For this reason, we recommend generating cuts at binary $\boldsymbol{z}$'s, which are often sparser than continuous $\boldsymbol{z}$'s. This recommendation can be relaxed when the separation problem can be solved efficiently, even for dense $\boldsymbol{z}$'s; for instance, in uncapacitated facility location problems, each subproblem is a knapsack problem that can be solved by sorting [106]. If possible, we recommend theoretically analyzing the sparsity of the optimal solution a priori, to derive an explicit cardinality or budget constraint on $\boldsymbol{z}$, and ensure the sparsity of each incumbent solution.

2. *Cut selection rule in the presence of degeneracy:* In the presence of degeneracy, selection criteria, such as Pareto optimality [168], have been proposed to accelerate convergence. However, these criteria are numerous, computationally expensive, and all in all can do more harm than good [188]. In an opposite direction, we recommend alleviating the burden of degeneracy by design by imposing a ridge regularizer whenever degeneracy hinders convergence.

3. *Rich root node analysis:* As suggested in [105], providing the solver with as much information as possible at the root node can drastically improve convergence of cutting-plane methods. This is the topic of the next two sections.

These ingredients, and especially the ability to generate cuts efficiently, dictate which problems could benefit the most from our approach and which regularizer to use. Problems with an explicit cardinality constraint, for instance, would require a small subproblem to be solved at each iteration. For network design problems, the network flow structure of the feasible set is a key numerical asset, so we intuit that ridge regularization, which leaves the feasible set unchanged, would be very efficient. On the other hand, for uncapacitated facility location, sub-problems with big-M regularization boils down to a knapsack problem and can be solved efficiently via sorting, as discussed in [106, Section 3.1].

2.5 Improving the Lower-Bound: A Relaxation

To certify optimality, high-quality lower bounds are of interest and can be obtained by relaxing the integrality constraint $\boldsymbol{z} \in \{0,1\}^n$ in the definition of $\mathcal{Z}$ to $\boldsymbol{z} \in [0,1]^n$. In this case, the Boolean relaxation of (2.2) is:

$$\min_{\boldsymbol{z} \in \text{Bool}(\mathcal{Z})} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}),$$

which can be solved via Kelley's method [146], a continuous analog of Algorithm 2.1.

Alternatively, the continuous minimization problem admits a reformulation

$$\min_{\boldsymbol{z} \in \text{Bool}(\mathcal{Z})} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^{n} z_i \, \Omega^\star(\alpha_j). \tag{2.20}$$

analogous to Problem (2.15). Under Assumption 2.2, we can further write the min-max relaxation formulation (2.20) as a non-smooth maximization problem

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} q(\boldsymbol{\alpha}), \quad \text{with} \quad q(\boldsymbol{\alpha}) := h(\boldsymbol{\alpha}) + \min_{\boldsymbol{z} \in \text{Bool}(\mathcal{Z})} \sum_{i=1}^{n} \left(c_i - \Omega^\star(\alpha_i)\right) z_i$$

and apply a projected sub-gradient ascent method.

The benefit from solving the Boolean relaxation with these algorithms is threefold. First, it provides a lower bound on the objective value of the discrete optimization problem (2.2). Second, it generates valid linear lower approximations of $f(\boldsymbol{z})$ to initiate the cutting-plane algorithm with. Finally, it supplies a sequence of continuous solutions that can be rounded and polished to obtain good binary solutions. Indeed, the Lipschitz continuity of $f(\boldsymbol{z})$ suggests that high-quality feasible binary solutions can be found in the neighborhood of a solution to the Boolean relaxation. We formalize this observation in the following theorem; a formal proof is available in [33]:

Theorem 2.2. *Let $\boldsymbol{z}^\star$ denote a solution to the Boolean relaxation* (2.20)*, $\mathcal{R}$ denote the indices of $\boldsymbol{z}^\star$ with fractional entries, and $\alpha^\star(\boldsymbol{z})$ denote a best choice of $\boldsymbol{\alpha}$ for a given $\boldsymbol{z}$. Suppose that for any $\boldsymbol{z} \in \mathcal{Z}$, $|\alpha^\star(\boldsymbol{z})_j| \leq L$. Then, a random rounding $\boldsymbol{z}$ of*

$\boldsymbol{z}^\star$, i.e., $z_j \sim$ Bernoulli$(z_j^\star)$, satisfies $0 \leq f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq \epsilon$ with probability at least $p = 1 - |\mathcal{R}| \exp\left(-\frac{\epsilon^2}{\kappa}\right)$, where

$$\kappa := 2M^2L^2|\mathcal{R}|^2 \quad \textit{for the big-M penalty,}$$
$$\kappa := \tfrac{1}{2}\gamma^2L^4|\mathcal{R}|^2 \quad \textit{for the ridge penalty.}$$

This result calls for multiple remarks:

- For $\varepsilon > \sqrt{\kappa \ln(|\mathcal{R}|)}$, we have that $p > 0$, which implies the existence of a binary ε-optimal solution in the neighborhood of $\boldsymbol{z}^\star$, which in turn bounds the integrality gap by ε. As a result, lower values of M or γ typically make the discrete optimization problem easier.
- A solution to the Boolean relaxation often includes some binary coordinates, i.e., $|\mathcal{R}| < n$. In this situation, it is tempting to fix $z_i = z_i^\star$ for $i \notin \mathcal{R}$ and solve the master problem (2.2) over coordinates in $\mathcal{R}$. In general, this approach provides sub-optimal solutions. However, Theorem 2.2 quantifies the price of fixing variables and bounds the optimality gap by $\sqrt{\kappa \ln(|\mathcal{R}|)}$.
- In the above high-probability bound, we do not account for the feasibility of the randomly rounded solution $\boldsymbol{z}$. Accounting for $\boldsymbol{z}$'s feasibility marginally reduces the probability given above, as shown for general discrete problems by [196].
- Rather than performing random rounding, one could also perform greedy rounding, i.e., round the k largest $z_i^\star$'s to 1 under a cardinality constraint, or otherwise round all $z_i^\star$'s above some threshold to 1. By the probabilistic method, greedy rounding yields solutions which are roughly as suboptimal as random rounding. However, it is deterministic, and therefore may be preferable in instances where evaluating the objective is expensive.

Proof. We only detail the proof for the big-M regularization case, as the ridge regularization case follows *mutatis mutandis*. From Proposition 2.1,

$$0 \leq f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq ML|\mathcal{R}| \max_{\boldsymbol{\alpha} \geq \mathbf{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i.$$

The polyhedron $\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \geq \mathbf{0}, \|\boldsymbol{\alpha}\|_1 \leq 1\}$ admits $|\mathcal{R}| + 1$ extreme points. However, if

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > t,$$

for some $t > 0$, then the maximum can only occur at some $\alpha > \mathbf{0}$ so that we can restrict our attention to the $|\mathcal{R}|$ positive extreme points. Applying tail bounds on the maximum of sub-Gaussian random variables over a polytope [see 203, Theorem 1.16], since $\|\boldsymbol{\alpha}\|_2 \leq \|\boldsymbol{\alpha}\|_1 \leq 1$, we have for any $t > 0$,

$$\mathbb{P}\left(\max_{\boldsymbol{\alpha} \geq \mathbf{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > t\right) \leq |\mathcal{R}| \exp\left(-\frac{t^2}{2}\right),$$

so that

$$\mathbb{P}\left(ML|\mathcal{R}| \max_{\boldsymbol{\alpha} \geq \mathbf{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > \varepsilon\right) \leq |\mathcal{R}| \exp\left(-\frac{\varepsilon^2}{2M^2L^2|\mathcal{R}|^2}\right).$$

□

Under specific problem structure, other strategies might be more efficient than Kelley's method or the subgradient algorithm. For instance, if $\text{Bool}(\mathcal{Z})$ is a polyhedron, then the inner minimization problem defining $q(\boldsymbol{\alpha})$ is a linear optimization problem that can be rewritten as a maximization problem by invoking strong duality.

2.6 Improving the Upper-Bound

To improve the quality of the upper-bound, i.e., the cost associated with the best feasible solution found so far, we implement two rounding and local-search strategies.

Our first strategy is a randomized rounding strategy, which is inspired by Theorem 2.2. Given $\boldsymbol{z}_0 \in \text{Bool}(\mathcal{Z})$, we generate randomly rounded vectors $\boldsymbol{z}$ by sampling $\boldsymbol{z}$ according to $z_i \sim \text{Bernoulli}(z_{0i})$ until $\boldsymbol{z} \in \mathcal{Z}$, which happens with high probability since $\mathbb{E}[\boldsymbol{z}] = \boldsymbol{z}_0$ satisfies all the constraints which describe $\mathcal{Z}$, besides integrality [196].

Our second strategy is a sequential rounding procedure, which is informed by the lower-approximation on $f(\boldsymbol{z})$, as laid out in Equation (2.17). Observing that the ith coordinate $\nabla f(\boldsymbol{z}_0)_i$ provides a first-order indication of how a change in z_i might impact the overall cost, we proceed in two steps. We first round down all coordinates such that $\nabla f(\boldsymbol{z}_0)_i(0 - \boldsymbol{z}_{0i}) < 0$. Once the linear approximation of f only suggests rounding up, we round all coordinates of $\boldsymbol{z}$ to 1 and iteratively bring some coordinates to 0 to restore feasibility.

If $\boldsymbol{z}_0$ is binary, we implement a comparable local search strategy. If $\boldsymbol{z}_{0i} = 0$, then switching the ith coordinate to one increases the cost by at least $\nabla f(\boldsymbol{z}_0)_i$. Alternatively, if $\boldsymbol{z}_{0i} = 1$, then switching it to zero increases the cost by at least $-\nabla f(\boldsymbol{z}_0)_i$. We therefore compute the one-coordinate change which provides the largest potential cost improvement. However, as we only have access to a lower approximation of f, we are not guaranteed to generate a cost-decreasing sequence. Therefore, we terminate the procedure as soon as it cycles. A second complication is that, due to the constraints defining $\mathcal{Z}$, the best change sometimes yields an infeasible $\boldsymbol{z}$. In practice, for simple constraints such as $\boldsymbol{\ell} \leq \boldsymbol{z} \leq \boldsymbol{u}$, we forbid switches which break feasibility; for cardinality constraints, we perform the best switch and then restore feasibility at minimal cost when necessary.

2.7 Relationship With Perspective Cuts

In this section, we connect the perspective cuts introduced by [110] with our framework and discuss the merits of both approaches, in theory and in practice. To the best of our knowledge, a connection between Boolean relaxations of the two approaches has only been made in the context of sparse regression, by [226]. That is, the general connection we make here between the discrete problems, as well as their respective cut generating procedures, is novel.

We first demonstrate that imposing the ridge regularization term $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ naturally leads to the perspective formulation of [110]:

Theorem 2.3. *Suppose that $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ and that Assumption 2.2 holds. Then,*

Problem (2.15) *is equivalent to the following optimization problem:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}}\ \min_{\boldsymbol{x}\in\mathbb{R}^n}\quad \boldsymbol{c}^\top\boldsymbol{z}+g(\boldsymbol{x})+\frac{1}{2\gamma}\sum_{i=1}^{n}\begin{cases}\dfrac{x_i^2}{z_i}, & \text{if } z_i>0,\\ 0, & \text{if } z_i=0 \text{ and } x_i=0,\\ \infty, & \text{otherwise.}\end{cases} \tag{2.21}$$

Proof. Let us fix $\boldsymbol{z}\in\mathcal{Z}$. Then, we have that:

$$\begin{aligned}\max_{\boldsymbol{\alpha}}\ h(\boldsymbol{\alpha})-\frac{\gamma}{2}\sum_{j=1}^{n}z_j\alpha_j^2 &= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ h(\boldsymbol{\alpha})-\frac{\gamma}{2}\sum_{j=1}^{n}z_j\,\beta_j^2 \text{ s.t. } \boldsymbol{\beta}=\boldsymbol{\alpha},\\ &= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ \min_{\boldsymbol{x}}\ h(\boldsymbol{\alpha})-\frac{\gamma}{2}\sum_{j=1}^{n}z_j\,\beta_j^2-\boldsymbol{x}^\top(\boldsymbol{\beta}-\boldsymbol{\alpha}),\\ &= \min_{\boldsymbol{x}}\ \underbrace{\max_{\boldsymbol{\alpha}}\ \left[h(\boldsymbol{\alpha})+\boldsymbol{x}^\top\boldsymbol{\alpha}\right]}_{(-h)^\star(\boldsymbol{x})=g(\boldsymbol{x})}+\sum_{i=1}^{n}\max_{\beta_i}\ \left[-\frac{\gamma}{2}z_i\,\beta_i^2-x_i\beta_i\right].\end{aligned}$$

Finally, observing that

$$\max_{\beta_i}\ \left[-\frac{\gamma}{2}z_i\,\beta_i^2-x_i\beta_i\right]=\begin{cases}\dfrac{x_i^2}{2\gamma z_i} & \text{if } z_i>0,\\ \max\limits_{\beta_i}\ x_i\beta_i & \text{if } z_j=0,\end{cases}$$

concludes the proof. □

Note that the equivalence stated in Theorem 2.3 also holds for $\boldsymbol{z}\in\text{Bool}(\mathcal{Z})$. Problem (2.21) can be formulated as a mixed-integer second-order cone problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{z}\in\mathcal{Z},\boldsymbol{\theta}\in\mathbb{R}^n}\quad \boldsymbol{c}^\top\boldsymbol{z}+g(\boldsymbol{x})+\sum_{i=1}^{n}\theta_i\quad\text{s.t.}\quad\left\|\begin{pmatrix}\sqrt{\frac{2}{\gamma}}x_i\\ \theta_i-z_i\end{pmatrix}\right\|_2\leq\theta_i+z_i\quad\forall i\in[n]. \tag{2.22}$$

and solved by linearizing the SOCP constraints into so-called perspective cuts, i.e., $\theta_i\geq\frac{1}{2\gamma}\bar{x}_i(2x_i-\bar{x}_iz_i)\ \forall\bar{x}\in\bar{\mathcal{X}}$, which have been extensively studied in the literature in the past fifteen years [110, 127, 89, 114, 7]. Observe that by separating Problem (2.21) into master and subproblems, their cutting-plane algorithm yields the same cut (2.17)

as in our scheme. In this regard, our approach supplies a new and insightful derivation of the perspective cut approach. It is worth noting that our proposal can easily be implemented within a standard integer optimization solver such as CPLEX or Gurobi using callbacks, while existing implementations of the perspective cut approach have required tailored branch-and-bound procedures [see, e.g., 110, Section 3.1].

Algorithmic Merits of Ridge, Big-M Regularization

We now summarize the relative merits of applying either ridge or big-M regularization from an algorithmic perspective:

- As noted in our randomized rounding guarantees in Section 2.5, the two regularization methods provide comparable bound gaps when $2M \approx \gamma L$, while if $2M \ll \gamma L$, big-M regularization provides smaller gaps, and if $2M \gg \gamma L$, ridge regularization provides smaller gaps.
- For linear problems, ridge regularization limits dual degeneracy, while big-M regularization does not. This benefit, however, has to be put in balance with the extra runtime and memory requirements needed for solving a quadratic, instead of linear, separation problem.

In summary, the benefits of applying either big-M or ridge regularization are largely even and depend on the specific instance to be solved. In the next section, we perform a sequence of numerical experiments on the problems studied in this chapter, to provide empirical guidance on which regularization approach works best when.

2.8 Numerical Experiments

In this section, we evaluate our single-tree cutting-plane algorithm, implemented in Julia 1.0 using CPLEX 12.8.0 and the Julia package `JuMP.jl` version 0.18.4 [91]. We compare our method against solving the natural big-M or MISOCP formulations directly, using CPLEX 12.8.0. All experiments were performed on one Intel Xeon $E5-2690$ v4 2.6GHz CPU core and using 32 GB RAM.

Overall Empirical Performance Versus State-of-the-Art

In this section, we compare our approach to state-of-the-art methods, and demonstrate that our approach outperforms the state-of-the-art for several problems.

Network design

We begin by evaluating the performance of our approach for the multi-commodity network design problem (2.5). We adapt the methodology of [126] and generate instances where each node $i \in [m]$ is the unique source of exactly one commodity ($k = m$). For each commodity $j \in [m]$, we generate demands according to $b^j_{j'} = \lfloor \mathcal{U}(5, 25) \rceil$ for $j' \neq j$ and $b^j_j = -\sum_{j' \neq j} b^j_{j'}$, where $\lfloor x \rceil$ is the closest integer to x and $\mathcal{U}(a, b)$ is a uniform random variable on $[a, b]$. We generate edge construction costs, c_e, uniformly on $\mathcal{U}(1, 4)$, and marginal flow circulation costs proportionally to each edge length[4]. The discrete set $\mathcal{Z}$ contains constraints of the form $\boldsymbol{z}_0 \leq \boldsymbol{z}$, where $\boldsymbol{z}_0$ is a binary vector which encodes existing edges. We generate graphs which contain a spanning tree plus pm additional randomly picked edges, with $p \in [4]$, so that the initial network is connected with $O(m)$ edges. We also impose a cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq (1 + 5\%)\boldsymbol{z}_0^\top \boldsymbol{e}$, which ensures that the network size increases by no more than 5%. For each edge, we impose a capacity $u_e \sim \lfloor \mathcal{U}(0.2, 1)B/A \rceil$, where $B = -\sum_{j=1}^m b^j_j$ is the total demand and $A = (1 + p)m$. We penalize the constraint $\boldsymbol{x} \leq \boldsymbol{u}$ with a penalty parameter $\lambda = 1,000$[5]. For big-M regularization, we set $M = \sum_j |b^j_j|$, and take $\gamma = \frac{2}{m(m-1)}$ for ridge regularization.

We apply our approach to large networks with 100s nodes, i.e., 10,000s edges, which is ten times larger than the state-of-the-art [132, 126], and compare the quality of the incumbent solutions after an hour, since no approach could terminate up to a satisfiable optimality gap within this time limit. Note that we define the quality of a solution as its cost in absence of regularization, although we might have augmented

[4]Nodes are uniformly distributed over the unit square $[0, 1]^2$. We fix the cost to be ten times the Euclidean distance.

[5]We do so to allow for a fair comparison between big-M and ridge regularization. By penalizing the capacity constraint, we remove a natural big-M regularization term and no regularization can be considered as more natural than the other.

the original formulation with a regularization term to compute the solution. As a result, we can compare the performance big-M and ridge regularization directly, despite the fact that the optimization problems they solve are actually different. On the other hand, performance metrics that depend on the function being minimized, such as the optimality gap, would not permit such a comparison. In 100 instances, our cutting plane algorithm with big-M regularization provides a better solution 94% of the time, by 9.9% on average, and by up to 40% for the largest networks. For ridge regularization, the cutting plane algorithm scales to higher dimensions than plain mixed-integer SOCP, returns solutions systematically better than those found by CPLEX (in terms of unregularized cost), by 11% on average. Also, ridge regularization usually outperforms big-M regularization, as reported in Table 2.3.

Given how numerically challenging these optimization problems are, the optimality gaps returned by all methods are often uninformative ($> 100\%$). Still, we observe that, with big-M regularization, CPLEX systematically returns tighter optimality gaps that the cutting-plane approach, while with ridge regularization, the gaps obtained by the cutting-plane algorithm are tighter 86% of the times. Even artificially added, ridge regularization improves the tractability of outer approximation.

Binary quadratic optimization

We study some of the binary quadratic optimization problems collated in the BQP library by [223]. Specifically, the bqp-$\{50, 100, 250, 500, 1000\}$ instances generated by [13], which have a cost matrix density of 0.1, and the be-100 and be-120.8 instances generated by [44], which respectively have cost matrix densities of 1.0 and 0.8. Note that these instances were generated as maximization problems, and therefore we consider a higher objective value to be better. We warm-start the cutting-plane approach with the best solution found after $10,000$ iterations of Goemans-Williamson rounding [see 124]. We also consider imposing triangle inequalities [85] via lazy callbacks, for they substantially tighten the continuous relaxations.

Within an hour, only the bqp-50 and bqp-100 instances could be solved by any approach considered here, in which case cutting-planes with big-M regularization

Table 2.3: Best solution found after one hour on network design instances with m nodes and $(1+p)m$ initial edges. We report improvement, i.e., the relative difference between the solutions returned by CPLEX and the cutting-plane. Values are averaged over five randomly generated instances. For ridge regularization, we report the "unregularized" objective value, that is we fix $\boldsymbol{z}$ to the best solution found and resolve the corresponding sub-problem with big-M regularization. A "$-$" indicates that the solver could not finish the root node inspection within the time limit (one hour), and "Imp." is an abbreviation of improvement.

			Big-M			Ridge			Overall
m	p	unit	CPLEX	Cuts	Imp.	CPLEX	Cuts	Imp.	Imp.
40	0	$\times 10^9$	1.17	**1.16**	0.86%	1.55	**1.16**	24.38%	1.74%
80	0	$\times 10^9$	8.13	7.52	6.99%	9.95	**7.19**	26.74%	10.85%
120	0	$\times 10^{10}$	3.03	2.10	29.94%	–	**1.94**	–%	35.30%
160	0	$\times 10^{10}$	5.90	4.32	26.69%	–	**4.07**	–%	30.91%
200	0	$\times 10^{10}$	11.45	**7.78**	31.45%	–	**7.50**	–%	32.32%
40	1	$\times 10^8$	5.53	5.47	1.07%	5.97	**5.45**	8.74%	1.41%
80	1	$\times 10^9$	2.99	**2.94**	1.81%	3.16	2.95	6.78%	1.89%
120	1	$\times 10^9$	8.38	**7.82**	6.69%	–	**7.82**	–%	6.86%
160	1	$\times 10^{10}$	1.64	**1.54**	5.98%	–	**1.54**	–%	6.03%
200	1	$\times 10^{10}$	2.60	2.54	2.33%	–	**2.26**	–%	12.98%
40	2	$\times 10^8$	4.45	4.38	1.62%	4.76	**4.36**	8.27%	2.06%
80	2	$\times 10^9$	2.44	**2.31**	5.39%	2.46	**2.31**	5.97%	5.40%
120	2	$\times 10^9$	6.23	**5.89**	5.55%	–	**5.89**	–%	5.75%
160	2	$\times 10^{11}$	1.22	1.16	4.74%	–	**0.71**	–%	19.33%
200	2	$\times 10^{10}$	2.06	1.43	30.46%	–	**1.01**	–%	73.43%
40	3	$\times 10^8$	3.91	**3.85**	1.58%	4.13	**3.85**	6.73%	1.78%
80	3	$\times 10^9$	2.06	**1.94**	5.76%	2.04	**1.94**	5.44%	5.85%
120	3	$\times 10^9$	5.43	5.15	5.31%	–	**4.2**	–%	12.35%
40	4	$\times 10^8$	3.32	3.28	1.35%	3.53	**3.26**	7.71%	1.85%
80	4	$\times 10^9$	1.88	**1.77**	5.59%	–	**1.77**	–%	5.64%

Table 2.4: Average runtime in seconds on binary quadratic optimization problems from the Biq-Mac library [223, 44]. Values are averaged over 10 instances. A "−" denotes an instance which was not solved because the approach did not respect the 32GB peak memory budget.

Instance	n	Average runtime (s)/Average optimality gap (%)			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
bqp-50	50	29.4	0.6	30.6	**0.4**
bqp-100	100	122.3	51.7	25.3%	**38.6**
bqp-250	250	1108.1%	83.5%	87.0%	**46.1%**
bqp-500	500	2055.8%	1783.3%	**157.3%**	410.7%
bqp-1000	1000	–	–	**260.9%**	–
be100	100	**79.7%**	208.0%	249.4%	201.2%
be120.8	120	**146.4%**	225.8%	264.1%	220.3%

is faster than CPLEX (see Table 2.4). For instances which cannot be solved to optimality, although CPLEX has an edge in producing tighter optimality gaps for denser cost matrices, as depicted in Table 2.4, the cutting-plane method provides tighter optimality gaps for sparser cost matrices, and provides higher-quality solutions than CPLEX for all instances, especially as n increases (see Table 2.5).

We remark that the cutting plane approach has low peak memory usage compared with the other methods: For the bqp-1000 instances, cutting-planes without triangle inequalities was the only method which respected the 32GB memory budget. This is another benefit of decomposing Problem (2.1) into master and sub-problems.

Table 2.5: Average incumbent objective value (higher is better) after 1 hour for medium-scale binary quadratic optimization problems from the Biq-Mac library [223, 44]. "−" denotes an instance which was not solved because the approach did not respect the 32GB peak memory budget. Values are averaged over 10 instances. Cuts-Triangle includes an extended formulation in the master problem.

Instance	n	Average objective value			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
bqp-250	250	9920.8	41843.4	**43774.9**	43701.5
bqp-500	500	19417.1	19659.0	**122879.3**	122642.4
bqp-1000	1000	–	–	**351450.7**	–
be100	100	16403.0	16985.0	17152.1	**17178.5**
be120.8	120	17943.2	19270.3	19307.7	**19371.2**

Evaluation of Different Ingredients in Numerical Recipe

We now consider the capacitated facility problem (2.6) on 112 real-world instances available from the OR-Library [13, 133], with the natural big-M and the ridge regularization with $\gamma = 1$. In both cases, the algorithms return the true optimal solution. Compared to CPLEX with big-M regularization, our cutting plane algorithm with big-M regularization is faster in 12.7% of instances (by 53.6% on average), and in 23.85% of instances (by 54.5% on average) when using a ridge penalty. This observation suggests that ridge regularization is better suited for outer-approximation, most likely because, as discussed in Section 2.4, a strongly convex ridge regularizer breaks the degeneracy of the separation problems. Note that our approach could benefit from multi-threading and restarting.

We take advantage of these instances to breakdown the independent contribution of each ingredient in our numerical recipe in Table 2.6. Although each ingredient contributes independently, jointly improving the lower and upper bounds provides the greatest improvement.

Table 2.6: Proportion of wins and relative improvement over CPLEX in terms of computational time on the 112 instances from the OR-library [13, 133] for different implementations of our method: an outer-approximation (OA) scheme with cuts generated at the root node using Kelley's method (OA + Kelley), OA with the local search procedure (OA + Local search) and OA with a strategy for both the lower and upper bound (OA + Both). Relative improvement is averaged over all "win" instances.

	Big-M		Ridge	
Algorithm	% wins	Relative improvement	% wins	Relative improvement
OA + Kelley	1.8%	36.6%	30.1%	91.6%
OA + Local search	1.9%	49.5%	19.4%	73.8%
OA + Both	12.7%	53.6%	92.5%	91.7%

Big-M Versus Ridge Regularization

In this section, our primary interest is in ascertaining conditions under which it is advantageous to solve a problem using big-M or ridge regularization, and argue that ridge regularization is preferable over big-M regularization as soon as the objective

is sufficiently strongly convex.

To illustrate this point, we consider large instances of the thermal unit commitment problem originally generated by [111], and multiply the quadratic coefficient a_i for each generator i by a constant factor $\alpha \in \{0.1, 1, 2, 5, 10\}$. Table 2.7 depicts the average runtime for CPLEX to solve both formulations to certifiable optimality, or provides the average bound-gap whenever CPLEX exceeds a time limit of 1 hour. Observe that when $\alpha \leq 1$, the big-M regularization is faster, but, when $\alpha > 1$ the MISOCP approach converges fast while the big-M approach does not converge within an hour. Consequently, ridge regularization performs more favorably whenever the quadratic term is sufficiently strong.

Table 2.7: Average runtime in seconds per approach, on data from [111] where the quadratic cost are multiplied by a factor of α. If the method did not terminate in one hour, we report the bound gap. n denotes the number of generators, each instances has 24 trade periods.

α	0.1		1		2		5		10	
n	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge
100	**93.6**	299.0	**16.2**	229.4	0.32%	**47.9**	1.68%	**4.6**	2.76%	**6.0**
150	**35.6**	352.1	**6.2**	28.3	0.25%	**33.4**	1.69%	**6.4**	2.82%	**8.0**
200	**56.3**	138.1	**3.3**	239.7	0.24%	**112.9**	1.62%	**16.7**	2.81%	**21.2**

Relative Merits of Big-M, Ridge Regularization: Experimental

We now conclude our comparison of big-M and ridge regularization, as initiated in Sections 2.3 and 2.7, by indicating the benefits of big-M and ridge regularization, from an experimental perspective:

- Big-M and ridge regularization play fundamentally the same role in reformulating logical constraints. This echoes our theoretical analysis in Section 2.2.
- As observed in the unit commitment problems studied in Section 2.8, ridge regularization should be the method of choice whenever the objective function contains a naturally occurring strongly convex term, which is sufficiently large.
- As observed for network design and capacitated facility location problems, ridge

regularization is usually more amenable to outer-approximation than big-M regularization, because it eliminates most degeneracy issues associated with outer-approximating MINLOs.

- The efficiency of outer-approximation schemes relies on the speed at which separation problems are solved. In this regard, special problem-structure or cardinality constraints on the discrete variable $\boldsymbol{z}$ drastically help. This has been the case in network design, sparse empirical risk minimization and sparse portfolio selection problems in Section 2.8.

2.9 Concluding Remarks

In this chapter, we proposed a new interpretation of the big-M method, as a regularization term rather than a modeling trick. By expanding this regularization interpretation to include ridge regularization, we considered a wide family of relevant problems from the Operations Research literature and derived equivalent reformulations as mixed-integer saddle-point problems, which naturally give rise to theoretical analysis and computational algorithms. Our framework provides provably near-optimal solutions in polynomial time via solving Boolean relaxations and performing randomized rounding as well as certifiably optimal solutions through an efficient branch-and-bound procedure, and, as we shall see in subsequent chapters of this thesis, indeed frequently outperforms the state-of-the-art in numerical experiments.

2.10 Appendix: Bounding the Lipschitz Constant

In our results, we relied on the observation that there exists some constant $L > 0$ such that, for any $\boldsymbol{z} \in \mathcal{Z}$, $\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\| \leq L$. Such an L always exists, since $\mathcal{Z}$ is a finite set. However, as our rounding results depend on L, explicit bounds are desirable.

We remark that while our interest is in the Lipschitz constant with respect to "$\boldsymbol{\alpha}$" in a generic setting, we have used different notation for some of the problems which fit in our framework, in order to remain consistent with the literature. In this sense,

we are also interested in obtaining a Lipschitz constant with respect to $\boldsymbol{w}$ for the portfolio selection problem (2.7), among others.

In this appendix, we bound the magnitude of L in a less conservative manner. Our first result provides a bound on L which holds whenever the function $h(\boldsymbol{\alpha})$ in Equation (2.15) is strongly concave in $\boldsymbol{\alpha}$, which occurs for the sparse ERM problem (2.8) with ordinary least-squares loss, the unit commitment problem (2.12), the portfolio selection (2.7), and network design problems whenever $\boldsymbol{\Sigma}$ (resp. $\boldsymbol{Q}$) is full-rank:

Lemma 2.1. *Let $h(\cdot)$ be a strongly concave function with parameter $\mu > 0$ [see 54, Chapter 9.1.2 for a general theory of strong convexity], and suppose that $\mathbf{0} \in dom(g)$ and $\boldsymbol{\alpha}^\star := \arg\max_{\boldsymbol{\alpha}} h(\boldsymbol{\alpha})$. Then, for any choice of $\boldsymbol{z}$, we have*

$$\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_2^2 \leq 8\,\frac{h(\boldsymbol{\alpha}^\star) - h(\mathbf{0})}{\mu},$$

i.e., $\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty \leq L$, where $L := 2\sqrt{2\frac{h(\boldsymbol{\alpha}^\star)-h(\mathbf{0})}{\mu}}$.

Proof. By the definition of strong concavity, for any $\boldsymbol{\alpha}$ we have

$$h(\boldsymbol{\alpha}) \leq h(\boldsymbol{\alpha}^\star) + \nabla h(\boldsymbol{\alpha}^\star)^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star) - \frac{\mu}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_2^2,$$

where $\nabla h(\boldsymbol{\alpha}^\star)^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star) \leq 0$ by the first-order necessary conditions for optimality, leading to

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_2^2 \leq 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\boldsymbol{\alpha})}{\mu}.$$

In particular for $\boldsymbol{\alpha} = \mathbf{0}$, we have

$$\|\boldsymbol{\alpha}^\star\|_2^2 \leq 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\mathbf{0})}{\mu},$$

and for $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star(\boldsymbol{z})$,

$$\|\boldsymbol{\alpha}^\star(\boldsymbol{z}) - \boldsymbol{\alpha}^\star\|_2^2 \leq 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\mathbf{0})}{\mu},$$

since

$$h(\boldsymbol{\alpha}^\star(\boldsymbol{z})) \geq h(\boldsymbol{\alpha}^\star(\boldsymbol{z})) - \sum_{j=1}^{n} z_j \Omega_j^\star(\alpha^\star(\boldsymbol{z})_j) \geq h(\mathbf{0}).$$

The result then follows by the triangle inequality. □

An important special case of the above result arises for the sparse ERM problem, as we demonstrate in the following corollary to Lemma 2.1:

Corollary 2.1. *For the sparse ERM problem* (2.8) *with OLS loss and a cardinality constraint* $\boldsymbol{e}^\top \boldsymbol{z} \leq k$*, a valid bound on the Lipschitz constant is*

$$\begin{aligned}\|\boldsymbol{\beta}^\star(\boldsymbol{z})\|_\infty = \|\mathrm{Diag}(\boldsymbol{Z})\boldsymbol{X}^\top \boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty &\leq \|\mathrm{Diag}(\boldsymbol{Z})\boldsymbol{X}^\top\|_\infty \|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty \\ &\leq \max_i \boldsymbol{X}_{i,[k]} \|\boldsymbol{\alpha}\|_2 \leq 2 \max_i \boldsymbol{X}_{i,[k]} \|\boldsymbol{y}\|_2,\end{aligned}$$

where $\boldsymbol{X}_{i,[k]}$ *is the sum of the* k *largest entries in the column* $\boldsymbol{X}_{i,[k]}$*.*

Proof. Applying Lemma 2.1 yields the bound

$$\|\boldsymbol{\alpha}\|_2 \leq 2\|\boldsymbol{y}\|_2,$$

after observing that we can parameterize this problem in $\boldsymbol{\alpha}$, and for this problem:

1. Setting $\boldsymbol{\alpha} = 0$ yields $h(\boldsymbol{\alpha}) = 0$.
2. $0 \leq h(\boldsymbol{\alpha}^\star) \leq \boldsymbol{y}^\top \boldsymbol{\alpha}^\star - \frac{1}{2}\boldsymbol{\alpha}^{\star\top}\boldsymbol{\alpha}^\star \leq \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{y}$.
3. $h(\cdot)$ is strongly concave in $\boldsymbol{\alpha}$, with concavity constant $\mu \geq 1$.

The result follows by applying the definition of the operator norm, and pessimizing. □

Chapter 3

Sparse Portfolio Selection

Since the Nobel-prize winning work of Markowitz [173], the problem of selecting an optimal portfolio of securities has received an enormous amount of attention from practitioners and academics alike. In a universe containing n distinct securities with expected marginal returns $\boldsymbol{\mu} \in \mathbb{R}^n$ and a variance-covariance matrix of the returns $\boldsymbol{\Sigma} \in \mathcal{S}^n_+$, the scalarized Markowitz model selects a portfolio which provides the highest expected return for a given amount of variance, via:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n_+} \ \frac{\sigma}{2} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{e}^\top \boldsymbol{x} = 1, \tag{3.1}$$

where $\sigma \geq 0$ is a parameter that controls the trade-off between the portfolios risk and return, and $\boldsymbol{e} \in \mathbb{R}^n$ denotes the vector of all ones.

To improve its realism, many authors have proposed augmenting Problem (3.1) with minimum investment, maximum investment, and cardinality constraints among others [see, e.g., 137, 192, 69]. Unfortunately, these constraints are disparate and imply each other, which makes defining a canonical portfolio model challenging.

Bienstock [42], Bertsimas et al. [30] defined a realistic portfolio selection model by augmenting Problem (3.1) with two sets of inequalities. The first set is a generic system of linear inequalites $\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}$ which, through an appropriate choice of data $\boldsymbol{l} \in \mathbb{R}^m, \boldsymbol{u} \in \mathbb{R}^m, \boldsymbol{A} \in \mathbb{R}^{m \times n}$, ensures that various real-world constraints such as allocating an appropriate amount of capital to each market sector hold. The

second inequality limits the number of non-zero positions held to $k \in \mathbb{N}$, by requiring that the portfolio is k-sparse, i.e., $\|\boldsymbol{x}\|_0 \leq k$. The sparsity constraint is important because (a) managers incur monitoring costs for each non-zero position, and (b) investors believe that portfolio managers who do not control the number of positions held perform index-tracking while charging active management fees [see 30, for an implementation of portfolio selection with sparsity constraints at a real-world asset management company]. Imposing the real-world constraints yields the following NP-hard—even without linear inequalities; see Gao and Li [120, Section E.C.1] for a proof—model:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n_+} \ \frac{\sigma}{2} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{x} \text{ s.t. } \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top \boldsymbol{x} = 1,\ \|\boldsymbol{x}\|_0 \leq k. \tag{3.2}$$

By introducing binary variables z_i which model whether x_i takes non-zero values by requiring that $x_i = 0$ if $z_i = 0$, we rewrite the above problem as a mixed-integer quadratic optimization problem:

$$\begin{aligned} \min_{\boldsymbol{z} \in \{0,1\}^n : \boldsymbol{e}^\top \boldsymbol{z} \leq k,\ \boldsymbol{x} \in \mathbb{R}^n_+} \quad & \frac{\sigma}{2} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top \boldsymbol{x} = 1, x_i = 0 \text{ if } z_i = 0\ \forall i \in [n]. \end{aligned} \tag{3.3}$$

In the past 20 years, a number of authors have proposed approaches for solving Problem (3.2) to certifiable optimality. However, no method has been shown to scale to real-world problem sizes where $20 \leq k \leq 50$ and $500 \leq n \leq 3,200$. This lack of scalability presents a challenge for practitioners and academics alike, because a scalable algorithm for Problem (3.2) has numerous financial applications, while algorithms which do not scale to this problem size are less practically useful.

Problem Formulation and Main Contributions

In this chapter, we provide two main contributions. Our first contribution is augmenting Problem (3.2) with a ridge regularization term to yield:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \frac{\sigma}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} + \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2 - \boldsymbol{\mu}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ \|\boldsymbol{x}\|_0 \leq k. \end{aligned} \tag{3.4}$$

This problem is more practically tractable than Problem (3.2), for two reasons. First, as we formally establish in Section 3.3, the duality gap between Problem (3.4) and its second-order cone relaxation decreases as we decrease γ and becomes 0 at some finite $\gamma > 0$. Second, as we numerically establish in Section 3.4, the algorithms developed here converge more rapidly when γ is smaller.

In addition to being more practically tractable, Problem (3.4) is a computationally useful surrogate for (3.2). Indeed, as we formally establish in Section 3.2, any optimal solution to Problem (3.4) is a $1/(2\gamma)$-optimal solution to (3.2). Moreover, one can find a solution to Problem (3.2) which is—often substantially—better than this, by (a) solving (3.4) and (b) solving a simple quadratic optimization problem over the set of securities with the same support as (3.4)'s solution and an unregularized objective. Indeed, since there are finitely many k-sparse binary support vectors, this strategy recovers an optimal solution to (3.2) for any sufficiently large γ.

Our second main contribution is a scalable outer-approximation algorithm for Problem (3.4). By utilizing Problem (3.4)'s regularization term, we question the modeling paradigm of writing the logical constraint "$x_i = 0$ if $z_i = 0$" as $x_i \leq z_i$ in Problem (3.4), by substituting the equivalent but non-convex term $x_i z_i$ for x_i and invoking strong duality and that $z_i^2 = z_i$ to obtain a convex mixed-integer quadratic reformulation of the problem. This allows us to propose a new outer-approximation algorithm in the spirit of the methods of [92, 107] but applied to a perspective reformulation [110, 127] of the problem which solves large-scale sparse portfolio selection problems to certifiable optimality.

Connection between regularization and robustness:

While we have introduced ridge regularization as a device which improves the problem's tractability in practice, one can actually interpret the regularizer as a robustification technique which improves the overall quality of the selected portfolio in an out-of-sample setting. Indeed, Ledoit and Wolf [155] [see also 66] have demonstrated that, in mean-variance portfolio selection problems, the largest eigenvalues of the sample covariance matrix $\mathbf{\Sigma}$ are systematically biased upwards and the smallest eigenvalues of $\mathbf{\Sigma}$ are systematically biased downwards. As a result, imposing a ridge regularization term (with a properly cross-validated γ) leads to portfolios which perform better out-of-sample. In a similar vein, DeMiguel et al. [83] has shown that the strategy of allocating an identical amount of capital to each security outperforms 13 other popular investment strategies. Since a ridge regularization term encourages investing a more equal amount in each security, DeMiguel et al. [83]'s work can be interpreted to imply that a ridge regularization term is beneficial.

3.1 Background and Literature Review

Our work touches on three different strands of the mixed-integer non-linear optimization literature, each of which propose certifiably optimal methods for solving Problem (3.2): (a) branch-and-bound methods which solve a sequence of relaxations, (b) decomposition methods which separate the discrete and continuous variables in Problem (3.2), and (c) perspective reformulation methods which obtain tight relaxations by linking the discrete and the continuous in a non-linear manner.

Branch-and-bound algorithms

A variety of branch-and-bound algorithms have been proposed for solving Mixed-Integer Nonlinear Optimization problems to certifiable optimality since the work of Glover [123], who proposed linearizing logical constraints "$x = 0$ if $z = 0$" by rewriting them as $-Mz \leq x \leq Mz$ for some $M > 0$. This is known as the big-M method.

The first branch-and-bound algorithm for solving Problem (3.2) to certifiable optimality was proposed by Bienstock [42]. This algorithm does not make use of binary variables. Instead, it reformulates the sparsity constraint implicitly, by recursively branching on subsets of the universe of buyable securities and obtaining relaxations by imposing constraints of the form $\sum_i \frac{x_i}{M_i} \leq K$, where M_i is an upper bound on x_i. Similar branch-and-bound schemes (which make use of binary variables) are studied in Bertsimas and Shioda [25], Bonami and Lejeune [48], who solve instances of Problem (3.2) with up to 50 (resp. 200) securities to certifiable optimality. Unfortunately, these methods do not scale well, because reformulating a sparsity constraint via the big-M method often yields weak relaxations in practice[1]

Motivated by the need to obtain tighter relaxations, more sophisticated branch-and-bound schemes have since been proposed, which obtain higher-quality bounds by lifting the problem to a higher-dimensional space. The first lifted approach was proposed by Vielma et al. [218], who successfully solved instances of Problem (3.2) with up to 100 securities to certifiable optimality, by taking efficient polyhedral relaxations of second order cone constraints. This approach has since been improved by Gao and Li [120], Cui et al. [76], who derive non-linear branch-and-bound schemes which use even tighter second order cone and semi-definite relaxations to solve problems with up to 300 securities to certifiable optimality.

Decomposition algorithms

A well-known method for solving MINLOs such as (3.2) is called outer approximation (OA), which was first proposed by Duran and Grossmann [92] (building on the work of Kelley [146], Benders [19], Geoffrion [122]), who prove its finite termination. OA separates a difficult MINLO into a finite sequence of *master* mixed-integer linear problems and non-linear *subproblems* (NLOs). This is often a good strategy, because linear integer and continuous conic solvers are more powerful than MINLO solvers.

[1]Indeed, if all securities are i.i.d. then investing $\frac{1}{k}$ in k randomly selected securities constitutes an optimal solution to Problem (3.2), but, as proven in [43], branch-and-bound must expand $2^{\frac{n}{10}}$ nodes to improve upon a naive sparsity-constraint free bound by 10%, and expand all 2^n nodes to certify optimality.

Unfortunately, OA has not yet been successfully applied to Problem (3.2), because it requires informative subgradient inequalities from each subproblem to attain a fast rate of convergence. Among others, Borchers and Mitchell [50], Fletcher and Leyffer [108] have compared OA to branch-and-bound, and found that branch-and-bound outperforms OA for Problem (3.2).

In the present chapter, by invoking strong duality, we derive a new subgradient inequality, redesign OA using this inequality, and solve Problem (3.4) to certifiable optimality via OA. The numerical success of our decomposition scheme can be explained by two ingredients: (a) the strength of the subgradient inequality, and (b) the tightness of our non-linear reformulation of a sparsity constraint, as further investigated in a more general setting in the previous chapter.

Perspective reformulation algorithms

An important aspect of solving Problem (3.2) is understanding its objective's convex envelope, since approaches which exploit the envelope perform better than approaches which use looser approximations of the objective [148]. An important step in this direction was taken by [110], who built on the work of Ceria and Soares [67] to derive Problem (3.2)'s convex envelope under an assumption that $\mathbf{\Sigma}$ is diagonal, and reformulated the envelope as a semi-infinite piecewise linear function. By splitting a generic covariance matrix into a diagonal matrix plus a positive semidefinite matrix, they subsequently derived a class of perspective cuts which provide bound gaps of $< 1\%$ for instances of Problem (3.2) with up to 200 securities. This approach was subsequently refined by [112, 113], who solved auxiliary SDOs to extract larger diagonal matrices, and thereby solve instances of Problem (3.2) with up to 400 securities.

The perspective reformulation approach has also been extended by other authors. An important work in the area is Aktürk et al. [4], who, building on the work of Ben-Tal and Nemirovski [17, p. 88, item 5], prove that if $\mathbf{\Sigma}$ is positive definite, i.e., $\mathbf{\Sigma} \succ \mathbf{0}$, then after extracting a diagonal matrix $\boldsymbol{D} \succ \mathbf{0}$ such that $\sigma\mathbf{\Sigma} - \boldsymbol{D} \succeq \mathbf{0}$, Problem (3.2)

is equivalent to the following mixed-integer second order cone optimization problem:

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{Z}_k^n,\ \boldsymbol{x}\in\mathbb{R}_+^n,\ \boldsymbol{\theta}\in\mathbb{R}_+^n} \quad & \frac{\sigma}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} + \frac{1}{2}\sum_{i=1}^n D_{i,i}\theta_i - \boldsymbol{\mu}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ x_i^2 \leq \theta_i z_i \quad \forall i \in [n]. \end{aligned} \tag{3.5}$$

In light of the above MISOCO, a natural question to ask is *what is the best matrix* $\boldsymbol{D}$ *to use?* This question was partially[2] answered by Zheng et al. [235], who demonstrated that the matrix $\boldsymbol{D}$ which yields the tightest continuous relaxation is computable via semidefinite optimization, and invoked this observation to solve problems with up to 400 securities to optimality [see also 89, who derive a similar perspective reformulation of sparse regression problems]. We refer the reader to Günlük and Linderoth [127] for a survey of perspective reformulation approaches.

Connection to our approach

An unchallenged assumption in *all* perspective reformulation approaches is that Problem (3.2) *must not be modified*. Under this assumption, perspective reformulation approaches separate $\boldsymbol{\Sigma}$ into a diagonal matrix $\boldsymbol{D} \succeq \boldsymbol{0}$ plus a positive semidefinite matrix $\boldsymbol{H}$, such that $\boldsymbol{D}$ is as diagonally dominant as possible. Recently, this approach was challenged by Bertsimas and van Parys [27]. Following a standard statistical learning theory paradigm, they imposed a ridge regularizer and set $\boldsymbol{D}$ equal to $1/\gamma \cdot \mathbb{I}$, where $\mathbb{I}$ denotes an identity matrix of appropriate dimension. Subsequently, they derived a cutting-plane method which exploits the regularizer to solve large-scale sparse regression problems to certifiable optimality. In the present chapter, we join Bertsimas and van Parys [27] in imposing a ridge regularizer, and derive a cutting-plane method which solves convex MIQOs *with constraints*. We also unify their approach with the perspective reformulation approach, in two steps. First, we note that Bertsimas and van Parys [27]'s algorithm can be improved by setting $\boldsymbol{D}$ equal to $1/\gamma \cdot \mathbb{I}$ *plus* a perspective reformulation's diagonal matrix, and this is particularly effective when $\boldsymbol{\Sigma}$ is diagonally dominant. Second, we observe that the cutting-plane approach also

[2]Weaker continuous relaxations may perform better after branching, as discussed by [87].

helps solve the unregularized problem, indeed, as mentioned previously it successfully supplies a $1/(2\gamma)$-optimal solution to Problem (3.2).

3.2 A Cutting-Plane Method

In this section, we present an efficient outer-approximation method for solving Problem (3.4), via its reformulation (3.8), as outlined in Chapter 2. To achieve this, we first take a Cholesky decomposition of $\boldsymbol{\Sigma}$ and complete the square. This is justified, because $\boldsymbol{\Sigma}$ is positive semidefinite and rank-r, meaning there exists an $\boldsymbol{X} \in \mathbb{R}^{r\times n}$: $\boldsymbol{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X}$. Therefore, by scaling $\boldsymbol{\Sigma} \leftarrow \sigma\boldsymbol{\Sigma}$ and letting:

$$\boldsymbol{y} := \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1}\boldsymbol{X}\boldsymbol{\mu}, \tag{3.6}$$

$$\boldsymbol{d} := \left(\boldsymbol{X}^\top\left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1}\boldsymbol{X} - \mathbb{I}\right)\boldsymbol{\mu}, \tag{3.7}$$

be the projection of the return vector $\boldsymbol{\mu}$ onto the span and nullspace of $\boldsymbol{X}$, completing the square yields the following equivalent problem, where we add the constant $\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{y}$ without loss of generality:

$$\begin{aligned}\min_{\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{X}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \boldsymbol{d}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \le \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ \|\boldsymbol{x}\|_0 \le k.\end{aligned}$$

We can then rewrite Problem (3.4) as the following MIO:

$$\min_{\boldsymbol{z}\in\mathcal{Z}_k^n}\Big[f(\boldsymbol{z})\Big], \tag{3.8}$$

$$\begin{aligned}\text{where } f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \frac{1}{2\gamma}\boldsymbol{x}^\top\boldsymbol{x} + \frac{1}{2}\|\boldsymbol{X}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \boldsymbol{d}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \le \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ \boldsymbol{x} \ge \boldsymbol{0}, x_i = 0 \text{ if } z_i = 0\ \forall i \in [n].\end{aligned} \tag{3.9}$$

After performing this reformulation, we follow Chapter 2 in rewriting Problem

(3.8) as a saddle-point problem, in the following theorem:

Theorem 3.1. *Suppose Problem* (3.8) *is feasible. Then, it is equivalent to:*

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{Z}_k^n} \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^r,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u\in\mathbb{R}^m_+,\ \lambda\in\mathbb{R}}} & -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \frac{\gamma}{2}\sum_i z_i w_i^2 + \boldsymbol{y}^\top\boldsymbol{\alpha} + \boldsymbol{\beta}_l^\top\boldsymbol{l} - \boldsymbol{\beta}_u^\top\boldsymbol{u} + \lambda \\ s.t. \quad & \boldsymbol{w} \geq \boldsymbol{X}^\top\boldsymbol{\alpha} + \boldsymbol{A}^\top(\boldsymbol{\beta}_l - \boldsymbol{\beta}_u) + \lambda\boldsymbol{e} - \boldsymbol{d}. \end{aligned} \tag{3.10}$$

Theorem 3.1 supplies objective function evaluations $f(\boldsymbol{z}_t)$ and subgradients $\boldsymbol{g}_t$ after solving a single convex quadratic optimization problem. We formalize this observation in the following corollary:

Corollary 3.1. *Let $\boldsymbol{w}^\star(\boldsymbol{z})$ be an optimal choice of $\boldsymbol{w}$ for a particular subset of securities $\boldsymbol{z}$. Then, a valid subgradient $\boldsymbol{g}_{\boldsymbol{z}} \in \partial f(\boldsymbol{z})$ has components given by the following expression for each $i \in [n]$:*

$$g_{\boldsymbol{z},i} = -\frac{\gamma}{2} w_i^\star(\boldsymbol{z})^2. \tag{3.11}$$

Corollary 3.1 shows that evaluating $f(\hat{\boldsymbol{z}})$ yields a first-order underestimator:

$$f(\boldsymbol{z}) \geq f(\hat{\boldsymbol{z}}) + \boldsymbol{g}_{\hat{\boldsymbol{z}}}^\top(\boldsymbol{z} - \hat{\boldsymbol{z}}) \tag{3.12}$$

at no additional cost. Consequently, a numerically efficient strategy for minimizing $f(\boldsymbol{z})$ is the outer-approximation (OA) method discussed in Chapter 2.

As Algorithm 2.1's rate of convergence for sparse portfolio selection problems depends heavily upon its implementation, we now discuss some practical aspects of implementing the method.

Practical Aspects of the Cutting-Plane Method

A computationally efficient subproblem strategy

For computational efficiency, we would like to solve subproblems which only involve active indices, i.e., indices where $z_i = 1$, since $k \ll n$. At a first glance, this does

not appear to be possible, because we must supply an optimal choice of w_i for all n indices in order to obtain valid subgradients. Fortunately, we can in fact supply a full OA cut after solving a subproblem in the active indices, by exploiting the structure of the saddle-point reformulation. Specifically, we optimize over the k indices where $z_i = 1$ and set $w_i = \max\left(\boldsymbol{X}_i^\top \boldsymbol{\alpha}^\star + \boldsymbol{A}_i^\top(\boldsymbol{\beta}_l^\star - \boldsymbol{\beta}_u^\star) + \lambda^\star - d_i, 0\right)$ for the remaining $n-k$ w_i's. This procedure yields an optimal choice of w_i for each index i, because it is a feasible choice and the remaining w_i's have a weight of 0 in the objective.

Extracting diagonal dominance:

In problems where $\boldsymbol{\Sigma}$ is diagonally dominant in the sense of Barker and Carlson [11], i.e., $\Sigma_{i,i} \geq \sum_{j \neq i} |\Sigma_{i,j}| \ \forall i \in [n]$, , the performance of Algorithm 2.1 can often be substantially improved by *boosting* the regularizer, i.e., selecting a diagonal matrix $\boldsymbol{D} \succeq \boldsymbol{0}$ such that $\sigma\boldsymbol{\Sigma} - \boldsymbol{D} \succeq \boldsymbol{0}$, replacing $\sigma\boldsymbol{\Sigma}$ with $\sigma\boldsymbol{\Sigma} - \boldsymbol{D}$, and using a different regularizer $\gamma_i := \left(\frac{1}{\gamma} + D_{i,i}\right)^{-1}$ for each index i. In general, selecting such a $\boldsymbol{D}$ involves solving a semidefinite optimization problem (SDO) [112, 235], which is fast when n is in the hundreds, but requires a prohibitive amount of memory when n is in the thousands. In the latter case, we recommend taking a second-order cone inner approximation of the SD cone and improving the approximation via column generation. Indeed, this approach provides high-quality solutions to large-scale SDOs [see 3, 23].

Copy of variables

In problems with complicating constraints, many feasibility cuts may be generated, which can hinder convergence greatly. If this occurs, we recommend introducing a copy of $\boldsymbol{x}$ in the master problem, and imposing the following master problem constraints:

$$\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top \boldsymbol{x} = 1,\ \boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{x} \leq \boldsymbol{z} \tag{3.13}$$

while keeping the subproblem the same. This approach performs well on the highly constrained problems studied in Section 3.4.

Modeling Minimum Investment Constraints

A frequently-studied extension to Problem (3.2) is to impose minimum investment constraints, which control transaction fees by requiring that $x_i \in \{0\} \cup [x_{i,\min}, u_i]$. We now extend our saddle-point reformulation to cope with them.

By letting z_i be a binary indicator variable which denotes whether we hold a non-zero position in the ith asset, we model these constraints via $z_i x_i \geq z_i x_{i,\min}\ \forall i \in [n]$. Moreover, we incorporate the upper bounds u_i within our algorithmic framework by "disappearing" the constraints $x_i \leq u_i$ into the general constraint set $\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}$.

Moreover, by letting ρ_i be the dual multiplier associated with the ith minimum investment constraint, and repeating the steps of our saddle-point reformulation, we retain efficient objective function and subgradient evaluations in the presence of these constraints. Specifically, including the constraints is equivalent to adding the term $\sum_{i=1}^{n} \rho_i \left(z_i x_{i,\min} - z_i x_i\right)$ to Problem (3.4)'s Lagrangian, which implies the saddle-point problem becomes:

$$\begin{aligned} \min_{\boldsymbol{z} \in \mathcal{Z}_k^n} \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^r,\ \boldsymbol{w} \in \mathbb{R}^n, \boldsymbol{\rho} \in \mathbb{R}_n^+ \\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u \in \mathbb{R}_+^m,\ \lambda \in \mathbb{R}}} & -\frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \frac{\gamma}{2}\sum_i z_i w_i^2 + \boldsymbol{y}^\top \boldsymbol{\alpha} + \boldsymbol{\beta}_l^\top \boldsymbol{l} - \boldsymbol{\beta}_u^\top \boldsymbol{u} + \lambda + \sum_i \rho_i z_i x_{i,\min} \\ \text{s.t.} \quad & \boldsymbol{w} \geq \boldsymbol{X}^\top \boldsymbol{\alpha} + \boldsymbol{A}^\top \left(\boldsymbol{\beta}_l - \boldsymbol{\beta}_u\right) + \lambda \boldsymbol{e} + \boldsymbol{\rho} - \boldsymbol{d}. \end{aligned} \tag{3.14}$$

Moreover, the subgradient with respect to each index i becomes

$$g_{\boldsymbol{z},i} = -\frac{\gamma}{2} w_i^\star(\boldsymbol{z})^2 + \rho_i x_{i,\min}. \tag{3.15}$$

Finally, if $z_i = 0$ then we can certainly set $\rho_i = 0$ without loss of optimality. Therefore, we recommend solving a subproblem in the k variables for which $z_i > 0$ and subsequently setting $\rho_i = 0$ for the remaining variables, in the manner discussed in the previous subsection. Indeed, setting $w_i = \max\left(\boldsymbol{X}_i^\top \boldsymbol{\alpha}^\star + \boldsymbol{A}_i^\top \left(\boldsymbol{\beta}_l^\star - \boldsymbol{\beta}_u^\star\right) + \lambda^\star + \rho_i^\star - d_i, 0\right)$ for each index i where $z_i = 0$, as discussed in the previous subsection, supplies the minimum absolute value of w_i.

Sensitivity Analysis

In this section, we study (3.8)'s dependence on the regularization parameter γ. This is an important issue in practice, because if we are interested in solving the unregularized problem, we can solve the regularized problem to obtain a support vector $\boldsymbol{z}$, and subsequently resolve the unregularized problem with the support fixed to $\boldsymbol{z}$. Therefore, we are interested in the suboptimality of $\boldsymbol{z}^\star$, an optimal solution to (3.4), if we perturb γ. We remark that the results in this section rely on basic sensitivity analysis proof techniques which can be found in most good optimization textbooks [e.g., 26, 205]. Nonetheless, we have included them, due to the central importance of regularization in this work, and because these results are not widely known.

Our first result demonstrates that the optimal support $\boldsymbol{z}$ for a larger value of γ can serve as a high-quality warm-start for a problem with less regularization:

Proposition 3.1. *Suppose that k is a fixed cardinality budget. Let $\boldsymbol{z}^\star(\gamma)$ denote an optimal solution to Problem* (3.8) *for a fixed regularizer γ, $f_\gamma(\boldsymbol{z})$ denote the optimal objective of Problem* (3.9) *for a fixed γ. Then, for any $\Delta > 0$:*

$$0 \leq f_{\gamma+\Delta}(\boldsymbol{z}^\star(\gamma)) - f_{\gamma+\Delta}(\boldsymbol{z}^\star(\gamma+\Delta)) \leq \frac{1}{2\gamma} - \frac{1}{2(\gamma+\Delta)}. \tag{3.16}$$

Proof. We have that

$$\begin{aligned} f_{\gamma+\Delta}(\boldsymbol{z}^\star(\gamma)) - f_{\gamma+\Delta}(\boldsymbol{z}^\star(\gamma+\Delta)) \leq & f_\gamma(\boldsymbol{z}^\star(\gamma)) - f_{\gamma+\Delta}(\boldsymbol{z}^\star(\gamma+\Delta)), \\ & \leq \left(\frac{1}{2\gamma} - \frac{1}{2(\gamma+\Delta)}\right) \|\boldsymbol{x}^\star(\boldsymbol{z}^\star(\gamma+\Delta))\|_2^2, \\ & \leq \left(\frac{1}{2\gamma} - \frac{1}{2(\gamma+\Delta)}\right), \end{aligned}$$

where the first inequality holds because decreasing the amount of regularization can only lower the optimal objective value, the second inequality holds because $\boldsymbol{x}^\star(\boldsymbol{z}^\star(\gamma+\Delta))$, an optimal choice of $\boldsymbol{x}$ with support indices $\boldsymbol{z}^\star(\gamma+\Delta)$, is a feasible solution with regularization parameter γ, specifically and the last inequality holds because all solutions $\boldsymbol{x}$ lie on the unit simplex. □

Observe that, by setting $\Delta \to \infty$, Proposition 3.1 supplies a formal proof of Section 1.1's claim that $\boldsymbol{z}^\star(\gamma)$ is a $1/(2\gamma)$-optimal solution for $\gamma \to +\infty$.

Our next result justifies our claim in the introduction that for a sufficiently large γ we recover the same optimal support from Problem (3.4) as the unregularized (3.2):

Proposition 3.2. *Let $\boldsymbol{z}^\star(\gamma)$ denote an optimal solution to Problem* (3.8) *for a fixed regularizer γ, and $f_\gamma(\boldsymbol{z})$ denote the optimal objective of Problem* (3.9) *for a fixed γ. Then, there exists some parameter $\gamma_0 > 0$ such that for any $\gamma \geq \gamma_0$:*

$$f_\gamma(\boldsymbol{z}^\star(\gamma_0)) = f_\gamma(z^\star(\gamma)). \tag{3.17}$$

Proof. Let us observe that, for each $\boldsymbol{z} \in \mathcal{Z}_n^k$, $f_\gamma(\boldsymbol{z})$ is concave in $1/\gamma$ as the pointwise minimum of functions which are linear in $1/\gamma$, and moreover $f_\gamma := \min_{\boldsymbol{z} \in \mathcal{Z}_n^k} f_\gamma(\boldsymbol{z})$ is also a concave function in $1/\gamma$. By this concavity, it is a standard result from sensitivity analysis [see, e.g., 26, Chapter 5.6] that the set of all γ's for which a particular $\boldsymbol{z}$ is optimal must form a (possibly open) interval. The result then follows directly from the finiteness of $\mathcal{Z}_n^k$. □

Our final result in this section shows that the optimal support of the portfolio remains unchanged for sufficiently small γ's (proof omitted, follows in the same fashion as Proposition 3.2):

Corollary 3.2. *Let $\boldsymbol{z}^\star(\gamma)$ denote an optimal solution to Problem* (3.8) *for a fixed regularizer γ, and $f_\gamma(\boldsymbol{z})$ denote the optimal objective of Problem* (3.9) *for a fixed γ. Then, there exists some parameter $\gamma_1 > 0$ such that for any $\gamma \leq \gamma_1$, we have:*

$$f_\gamma(\boldsymbol{z}^\star(\gamma_1)) = f_\gamma(\boldsymbol{z}^\star(\gamma)). \tag{3.18}$$

3.3 Improving the Cutting-Plane Method

In portfolio rebalancing applications, practitioners often require a high-quality solution to Problem (3.4) within a fixed time budget. Unfortunately, Algorithm 2.1 is ill-suited to this task: while it always identifies a certifiably optimal solution, it

does not always do so within a time budget. In this section, we propose alternative techniques which sacrifice some optimality for speed, and discuss how they can be applied to improve the performance of Algorithm 2.1. In Section 3.3 we propose a warm-start heuristic which supplies a high-quality solution to Problem (3.4) a priori, and in Section 3.3 we derive a second order cone representable lower bound which is often very tight in practice. Taken together, these techniques supply a certifiably near optimal solution quickly, which can often be further improved by running Algorithm 2.1 for a short amount of time.

An ADMM Warm-Start Heuristic

In branch-and-cut methods, a frequently observed source of inefficiency is that solvers explore highly suboptimal regions of the search space in considerable depth. To discourage this behavior, optimizers frequently supply a high-quality feasible solution, which is installed as an incumbent by the solver. Warm-starts are beneficial for two reasons. First, they improve Algorithm 2.1's upper bound. Second, they allow Algorithm 2.1 to prune vectors of partial solutions which are provably worse than the warm-start, which in turn improves Algorithm 2.1's bound quality, by reducing the set of feasible binaries which can be selected at each subsequent iteration. Indeed, by pruning suboptimal solutions, warm-starts encourage branch-and-cut methods to focus on regions of the search space which contain near-optimal solutions.

We now describe a heuristic which supplies high-quality solutions for Problem (3.4), inspired by a heuristic due to Bertsimas et al. [38, Algorithm 1]. The heuristic works under the assumption that $f(\boldsymbol{z})$ is L-Lipschitz continuous in $\boldsymbol{z}$, with Lipschitz continuous gradient $g_{\boldsymbol{z}}$ such that

$$\|\boldsymbol{g}_{\boldsymbol{z}_1} - \boldsymbol{g}_{\boldsymbol{z}_2}\|_2 \leq L\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_2 \quad \forall \boldsymbol{z}_1, \boldsymbol{z}_2 \in \operatorname{Conv}\left(\mathcal{Z}_k^n\right). \tag{3.19}$$

This assumption is justified whenever the optimal dual variables are bounded. Under this assumption, the heuristic approximately minimizes $f(\boldsymbol{z})$ by iteratively minimizing a quadratic approximation of $f(\boldsymbol{z})$ at $\boldsymbol{z}_{\text{old}}$, namely $f(\boldsymbol{z}) \approx \|\boldsymbol{z} - \boldsymbol{x}^\star(\boldsymbol{z}_{\text{old}}) - \frac{1}{L} g_{\boldsymbol{z}_{\text{old}}}\|_2^2$.

This idea is algorithmized as follows: given a sparsity pattern $\boldsymbol{z}_{\text{old}} \in \mathcal{Z}_k^n$ and an optimal sparse portfolio for this given sparsity pattern $\boldsymbol{x}^\star(\boldsymbol{z}_{\text{old}})$, the method iteratively solve the following problem, which ranks the differences between each security's contribution to the portfolio, $x_i^\star(\boldsymbol{z}_{\text{old}})$, and its subgradient $g_{\boldsymbol{z}_{\text{old}},i}$:

$$\boldsymbol{z}_{\text{new}} := \arg\min_{\boldsymbol{z}\in\mathcal{Z}_k^n} \left\| \boldsymbol{z} - \boldsymbol{x}^\star(\boldsymbol{z}_{\text{old}}) + \frac{1}{L}\boldsymbol{g}_{\boldsymbol{z}_{\text{old}}} \right\|_2^2. \tag{3.20}$$

Note that, given $\boldsymbol{z}_{\text{old}}$, $\boldsymbol{z}_{\text{new}}$ can be obtained by setting $z_i = 1$ for k of the indices where $\left|-x_i^\star(\boldsymbol{z}_{\text{old}}) + \frac{1}{L}g_{\boldsymbol{z}_{\text{old}},i}\right|$ is largest [cf. 38, Proposition 3]. We formalize this warm-start procedure in Algorithm 3.1.

Algorithm 3.1 A discrete ADMM heuristic

$t \leftarrow 1$
$\boldsymbol{z}_1 \leftarrow$ randomly generated k-sparse binary vector.
while $\boldsymbol{z}_t \neq \boldsymbol{z}_{t-1}$ **and** $t < T$ **do**
Set $\boldsymbol{w}_t$ optimal solution to:

$$\begin{aligned} \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^r,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u\in\mathbb{R}^m_+,\ \lambda\in\mathbb{R}}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \frac{\gamma}{2}\sum_i z_{i,t}w_i^2 + \boldsymbol{y}^\top\boldsymbol{\alpha} + \boldsymbol{\beta}_l^\top\boldsymbol{l} - \boldsymbol{\beta}_u^\top\boldsymbol{u} + \lambda \\ \text{s.t.} \quad & \boldsymbol{w} \geq \boldsymbol{X}^\top\boldsymbol{\alpha} + \boldsymbol{A}^\top(\boldsymbol{\beta}_l - \boldsymbol{\beta}_u) + \lambda\boldsymbol{e} - \boldsymbol{d}. \end{aligned}$$

Average multipliers via $\boldsymbol{w}^\star \leftarrow \frac{1}{t}\boldsymbol{w}_t + \frac{t-1}{t}\boldsymbol{w}^\star$.
Set $g_{\boldsymbol{z},i} = \frac{-\gamma}{2}w_i^{\star 2} \quad \forall i \in [n]$, $x_{i,t} = \gamma w_i^\star \quad \forall i \in [n] : z_{i,t} = 1$,
$\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z}\in\mathcal{Z}_k^n} \left\|\boldsymbol{z} - \boldsymbol{x}_t + \frac{1}{L}\boldsymbol{g}_{\boldsymbol{z}_t}\right\|_2^2$
$t \leftarrow t + 1$
return $\boldsymbol{z}_t$

Some remarks on Algorithm 3.1 are now in order:

- In our numerical experiments, we run Algorithm 3.1 from five different randomly generated k-sparse binary vectors, to increase the probability that it identifies a high-quality solution.
- Averaging the dual multipliers across iterations, as suggested in the pseudocode, improves the method's performance. Observe that the contribution of each $\boldsymbol{w}_t$ to $\boldsymbol{w}^\star$ is $\frac{1}{t}\prod_{i=t+1}^{T_{final}}\frac{i-1}{i} = \frac{1}{T_{final}}$, where T_{final} is the total number of iterations completed by Algorithm 3.1 and we initially set $\boldsymbol{w}^\star = \boldsymbol{0}$ in order that $\boldsymbol{w}^\star$ is

defined when we perform the averaging step at the first iteration. Also, note that when $t = 1$ we have $(t-1)/t = 0$ so the initialization is unimportant.

- Each $\boldsymbol{w}_t$ is the optimal solution of a convex quadratic optimization problem which can be reformulated as a (rotated) second-order cone program. Therefore, each $\boldsymbol{w}_t$ can be obtained via a standard second-order cone solver such as `CPLEX`, `Gurobi` or `Mosek`.

- The Lipschitz constant L is motivated as an upper bound on an entry in a subgradient of $f(\boldsymbol{z})$, $\frac{\gamma}{2}w_i^2$. However, Algorithm 3.1 is ultimately a heuristic method. Therefore, we recommend picking L by cross-validating to minimize the objective obtained by Algorithm 3.1. In practice, setting $L = 10$ was sufficient to reliably obtain high-quality solutions in Section 3.4, because Algorithm 3.2 invokes a judicious combination of outer-approximation cuts and this warm-start to convert this warm-start into an optimal solution within seconds. Therefore, we set $L = 10$ throughout Section 3.4, although it may be appropriate to cross-validate L if running the method on new data.

A Second-Order Cone Relaxation

In financial applications, we sometimes require a certifiably near-optimal solution quickly but do not have time to certify optimality. Therefore, we now derive near-exact lower bounds which can be computed in polynomial time. Immediately, we see that we obtain a valid lower bound by relaxing the constraint $\boldsymbol{z} \in \mathcal{Z}_k^n$ to $\boldsymbol{z} \in \text{Conv}(\mathcal{Z}_k^n)$ in Problem (3.4). By invoking strong duality, we now demonstrate that this lower bound can be obtained by solving a single second order cone problem.

Theorem 3.2. *Suppose that Problem* (3.4) *is feasible. Then, the following three optimization problems attain the same optimal value:*

$$\begin{array}{ll} \min\limits_{\boldsymbol{z}\in\text{Conv}\left(\mathcal{Z}_k^n\right)} & \max\limits_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^r,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u\in\mathbb{R}^m_+,\ \lambda\in\mathbb{R}}} \quad -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \frac{\gamma}{2}\sum_i z_i w_i^2 + \boldsymbol{y}^\top\boldsymbol{\alpha} + \boldsymbol{\beta}_l^\top\boldsymbol{l} - \boldsymbol{\beta}_u^\top\boldsymbol{u} + \lambda \\ & \qquad s.t. \quad \boldsymbol{w} \geq \boldsymbol{X}^\top\boldsymbol{\alpha} + \lambda\boldsymbol{e} + \boldsymbol{A}^\top(\boldsymbol{\beta}_l - \boldsymbol{\beta}_u) - \boldsymbol{d}. \end{array} \tag{3.21}$$

$$\begin{aligned}\max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^r,\ \boldsymbol{v}\in\mathbb{R}^n_+,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u\in\mathbb{R}^m_+,\ \lambda\in\mathbb{R},\ t\in\mathbb{R}_+}} & -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha}+\boldsymbol{y}^\top\boldsymbol{\alpha}+\boldsymbol{\beta}_l^\top\boldsymbol{l}-\boldsymbol{\beta}_u^\top\boldsymbol{u}+\lambda-\boldsymbol{e}^\top\boldsymbol{v}-kt\\ s.t.\quad & \boldsymbol{w}\geq\boldsymbol{X}^\top\boldsymbol{\alpha}+\lambda\boldsymbol{e}+\boldsymbol{A}^\top(\boldsymbol{\beta}_l-\boldsymbol{\beta}_u)-\boldsymbol{d},\\ & v_i\geq\frac{\gamma}{2}w_i^2-t\quad\forall i\in[n].\end{aligned}\tag{3.22}$$

$$\begin{aligned}\min_{\boldsymbol{z}\in\mathrm{Conv}\left(\mathcal{Z}_k^n\right)}\ \min_{\boldsymbol{x}\in\mathbb{R}^n_+,\boldsymbol{\theta}\in\mathbb{R}^n_+}\quad & \frac{1}{2}\left\|\boldsymbol{X}\boldsymbol{x}-\boldsymbol{y}\right\|_2^2+\frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta}+\boldsymbol{d}^\top\boldsymbol{x}\\ s.t.\quad & \boldsymbol{l}\leq\boldsymbol{A}\boldsymbol{x}\leq\boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x}=1,\ x_i^2\leq z_i\theta_i\quad\forall i\in[n].\end{aligned}\tag{3.23}$$

Remark 4. *We recognize Problem* (3.23) *as a perspective relaxation of Problem* (3.4) *[see 127, for a survey]. As perspective relaxations are often near-exact in practice [110, 113] this explains why the second-order cone bound is high-quality.*

Proof. Problem (3.21) is strictly feasible, since the interior of $\mathrm{Conv}(\mathcal{Z}_k^n)$ is non-empty and $\boldsymbol{w}$ can be increased without bound. Therefore, the Sion-Kakutani minimax theorem [17, Appendix D.4.] holds, and we can exchange the minimum and maximum operators in Problem (3.21), to yield:

$$\begin{aligned}\max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^r,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u\in\mathbb{R}^m_+,\ \lambda\in\mathbb{R}}} & -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha}+\boldsymbol{y}^\top\boldsymbol{\alpha}+\boldsymbol{\beta}_l^\top\boldsymbol{l}-\boldsymbol{\beta}_u^\top\boldsymbol{u}+\lambda-\frac{\gamma}{2}\max_{\boldsymbol{z}\in\mathrm{Conv}\left(\mathcal{Z}_k^n\right)}\sum_i z_i w_i^2\\ \text{s.t.}\quad & \boldsymbol{w}\geq\boldsymbol{X}^\top\boldsymbol{\alpha}+\lambda\boldsymbol{e}+\boldsymbol{A}^\top(\boldsymbol{\beta}_l-\boldsymbol{\beta}_u)-\boldsymbol{d}.\end{aligned}\tag{3.24}$$

Next, fixing $\boldsymbol{w}$ and applying strong duality between the inner primal problem

$$\max_{\boldsymbol{z}\in\mathrm{Conv}\left(\mathcal{Z}_k^n\right)}\sum_i\frac{\gamma}{2}z_i w_i^2\ =\max_{\boldsymbol{z}}\ \sum_i\frac{\gamma}{2}z_i w_i^2\quad\text{s.t.}\quad\boldsymbol{0}\leq\boldsymbol{z}\leq\boldsymbol{e},\ \boldsymbol{e}^\top\boldsymbol{z}\leq k,$$

and its dual problem

$$\min_{\boldsymbol{v}\in\mathbb{R}^n_+,t\in\mathbb{R}_+}\ \boldsymbol{e}^\top\boldsymbol{v}+kt\quad\text{s.t.}\quad v_i+t\geq\frac{\gamma}{2}w_i^2\quad\forall i\in[n]$$

proves that strong duality holds between Problems (3.21)-(3.22).

Next, we observe that (3.22)-(3.23) are dual, as can be seen by applying

$$bc \geq a^2, b, c \geq 0 \iff \left\| \begin{pmatrix} 2a \\ b-c \end{pmatrix} \right\| \leq b + c$$

to rewrite Problem (3.22) as an SOCO in standard form, and applying SOCO duality [see, e.g., 54, Exercise 5.43]. Moreover, since Problem (3.22) is strictly feasible (as $\boldsymbol{v}$, $\boldsymbol{w}$ are unbounded from above) strong duality must hold between these problems. □

Having derived Problem (3.4)'s bidual, namely Problem (3.23), it follows from a direct application of convex analysis that the duality gap between Problem (3.4) and (3.23), Δ_γ, decreases as we decrease γ and becomes 0 at some finite $\gamma > 0$. Note however that this Δ_γ will, in general, depend upon the problem data [see 131, Theorem XII.5.2.2]. This observation justifies our claim in the introduction that decreasing γ makes Problem (3.4) easier.

We now derive conditions under which Problem (3.22) provides an optimal solution to Problem (3.4) a priori.

Corollary 3.3. *Let there exist some $\boldsymbol{z} \in \mathcal{Z}_k^n$ and set of dual multipliers $(\boldsymbol{v}^\star, \boldsymbol{w}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\beta}_l^\star, \boldsymbol{\beta}_u^\star, \lambda^\star)$ which solve Problem* (3.22)*, such that these two quantities collectively satisfy the following conditions:*

$$\gamma \sum_i z_i w_i^\star = 1,\ \boldsymbol{l} \leq \gamma \sum_i \boldsymbol{A}_i w_i^\star z_i \leq \boldsymbol{u},\ z_i w_i \geq 0 \quad \forall i \in [n],\ v_i^\star = 0 \quad \forall i \in [n] \ :\ z_i = 0. \tag{3.25}$$

Then, Problem (3.22)*'s lower bound is exact. Moreover, let $|w^\star|_{[k]}$ denote the kth largest entry in $\boldsymbol{w}^\star$* by absolute magnitude. *If $|w^\star|_{[k]} > |w^\star|_{[k+1]}$ in Problem* (3.22) *then setting*

$$z_i = 1 \quad \forall i : |w_i^\star| \geq |w^\star|_{[k]},\ z_i = 0 \quad \forall i : |w_i^\star| < |w^\star|_{[k]}$$

supplies a $\boldsymbol{z} \in \mathcal{Z}_k^n$ *which satisfies the above condition and hence solves Problem* (3.4).

Proof. Let there exist some $(\boldsymbol{v}^\star, \boldsymbol{w}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\beta}_l^\star, \boldsymbol{\beta}_u^\star, \lambda^\star)$ which solve Problem (3.22), and binary vector $\boldsymbol{z} \in \mathcal{Z}_k^n$, such that these two quantities collectively satisfy the conditions encapsulated in Expression (3.25). Then, this optimal solution to Problem (3.22) provides the following lower bound for Problem (3.4):

$$-\frac{1}{2}\boldsymbol{\alpha}^{*\top}\boldsymbol{\alpha}^\star + \boldsymbol{y}^\top\boldsymbol{\alpha}^\star + \boldsymbol{\beta}_l^{*\top}\boldsymbol{l} - \boldsymbol{\beta}_u^{*\top}\boldsymbol{u} + \lambda^\star - \boldsymbol{e}^\top\boldsymbol{v}^\star - kt^\star.$$

Moreover, let $\hat{\boldsymbol{x}}$ be a candidate solution to Problem (3.4) defined by $\hat{x}_i := \gamma w_i z_i$. Then, $\hat{\boldsymbol{x}}$ is feasible for Problem (3.4), since $\boldsymbol{l} \leq A\hat{\boldsymbol{x}} \leq \boldsymbol{u}$, $\boldsymbol{e}^\top\hat{\boldsymbol{x}} = 1$, $\hat{\boldsymbol{x}} \geq \boldsymbol{0}$ and $\|\hat{\boldsymbol{x}}\|_0 \leq k$ by Expression (3.25) and the definition of $\boldsymbol{z}$. Additionally, since an optimal choice of t is the kth largest value of $\frac{\gamma}{2}w_i^2$, i.e., $\frac{\gamma}{2}w_{[k]}^2$ [see 229, Lemma 1], at optimality we have that $\boldsymbol{e}^\top\boldsymbol{v} + kt = \frac{1}{2\gamma}\hat{\boldsymbol{x}}^\top\hat{\boldsymbol{x}}$. Therefore, Problem (3.4)'s objective when $\boldsymbol{x} = \hat{\boldsymbol{x}}$ is given by:

$$-\frac{1}{2}\boldsymbol{\alpha}^{*\top}\boldsymbol{\alpha}^\star + \boldsymbol{y}^\top\boldsymbol{\alpha}^\star + \boldsymbol{\beta}_l^{*\top}\boldsymbol{l} - \boldsymbol{\beta}_u^{*\top}\boldsymbol{u} + \lambda^\star - \frac{1}{2\gamma}\hat{\boldsymbol{x}}^\top\hat{\boldsymbol{x}},$$

which is less than or equal to (3.22)'s objective, since $v_i^\star = 0 \quad \forall i \in [n] \ : \ z_i = 0$.

Finally, let $|w^\star|_{[k]} > |w^\star|_{[k+1]}$ and let S denote the set of indices such that $|w_i^\star| \geq |w^\star|_{[k]}$. Then, as the primal-dual KKT conditions for max-k norms [see, e.g., 229, Lemma 1] imply that an optimal choice of t is given by $t^\star = \frac{\gamma}{2}w_{[k]}^{\star 2}$, we can set $t^\star = \frac{\gamma}{2}w_{[k]}^{\star 2}$ without loss of generality. Note that, in general, this choice is not unique. Indeed, any $t \in [\frac{\gamma}{2}w_{[k+1]}^\star, \frac{\gamma}{2}w_{[k]}^\star]$ constitutes an optimal choice [229].

We then have that $v_i^\star = 0 \ \forall i \notin S$, which implies that the constraint $v_i + t \geq \frac{\gamma}{2}w_i^2$ holds strictly for any $i \notin S$. Therefore, the dual multipliers associated with these constraints must take value 0. But these constraints' dual multipliers are precisely $\boldsymbol{z} \in \text{Conv}(\mathcal{Z}_k^n)$, which implies that $z_i = 1 \forall i \in S$ gives a valid set of dual multipliers. Moreover, setting $\boldsymbol{x}_i = \gamma z_i w_i^\star$ supplies an optimal (and thus feasible) choice of $\boldsymbol{x}$ for this fixed $\boldsymbol{z}$. Therefore, this primal-dual pair satisfies (3.25). □

We now apply Theorem 3.2 to prove that if $\boldsymbol{\Sigma}$ is a diagonal matrix, $\boldsymbol{\mu}$ is a multiple

of the vector of all ones and the matrix $\boldsymbol{A}$ is empty then Problem (3.4) is solvable in closed-form. Let us first observe that under these conditions (3.4) is equivalent to

$$\min \quad \sum_i \frac{1}{2\gamma_i} x_i^2 \quad \text{s.t.} \quad \boldsymbol{e}^\top \boldsymbol{x} = 1, \boldsymbol{x} \geq \boldsymbol{0}, \|\boldsymbol{x}\|_0 \leq k.$$

We now have the following result:

Corollary 3.4. *Let $0 < \gamma_n \leq \gamma_{n-1} \leq ...\gamma_1$. Then, strong duality holds between*

$$\min \quad \sum_i \frac{1}{2\gamma_i} x_i^2 \quad \text{s.t.} \quad \boldsymbol{e}^\top \boldsymbol{x} = 1, \boldsymbol{x} \geq \boldsymbol{0}, \|\boldsymbol{x}\|_0 \leq k \tag{3.26}$$

and its second-order cone relaxation:

$$\begin{array}{ll} \max\limits_{\substack{\boldsymbol{v}\in\mathbb{R}^+_n,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \lambda\in\mathbb{R},\ t\in\mathbb{R}_+}} & \lambda - \boldsymbol{e}^\top \boldsymbol{v} - kt \\ \text{s.t.} & \boldsymbol{w} \geq \lambda \boldsymbol{e}, \quad v_i \geq \dfrac{\gamma_i}{2} w_i^2 - t \quad \forall i \in [n]. \end{array} \tag{3.27}$$

Moreover, an optimal solution to (3.26) *is $x_i = \frac{\gamma_i}{\sum_{i=1}^k \gamma_i}$ for $i \leq k$, $x_i = 0$ for $i > k$.*

Proof. By Theorem 3.2, a valid lower bound to Problem (3.26) is given by the SOCO

$$\begin{array}{ll} \max\limits_{\substack{\boldsymbol{v}\in\mathbb{R}^+_n,\ \boldsymbol{w}\in\mathbb{R}^n,\\ \lambda\in\mathbb{R},\ t\in\mathbb{R}_+}} & \lambda - \boldsymbol{e}^\top \boldsymbol{v} - kt \\ \text{s.t.} & \boldsymbol{w} \geq \lambda \boldsymbol{e}, \quad v_i \geq \dfrac{\gamma_i}{2} w_i^2 - t \quad \forall i \in [n]. \end{array} \tag{3.28}$$

Let us assume that $\lambda^\star \geq 0$ (otherwise the objective value cannot exceed 0, which is certainly suboptimal). Then, we can let the constraint $w_i \geq \lambda$ be binding without loss of optimality for each index i, i.e., set $\boldsymbol{w} = \lambda \boldsymbol{e}$ for some λ. This allows us to simplify this problem to:

$$\begin{array}{ll} \max\limits_{\boldsymbol{v}\in\mathbb{R}^+_n,\ \lambda\in\mathbb{R},\ t\in\mathbb{R}+} & \lambda - \boldsymbol{e}^\top \boldsymbol{v} - kt \\ \text{s.t.} & v_i \geq \dfrac{\gamma_i}{2} \lambda^2 - t \quad \forall i \in [n]. \end{array} \tag{3.29}$$

The KKT conditions for max-k norms [see, e.g., 229, Lemma 1] then reveal that

an optimal choice of t is given by the kth largest value of $\frac{\gamma_i}{2}\lambda^2$, i.e., $t^\star = \frac{\gamma_k}{2}\lambda^2$ and an optimal choice of v_i is given by $v_i = \max\left(\frac{\gamma_i}{2}\lambda^2 - t, 0\right)$, i.e.,

$$v_i^\star = \begin{cases} \frac{\gamma_i - \gamma_k}{2}\lambda^2 & \forall i \leq k, \\ 0 & \forall i > k. \end{cases}$$

Substituting these terms into the objective function gives an objective of

$$\lambda - \sum_{i=1}^{k} \frac{\gamma_i}{2}\lambda^2,$$

which implies that an optimal choice of λ is $\lambda = {}^1/\sum_{i=1}^k \gamma_i$. Next, substituting the expression $\lambda = {}^1/\sum_{i=1}^k \gamma_i$ into the objective function gives an objective value of ${}^\lambda/_2$, which implies that a lower bound on Problem (3.26)'s objective is ${}^1/_2 \sum_{i=1}^k \gamma_i$.

Finally, we construct a primal solution via $z_i = 1\ \forall i \leq k$, and the primal-dual KKT condition $x_i = \gamma_i z_i w_i = \gamma_i z_i \lambda = {}^{\gamma_i z_i}/\sum_{i=1}^k \gamma_i$. This is feasible, by inspection. Moreover, it has an objective of

$$\sum_{i=1}^{k} \frac{1}{2\gamma_i}(\gamma_i \lambda)^2 = \frac{\lambda}{2}\sum_{i=1}^{k} \gamma_i \lambda = \frac{\lambda}{2},$$

and therefore is optimal. □

An Improved Cutting-Plane Method

We close this section by combining Algorithm 2.1 with the improvements discussed in this section, to obtain an efficient numerical approach to Problem (3.4), which we present in Algorithm 3.2. Note that we use the larger of θ_t and the second-order cone lower bound in our termination criterion, as the second-order cone gap is sometimes less than ϵ.

Figure 3-1 depicts the algorithm's convergence on the problem *port2* with a cardinality value $k = 5$ and a minimum return constraint, as described in Section 3.4. Note that we did not use the second-order cone lower bound when generating this

Algorithm 3.2 A refined cutting-plane method for Problem (3.4).

Require: Initial warm-start solution $\boldsymbol{z}_1$
  $t \leftarrow 1$
  Set θ_{SOCO} optimal objective value of Problem (3.22)
  repeat
    Compute $\boldsymbol{z}_{t+1}, \theta_{t+1}$ solution of:

$$\min_{\boldsymbol{z} \in \mathcal{Z}_k^n, \theta} \theta \quad \text{s.t.} \quad \theta \geq f(\boldsymbol{z}_i) + g_{\boldsymbol{z}_i}^\top (\boldsymbol{z} - \boldsymbol{z}_i) \quad \forall i \in [t].$$

    Compute $f(\boldsymbol{z}_{t+1})$ and $g_{\boldsymbol{z}_{t+1}} \in \partial f(\boldsymbol{z}_{t+1})$
    $t \leftarrow t + 1$
  until $f(\boldsymbol{z}_t) - \max(\theta_t, \theta_{\text{SOCO}}) \leq \varepsilon$ **return** $\boldsymbol{z}_t$

plot; the second-order cone lower bound is 0.009288 in this instance, and Algorithm 3.2 requires 1225 cuts to improve upon this bound.

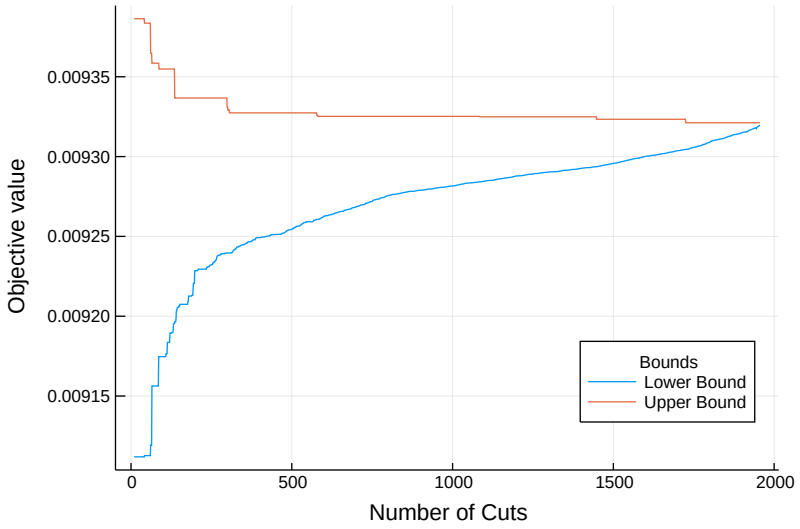


Figure 3-1: Convergence of Algorithm 3.2 on the OR-library problem *port2* with a minimum return constraint and a cardinality constraint $\|\boldsymbol{x}\|_0 \leq 5$. The behavior shown here is typical.

3.4 Experiments on Real-World Data

In this section, we evaluate our outer-approximation method, implemented in `Julia` 1.1 using the `JuMP.jl` package version 0.18.5 and solved using `CPLEX` version 12.8.0 for the master problems, and `Mosek` version 9.0 for the continuous quadratic subproblems.

We compare the method against big-M and MISOCO formulations of Problem (3.4), solved in `CPLEX`. To bridge the gap between theory and practice, we have made our code freely available on `Github`[3]

All experiments were performed on a MacBook Pro with a 2.9GHz i9 Intel®CPU and 16GB DDR4 Memory. For simplicity, we ran all methods on one thread, using default `CPLEX` parameters.

In all experiments, we solve the following optimization problem, which places a multiplier κ on the return term but is mathematically equivalent to (3.4):

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} + \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2 - \kappa\boldsymbol{\mu}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ \|\boldsymbol{x}\|_0 \leq k. \end{aligned} \tag{3.30}$$

Note that we only consider cases where $\kappa = 0$ or $\kappa = 1$, depending on whether we are penalizing low expected return portfolios in the objective or constraining the portfolios expected return.

We aim to answer the following questions:

1. How does Algorithm 3.2 compare to existing codes such as `CPLEX`?
2. How do constraints affect Algorithm 3.2's scalability?
3. How does Algorithm 3.2 scale as a function of the number of securities n?
4. How sensitive are optimal solutions to (3.4) to the hyperparameters κ, γ, k?

Comparison Between Algorithm 3.2, Existing Codes

We now present a direct comparison of Algorithm 3.2 with `CPLEX` version 12.8.0, where `CPLEX` uses the MISOCO formulations of Problem (3.4). Note that the MISOCO

[3] github.com/ryancorywright/SparsePortfolioSelection.jl.

formulation which we pass directly to CPLEX is [cf. 17, 4]:

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{Z}_k^n,\boldsymbol{x}\in\mathbb{R}_+^n,\boldsymbol{\theta}\in\mathbb{R}_+^n} \quad & \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} + \frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta} - \kappa\boldsymbol{\mu}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{e}^\top\boldsymbol{x} = 1,\ x_i^2 \leq z_i\theta_i \quad \forall i \in [n]. \end{aligned} \tag{3.31}$$

We compare the approaches in two distinct situations. First, when no constraints are applied and the system $\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}$ is empty, and second when a minimum return constraint is applied, i.e., $\boldsymbol{\mu}^\top\boldsymbol{x} \geq \bar{r}$. In the former case we set $\kappa = 1$, while in the latter case we set $\kappa = 0$ and as suggested by Cesarone et al. [68], Zheng et al. [235] we set $\bar{r}$ in the following manner: Let

$$r_{\min} = \boldsymbol{\mu}^\top\boldsymbol{x}_{\min} \text{ where } \boldsymbol{x}_{\min} = \arg\min_{\boldsymbol{x}}\ \frac{1}{2}\boldsymbol{x}^\top\left(\frac{1}{\gamma}\mathbb{I} + \boldsymbol{\Sigma}\right)\boldsymbol{x} \text{ s.t. } \boldsymbol{e}^\top\boldsymbol{x} = 1, \boldsymbol{x} \geq \boldsymbol{0},$$
$$r_{\max} = \boldsymbol{\mu}^\top\boldsymbol{x}_{\max} \text{ where } \boldsymbol{x}_{\max} = \arg\max_{\boldsymbol{x}}\ \boldsymbol{\mu}^\top\boldsymbol{x} - \frac{1}{2\gamma}\boldsymbol{x}^\top\boldsymbol{x} \text{ s.t. } \boldsymbol{e}^\top\boldsymbol{x} = 1, \boldsymbol{x} \geq \boldsymbol{0},$$

and set $\bar{r} = r_{\min} + 0.3(r_{\max} - r_{\min})$.

Table 3.1 (resp. Table 3.2) depicts the time required for both approaches to determine an optimal allocation of funds without (resp. with) the minimum return constraint. The problem data is taken from the 5 mean-variance portfolio optimization problems described by [69] and included in the OR-library test set by Beasley [13]. Note that we turned off the second-order cone lower bound for these tests, and ensured feasibility in the master problem by imposing $\sum_{i\in[n]:\mu_i\geq\bar{r}} z_i \geq 1$ when running Algorithm 3.2 on the instances with a minimum return constraint.

Table 3.2 indicates that some instances of *port2-port4* cannot be solved to certifiable optimality by any approach within an hour, in the presence of a minimum return constraint. Nonetheless, both Algorithm 3.2 and CPLEX's MISOCO method obtain solutions which are certifiably within 1% of optimality very quickly. Indeed, Table 3.3 depicts the bound gaps of all 3 approaches at 120s on these problems; Algorithm 3.2 never has a bound gap larger than 0.5%.

The experimental results illustrate that our approach is typically more efficient than the MISOCO approach. Moreover, our approach's edge over CPLEX increases

Table 3.1: Runtime in seconds per approach with $\kappa = 1$, $\gamma = \frac{100}{\sqrt{n}}$ and no constraints in the system $\boldsymbol{l} \leq \boldsymbol{Ax} \leq \boldsymbol{u}$. We impose a time limit of 300s and run all approaches on one thread. If a solver fails to converge, we report the number of explored nodes at the time limit.

Problem	n	k	Algorithm 3.2			CPLEX MISOCO	
			Time	Nodes	Cuts	Time	Nodes
port 1	31	5	0.17	0	4	**0.03**	0
		10	0.16	0	4	**0.01**	0
		20	0.14	0	4	0.03	0
port 2	85	5	**0.01**	0	4	0.11	0
		10	**0.01**	0	4	0.12	0
		20	**0.01**	0	4	0.29	0
port 3	89	5	**0.01**	0	8	0.38	0
		10	**0.01**	0	4	0.41	0
		20	**0.02**	0	4	0.11	0
port 4	98	5	**0.03**	0	8	0.41	0
		10	**0.02**	0	8	2.74	3
		20	**0.03**	0	9	0.38	0
port 5	225	5	**0.15**	0	9	11.17	9
		10	**0.02**	0	4	3.04	0
		20	**0.03**	0	7	2.88	0

with the problem size.

Our main findings from this set of experiments are as follows:

1. MISOCO approaches perform competitively, and are often a computationally reasonable approach for small to medium sized instances of Problem (3.4), as they are easy to implement and typically have bound gaps of $< 1\%$ in instances where they fail to converge within the time budget.

2. Varying the cardinality of the optimal portfolio does not affect solve times substantially without a minimum return constraint, although it has a nonlinear effect with this constraint.

Benchmarking With Threshold Constraints

In this section, we explore Algorithm 3.2's scalability in the presence of minimum investment constraints, by solving the problems generated by Frangioni and Gentile

Table 3.2: Runtime in seconds per approach with $\kappa = 0$, $\gamma = \frac{100}{\sqrt{n}}$ and a minimum return constraint $\boldsymbol{\mu}^\top \boldsymbol{x} \geq \bar{r}$. We impose a time limit of 3600s and run all approaches on one thread. If a solver fails to converge, we report the number of explored nodes at the time limit.

Problem	n	k	Algorithm 3.2			CPLEX	
			Time	Nodes	Cuts	Time	Nodes
port 1	31	5	**0.22**	161	32	0.83	47
		10	**0.20**	159	28	0.84	44
		20	0.16	0	7	**0.05**	0
port 2	85	5	**48.29**	73,850	1,961	91.98	1,163
		10	807.3	243,500	6,433	**82.44**	902
		20	**10.52**	12,260	1,224	24.54	210
port 3	89	5	**175.2**	132,700	3,187	213.3	2,528
		10	$> 3,600$	439,400	9,851	**531.3**	5,776
		20	119.5	65,180	4,473	**21.32**	170
port 4	98	5	**2,690**	479,700	11,320	2,779	25,180
		10	$> 3,600$	311,200	12,400	$> 3,600$	30,190
		20	1,638	241,600	10,710	**148.9**	1,115
port 5	225	5	**0.85**	1,489	202	28.3	22
		10	**0.60**	73	41	3.33	0
		20	**0.39**	63	52	115.02	90

[110] and subsequently solved by [112, 113, 235] among others[4]. These problems have minimum investment, maximum investment, and minimum return constraints, which render many entries in $\mathcal{Z}_k^n$ infeasible. Therefore, to avoid generating an excessive number of feasibility cuts, we use the copy of variables technique when running Algorithm 3.2.

Additionally, as the covariance matrices in these problems are highly diagonally dominant (with much larger on-diagonal entries than off-diagonal entries), the method does not converge quickly if we do not extract any diagonal dominance. Therefore, we first preprocess the covariance matrices to extract more diagonal dominance. Note that we need not actually solve any SDOs to preprocess the data, as high quality diagonal matrices for this problem data have been made publicly available by Frangioni et al. [115]. After reading in their diagonal matrix $\boldsymbol{D}$, we replace $\boldsymbol{\Sigma}$ with $\boldsymbol{\Sigma} - \boldsymbol{D}$ and use the regularizer γ_i for each index i, where $\gamma_i = \left(\frac{1}{\gamma} + D_{i,i}\right)^{-1}$.

We now compare the times for Algorithm 3.2 and `CPLEX`'s MISOCO routines to

[4]This problem data is available at www.di.unipi.it/optimize/Data/MV.html

Table 3.3: Bound gap at 120s per approach with $\kappa = 0$, $\gamma = \frac{100}{\sqrt{n}}$ and a minimum return constraint $\boldsymbol{\mu}^\top \boldsymbol{x} \geq \bar{r}$. We run all approaches on one thread.

Problem	n	k	Algorithm 3.2			CPLEX MISOCO	
			Gap (%)	Nodes	Cuts	Gap (%)	Nodes
port 2	85	5	0	73,850	1,961	0	1,163
		10	0.26	90,670	3,463	0	902
		20	0	12,260	1,224	0	210
port 3	89	5	0.1	123,100	2,308	0.27	1,247
		10	0.29	65,180	4,473	0.19	1,246
		20	0	60,090	3,237	0	170
port 4	98	5	0.18	55,460	3,419	0.60	888
		10	0.46	51,500	3,704	0.29	977
		20	0.17	57,990	3,393	0.05	846

solve the diagonally dominant instances in the dataset generated by Frangioni and Gentile [110], along with a variant of Algorithm 3.2 where we use the `in-out` method at the root node. In all cases, we take $\gamma = \frac{1000}{n}$, which ensures that $\gamma_i \approx \frac{1}{D_{i,i}}$, since on this dataset $\frac{1}{2\gamma}$ is around 5 orders of magnitude smaller than $D_{i,i}$ and thus the net contribution of the regularization term to the objective is negligible. Table 3.4 depicts the average time taken by each approach, and demonstrates that Algorithm 3.2 substantially outperforms `CPLEX`, particularly for problems without a cardinality constraint.

Our main findings from this experiment are as follows:

- Algorithm 3.2 outperforms `CPLEX` in the presence of minimum investment constraints, possibly because the master problems solved by Algorithm 3.2 are cardinality constrained LOs, rather than SOCOs, and therefore the method can quickly expand larger branch-and-bound trees.
- With a cardinality constraint, Algorithm 3.2's solve times are comparable to those reported by Zheng et al. [235], Frangioni et al. [114]. This can be explained by the fact that all three methods solve these problems in 10s of seconds, and thus these problems can be viewed as "easy". Without an explicit cardinality constraint (but with minimum investment constraints which impose an implicit cardinality constraint), our solve times are two orders of magnitude faster than

Table 3.4: Average runtime in seconds per approach with $\kappa = 0$, $\gamma = \frac{1000}{n}$ for the problems generated by Frangioni and Gentile [110]. We impose a time limit of 600s and run all approaches on one thread. If a solver fails to converge, we use 600s in lieu of the solve time. Note that the minimum investment constraints impose an implicit cardinality constraint with $k \approx 20$.

Problem	k	Algorithm 3.2			Algorithm 3.2 + in-out			CPLEX MISOCO	
		Time	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes
200+	6	**1.55**	1,298	236.3	1.77	1,262	209.4	87.74	95.3
200+	8	**1.95**	1,968	260.3	2.30	1,626	217	73.42	79.8
200+	10	7.74	7,606	509.7	**4.33**	3,686	298.9	161.9	184
200+	12	25.57	28,830	203.8	**2.06**	1,764	71.6	353.1	398.1
200+	200	18.71	23,190	208.4	**2.79**	2,288	92	599.3	735.1
300+	6	**16.83**	9,141	974.2	23.59	8,025	864.1	434.5	157.6
300+	8	**44.68**	21,050	1,577	64.46	19,682	1457.8	489.5	174.0
300+	10	88.57	44,160	1,901	**78.05**	33,253	1438.4	472.0	171.9
300+	12	16.16	13,880	262.7	**4.65**	3,181	127.4	401.5	158.2
300+	300	21.36	18,140	262.1	**9.24**	6,288	191.9	600.0	219.2
400+	6	**54.47**	13,330	1,717	66.52	12,160	1,619	531.7	84.0
400+	8	173.8	35,390	2,828	**160.9**	32,930	2,709	534.0	80.8
400+	10	158.0	55,490	1,669	104.5	32,314	1369.7	517.9	74.8
400+	12	3.97	4,324	116.6	**1.9**	1,214	48.6	478.0	75.3
400+	400	8.68	7,540	120.5	**5.19**	3,539	88.8	600.0	74.2

those reported by Zheng et al. [235]'s (an average of 580s for 400+), and an order of magnitude faster than those reported by Frangioni et al. [114] (an average of 52s for 400+).

Exploring the Scalability of Algorithm 3.2

In this section, we explore Algorithm 3.2's scalability with respect to the number of securities in the buyable universe, by measuring the time required to solve several large-scale sparse portfolio selection problems to provable optimality: the S&P 500, the Russell 1000, and the Wilshire 5000. In all three cases, the problem data is taken from daily closing prices from January 3 2007 to December 29 2017, which are obtained from Yahoo! Finance via the R package *quantmod* (see [207]), and rescaled to correspond to a holding period of one month. We apply Singular Value Decomposition to obtain low-rank estimates of the correlation matrix, and rescale the low-rank correlation matrix by each asset's variance to obtain a low-rank covariance

matrix $\mathbf{\Sigma}$. We also omit days with a greater than 20% change in closing prices when computing the mean and covariance for the Russell 1000 and Wilshire 5000, since these changes occur on low-volume trading and typically reverse the next day.

Tables 3.5–3.6 depict the times required for Algorithm 3.2 and `CPLEX MISOCO` to solve the problem to provable optimality for different choices of γ, k, and Rank($\mathbf{\Sigma}$). In particular, they depict the time taken to solve a constrained problem where $\kappa = 0$, and containing a minimum return constraint computed in the same fashion as in Section 3.4.

Table 3.5: Runtimes in seconds per approach for the S&P 500 with $\kappa = 0$ and a minimum return constraint, a one-month holding period and a runtime limit of 600s. For instances with a minimum return constraint where $\gamma = \frac{100}{\sqrt{n}}$, we run the in-out method at the root node before running Algorithm 3.2. We run all approaches on one thread. When a method fails to converge, we report the bound gap at 600s.

γ	Rank($\mathbf{\Sigma}$)	k	Algorithm 3.2			CPLEX MISOCO	
			Time	Nodes	Cuts	Time	Nodes
$\frac{1}{\sqrt{n}}$	50	10	0.01	0	3	73.28	210
		50	0.28	108	45	78.59	499
		100	0.05	7	7	0.97	0
		200	0.08	1	5	53.53	300
$\frac{1}{\sqrt{n}}$	200	10	5.20	2,804	450	345.0	171
		50	0.49	86	47	337.7	210
		100	0.15	5	8	104.2	40
		200	0.10	0	3	46.18	10
$\frac{100}{\sqrt{n}}$	50	10	0.09%	70,200	3,855	0.10%	1,600
		50	0.77	309	113	268.5	841
		100	0.09	0	8	1.66	0
		200	0.16	0	4	15.26	10
$\frac{100}{\sqrt{n}}$	200	0	0.45%	56,100	4,336	0.36%	280
		50	0.20	1	19	0.35%	256
		100	0.15	0	5	104.2	40
		200	0.18	0	4	76.80	10

Our main finding from this set of experiments is that Algorithm 3.2 is substantially faster than `CPLEX`'s MISOCO routine, particularly as the rank of $\mathbf{\Sigma}$ increases. The relative numerical success of Algorithm 3.2 in this section, compared to the previous section, can be explained by the differences in the problems solved: (a) in this section, we optimize over a sparse unit simplex, while in the previous section we optimized

Table 3.6: Runtimes in seconds per approach for the Wilshire 5000 with $\kappa = 0$ and a minimum return constraint, a one-month holding period and a runtime limit of 600s. For instances with a minimum return constraint where $\gamma = \frac{100}{\sqrt{n}}$, we run the in-out method at the root node before running Algorithm 3.2. We run all approaches on one thread. When a method fails to converge, we report the bound gap at 600s (using the symbol "–" to denote that a method failed to produce a feasible solution).

γ	Rank($\mathbf{\Sigma}$)	k	Algorithm 3.2			CPLEX MISOCO	
			Time	Nodes	Cuts	Time	Nodes
$\frac{1}{\sqrt{n}}$	100	10	1.95	0	2	50.0%	122
		50	2.32	0	2	32.0%	132
		100	0.59	10	9	62.0%	127
		200	0.27	0	6	44.5%	100
$\frac{1}{\sqrt{n}}$	1,000	10	0.01%	40,500	1,130	–	2
		50	0.02%	56,800	937	–	2
		100	0.02%	25,040	523	–	2
		200	2.61	1	12	–	2
$\frac{100}{\sqrt{n}}$	100	10	0.28%	24,870	1,178	50.1%	91
		50	0.38%	45,810	636	62.1%	82
		100	0.12%	55,700	912	45.1%	80
		200	0.49	0	10	22.1%	91
$\frac{100}{\sqrt{n}}$	1,000	10	1.02%	6,7600	1,108	–	2
		50	0.26	33,930	1,122	–	0
		100	1.85%	53,500	804	–	2
		200	1.28	1	7	–	2

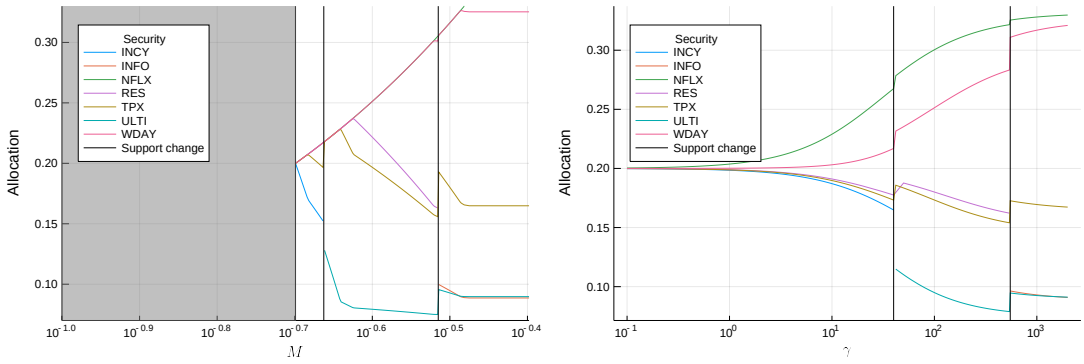
over minimum-return and minimum-investment constraints, (b) in this section, we use data taken directly from stock markets, while in the previous section we used less realistic synthetic data, which evidently made the problem harder.

Exploring Sensitivity to Hyperparameters

Our next set of experiments explores Problem (3.4)'s stability to changes in its hyperparameter γ. We first explore optimizing over a rank–200 approximation of the Russell 1000 with a one month holding period, a sparsity budget $k = 5$ and a weight $\kappa = 1$, using either the ridge regularization term proposed in this chapter or the big-M regularizer explored in Chapter 2.

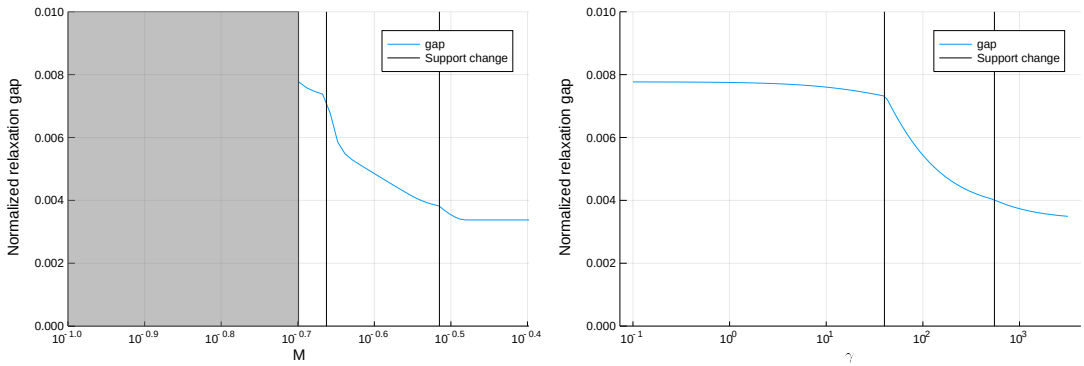
Figure 3-2 depicts the relationship between the optimal allocation of funds $\boldsymbol{x}^\star$ and the regularization parameter M (left) and γ (right), and Figure 3-3 depicts the

magnitude of the gap between the optimal objective and the Boolean relaxation's objective, normalized by the unregularized objective. The two investment profiles are comparable, selecting the same stocks. Yet, we observe two main differences: First, setting $M < \frac{1}{k}$ renders the entire problem infeasible, while the problem remains feasible for any $\gamma > 0$. This is a serious practical concern in cases where a lower bound on the value of M is not known apriori. Second, the profile for ridge regularization seems smoother than its equivalent with big-M.



(a) Big-M regularization **(b)** Ridge regularization

Figure 3-2: Optimal allocation of funds between securities as the regularization parameter (M or γ) increases. Data is obtained from the Russell 1000, with a cardinality budget of 5, a rank-200 approximation of the covariance matrix, a one-month holding period and an Arrow-Pratt coefficient of 1, as in [24]. Setting $M < \frac{1}{k}$ renders the entire problem infeasible.



(a) Big-M regularization **(b)** Ridge regularization

Figure 3-3: Magnitude of the normalized absolute bound gap as the regularization parameter (M or γ) increases, for the portfolio selection problem studied in Figure 3-2

Our final experiment studies the impact of the regularizer γ on both solve times and the number of cuts generated, in order to justify our assertion in the introduction that increasing the amount of regularization in the problem makes the problem easier. In this direction, we solve the ten 300+ and 400+ instances with minimum investment and minimum return constraints studied in Section 3.4 for different values of γ (with the copy of variables technique and in-out method on). We report the average runtime and number of cuts generated by Algorithm 3.2 across the 10 instances for each n in Figures 3-4 ($n = 300$) and 3-5 ($n = 400$).

Observe that the average runtime is essentially non-decreasing in γ, and for both values of n there exists a finite $\gamma > 0$ at which all instances can be solved using a single cut. This empirically verifies Section 3.2's sensitivity analysis findings.

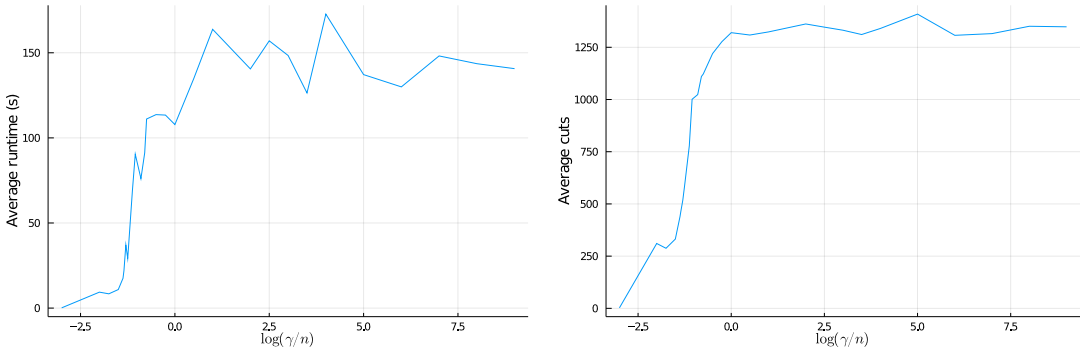


Figure 3-4: Average runtime (left) and number of cuts (right) vs. $\log(\gamma)$ for the 300+ instances with buy-in and minimum return constraints with a cardinality budget of $k = 10$.

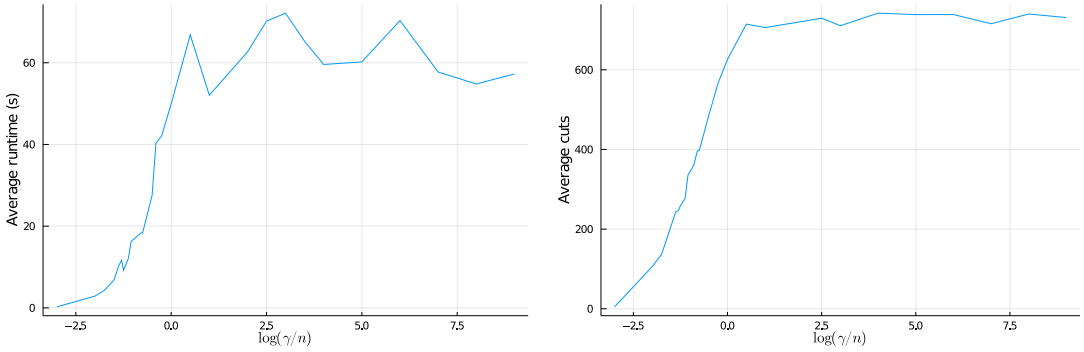


Figure 3-5: Average runtime (left) and number of cuts (right) vs. $\log(\gamma)$ for the 400+ instances with buy-in and minimum return constraints with a cardinality budget of $k = 10$.

Summary of Findings From Numerical Experiments

We are now in a position to answer the four questions introduced at the start of this section. Our findings are as follows:

1. In the absence of complicating constraints, Algorithm 3.2 is substantially more efficient than state-of-the-art MIQO solvers such as `CPLEX`. This efficiency improvement can be explained by (a) our ability to generate stronger and more informative lower bounds via dual subproblems, and (b) our dual representation of the problems' subgradients. Indeed, the method did not require more than one second to solve any of the constraint-free problems considered here, although this phenomenon can be partially attributed to the problem data used.

2. Although imposing complicating constraints, such as minimum investment constraints, slows Algorithm 3.2, the method performs competitively in the presence of these constraints. Moreover, running the `in-out` cutting-plane method at the root node substantially reduces the initial bound gap, and allows the method to supply a certifiably near-optimal (if not optimal) solution in seconds. This suggests that running the `in-out` method at the root node should be considered as a viable and more scalable alternative to existing root node techniques, particularly in the presence of complicating constraints such as minimum investment constraints, or if the cardinality budget is at least 10 (although it can do more harm than good for easier problems).

3. Algorithm 3.2 scales to solve real-world problem instances which comprise selecting assets from universes with thousands of securities, such as the Russell 1000 and the Wilshire 5000, while existing state-of-the-art approaches such as `CPLEX` either solve these problems much more slowly or do not successfully solve them, because they cannot attain sufficiently strong lower bounds quickly.

4. Solutions to Problem (3.4) are stable with respect to γ.

3.5 Conclusion and Extensions

This chapter describes a scalable algorithm for solving quadratic optimization problems subject to sparsity constraints, and applies it to the problem of sparse portfolio selection. Although sparse portfolio selection is NP-hard, and therefore considered to be intractable, our algorithm provides provably optimal portfolios even when the number of securities is in the thousands.

Our algorithm, which solves ridge-regularized sparse portfolio selection problems with mean-variance objectives, could be generalized to incorporate other risk measures, such as mean-CVaR risk measures, as developed in [150].

3.6 Appendix: Supplementary Material

In this section, we present supplementary experimental results pertaining to the experiments conducted in Section 3.4, in the order in which the experiments were conducted. For the sake of conciseness, further supplementary material regarding these experiments can be found in the online supplement to [24].

We first present the aggregate runtimes (in seconds) for all instances generated by Frangioni and Gentile [110], when we run `CPLEX`'s MISOCP solver after first supplying the cuts generated by the `in-out` method in Table 3.7. Note that, to allow `CPLEX` to benefit from the `in-out` cuts, we introduce an auxiliary variable τ, change the objective to minimizing τ, impose the constraint $\tau \geq \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} + \frac{1}{2\gamma}\boldsymbol{e}^\top \boldsymbol{\theta} - \kappa \boldsymbol{\mu}^\top \boldsymbol{x}$ to model the MISOCO objective, and impose the cuts from the in-out method using the epigraph variable τ.

Next, we present the instance-wise runtimes (in seconds) for the smallest instances generated by Frangioni and Gentile [110], with the diagonal matrix extraction technique proposed by Zheng et al. [235], and $k \in \{10, n\}$ (we restrict the values k can take to use the diagonal matrices pre-computed by Frangioni et al. [115]). Table 3.8 demonstrates that using the diagonal matrix extraction technique proposed by Zheng et al. [235] substantially slows our approach; the results for $n \in \{300, 400\}$ are similar.

Indeed, this technique is only faster for the pard200-1 problem with $k = 10$, and is slower in the other 95% of instances (sometimes substantially so).

Table 3.7: Average runtime in seconds per approach with $\kappa = 0$, $\gamma = \frac{1000}{n}$ for the problems generated by Frangioni and Gentile [110]. We impose a time limit of 600s and run all approaches on one thread. If a solver fails to converge, we report the number of explored nodes at the time limit, use 600s in lieu of the solve time, and report the number of failed instances (out of 10) next to the solve time in brackets. Note that the minimum investment constraints impose an implicit cardinality constraint with $k \approx 13$.

Problem	k	CPLEX MISOCO in-out	
		Time	Nodes
200+	6	255.5 (1)	92.3
200+	8	272.2 (1)	98.4
200+	10	375.2 (3)	116.4
200+	12	447.8 (6)	216
200+	200	$>$ 600 (10)	332.1
300+	6	573.4 (9)	92
300+	8	578.4 (9)	89.2
300+	10	$>$ 600 (10)	92.1
300+	12	575.65 (9)	117.3
300+	200	$>$ 600 (10)	108.5
400+	6	569.4 (9)	43.3
400+	8	563.3 (9)	45.1
400+	10	562.4 (9)	45.7
400+	12	593.8 (9)	58.3
400+	200	$>$ 600 (10)	41.4

Table 3.8: Performance of the outer-approximation method on the 200^+ instances generated by Frangioni and Gentile [110], with a time budget of 600s per approach, $\kappa = 0$, $\gamma = \frac{1000}{n}$, and the diagonal matrix extraction technique proposed by Zheng et al. [235]. We run all approaches on one thread. Note that "nc" refers to an instance without an explicit cardinality constraint.

Problem	k	Algorithm 3.2			Algorithm 3.2 + in-out			Algorithm 3.2 + in-out + 50		
		Time	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes	Cuts
pard200-1	10	0.74	130	30	0.03	0	4	0.03	0	4
pard200-1	nc	> 600	887,400	1,386	> 600	539,900	1,070	> 600	369,500	675
pard200-2	10	234.3	239,500	875	57.14	45,230	193	78.88	39,000	196
pard200-2	nc	> 600	995,900	823	> 600	157,100	135	> 600	226,100	66
pard200-3	10	245.1	207,200	1,195	71.95	55,050	365	76.64	30,910	249
pard200-3	nc	> 600	903,500	888	> 600	357,200	246	> 600	268,400	259
pard200-4	10	> 600	442,500	1,967	344.7	223,700	1,053	228.9	135,500	760
pard200-4	nc	535.1	913,500	1,092	> 600	297,600	94	529.8	212,600	94
pard200-5	10	> 600	439,900	4,965	48.87	70,200	206	69.71	52,300	204
pard200-5	nc	> 600	1,340,000	1,314	> 600	531,400	1,408	> 600	573,200	1,358
pard200-6	10	> 600	311,800	4,922	6.54	6,382	116	36.29	12,370	107
pard200-6	nc	> 600	1,280,000	1,016	> 600	479,900	789	> 600	557,600	580
pard200-7	10	549.8	389,000	2,542	515.6	228,100	1,336	292.4	105,900	743
pard200-7	nc	> 600	1,245,000	522	> 600	496,200	183	> 600	502,600	119
pard200-8	10	> 600	399,200	3,419	2.32	1,638	46	20.19	1,716	45
pard200-8	nc	> 600	1,337,000	862	> 600	674,300	552	> 600	507,900	420
pard200-9	10	589.7	576,100	1,756	6.31	8,746	122	26.53	8,290	143
pard200-9	nc	> 600	1,264,000	1,941	> 600	703,900	1,977	> 600	498,000	1,970
pard200-10	10	> 600	416,200	3,798	288.4	160,300	1,070	422.0	192,800	1,002
pard200-10	nc	> 600	974,400	1,584	> 600	498,100	1,513	> 600	313,400	1482

Chapter 4

Sparse Principal Component Analysis

In the era of big data, interpretable methods for compressing a high-dimensional dataset into a lower dimensional set which shares the same essential characteristics are imperative. Principal component analysis (PCA) is one of the most popular approaches for completing this task. Given data $\boldsymbol{A} \in \mathbb{R}^{n \times p}$ and its sample covariance matrix $\boldsymbol{\Sigma} := \frac{1}{n-1}\boldsymbol{A}\boldsymbol{A}^\top \in \mathbb{R}^{p \times p}$, PCA selects the leading eigenvectors, or principal components, of $\boldsymbol{\Sigma}$ and subsequently projects $\boldsymbol{A}$ onto these eigenvectors, by multiplying $\boldsymbol{A}$ by the leading principal components in order to obtain a lower-dimensional matrix.

The mathematically simplest approach to PCA is an iterative process. First, the leading eigenvector, or principal component, can be found by solving the following quadratic optimization problem, which can be addressed in a number of ways including via the power method [227]:

$$\max_{\boldsymbol{x} \in \mathbb{R}^p : \|\boldsymbol{x}\|_2 = 1} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x}.$$

After obtaining $\boldsymbol{x}_1$, an optimal solution to this problem which explains the most variance in $\boldsymbol{\Sigma}$ of any vector, we "project out" this direction from the covariance matrix by setting $\boldsymbol{\Sigma}_{\text{new}} := (\mathbb{I} - \boldsymbol{x}_1\boldsymbol{x}_1^\top)\boldsymbol{\Sigma}(\mathbb{I} - \boldsymbol{x}_1\boldsymbol{x}_1^\top)$, and resolve the quadratic problem to explain the optimal amount of variance once $\boldsymbol{x}_1$ is accounted for. Repeating this process iteratively d times, where $d \leq n$, supplies an orthogonal basis $\{\boldsymbol{x}_1, \ldots \boldsymbol{x}_d\}$. Moreover, this basis explains the most variance in $\boldsymbol{\Sigma}$ of any basis with d components,

and therefore multiplying $\boldsymbol{A}$ by this basis supplies a lower-dimensional dataset which shares the same essential characteristics.

A popular and unified approach for performing principal component analysis is via the singular value decomposition. Indeed, PCA can be achieved in $O(p^3)$ time by taking a singular value decomposition $\boldsymbol{\Sigma} = \boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^\top$, and projecting $\boldsymbol{A}$ onto the d leading eigenvalues of $\boldsymbol{S}$, $\boldsymbol{S}_{[1:k]}$, via $\boldsymbol{A}_{\text{new}} := \boldsymbol{S}_{[1:d]}\boldsymbol{A}$.

A common criticism of PCA is the columns of $\boldsymbol{S}$ are not interpretable, since each eigenvector is a linear combination of all features. This causes difficulties because:

- In medical diagnostic applications such as cancer detection, downstream decisions taken using principal component analysis need to be interpretable.
- In scientific applications such as protein folding, each original co-ordinate axis has a physical interpretation, and the reduced set of co-ordinate axes should also have this property.
- In financial applications such as investing capital across a set of index funds, each non-zero entry in each eigenvector incurs a transaction cost.

One common method for obtaining interpretable principal components is to stipulate that they are sparse. This leads to the following problem:

$$\max_{\boldsymbol{x}\in\mathbb{R}^p} \quad \boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x}^\top\boldsymbol{x} = 1, \quad ||\boldsymbol{x}||_0 \leq k, \tag{4.1}$$

where the constraint $||\boldsymbol{x}||_0 \leq k$ forces variance to be explained in a compelling way.

In this chapter, we demonstrate that Problem (4.1) can be reformulated as a mixed-integer semidefinite optimization problem. This is significant because prior works characterizes (4.1) as a low-rank—rather than mixed-integer semidefinite—problem, and the framework developed in Chapter 2 can handle integer but not low-rank constraints. After casting (4.1) as a MISDO, we adapt the framework from Chapter 2 to solve this problem to certifiable optimality in a tractable fashion. In addition, we propose eigenvalue bounds which improve the quality of the master problem formulation, and scalable semidefinite and second-order cone relaxations and randomized

rounding methods which supply certifiably near-optimal solutions when $p = 1000$s.

4.1 Background and Literature Review

Owing to sparse PCA's fundamental importance in a variety of applications including best subset selection [81], natural language processing [233], compressed sensing [61], and clustering [165] among others, three distinct classes of methods for addressing Problem (4.1) have arisen. Namely, (a) heuristic methods which obtain high-quality sparse PCs in an efficient fashion but do not supply guarantees on the quality of the solution, (b) convex relaxations which obtain certifiably near-optimal solutions by solving a convex relaxation and rounding, and (c) exact methods which obtain certifiably optimal solutions, albeit in exponential time.

Heuristic approaches

The importance of identifying a small number of interpretable principal components has been well-documented in the literature since the work of Hotelling [136] [see also 139], giving rise to many distinct heuristic approaches for obtaining high-quality solutions to Problem (4.1). Two interesting such approaches are to rotate dense principal components to promote sparsity [144, 201, 140], or apply an ℓ_1 penalty term as a convex surrogate to the cardinality constraint [141, 236]. Unfortunately, the former approach does not provide performance guarantees, while the latter approach still results in a non-convex optimization problem.

More recently, motivated by the need to rapidly obtain high-quality sparse principal components at scale, a wide variety of first-order heuristic methods have emerged. The first such *modern* heuristic was developed by Journée et al. [143], and involves combining the power method with thresholding and re-normalization steps. By pursuing similar ideas, several related methods have since been developed [see 224, 130, 202, 166, 228]. Unfortunately, while these methods are often very effective in practice, they sometimes badly fail to recover an optimal sparse principal component, and a practitioner using a heuristic method typically has no way of knowing when this has

occurred. Indeed, Berk and Bertsimas [20] recently compared 7 heuristic methods, including most of those reviewed here, on 14 instances of sparse PCA, and found that none of the heuristic methods successfully recovered an optimal solution in all 14 cases (i.e., no heuristic was right all the time).

Convex relaxations

Motivated by the shortcomings of heuristic approaches on high-dimensional data sets, and the successful application of semi-definite optimization in obtaining high-quality approximation bounds in other applications [see 124, 225], a variety of convex relaxations have been proposed for sparse PCA. The first such convex relaxation was proposed by d'Aspremont et al. [79], who reformulated sparse PCA as the rank-constrained mixed-integer semidefinite optimization problem (MISDO)

$$\max_{\boldsymbol{X} \succeq \mathbf{0}} \quad \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1,\ \|\boldsymbol{X}\|_0 \leq k^2,\ \operatorname{Rank}(\boldsymbol{X}) = 1, \tag{4.2}$$

where $\boldsymbol{X}$ models the outer product $\boldsymbol{x}\boldsymbol{x}^\top$. Note that, for a rank-one matrix $\boldsymbol{X}$, the constraint $\|\boldsymbol{X}\|_0 \leq k^2$ in (4.2) is equivalent to the constraint $\|\boldsymbol{x}\|_0 \leq k$ in (4.1), since a vector $\boldsymbol{x}$ is k-sparse if its outer product $\boldsymbol{x}\boldsymbol{x}^\top$ is k^2-sparse. After performing this reformulation, d'Aspremont et al. [79] relaxed both the cardinality and rank constraints and instead solved

$$\max_{\boldsymbol{X} \succeq \mathbf{0}} \quad \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1,\ \|\boldsymbol{X}\|_1 \leq k, \tag{4.3}$$

which supplies a valid upper bound on Problem (4.1)'s objective.

The semidefinite approach has since been refined in a number of follow-up works. Among others, d'Aspremont et al. [81], building upon the work of Ben-Tal and Nemirovski [18], proposed a different semidefinite relaxation which supplies a sufficient condition for optimality via the primal-dual KKT conditions, and d'Aspremont et al. [82] analyzed the quality of the semidefinite relaxation in order to obtain high-quality approximation bounds. A common theme in these approaches is that they require

solving large-scale semidefinite optimization problems. This presents difficulties for practitioners because state-of-the-art implementations of interior point methods such as `Mosek` require $O(p^6)$ memory to solve Problem (4.3), and therefore currently cannot solve instances of Problem (4.3) with $p \geq 300$ [see 23, for a recent comparison]. Techniques other than interior point methods, e.g., ADMM or augmented Lagrangian methods as reviewed in [169] could also be used to solve Problem (4.3), although they tend to require more runtime than IPMs to obtain a solution of a similar accuracy and be unstable for problem sizes where IPMs run out of memory [169].

A number of works have also studied the statistical estimation properties of Problem (4.3), by assuming an underlying probabilistic model. Among others, Amini and Wainwright [6] have demonstrated the asymptotic consistency of Problem (4.3) under a spiked covariance model once the number of samples used to generate the covariance matrix exceeds a certain threshold; see [219, 21, 221] for further results in this direction, [175] for a recent survey.

In an complementary direction, Dey et al. [84] has recently questioned the modeling paradigm of lifting $\boldsymbol{x}$ to a higher dimensional space by instead considering the following (tighter) relaxation of sparse PCA in the original problem space

$$\max_{\boldsymbol{x} \in \mathbb{R}^p} \quad \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 = 1, \|\boldsymbol{x}\|_1 \leq \sqrt{k}. \tag{4.4}$$

Interestingly, Problem (4.4)'s relaxation provides a $\left(1 + \sqrt{\frac{k}{k+1}}\right)^2$-factor bound approximation of Problem (4.1)'s objective, while Problem (4.3)'s upper bound may be exponentially larger in the worst case [6]. This additional tightness, however, comes at a price: Problem (4.4) is NP-hard to solve—indeed, providing a constant-factor guarantee on sparse PCA is NP-hard [167]—and thus (4.4) is best formulated as a MIO, while Problem (4.3) can be solved in polynomial time.

More recently, by building on the work of Kim and Kojima [147], Bertsimas and Cory-Wright [23] introduced a second-order cone relaxation of (4.2) which scales to $p = 1000s$, and matches the semidefinite bound after imposing a small number of cuts. Moreover, it typically supplies bound gaps of less than 5%. However, it does

not supply an *exact* certificate of optimality, which is often desirable.

A fundamental drawback of existing convex relaxation techniques is that they are not coupled with rounding schemes for obtaining high-quality feasible solutions. This is problematic, because optimizers are typically interested in obtaining high-quality solutions, rather than certificates. In this chapter, we take a step in this direction, by deriving new convex relaxations that naturally give rise to greedy and random rounding schemes. The fundamental point of difference between our relaxations and existing relaxations is that we derive our relaxations by rewriting sparse PCA as a MISDO and dropping an integrality constraint, rather than using ad-hoc techniques.

Exact methods

Motivated by the successful application of mixed-integer optimization for solving statistical learning problems such as best subset selection [27] and sparse classification [36], several exact methods for solving sparse PCA to certifiable optimality have been proposed. The first branch-and-bound algorithm for solving Problem (4.1) was proposed by Moghaddam et al. [176], by applying norm equivalence relations to obtain valid bounds. However, Moghaddam et al. [176] did not couple their approach with high-quality initial solutions and tractable bounds to prune partial solutions. Consequently, they could not scale their approach beyond $p = 40$.

A more sophisticated branch-and-bound scheme was recently proposed by Berk and Bertsimas [20], which couples tighter Gershgorin Circle Theorem bounds [135, Chapter 6] with a fast heuristic due to [228] to solve problems up to $p = 250$. However, their method cannot scale beyond $p = 100$s, because the bounds obtained are too weak to avoid enumerating a sizeable portion of the tree.

In Chapter 2, we developed a framework for reformulating convex mixed-integer optimization problems with logical constraints, and demonstrated that this framework allows a number of problems of practical relevance to be solved to certifiably optimality via a cutting-plane method. In this chapter, we build upon this work by reformulating Problem (4.1) as a *convex* mixed-integer semidefinite optimization problem, and leverage this reformulation to design a cutting-plane method which

solves sparse PCA to certifiable optimality. A key feature of our approach is that we need not solve any semidefinite subproblems. Rather, we use concepts from SDO to design a semidefinite-free approach which uses simple linear algebra techniques.

Concurrently to our initial submission of the paper this chapter is based upon (see [37]), Li and Xie [158] also attempted to reformulate sparse PCA as an MISDO, and proposed valid inequalities for strengthening their formulation and local search algorithms for obtaining high-quality solutions at scale. Our work differs in the following two ways. First, we propose strengthening the MISDO formulation using the Gershgorin circle theorem and demonstrate that this allows our MISDO formulation to scale to problems with $p = 100$s of features, while they do not, to our knowledge, solve any MISDOs to certifiable optimality where $p > 13$. Second, we develop tractable second-order cone relaxations and greedy rounding schemes which allow practitioners to obtain certifiably near optimal sparse principal components even in the presence of $p = 1,000$s of features. More remarkable than the differences between the works however is the similarities: more than 15 years after d'Aspremont et al. [79]'s landmark paper first appeared, both works proposed reformulating sparse PCA as an MISDO less than a week apart. In our view, this demonstrates that the ideas contained in both works transcend sparse PCA, and can perhaps be applied to other problems in the optimization literature which have not yet been formulated as MISDOs.

4.2 A Mixed-Integer Semidefinite Reformulation

In this section, we reformulate Problem (4.1) as a convex mixed-integer semidefinite convex optimization problem, before supplying a formal proof that our reformulation is indeed equivalent to Problem (4.1). Let us introduce a rank one positive semidefinite matrix $\boldsymbol{X}$ which models the outer product $\boldsymbol{x}\boldsymbol{x}^\top$, and rewrite Problem (4.1) as the following non-convex problem:

$$\max_{\boldsymbol{X} \in S_+^p} \quad \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \quad \text{tr}(\boldsymbol{X}) = 1, \ \text{Card}(\boldsymbol{X}) \leq k^2, \ \text{Rank}(\boldsymbol{X}) = 1.$$

Starting from the rank-constrained SDO formulation (4.2), we introduce binary variables z_i to model whether $X_{i,j}$ is non-zero, via the logical constraint $X_{i,j} = 0$ if $z_i = 0$; note that we need not require that $X_{i,j} = 0$ if $z_j = 0$, since $\boldsymbol{X}$ is a symmetric matrix. By enforcing the logical constraint via $-M_{i,j}z_i \leq X_{i,j} \leq M_{i,j}z_i$ for sufficiently large $M_{i,j} > 0$, Problem (4.2) becomes

$$\max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\in S^p_+} \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle$$
$$\text{s.t. } \mathrm{tr}(\boldsymbol{X}) = 1,\ -M_{i,j}z_i \leq X_{i,j} \leq M_{i,j}z_i\ \forall i,j \in [p],\ \mathrm{rank}(\boldsymbol{X}) = 1.$$

To obtain a MISDO, we omit the rank constraint. In general, omitting a rank constraint generates a relaxation and induces a loss of optimality. Remarkably, this omission is without loss of optimality in this case. Indeed, the objective is convex and therefore some rank-one extreme matrix $\boldsymbol{X}$ is optimal. We formalize this observation in the following theorem; note that a similar result—although in the context of computing Restricted Isometry constants and with a different proof—exists [117]:

Theorem 4.1. *Problem* (4.1) *attains the same optimal objective value as the following problem, where we associate a dual multiplier with each constraint in square brackets:*

$$\begin{aligned}
\max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\in S^p_+} \quad & \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle & \\
s.t. \quad & \mathrm{tr}(\boldsymbol{X}) = 1 & [\lambda], \\
& X_{i,j} \leq M_{i,j}z_i & [\alpha^+_{i,j}]\ \forall i,j \in [p], \\
& -X_{i,j} \leq M_{i,j}z_i & [\alpha^-_{i,j}]\ \forall i,j \in [p],
\end{aligned} \tag{4.5}$$

and $M_{i,i} = 1$, $M_{i,j} = \frac{1}{2}$ *if* $j \neq i$.

Remark 5. *If we set* $M_{i,j} = 1\ \forall i,j \in [p]$ *in Problem* (4.5) *then the optimal value of the continuous relaxation is* $\lambda_{\max}(\boldsymbol{\Sigma})$*, i.e., the same value as if we did not impose a cardinality constraint. Indeed, letting* $\boldsymbol{x}$ *be a leading eigenvector of the unconstrained problem (where* $\|\boldsymbol{x}\|_2 = 1$*), we can set* $z_i = |x_i| \geq |x_i||x_j|$ *and* $X_{i,j} = x_i x_j$*, meaning* $\sum_i z_i = \|\boldsymbol{x}\|_1 \leq k$ *and thus* $(\boldsymbol{X}, \boldsymbol{z})$ *solves this continuous relaxation. Therefore,*

setting $M_{i,j} = \frac{1}{2}$ if $j \neq i$ is a necessary condition to obtain non-trivial relaxations.

Proof. It suffices to demonstrate that for any feasible solution to (4.1) we can construct a feasible solution to (4.5) with an equal or greater payoff, and vice versa.

- Let $\boldsymbol{x} \in \mathbb{R}^p$ be a feasible solution to (4.1). Then, since $\|\boldsymbol{x}\|_1 \leq \sqrt{k}$, $(\boldsymbol{X} := \boldsymbol{x}\boldsymbol{x}^\top, \boldsymbol{z})$ is a feasible solution to (4.5) with equal cost, where $z_i = 1$ if $|x_i| > 0$, $z_i = 0$ otherwise.
- Let $(\boldsymbol{X}, \boldsymbol{z})$ be a feasible solution to Problem (4.5), and let $\boldsymbol{X} = \sum_{i=1}^p \sigma_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$ be a Cholesky decomposition of $\boldsymbol{X}$, where $\boldsymbol{e}^\top \boldsymbol{\sigma} = 1, \boldsymbol{\sigma} \geq \boldsymbol{0}$, and $\|\boldsymbol{x}_i\|_2 = 1 \; \forall i \in [p]$. Observe that $\|\boldsymbol{x}_i\|_0 \leq k \; \forall i \in [p]$, since we can perform the Cholesky decomposition on the submatrix of $\boldsymbol{X}$ induced by $\boldsymbol{z}$, and "pad" out the remaining entries of each $\boldsymbol{x}_i$ with 0s to obtain the decomposition of $\boldsymbol{X}$. Therefore, let us set $\hat{\boldsymbol{x}} := \arg\max_i [\boldsymbol{x}_i^\top \boldsymbol{\Sigma} \boldsymbol{x}_i]$. Then, $\hat{\boldsymbol{x}}$ is a feasible solution to (4.1) with an equal or greater payoff.

Finally, we let $M_{i,i} = 1$, $M_{i,j} = \frac{1}{2}$ if $i \neq j$, as the 2×2 minors imply $X_{i,j}^2 \leq X_{i,i} X_{j,j} \leq \frac{1}{4}$ whenever $i \neq j$ [c.f. 117, Lemma 1]. □

Theorem 4.1 reformulates Problem (4.1) as a mixed-integer SDO. Therefore, we can solve Problem (4.5) using general branch-and-cut techniques for semidefinite optimization problems [see 118, 149]. However, this approach is not scalable, as it comprises solving a large number of semidefinite subproblems and the community does not know how to efficiently warm-start IPMs for SDOs.

We now invoke Theorem 2.1 to propose a saddle-point reformulation of Problem (4.5) which avoids the computational difficulty in solving a large number of SDOs by exploiting problem structure, as we will show in Theorem 4.2. Our reformulation allows us to propose a branch-and-cut method which solves each subproblem using linear algebra techniques. We have the following result:

Proposition 4.1. *Problem* (4.5) *attains the same optimal value as:*

$$\begin{aligned} \max_{\boldsymbol{z}\in\{0,1\}^p:\ \boldsymbol{e}^\top\boldsymbol{z}\leq k} \quad & f(\boldsymbol{z}) \\ \textit{where} \quad & f(\boldsymbol{z}) := \min_{\lambda\in\mathbb{R},\boldsymbol{\alpha}\in\mathbb{R}^{p\times p}} \quad \lambda + \sum_{i=1}^p z_i \left(|\alpha_{i,i}| + \frac{1}{2}\sum_{j=1,j\neq i}^p |\alpha_{i,j}| \right) \\ & \textit{s.t.} \quad \lambda\mathbb{I} + \boldsymbol{\alpha} \succeq \boldsymbol{\Sigma}, \end{aligned} \tag{4.6}$$

where the variables in the inner minimization problem precisely correspond to the dual multipliers associated with the maximization problem (4.5)*.*

Proof. Let us introduce auxiliary variables $U_{i,j}$ to model the absolute value of $X_{i,j}$ and rewrite the inner optimization problem of (4.5) as

$$\begin{aligned} f(\boldsymbol{z}) := \quad & \max_{\boldsymbol{X}\succeq\boldsymbol{0},\boldsymbol{U}} && \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle \\ & \text{s.t.} \quad \operatorname{tr}(\boldsymbol{X}) = 1, && [\lambda] \\ & && U_{i,j} \leq M_{i,j} z_i \qquad \forall i,j \in [p], \quad [\sigma_{i,j}] \\ & && |X_{i,j}| \leq U_{i,j} \qquad \forall i,j \in [p], \quad [\alpha_{i,j}] \\ & && \sum_{j=1}^p U_{i,j} \leq \sqrt{k} z_i \qquad \forall i \in [p], \quad [\beta_i] \end{aligned} \tag{4.7}$$

where we associate dual constraint multipliers with primal constraints in square brackets. For $\boldsymbol{z}$ such that $\boldsymbol{e}^\top\boldsymbol{z} \geq 1$, the maximization problem induced by $f(\boldsymbol{z})$ satisfies Slater's condition [see, e.g., 54, Chapter 5.2.3], strong duality applies and leads to

$$\begin{aligned} f(\boldsymbol{z}) = \quad & \min_{\substack{\lambda \\ \boldsymbol{\sigma},\boldsymbol{\alpha},\boldsymbol{\beta}\geq\boldsymbol{0}}} \quad \lambda + \sum_{i,j} \sigma_{i,j} M_{i,j} z_i + \sum_{i=1}^p \beta_i \sqrt{k} z_i \\ & \text{s.t.} \quad \lambda\mathbb{I} + \boldsymbol{\alpha} \succeq \boldsymbol{\Sigma}, |\alpha_{i,j}| \leq \sigma_{i,j} + \beta_i. \end{aligned}$$

We eliminate $\boldsymbol{\sigma}$ above by optimizing over $\sigma_{i,j}$ and setting $\sigma^\star_{i,j} = \max(0, |\alpha_{i,j}| - \beta_i)$.

For $\boldsymbol{z} = \boldsymbol{0}$, the primal subproblem is infeasible and the dual subproblem has objective $-\infty$, but this can safely be ignored since $\boldsymbol{z} = \boldsymbol{0}$ is certainly suboptimal. □

Proposition 4.1 reformulates (4.1) as a special case of the framework developed in Chapter 2. Therefore, we can solve this problem to certifiable optimality using the cutting-plane method laid out in Algorithm 2.1 (this is a maximization, rather than a minimization, problem, so the algorithm needs to be adjusted accordingly).

A key drawback in applying Algorithm 2.1 "out-of-the-box" is that it involves solving a large number of semidefinite subproblems. This is not a good idea in practice, because semidefinite optimization problems are expensive to solve. Therefore, we now derive a computationally efficient subproblem strategy which crucially does not require solving *any* semidefinite programs. Formally, we have the following result, a proof of which can be found in [37]:

Theorem 4.2. *For any $\boldsymbol{z} \in \{0,1\}^p$, optimal dual variables in* (4.6) *are*

$$\lambda = \lambda_{\max}\left(\boldsymbol{\Sigma}_{1,1}\right), \boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_{1,1} & \boldsymbol{\alpha}_{1,2} \\ \boldsymbol{\alpha}_{1,2}^\top & \boldsymbol{\alpha}_{2,2} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{2,2} - \lambda \mathbb{I} + \boldsymbol{\Sigma}_{1,2}^\top \left(\lambda \mathbb{I} - \boldsymbol{\Sigma}_{1,1}\right)^\dagger \boldsymbol{\Sigma}_{1,2} \end{pmatrix}, \tag{4.8}$$

where $\lambda_{\max}(\cdot)$ denotes the leading eigenvalue of a matrix, $\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_{1,1} & \boldsymbol{\alpha}_{1,2} \\ \boldsymbol{\alpha}_{1,2}^\top & \boldsymbol{\alpha}_{2,2} \end{pmatrix}$ is a decomposition such that $\boldsymbol{\alpha}_{1,1}$ (resp. $\boldsymbol{\alpha}_{2,2}$) denotes the entries of $\boldsymbol{\alpha}$ where $z_i = z_j = 1$ ($z_i = z_j = 0$); $\boldsymbol{\Sigma}$ is similar.

Remark 6. *By Theorem 4.2, we can obtain an optimal set of dual variables by computing the leading eigenvalue of $\boldsymbol{\Sigma}_{1,1}$ and solving a linear system. This justifies our claim that we need not solve any SDOs in our implementation of Algorithm 2.1.*

4.3 Sparse PCA Under Ridge Regularization

In this section, we explore enforcing the logical relation $X_{i,j} = 0$ if $z_i = 0$ using ridge, rather than big-M regularization, as proposed in Chapter 2.

By following the analysis in Chapter 2, and also imposing the constraint $X_{i,j} = 0$ if $z_j = 0$ (unlike the big-M case, imposing both logical constraints is helpful for

developing our subproblem strategy) we obtain the following problem:

$$\begin{aligned} \max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\in S^p_+} \quad & \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle - \frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \operatorname{tr}(\boldsymbol{X}) = 1, X_{i,j} = 0 \text{ if } z_i = 0 \text{ or } z_j = 0 \ \forall i,j \in [p], \end{aligned} \tag{4.9}$$

which, by strong semidefinite duality—which holds for the inner maximization problem for any $\boldsymbol{z} \neq \boldsymbol{0}$ since the inner problem has non-empty relative interior with respect to the non-affine constraints [see, e.g, 54, Chapter 5.2.3]—is equivalent to the saddle-point problem

$$\max_{\boldsymbol{z}\in\{0,1\}^p:\ \boldsymbol{e}^\top\boldsymbol{z}\leq k} \quad f(\boldsymbol{z}) \tag{4.10}$$

$$\text{where} \quad f(\boldsymbol{z}) := \min_{\lambda\in\mathbb{R},\boldsymbol{\alpha}\in\mathbb{R}^{p\times p},\boldsymbol{\beta}\in\mathbb{R}^{p\times p}} \quad \lambda + \frac{\gamma}{2}\sum_{i=1}^p z_i \sum_{j=1}^p (\alpha_{i,j} + \beta_{j,i})^2 \tag{4.11}$$

$$\text{s.t.} \quad \lambda\mathbb{I} + \boldsymbol{\alpha} + \boldsymbol{\beta} \succeq \boldsymbol{\Sigma},$$

and can be addressed by a cutting-plane method such as Algorithm 2.1.

It should be noted however that Problem (4.9) does not supply a rank-one matrix $\boldsymbol{X}^\star$, due to the ridge regularizer. Therefore, under Frobenius norm regularization, we first solve Problem (4.10) to obtain an optimal set of indices $\boldsymbol{z}$, and subsequently solve for an optimal $\boldsymbol{X}$ for this $\boldsymbol{z}$ in (4.5).

This perturbation strategy necessarily gives rise to some loss of optimality. However, this loss can be bounded. Indeed, the difference in optimal objectives between Problems (4.5) and (4.9) is at most $\frac{1}{2\gamma}\|\boldsymbol{X}^\star\|_F^2$, where $\boldsymbol{X}^\star$ is an optimal $\boldsymbol{X}$ in Problem (4.5). Moreover, since

$$\frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2 = \frac{1}{2\gamma}\sum_i\sum_j X_{i,j}^2 \leq \frac{1}{2\gamma}\sum_i X_{i,i}\sum_j X_{j,j} = \frac{1}{2\gamma},$$

where the inequality follows from the 2×2 minors in $\boldsymbol{X} \succeq \boldsymbol{0}$ [c.f. 23, Proposition 3], the difference in objectives between Problems (4.5) and (4.9) is at most $\frac{1}{2\gamma}$ and becomes negligible as $\gamma \to \infty$.

We will make use of both types of regularization in our algorithmic results, and therefore derive an efficient subproblem strategy under ridge regularization as well:

Theorem 4.3. *For any $\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k$, optimal dual variables in* (4.10) *are*

$$\lambda = \arg\min_{\lambda} \left\{ \lambda + \frac{\gamma}{2} \|(\boldsymbol{\Sigma}_{1,1} - \lambda \mathbb{I})_+\|_F^2 \right\}, \tag{4.12}$$
$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_{1,1} & \boldsymbol{\alpha}_{1,2} \\ \boldsymbol{\alpha}_{2,1} & \boldsymbol{\alpha}_{2,2} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} (\boldsymbol{\Sigma}_{1,1} - \lambda \mathbb{I})_+ & \boldsymbol{0} \\ 2\boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} - \lambda \mathbb{I} \end{pmatrix}, \boldsymbol{\beta} = \boldsymbol{\alpha}^\top$$

where $(\boldsymbol{X})_+$ denotes the positive semidefinite component of $\boldsymbol{X}$, i.e., if $\boldsymbol{X} = \sum_{i=1}^p \sigma_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is an eigendecomposition of $\boldsymbol{X}$ then $(\boldsymbol{X})_+ = \sum_{i=1}^p \max(\sigma_i, 0) \boldsymbol{x}_i \boldsymbol{x}_i^\top$, $\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_{1,1} & \boldsymbol{\alpha}_{1,2} \\ \boldsymbol{\alpha}_{1,2}^\top & \boldsymbol{\alpha}_{2,2} \end{pmatrix}$ is a decomposition of $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha}_{1,1}$ (resp. $\boldsymbol{\alpha}_{2,2}$) denotes the entries of $\boldsymbol{\alpha}$ where $z_i = z_j = 1$ (resp. $z_i = z_j = 0$), and $\boldsymbol{\beta}$, $\boldsymbol{\Sigma}$ are similar.

Proof. Observe that if $z_i = 0$ then $\alpha_{i,j}$ does not contribute to the objective, while if $z_j = 0$, $\beta_{i,j}$ does not contribute to the objective. Therefore, if $z_i = 0$ and $z_j = 1$ we can set $\beta_{i,j} = 0$ and $\alpha_{i,j}$ to be any dual-feasible value, and vice versa. As a result, it suffices to solve $\boldsymbol{\alpha}_{1,1}$, $\boldsymbol{\beta}_{1,1}, \lambda$, as we can subsequently pick the remaining components of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in order that they are feasible and satisfy the aforementioned condition. Moreover, observe that we can set $\boldsymbol{\alpha} = \boldsymbol{\beta}^\top$ without loss of generality, since, in the derivation of the dual problem, $\boldsymbol{\alpha}$ is a matrix of dual variables associated with a constraint of the form $\boldsymbol{V} = \mathrm{Diag}(\boldsymbol{z})\boldsymbol{X}$, while $\boldsymbol{\beta}$ is a matrix of dual variables associated with a constraint of the form $\boldsymbol{V} = \boldsymbol{X}\mathrm{Diag}(\boldsymbol{z})$ [c.f. 33, Theorem 1].

Let us substitute $\hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}_{1,1} + \boldsymbol{\beta}_{1,1}$ and consider the reduced inner dual problem

$$\min_{\hat{\boldsymbol{\alpha}}, \lambda} \quad \lambda + \frac{\gamma}{2} \|\hat{\boldsymbol{\alpha}}\|_F^2 \text{ s.t. } \lambda \mathbb{I} + \hat{\boldsymbol{\alpha}} \succeq \boldsymbol{\Sigma}_{1,1}.$$

In this problem, for any λ, an optimal choice of $\hat{\boldsymbol{\alpha}}$ is given by projecting (with respect to the Frobenius distance) onto a positive semidefinite cone centered at $\boldsymbol{\Sigma}_{1,1} - \lambda \mathbb{I}$. Therefore, an optimal choice of $\hat{\boldsymbol{\alpha}}$ is given by $\hat{\boldsymbol{\alpha}} = (\boldsymbol{\Sigma}_{1,1} - \lambda \mathbb{I})_+$ [see 54, Chapter 8.1.1]. Moreover, we have verified that an optimal choice of λ is indeed given by

solving (4.12), and therefore the result follows. □

We now derive an efficient technique for computing an optimal λ in (4.12):

Corollary 4.1. *Let $\mathbf{\Sigma}_{1,1}$ be a submatrix containing the entries of $\mathbf{\Sigma}$ where $z_i = z_j = 1$, and let $\sigma_1 \geq \ldots \geq \sigma_k$ denote the ordered eigenvalues of $\mathbf{\Sigma}_{1,1}$. Then, any λ which solves the following optimization problem is an optimal dual variable in* (4.12)*:*

$$\min_{\lambda \in \mathbb{R}, \boldsymbol{\theta} \in \mathbb{R}^k_+} \quad \lambda + \frac{\gamma}{2} \sum_{i=1}^{k} \theta_i^2 \ \textit{s.t.}\ \boldsymbol{\theta} \geq \boldsymbol{\sigma} - \lambda \boldsymbol{e}.$$

Moreover, suppose $\sigma_l \geq \lambda \geq \sigma_{l+1}$, where $\lambda := \frac{1}{\gamma l} + \frac{1}{l} \sum_{i=1}^{l} \sigma_i$. Then λ is optimal.

Proof. Recall from Theorem 4.3 that any λ solving $\arg\min_\lambda \left\{ \lambda + \frac{\gamma}{2} \|(\mathbf{\Sigma}_{1,1} - \lambda \mathbb{I})_+\|_F^2 \right\}$ is optimal. Since $\| \left(\Sigma_{1,1} - \lambda \mathbb{I}\right)_+ \|_F^2 = \sum_{i=1}^{k} (\sigma_i - \lambda)_+^2$, this is equivalent to solving

$$\arg\min_{\lambda} \left\{ \lambda + \frac{\gamma}{2} \sum_{i=1}^{k} (\sigma_i - \lambda)_+^2 \right\}.$$

The result follows by solving the latter problem. □

As the quadratic optimization problem has a piecewise convex objective, some optimal choice of λ is either an endpoint of an interval $[\sigma_i, \sigma_{i+1}]$ or a solution of the form $\lambda := \frac{1}{\gamma l} + \frac{1}{l} \sum_{i=1}^{l} \sigma_i$ for some l. Therefore, we need only check at most $2k$ different values of λ. Moreover, since the objective function is convex in λ, we can check these points via bisection search, in $O(\log k)$ time. Alternatively, we can cast the subproblem as a second-order cone problem and invoke a conic solver, e.g., `Mosek`.

Observe that the value of the regularization term is always at least $\frac{1}{2\gamma k}$, since

$$\min_{\boldsymbol{X} \succeq \mathbf{0}} \ \frac{1}{2\gamma} \|\boldsymbol{X}\|_F^2 \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1$$

is minimized by setting $\boldsymbol{X} = \frac{1}{n} \boldsymbol{e}\boldsymbol{e}^\top$, and we have the constraint $X_{i,j} = 0$ if $z_i = 0$, $\boldsymbol{e}^\top \boldsymbol{z} \leq k$. Therefore, we can subtract $\frac{1}{2\gamma k}$ from our bound under ridge regularization.

4.4 Strengthening the Master Problem

As Algorithm 2.1's rate of convergence rests heavily upon its implementation, we now propose a practical technique for accelerating Algorithm 2.1. Namely, we strengthen the master problem by imposing bounds from the Gershgorin circle theorem. Formally, we have the following result, which can be deduced from [134, Theorem 6.1.1]:

Theorem 4.4. *For any vector $\boldsymbol{z} \in \{0,1\}^p$ we have the following upper bound:*

$$f(\boldsymbol{z}) \leq \max_{i \in [p]: z_i = 1} \sum_{j \in [p]} z_j |\Sigma_{i,j}|.$$

Observe that this bound cannot be used to *directly* strengthen Algorithm 2.1's master problem, since the bound is not convex in $\boldsymbol{z}$. Nonetheless, it can be successfully applied if we (a) impose a big-M assumption on Problem (4.1)'s optimal objective and (b) introduce p additional binary variables $\boldsymbol{s} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{s} = 1$ which model whether the ith Gershgorin disc is active; recall that each eigenvalue is contained in the union of the discs. Formally, we impose the following valid inequalities:

$$\exists \boldsymbol{s} \in \{0,1\}^p : \theta \leq \sum_{i \in [p]} z_i |\Sigma_{i,j}| + M(1 - s_j) \ \forall j \in [p], \boldsymbol{e}^\top \boldsymbol{s} = 1, \boldsymbol{s} \leq \boldsymbol{z}, \tag{4.13}$$

where θ is the epigraph variable maximized in the master problem stated in Algorithm 2.1, and M is an upper bound on the sum of the k largest absolute entries in any column of $\boldsymbol{\Sigma}$. Note that we set $\boldsymbol{s} \leq \boldsymbol{z}$ since if $z_i = 0$ the ith column of $\boldsymbol{\Sigma}$ does not feature in the relevant submatrix of $\boldsymbol{\Sigma}$. In the above inequalities, a valid M is given by any bound on the optimal objective. Since Theorem (4.4) supplies one such bound for any given $\boldsymbol{z}$, we can compute

$$M := \max_{j \in [p]} \max_{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k} \sum_{i \in [p]} z_i |\Sigma_{i,j}|, \tag{4.14}$$

which can be done in $O(p^2)$ time.

To further improve Algorithm 2.1, we also make use of the Gershgorin circle

theorem before generating each cut. Namely, at a given node in a branch-and-bound tree, there are indices i where z_i has been fixed to 1, indices i where z_i has been fixed to 0, and indices i where z_i has not yet been fixed. Accordingly, we compute the worst-case Gershgorin bound—by taking the worst-case bound over each index j such that z_j has not yet been fixed to 0, i.e.,

$$\max_{j:z_j\neq 0}\left\{\max_{\boldsymbol{s}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{s}\leq k}\left\{\sum_{i\in[p]}s_i|\Sigma_{i,j}|\text{ s.t. }s_i=0\text{ if }z_i=0, s_i=1\text{ if }z_i=1\right\}\right\}.$$

If this bound is larger than our incumbent, then we generate an outer-approximation cut, otherwise the entire subtree rooted at this node does not contain an optimal solution and we use instruct the solver to avoid exploring this node via a `callback`.

Our numerical results in Chapter 4.6 echo the empirical findings of Berk and Bertsimas [20] and indicate that Algorithm 2.1 performs substantially better when the Gershgorin bound is supplied in the master problem. Therefore, it is interesting to theoretically investigate the tightness, or at least the quality, of Gershgorin's bound. We supply some results in this direction in the following proposition:

Proposition 4.2. *Suppose that $\boldsymbol{\Sigma}$ is a scaled diagonally dominant matrix as defined by [47], i.e., there exists some vector $\boldsymbol{d}>0$ such that*

$$d_i\Sigma_{i,i}\geq\sum_{j\in[p]:j\neq i}d_j|\Sigma_{i,j}|\ \forall i\in[p].$$

Then, letting $\rho:=\max_{i,j\in[p]}\{\frac{d_i}{d_j}\}$, the Gershgorin circle theorem provides a $(1+\rho)$-factor approximation, i.e.,

$$f(\boldsymbol{z})\leq\max_{j\in[p]}\{\sum_{i\in[p]}z_i|\Sigma_{i,j}|\}\leq(1+\rho)f(\boldsymbol{z})\quad\forall\boldsymbol{z}\in\{0,1\}^p. \tag{4.15}$$

Proof. Scaled diagonally dominant matrices have scaled diagonally dominant princi-

pal minors—this is trivially true because

$$d_i\Sigma_{i,i} \geq \sum_{j\in[p]:j\neq i} d_j|\Sigma_{i,j}| \ \forall i \in [p] \implies d_i\Sigma_{i,i} \geq \sum_{j\in[p]:j\neq i} d_j z_j|\Sigma_{i,j}| \ \forall i \in [p] : z_i = 1$$

for the same vector $\boldsymbol{d} > \mathbf{0}$ and therefore the following chain of inequalities holds

$$\begin{aligned} f(\boldsymbol{z}) \leq & \max_{j\in[p]}\{\sum_{i\in[p]} z_i|\Sigma_{i,j}|\} = \max_{j\in[p]}\{z_j\Sigma_{j,j} + \sum_{i\in[p]:j\neq i} z_i|\Sigma_{i,j}|\} \\ & \leq \max_{j\in[p]}\{z_j\Sigma_{j,j} + \sum_{i\in[p]:j\neq i} \rho\frac{d_i}{d_j} z_i|\Sigma_{i,j}|\} \\ & \leq (1+\rho)\max_{j\in[p]}\{z_j\Sigma_{j,j}\} \leq (1+\rho) f(\boldsymbol{z}) \quad \forall \boldsymbol{z} \in \{0,1\}^p, \end{aligned}$$

where the second inequality follows because $\rho \geq \frac{d_i}{d_j}$, the third follows from the scaled diagonal dominance of the principal submatrices of $\boldsymbol{\Sigma}$, and the fourth holds because the leading eigenvalue of a PSD matrix is at least as large as each diagonal entry. □

To make clear how our numerical success depends upon Theorem 4.4, our results in Section 4.6 present implementations of Algorithm 2.1 with and without the bound.

Beyond Gershgorin: Strengthening via Brauer's Ovals of Cassini

Given the relevance of Gershgorin's bound, we propose, in this section, a stronger —yet more expensive to implement— upper bound, based on an generalization of the Gershgorin Circle theorem, namely Brauer's ovals of Cassini.

First, we derive a new upper-bound on $f(\boldsymbol{z})$ that is at least as strong as the one presented in Theorem 4.4 and often strictly stronger [135, Chapter 6]:

Theorem 4.5. *For any vector $\boldsymbol{z} \in \{0,1\}^p$, we have the following upper bound:*

$$f(\boldsymbol{z}) \leq \max_{i,j\in[p]:i>j,z_i=z_j=1} \left\{ \frac{\Sigma_{i,i}+\Sigma_{j,j}}{2} + \frac{\sqrt{(\Sigma_{i,i}-\Sigma_{j,j})^2 + 4R_i(\boldsymbol{z})R_j(\boldsymbol{z})}}{2} \right\}, \qquad (4.16)$$

where $R_i(\boldsymbol{z}) := \sum_{j\in[p]:j\neq i} z_j|\Sigma_{i,j}|$ is the absolute sum of off-diagonal entries in the ith column of the submatrix of $\boldsymbol{\Sigma}$ induced by $\boldsymbol{z}$.

Proof. Let us first recall that, per Brauer [56]'s original result, all eigenvalues of a matrix $\mathbf{\Sigma} \in S_+^p$ are contained in the union of the following $p(p-1)/2$ ovals of Cassini:

$$\bigcup_{i\in[p],j\in[p]:i<j} \{\lambda \in \mathbb{R}_+ : |\lambda - \Sigma_{i,i}||\lambda - \Sigma_{j,j}| \leq R_i R_j\},$$

where $R_i := \sum_{j\in[p]:j\neq i} |\Sigma_{i,j}|$ is the absolute sum of off-diagonal entries in the ith column of $\mathbf{\Sigma}$. Next, let us observe that, if λ is a dominant eigenvalue of a PSD matrix $\mathbf{\Sigma}$ then $\lambda \geq \Sigma_{i,i}\ \forall i$ and, in the (i,j)th oval, the bound reduces to

$$\lambda^2 - \lambda(\Sigma_{i,i} + \Sigma_{j,j}) + \Sigma_{i,i}\Sigma_{j,j} - R_i R_j \leq 0, \tag{4.17}$$

which, by the quadratic formula, implies an upper bound is $\frac{\Sigma_{i,i}+\Sigma_{j,j}}{2} + \frac{\sqrt{(\Sigma_{i,i}-\Sigma_{j,j})^2+4R_iR_j}}{2}$. The result follows because if $z_i = 0$ the ith row of $\mathbf{\Sigma}$ cannot be used to bound $f(\boldsymbol{z})$. □

Theorem 4.5's inequality can be enforced numerically as mixed-integer second order cone constraints. Indeed, the square root term in (4.16) can be modeled using second-order cone, and the bilinear terms only involve binary variables and can be linearized. Completing the square in Equation (4.17), (4.16) is equivalent to the following system of $p(p-1)/2$ mixed-integer second-order cone inequalities:

$$\left(\theta - \frac{1}{2}(\Sigma_{i,i} + \Sigma_{j,j})\right)^2 \leq \sum_{s,t\in[p]:s\neq i,t\neq j} W_{s,t}|\Sigma_{i,s}\Sigma_{j,t}| - \frac{3}{4}\Sigma_{i,i}\Sigma_{j,j} + M(1 - s_{i,j}),$$

$$\sum_{i,j\in[p]:i<j} s_{i,j} = 1, s_{i,j} \leq \min(z_i, z_j),$$

$$s_{i,j} \in \{0,1\} \quad \forall i,j \in [p] : i < j.$$

where $W_{i,j} = z_i z_j$ is a product of binary variables which can be modeled using, e.g., the McCormick inequalities $\max(0, z_i + z_j - 1) \leq W_{i,j} \leq \min(z_i, z_j)$, and M is an upper bound on the right-hand-side of the inequality for any $i, j : i \neq j$, which can be computed in $O(p^3)$ time in much the same manner as a big-M constant was computed in the previous section. Note that we do not make use of these inequalities directly in our numerical experiments, due to their high computational cost. However, an

interesting extension would be to introduce the binary variables dynamically, via branch-and-cut-and-price [12].

Since the bound derived from the ovals of Cassini (Theorem 4.5) is at least as strong as the Gershgorin circle's one (Theorem 4.4), it satisfies the same approximation guarantee (Proposition 4.2). In particular, it is tight when $\mathbf{\Sigma}$ is diagonal and provides a 2−factor approximation for diagonally dominant matrices. Actually, we now prove a stronger result and demonstrate that Theorem 4.5 provides a 2−factor bound on $f(\boldsymbol{z})$ for doubly diagonally dominant matrices—a broader class of matrices than diagonally dominant matrices [see 157, for a general theory]:

Proposition 4.3. *Let $\mathbf{\Sigma} \in S_+^p$ be a doubly diagonally dominant matrix, i.e.,*

$$\Sigma_{i,i}\Sigma_{j,j} \geq R_i R_j \ \forall i, j \in [p] : i > j,$$

where $R_i := \sum_{j\in[p]:j\neq i} |\Sigma_{i,j}|$ is the sum of the off-diagonal entries in the ith column of $\mathbf{\Sigma}$. Then, we have that

$$f(\boldsymbol{z}) \leq \max_{i,j\in[p]:i>j,z_i=z_j=1} \left\{ \frac{\Sigma_{i,i}+\Sigma_{j,j}}{2} + \frac{\sqrt{(\Sigma_{i,i}-\Sigma_{j,j})^2 + 4R_i(\boldsymbol{z})R_j(\boldsymbol{z})}\}}{2} \right\} \leq 2f(\boldsymbol{z}). \tag{4.18}$$

Proof. Observe that if $\Sigma_{i,i}\Sigma_{j,j} \geq R_i R_j$ then

$$\sqrt{(\Sigma_{i,i}-\Sigma_{j,j})^2 + 4R_i R_j} \leq \sqrt{(\Sigma_{i,i}-\Sigma_{j,j})^2 + 4\Sigma_{i,i}\Sigma_{j,j}} = \Sigma_{i,i}+\Sigma_{j,j}.$$

The result then follows in essentially the same fashion as Proposition 4.2. □

4.5 Convex Relaxations and Rounding Methods

For large-scale instances, high-quality solutions can be obtained by solving a convex relaxation of Problem (4.5) and rounding the optimal solution. Therefore, we first propose relaxing $\boldsymbol{z} \in \{0,1\}^p$ in (4.5) to $\boldsymbol{z} \in [0,1]^p$ and applying a greedy rounding

scheme. We then further tighten this relaxation using second-order cone constraints.

A Boolean relaxation and a greedy rounding method

We first consider a Boolean relaxation of (4.5), which we obtain by relaxing $\boldsymbol{z} \in \{0,1\}^p$ to $\boldsymbol{z} \in [0,1]^p$. This gives $\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} f(\boldsymbol{z})$, i.e,

$$\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\succeq\boldsymbol{0}} \langle\boldsymbol{\Sigma},\boldsymbol{X}\rangle \text{ s.t. } \mathrm{tr}(\boldsymbol{X})=1, |X_{i,j}| \leq M_{i,j} z_i \ \forall i,j\in[p]. \tag{4.19}$$

A useful strategy for obtaining a high-quality feasible solution is to solve (4.19) and set $z_i = 1$ for k indices corresponding to the largest $\boldsymbol{z}_j$'s in (4.19). We formalize this in Algorithm 4.1.

Algorithm 4.1 A greedy rounding method for Problem (4.1)

Require: Covariance matrix $\boldsymbol{\Sigma}$, sparsity parameter k
Compute $\boldsymbol{z}^\star$ solution of (4.19) or (4.20)
Construct $\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top\boldsymbol{z} = k$ such that $z_i \geq z_j$ if $z_i^\star \geq z_j^\star$.
Compute $\boldsymbol{X}$ solution of

$$\max_{\boldsymbol{X}\in S_+^p} \langle\boldsymbol{\Sigma},\boldsymbol{X}\rangle \text{ s.t. } \mathrm{tr}(\boldsymbol{X})=1, X_{i,j}=0 \text{ if } z_i z_j = 0 \quad \forall i,j\in[p].$$

return $\boldsymbol{z}, \boldsymbol{X}$.

Remark 7. *Our numerical results in Chapter 4.6 reveal that explicitly imposing a PSD constraint on $\boldsymbol{X}$ in the relaxation* (4.19)*—or the ones derived later in the following section—prevents our approximation algorithm from scaling to larger problem sizes than the exact Algorithm 2.1 can already solve. Therefore, to improve scalability, the semidefinite cone can be safely approximated via its second-order cone relaxation, $X_{i,j}^2 \leq X_{i,i}X_{j,j}\ \forall i,j\in[p]$, plus a small number of cuts of the form $\langle\boldsymbol{X}, \boldsymbol{x}_t\boldsymbol{x}_t^\top\rangle \geq 0$ as presented in [23].*

Remark 8. *Rather than relaxing and greedily rounding $\boldsymbol{z}$, one could consider a higher dimensional relax-and-round scheme where we let $\boldsymbol{Z}$ model the outer product $\boldsymbol{z}\boldsymbol{z}^\top$ via $\boldsymbol{Z} \succeq \boldsymbol{z}\boldsymbol{z}^\top$, $\max(0, z_i + z_j - 1) \leq Z_{i,j} \leq \min(z_i, z_j)\ \forall i,j\in[p]$, $Z_{i,i} = z_i$, and require*

that $\sum_{i,j\in[p]} Z_{i,j} \leq k^2$. *Indeed, a natural "round" component of such a relax-and-round scheme is precisely Goemans-Williamson rounding [124, 29], which performs at least as well as greedy rounding in both theory and practice. Unfortunately, some preliminary numerical experiments indicated that Goemans-Williamson rounding is not actually much better than greedy rounding in practice, and is considerably more expensive to implement. Therefore, we do not consider it any further in this thesis.*

Valid inequalities for convex relaxation

We now propose valid inequalities which allow us to improve the quality of the convex relaxations discussed previously. Note that as convex relaxations and random rounding methods are two sides of the same coin [10], applying these valid inequalities also improves the quality of the randomly rounded solutions.

Theorem 4.6. *Let* $\mathcal{P}_{strong}$ *denote the optimal objective value of the problem:*

$$\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\in S^p_+} \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle \ s.t. \quad \mathrm{tr}(\boldsymbol{X}) = 1, |X_{i,j}| \leq M_{i,j} z_i \ \forall i,j \in [p], \qquad (4.20)$$
$$\sum_{j\in[p]} X_{i,j}^2 \leq X_{i,i} z_i, \|\boldsymbol{X}\|_1 \leq k.$$

Then, (4.20) *is a stronger relaxation than* (4.19)*, i.e., the following holds:*

$$\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} f(\boldsymbol{z}) \geq \mathcal{P}_{strong} \geq \max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} f(\boldsymbol{z}). \qquad (4.21)$$

Moreover, suppose that an optimal solution to (4.20) *is of rank one. Then:*

$$\mathcal{P}_{strong} = \max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} f(\boldsymbol{z}),$$

i.e., the relaxation is tight.

Proof. The first inequality $\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} f(\boldsymbol{z}) \geq \mathcal{P}_{strong}$ is trivial. The second inequality holds because $\mathcal{P}_{strong}$ is indeed a valid relaxation of Problem (4.1). Indeed, $\|\boldsymbol{X}\|_1 \leq k$ follows from the cardinality and big-M constraints. The semidefinite

constraint $\boldsymbol{X} \succeq 0$ impose second-order cone constraints on the 2×2 minors of $\boldsymbol{X}$, $X_{i,j}^2 \leq z_i X_{i,i} X_{j,j}$, which can be aggregated into $\sum_{j \in [p]} X_{i,j}^2 \leq X_{i,i} z_i$ [see 23].

Finally, suppose that an optimal solution to Problem (4.20) is of rank one, i.e., the optimal matrix $\boldsymbol{X}$ can be decomposed as $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. Then, the SOCP inequalities imply that $\sum_{j \in [p]} x_i^2 x_j^2 \leq x_i^2 z_i$. However, $\sum_{j \in [p]} x_j^2 = \mathrm{tr}(\boldsymbol{X}) = 1$, which implies that $x_i^2 \leq x_i^2 z_i$, i.e., $z_i = 1$ for any index i such that $|x_i| > 0$. Since $\boldsymbol{e}^\top \boldsymbol{z} \leq k$, this implies that $\|\boldsymbol{x}\|_0 \leq k$, i.e., $\boldsymbol{X}$ also solves Problem (4.2). □

As our numerical experiments demonstrate and despite the simplicity of our rounding mechanism in Algorithm 4.1, the relaxation (4.20) provides high-quality solutions to the original sparse PCA problem (4.1), without introducing additional variables.

4.6 Numerical Results

We now assess the numerical behavior of the algorithms proposed in Chapter 4.2 and 4.5. To bridge the gap between theory and practice, we present a `Julia` code which implements the described convex relaxation and greedy rounding procedure on GitHub[1]. The code requires a conic solver such as `Mosek` and several open source Julia packages to be installed.

Performance of exact methods

In this section, we apply Algorithm 2.1 to medium and large-scale sparse PCA problems, with and without Gershgorin circle theorem bounds in the master problem. All experiments were implemented in `Julia` 1.2, using `CPLEX` 12.10 and `JuMP.jl` 0.18.6, and performed on a standard Macbook Pro laptop, with a 2.9GHz 6-Core Intel i9 CPU, using 16 GB DDR4 RAM. We benchmark our approach on the UCI `pitprops`, `wine`, `miniboone`, `communities`, `arrythmia` and `micromass` datasets, both in terms of runtime and the number of nodes expanded; we refer to [20, 23] for descriptions of these datasets. Note that we normalized all datasets before running the method

[1] https://github.com/ryancorywright/ScalableSPCA.jl

(i.e., we compute the leading sparse principal components of correlation matrices). Additionally, we warm-start the methods with the solution from the method of [228].

Table 4.1 reports the time for Algorithm 2.1 (with and without Gershgorin circle theorem bounds in the master problem) to identify the leading k-sparse principal component for $k \in \{5, 10, 20\}$, along with the number of nodes expanded, and the number of outer approximation cuts generated.

Table 4.1: Runtime in seconds (T), Nodes expanded (N) and cuts generated (C) per approach. We run all approaches on one thread, and impose a time limit of 600s. If a solver fails to converge, we report the relative gap (%) at termination in brackets, and the no. explored nodes and cuts at the time limit.

Dataset	p	k	Alg. 2.1			Alg. 2.1+ Circle Thm.		
			T. (s)	N.	C.	T. (s)	N.	C.
Pitprops	13	5	0.38	2,211	4,359	0.06	38	27
		10	0.08	304	763	0.02	18	127
Wine	13	5	0.53	2,952	6,043	0.02	46	31
		10	0.12	319	797	0.08	418	965
Miniboone	50	5	0.01	0	8	0.02	1	84
		10	0.01	0	74	0.01	0	0
		20	0.02	1	26	0.01	0	5
Communities	101	5	(2.87%)	23,329	22,393	0.20	297	730
		10	(13.3%)	23,427	22,010	0.32	406	117
		20	(39.6%)	26,270	24,020	(10.8%)	42,780	9,176
Arrhythmia	274	5	(18.1%)	35,780	10,449	3.67	1,287	3,020
		10	(32.6%)	27,860	12,670	(2.47%)	15,115	18,422
		20	(74.4%)	33,773	12,374	(24.2%)	27,507	61,915
Micromass	1300	5	33.99	1,000	509	163.48	2,189	6,285
		10	(107%)	4,380	33,660	241.6	4,603	16,898
		20	(35.9%)	4,945	10,330	(35.9%)	5,085	1,210

Our main findings from these experiments are as follows:

- For smaller problems, the strength of Algorithm 2.1's cuts allows it to obtain certifiably optimal solutions in seconds.
- For larger problem sizes, our method obtains certifiably near-optimal—yet not always optimal—solutions in hundreds of seconds, which suggests that running the method for a short period of time and returning the best solution found could be a powerful heuristic for problem sizes where Algorithm 2.1 fails to

converge in a reasonable amount of time.

- Generating outer-approximation cuts and upper bounds from the Gershgorin circle theorem are both powerful ideas, but the greatest aggregate power arises from intersecting these bounds, rather than using one bound alone.
- The aggregate time in user callbacks did not exceed 0.1 seconds in any problem instance considered, which suggests the subproblem strategy is very efficient.

Convex relaxations and rounding methods

In this section, we apply Algorithm 4.1 to obtain high quality convex relaxations and feasible solutions for the datasets studied in the previous subsection, and compare the relaxation to a difference convex relaxation developed by d'Aspremont et al. [81], in terms of the quality of the upper bound and the resulting greedily rounded solutions. All experiments were implemented using the same specifications as the previous section. Note that [81]'s upper bound which we compare against is:

$$\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \; \max_{\boldsymbol{X}\succeq\boldsymbol{0},\boldsymbol{P}_i\succeq\boldsymbol{0}\;\forall i\in[p]} \quad \sum_{i\in[p]} \langle \boldsymbol{a}_i\boldsymbol{a}_i^\top, \boldsymbol{P}_i\rangle \tag{4.22}$$
$$\text{s.t.} \quad \text{tr}(\boldsymbol{X}) = 1,\; \text{tr}(\boldsymbol{P}_i) = z_i,\; \boldsymbol{X}\succeq\boldsymbol{P}_i\;\forall i\in[p],$$

where $\boldsymbol{\Sigma} = \sum_{i=1}^p \boldsymbol{a}_i\boldsymbol{a}_i^\top$ is a Cholesky decomposition of $\boldsymbol{\Sigma}$, and we obtain feasible solutions from this relaxation by greedily rounding an optimal $\boldsymbol{z}$ in the bound *a la* Algorithm 4.1. To allow for a fair comparison, we also consider augmenting this formulation with the inequalities derived in the previous section to obtain the following stronger yet more expensive to solve relaxation:

$$\max_{\boldsymbol{z}\in[0,1]^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \; \max_{\substack{\boldsymbol{X}\succeq\boldsymbol{0},\\ \boldsymbol{P}_i\succeq\boldsymbol{0}\;\forall i\in[p]}} \quad \sum_{i\in[p]} \langle \boldsymbol{a}_i\boldsymbol{a}_i^\top, \boldsymbol{P}_i\rangle \tag{4.23}$$
$$\text{s.t.} \quad \text{tr}(\boldsymbol{X}) = 1,\; \text{tr}(\boldsymbol{P}_i) = z_i,\; \boldsymbol{X}\succeq\boldsymbol{P}_i\;\forall i\in[p],$$
$$\sum_{j\in[p]} X_{i,j}^2 \leq X_{i,i}z_i, \|\boldsymbol{X}\|_1 \leq k.$$

We first apply these relaxations on datasets where Algorithm 2.1 terminates, hence the optimal solution is known and can be compared against. We report the quality of both methods with and without the additional inequalities discussed in the previous section, in Tables 4.2-4.3 respectively.

Table 4.2: Quality of relaxation gap (upper bound vs. optimal solution-denoted R.), objective gap (rounded solution vs. optimal solution-denoted O.) and runtime in seconds per method.

Dataset	p	k	Alg. 4.1 with (4.19)			Alg. 4.1 with (4.22)		
			R. (%)	O. (%)	T. (s)	R. (%)	O. (%)	T. (s)
Pitprops	13	5	23.8	0.00	0.02	23.8	16.1	0.46
		10	1.10	0.30	0.03	1.10	1.33	0.46
Wine	13	5	36.8	0.00	0.02	36.8	40.4	0.433
		10	2.43	0.26	0.03	2.43	15.0	0.463
Miniboone	50	5	781	236	7.37	781	34.7	1,191
		10	341	118	7.50	341	44.9	1,103
		20	120.3%	38.08%	6.25	120.3%	31.9%	1,140.2

Table 4.3: Quality of relaxation gap (upper bound vs. optimal solution-denoted R.), objective gap (rounded solution vs. optimal solution-denoted O.) and runtime in seconds, with additional inequalities from Chap. 4.5.

Dataset	p	k	Alg. 4.1 with (4.20)			Alg. 4.1 with (4.23)		
			R. (%)	O. (%)	T. (s)	R. (%)	O. (%)	T. (s)
Pitprops	13	5	0.71	0.00	0.17	1.53	0.00	0.55
		10	0.12	0.00	0.27	1.10	0.00	3.27
Wine	13	5	1.56	0.00	0.24	2.98	15.0	0.95
		10	0.40	0.00	0.22	2.04	0.00	1.15
Miniboone	50	5	0.00	0.00	163	0.00	0.01	501
		10	0.00	0.00	149	0.00	0.02	490
		20	0.00%	0.00%	194.5	0.00%	0.00%	776.3

Observe that applying Algorithm 4.1 without the additional inequalities (Table 4.2) yields rather poor relaxations and randomly rounded solutions. However, by intersecting our relaxations with the additional inequalities from Chapter 4.5 (Table 4.3), we obtain extremely high quality relaxations. Indeed, with the additional inequalities, Algorithm 4.1 (using Problem (4.20)) identifies the optimal solution in all instances, and always supplies a bound gap of less than 2%. Moreover, in terms of

obtaining high-quality solutions, the new inequalites allow Problem (4.20) to perform as well or better as Problem (4.22), despite optimizing over one semidefinite matrix, rather than $p+1$ semidefinite matrices. This suggests that Problem (4.20) should be considered as a viable, more scalable and more accurate alternative to existing SDO relaxations such as Problem (4.22). For this reason, we shall only consider using Problem (4.20)'s formulation for the rest of the chapter.

We remark however that the key drawback of applying these methods is that, as implemented in this section, they do not scale to sizes beyond which Algorithm 2.1 successfully solves. This is a drawback because Algorithm 2.1 supplies an exact certificate of optimality, while these methods do not. In the following set of experiments, we investigate numerical techniques to improve the scalability of Algorithm 4.1.

Scalable dual bounds and rounding methods

To improve the scalability of Algorithm 4.1, we relax the PSD constraint on $\boldsymbol{X}$ in (4.19) and (4.20). With these enhancements, we demonstrate that Algorithm 4.1 can be successfully scaled to generate high-quality bounds for $1000s \times 1000s$ matrices. As discussed in Remark 7, we can replace the PSD constraint $\boldsymbol{X} \succeq \mathbf{0}$ by requiring that the $p(p-1)/2$ two by two minors of $\boldsymbol{X}$ are non-negative: $X_{i,j}^2 \leq X_{i,i}X_{j,j}$. Second, we consider adding 20 linear inequalities of the form $\langle \boldsymbol{X}, \boldsymbol{x}_t\boldsymbol{x}_t^\top \rangle \geq 0$, for some vector $\boldsymbol{x}_t$ [see 23, for a discussion]. Table 4.4 reports the performance of Algorithm 4.1 (with the relaxation (4.20)) with these two approximations of the positive semidefinite cone, "Minors" and "Minors + 20 inequalities" respectively. Note that we report the entire duality gap (i.e., do not break the gap down into its relaxation and objective gap components) since, as reflected in Table 4.1, some of these instances are currently too large to solve to optimality.

Observe that if we impose constraints on the 2×2 minors only then we obtain a solution within 1% of optimality and provably within 15% of optimality in seconds (resp. minutes) for $p = 100$s (resp. $p = 1000$s). Moreover, adding 20 linear inequalities, we obtain a solution within 0.3% of optimality and provably within 2% of optimality in minutes (resp. hours) for $p = 100$s (resp. $p = 1000$s).

Table 4.4: Quality of bound gap (rounded solution vs. upper bound) and runtime of Algorithm 4.1 with (4.20), outer-approximation of the PSD cone.

Dataset	p	k	Minors		Minors + 20 inequalities	
			Gap (%)	Time(s)	Gap (%)	Time(s)
Pitprops	13	5	1.51%	0.02	0.72%	0.36
		10	5.29%	0.02	1.12%	0.36
Wine	13	5	2.22%	0.02	1.59%	0.38
		10	3.81%	0.02	1.50%	0.37
Miniboone	50	5	0.00%	0.11	0.00%	0.11
		10	0.00%	0.12	0.00%	0.12
		20	0.00%	0.39	0.00%	0.39
Communities	101	5	0.07%	0.67	0.07%	14.8
		10	0.66%	0.68	0.66%	14.4
		20	3.32%	1.84	2.23%	33.5
Arrhythmia	274	5	3.37%	27.2	1.39%	203.6
		10	3.01%	25.6	1.33%	184.0
		20	8.87%	21.8	4.48%	426.8
Micromass	1300	5	0.04%	239.4	0.01%	4,639
		10	0.63%	232.6	0.32%	6,392
		20	13.1%	983.5	5.88%	16,350

To conclude this section, we explore Algorithm 4.1's ability to scale to even higher dimensional datasets in a high performance setting, by running the method on one Intel Xeon E5–2690 v4 2.6GHz CPU core using 600 GB RAM. Table 4.5 reports the methods scalability and performance on the Wilshire 5000, and `Arcene` UCI datasets. For the `Gisette` dataset, we report on the methods performance when we include the first $3,000$ and $4,000$ rows/columns (as well as all $5,000$ rows/columns). Similarly, for the `Arcene` dataset we report on the method's performance when we include the first $6,000$, $7,000$ or $8,000$ rows/columns. We do not report results for the `Arcene` dataset for $p > 8,000$, as computing this requires more memory than was available (i.e. $>$ 600 GB RAM). We do not report the method's performance when we impose linear inequalities for the PSD cone, as solving the relaxation without them is already rather time consuming. Moreover, we do not impose the 2×2 minor constraints to save memory, do not impose $|X_{i,j}| \leq M_{i,j} z_i$ when $p \geq 4000$ to save even more memory, and report the overall bound gap, as improving upon the randomly rounded solution is challenging in a high-dimensional setting.

Table 4.5: Quality of bound gap (rounded solution vs. upper bound).

Dataset	p	k	Algorithm 4.1 (SOC relax)+Inequalities	
			Bound gap (%)	Time(s)
Wilshire 5000	2130	5	0.38%	1,036
		10	0.24%	1,014
		20	0.36%	1,059
Gisette	3000	5	1.67%	2,249
		10	35.81%	2,562
		20	10.61%	3,424
Gisette	4000	5	1.55%	1,402
		10	54.4%	1,203
		20	11.84%	1,435
Gisette	5000	5	1.89%	2,169
		10	2.22%	2,455
		20	7.16%	2,190
Arcene	6000	5	0.01%	3,333
		10	0.06%	3,616
		20	0.14%	3,198
Arcene	7000	5	0.03%	4,160
		10	0.05%	4,594
		20	0.25%	4,730
Arcene	8000	5	0.02%	6,895
		10	0.17%	8,479
		20	0.21%	6,335

These results suggest that if we solve the SOC relaxation using a first-order method rather than an interior point method, our approach could successfully generate certifiably near-optimal PCs when $p = 10,000$s.

Performance of Methods on Synthetic Data

We now compare the exact and approximate methods against existing state-of-the-art methods in a spiked covariance matrix setting. We use the experimental setup laid out in d'Aspremont et al. [81, Section 7.1]. We recover the leading principal component of a test matrix $\mathbf{\Sigma} \in S_+^p$, where $p = 150$, $\mathbf{\Sigma} = \frac{1}{n}\boldsymbol{U}^\top\boldsymbol{U} + \frac{\sigma}{\|\boldsymbol{v}\|_2^2}\boldsymbol{v}\boldsymbol{v}^\top$, $\boldsymbol{U} \in [0,1]^{150\times 150}$ is

a noisy matrix with i.i.d. standard uniform entries, $\boldsymbol{v} \in \mathbb{R}^{150}$ is a vector of signals:

$$v_i = \begin{cases} 1, & \text{if } i \leq 50, \\ \frac{1}{i-50}, & \text{if } 51 \leq i \leq 100, \\ 0, & \text{otherwise,} \end{cases} \tag{4.24}$$

and $\sigma = 2$ is the signal-to-noise ratio. The methods which we compare are:

- **Exact**: Algorithm 2.1 with Gershgorin inequalities and a time limit of 600s.
- **Approximate:** Algorithm 4.1 with Problem (4.20), the SOC outer approximation of the PSD cone, no PSD cuts, and the additional SOC inequalities.
- **Greedy:** as proposed by [176] and laid out in [81, Algorithm 1], start with a solution $\boldsymbol{z}$ of cardinality 1 and iteratively augment this solution vector with the index which gives the maximum variance contribution. Note that [81] found this method outperformed the 3 other methods (approximate greedy, thresholding and sorting) they considered in their work.
- **Truncated Power Method:** as proposed by [228], alternate between applying the power method to the solution vector and truncating the vector to ensure that it is k-sparse. Note that [20] found that this approach performed better than five other state-of-the-art methods across the real-world datasets studied in the previous section of this chapter and often matched the performance of the method of [20]—indeed, it functions as a warm-start for the later method.
- **Sorting:** sort the entries of $\boldsymbol{\Sigma}_{i,i}$ by magnitude and set $z_i = 1$ for the k largest entries of $\boldsymbol{\Sigma}$, as studied in [81]. This naive method serves as a benchmark for the value of optimization in the more sophisticated methods considered here.

Figures 4-1 depicts the ROC curve (true positive rate vs. false positive rate for recovering the support of $\boldsymbol{v}$) over 20 synthetic random instances, as we vary k for each instance. We observe that among all methods, the sorting method is the least accurate, with a substantially larger false detection rate for a given true positive rate than the remaining methods (AUC= 0.7028). The truncated power method and

our exact method[2] then offer a substantial improvement over sorting, with respective AUCs of 0.7482 and 0.7483. The greedy method then offers a modest improvement over them (AUC= 0.7561) and the approximate relax+round method is the most accurate (AUC= 0.7593).

In addition to support recovery, Figure 4-2 reports average runtime (left panel) and average optimality gap (right panel) over the same instances. Observe that among all methods, only the exact and the approximate relax+round methods provide optimality gaps, i.e., certificate of near optimality. On this metric, relax+round supplies average bound gaps of 1% or less on all instances, while the exact method typically supplies bound gaps of 30% or more. Moreover, the relax+round method converges in less than one minute on all instances. All told, the relax+round method is the best performing method overall, although if k is set to be sufficiently close to 0 or p all methods behave comparably. In particular, the relax+round method should be preferred over the exact method, even though the exact method performs better at smaller problem sizes.

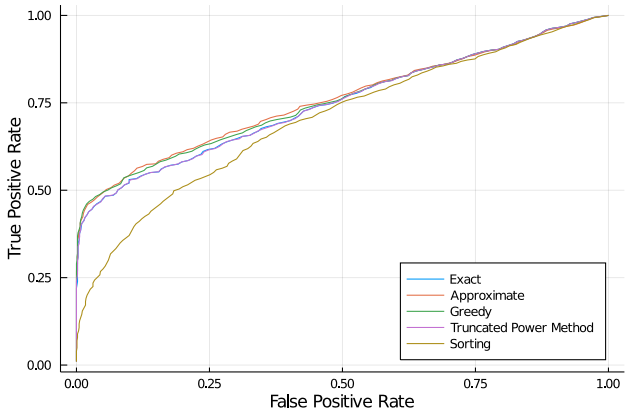


Figure 4-1: ROC curve over 20 instances where $p = 150$, $k_{\text{true}} = 100$ is unspecified.

[2]The exact method would dominate the remaining methods if given an unlimited runtime budget. Its poor performance reflects its inability to find the true optimal solution within 600 seconds.

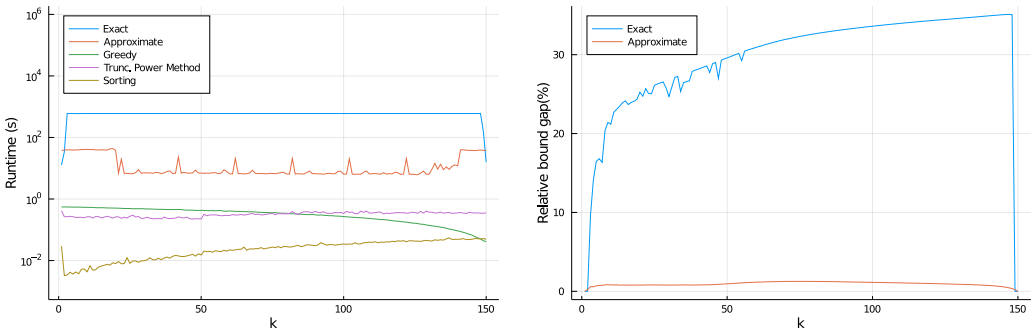


Figure 4-2: Average time to compute solution, optimality gap and in-sample variance ratio over 20 instances where $p = 150$, $k_{\text{true}} = 100$ unspecified.

4.7 Conclusion and Extensions

In this chapter, we developed a MISDO formulation of sparse PCA and provided techniques for solving it to certifiable optimality or near optimality at scale. We have also demonstrated that our relaxations are both more scalable and more accurate than existing state-of-the-art relaxations such as the relaxation of [81]. We now conclude by discussing three extensions of sparse PCA where our methodology applies.

Non-Negative Sparse PCA

One potential extension to this chapter would be to develop a certifiably optimal algorithm for non-negative sparse PCA [see 230, for a discussion], i.e., develop a tractable reformulation of

$$\max_{\boldsymbol{x}\in\mathbb{R}^p} \quad \langle \boldsymbol{x}\boldsymbol{x}^\top, \boldsymbol{\Sigma}\rangle \text{ s.t. } \boldsymbol{x}^\top\boldsymbol{x} = 1, \boldsymbol{x} \geq \boldsymbol{0}, \|\boldsymbol{x}\|_0 \leq k.$$

Unfortunately, we cannot develop a MISDO reformulation of non-negative sparse PCA *mutatis mutandis* Theorem 4.1. Indeed, while we can set $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ and relax the rank-one constraint, if we do so then, by the non-negativity of $\boldsymbol{x}$, lifting yields

$$\begin{aligned} \max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \quad & \max_{\boldsymbol{X}\in\mathcal{C}_n} \quad \langle \boldsymbol{\Sigma}, \boldsymbol{X}\rangle \\ & \text{s.t.} \quad \operatorname{tr}(\boldsymbol{X}) = 1,\ X_{i,j} = 0 \text{ if } z_i = 0,\ X_{i,j} = 0 \text{ if } z_j = 0\ \forall i,j \in [p]. \end{aligned} \tag{4.25}$$

where $\mathcal{C}_n := \{\boldsymbol{X} : \exists\, \boldsymbol{U} \geq \boldsymbol{0}, \boldsymbol{X} = \boldsymbol{U}^\top \boldsymbol{U}\}$ denotes the completely positive cone, which is NP-hard to separate over and cannot currently be optimized over tractably [88]. Nonetheless, we can develop relatively tractable mixed-integer conic upper and lower bounds for non-negative sparse PCA. Indeed, we can obtain a fairly tight upper bound by replacing the completely positive cone with the larger doubly non-negative cone $\mathcal{D}_n := \{\boldsymbol{X} \in S_+^p : \boldsymbol{X} \geq \boldsymbol{0}\}$, which is a high-quality outer-approximation of $\mathcal{C}_n$, indeed exact when $k \leq 4$ [60].

Unfortunately, this relaxation is strictly different in general, since the extreme rays of the doubly non-negative cone are not necessarily rank-one when $k \geq 5$ [60]. Nonetheless, to obtain feasible solutions which supply lower bounds, we could inner approximate the completely positive cone with the cone of non-negative scaled diagonally dominant matrices [see 2, 51].

Sparse PCA on Rectangular Matrices

A second extension would be to extend our methodology to the non-square case:

$$\max_{\boldsymbol{x} \in \mathbb{R}^m, \boldsymbol{y} \in \mathbb{R}^n} \quad \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} \text{ s.t. } \|\boldsymbol{x}\|_2 = 1, \|\boldsymbol{y}\|_2 = 1, \|\boldsymbol{x}\|_0 \leq k, \|\boldsymbol{y}\|_0 \leq k. \tag{4.26}$$

Observe that computing the spectral norm of a matrix $\boldsymbol{A}$ is equivalent to

$$\max_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} \quad \langle \boldsymbol{A}, \boldsymbol{X} \rangle \text{ s.t. } \begin{pmatrix} \boldsymbol{U} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{V} \end{pmatrix} \succeq \boldsymbol{0}, \operatorname{tr}(\boldsymbol{U}) + \operatorname{tr}(\boldsymbol{V}) = 2, \tag{4.27}$$

where, in an optimal solution, $\boldsymbol{U}$ stands for $\boldsymbol{x}\boldsymbol{x}^\top$, $\boldsymbol{V}$ stands for $\boldsymbol{y}\boldsymbol{y}^\top$ and $\boldsymbol{X}$ stands for $\boldsymbol{x}\boldsymbol{y}^\top$—this can be seen by taking the dual of [198, Equation 2.4].

Therefore, by using the same argument as in the positive semidefinite case, we

can rewrite sparse PCA on rectangular matrices as the following MISDO:

$$\begin{aligned}\max_{\boldsymbol{w}\in\{0,1\}^m,\boldsymbol{z}\in\{0,1\}^n}\ \max_{\boldsymbol{X}\in\mathbb{R}^{n\times m}}\quad & \langle \boldsymbol{A},\boldsymbol{X}\rangle \\ & \text{s.t. } \begin{pmatrix}\boldsymbol{U} & \boldsymbol{X}\\ \boldsymbol{X}^\top & \boldsymbol{V}\end{pmatrix}\succeq \boldsymbol{0}, \mathrm{tr}(\boldsymbol{U})+\mathrm{tr}(\boldsymbol{V})=2, \\ & U_{i,j}=0 \text{ if } w_i=0\ \forall i,j\in[m], \\ & V_{i,j}=0 \text{ if } z_i=0\ \forall i,j\in[n], \boldsymbol{e}^\top\boldsymbol{w}\leq k, \boldsymbol{e}^\top\boldsymbol{z}\leq k.\end{aligned} \tag{4.28}$$

Sparse PCA with Multiple Principal Components

A third extension where our methodology is applicable is the problem of obtaining multiple principal components simultaneously, rather than deflating $\boldsymbol{\Sigma}$ after obtaining each principal component. As there are multiple definitions of this problem, we now discuss the extent to which our framework encompasses each case.

Common Support: Perhaps the simplest extension of sparse PCA to a multi-component setting arises when all r principal components have common support. By retaining the vector of binary variables $\boldsymbol{z}$ and employing the Ky-Fan theorem [c.f. 225, Theorem 2.3.8] to cope with multiple principal components, we obtain the following formulation in much the same manner as previously:

$$\max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k}\ \max_{\boldsymbol{X}\in S^p_+}\quad \langle \boldsymbol{X},\boldsymbol{\Sigma}\rangle \text{ s.t. } \boldsymbol{0}\preceq\boldsymbol{X}\preceq\mathbb{I},\ \mathrm{tr}(\boldsymbol{X})=r,\ X_{i,j}=0 \text{ if } z_i=0\ \forall i\in[p]. \tag{4.29}$$

The logical constraint $X_{i,j}=0$ if $z_i=0$, which formed the basis of our subproblem strategy, still successfully models the sparsity constraint. This suggests that (a) one can derive an equivalent subproblem strategy under common support, and (b) a cutting-plane method for common support should scale as well as a single component.

Disjoint Support: In a sparse PCA problem with disjoint support [219], simultaneously computing the first r principal components is equivalent to solving

$$\max_{\substack{\boldsymbol{z}\in\{0,1\}^{p\times r}:\boldsymbol{e}^\top\boldsymbol{z}_t\leq k\ \forall t\in[r],\\ \boldsymbol{z}\boldsymbol{e}\leq\boldsymbol{e}}} \max_{\boldsymbol{W}\in\mathbb{R}^{p\times r}} \quad \langle \boldsymbol{W}\boldsymbol{W}^\top, \boldsymbol{\Sigma}\rangle$$
$$\boldsymbol{W}^\top\boldsymbol{W} = \mathbb{I}_r,\ W_{i,j} = 0 \text{ if } z_{i,t} = 0\ \forall i \in [p], t \in [r], \tag{4.30}$$

where $z_{i,t}$ is a binary variable denoting whether feature i is a member of the tth principal component. By applying the technique used to derive Theorem 4.1 *mutatis mutandis*, and invoking the Ky-Fan theorem [c.f. 225, Theorem 2.3.8] to cope with the rank-r constraint, we obtain

$$\max_{\boldsymbol{z}\in\{0,1\}^p:\boldsymbol{e}^\top\boldsymbol{z}\leq k} \max_{\boldsymbol{X}\in S^p} \quad \langle \boldsymbol{X}, \boldsymbol{\Sigma}\rangle$$
$$\mathbf{0} \preceq \boldsymbol{X} \preceq \mathbb{I},\ \mathrm{tr}(\boldsymbol{X}) = r,\ X_{i,j} = 0 \text{ if } Y_{i,j} = 0\ \forall i \in [p], \tag{4.31}$$

where $Y_{i,j} = \sum_{t=1}^r z_{i,t}z_{j,t}$ is a binary matrix denoting whether features i and j are members of the same principal component; this problem can be addressed by a cutting-plane method in much the same manner as when $r = 1$.

Part II

Rank Constraints

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Mixed-Projection Conic Optimization

Many problems in optimization, machine learning, and control theory are equivalent to optimizing a low-rank matrix over a convex set. For instance, rank constraints successfully model notions of minimal complexity, low dimensionality, or orthogonality in a system. However, while rank constraints offer unparalleled modeling flexibility, no generic code currently solves these problems to certifiable optimality at even moderate sizes. This state of affairs has led influential works on low-rank optimization [62, 198] to characterize low-rank optimization as intractable and advocate convex relaxations or heuristics which do not enjoy assumption-free optimality guarantees.

The manner in which low-rank optimization is regarded today is reminiscent of how mixed-integer optimization (MIO), which can model NP-complete problems, was originally considered. After decades of research effort, however, algorithms and software for MIO are now widely available [see, e.g., 49, 73] and solve large instances of disparate non-convex problems such as best subset selection [27] or the Traveling Salesperson Problem [187] to optimality. Unfortunately, rank constraints cannot be represented using MIO [161, Lemma 4.1] and do not benefit from these advances.

In this chapter, we characterize the complexity of rank constrained optimization and propose a new, more general framework, which we term *Mixed-Projection Conic Optimization* (MPCO). Our proposal generalizes MICO, by replacing binary variables z which satisfy $z^2 = z$ with symmetric orthogonal projection matrices $\boldsymbol{Y}$ which satisfy $\boldsymbol{Y}^2 = \boldsymbol{Y}$, and offers the following advantages over existing state-of-the-art methods:

First, it supplies certificates of (near) optimality for low-rank problems. Second, it demonstrates that some of the best ideas in MICO, such as decomposition methods, cutting-planes, relaxations, and random rounding schemes, admit straightforward extensions to MPCO. Finally, we implement a near-optimal rounding strategy and a globally optimal cutting-plane algorithm that improve upon the state-of-the-art for matrix completion and sensor location problems. We hope that MPCO gives rise to exciting new challenges for the optimization community to tackle.

Scope of the Framework

Formally, we consider the problem:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} \lambda \cdot \text{Rank}(\boldsymbol{X}) + \langle \boldsymbol{C}, \boldsymbol{X}\rangle \text{ s.t. } \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B},\ \text{Rank}(\boldsymbol{X}) \leq k,\ \boldsymbol{X} \in \mathcal{K}, \tag{5.1}$$

where λ (resp. k) prices (bounds) the rank of $\boldsymbol{X}$, $(\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}^{\ell\times n} \times \mathbb{R}^{\ell\times m}$ defines an affine subspace, and $\mathcal{K}$ is a proper cone in the sense of [54], i.e., closed, convex, solid and pointed. Observe that Problem (5.1) offers significant modeling flexibility, as it allows arbitrary conic constraints on $\boldsymbol{X}$. As a result, linear, convex quadratic, semidefinite, exponential, and power constraints and objectives can be captured by letting $\mathcal{K}$ be an appropriate product of different cones.

We now present some problems from the optimization and machine learning literature which admit low-rank formulations and fall within our framework.

Low-rank matrix completion

Given a sub-sample $(A_{i,j} : (i,j) \in \mathcal{I} \subseteq [n] \times [p])$ of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$, the matrix completion problem is to recover the entire matrix, by assuming $\boldsymbol{A}$ is low rank and seeking a rank-k matrix $\boldsymbol{X}$ which approximately fits the observed values. This problem arises in recommender system applications and admits the formulation:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times p}} \quad \frac{1}{2}\sum_{(i,j)\in\mathcal{I}} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \text{Rank}(\boldsymbol{X}) \leq k. \tag{5.2}$$

Minimum dimension Euclidean distance embedding

Given a set of pairwise distances $d_{i,j}$, the Euclidean Distance Embedding (EDM) problem is to determine the lowest dimensional space which the distances can be embedded in, such that the distances correspond to Euclidean distances. This problem arises in protein folding, network sensor location, and satellite ranging applications among others [45, 159]. By Blekherman et al. [46] Theorem 2.49, a set of distances $d_{i,j}$ can be embedded in a Euclidean space of dimension k if and only if there exists some Gram matrix $\boldsymbol{G} \succeq \boldsymbol{0}$ of rank k such that $d_{i,j}^2 = G_{i,i} + G_{j,j} - 2G_{i,j}$, on all pairs (i,j) where $d_{i,j}$ is supplied. Denoting $D_{i,j} = d_{i,j}^2$, we write these constraints in matrix form, $\boldsymbol{D} = \text{Diag}(\boldsymbol{G})\boldsymbol{e}^\top + \boldsymbol{e}\text{Diag}(\boldsymbol{G})^\top - 2\boldsymbol{G}$, where the equality is implicitly imposed only for pairs (i,j) where $d_{i,j}$ is supplied. This is equivalent to:

$$\min_{\boldsymbol{G} \in S_+^n} \text{Rank}(\boldsymbol{G}) \quad \text{s.t.} \quad \text{Diag}(\boldsymbol{G})\boldsymbol{e}^\top + \boldsymbol{e}\text{Diag}(\boldsymbol{G})^\top - 2\boldsymbol{G} = \boldsymbol{D}. \tag{5.3}$$

Given a solution $\boldsymbol{G}$, we can obtain the matrix of coordinates of the underlying points $\boldsymbol{X}$ (up to a rotation and translation of the points) by performing a Cholesky decomposition of $\boldsymbol{G}$, $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^\top$. Post decomposition, $\boldsymbol{X}$ is a $n \times k$ rectangular matrix which contains the coordinates of the underlying points.

Quadratically constrained quadratic optimization

Quadratically constrained quadratic optimization (QCQO) seeks an $\boldsymbol{x} \in \mathbb{R}^n$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \boldsymbol{x}^\top \boldsymbol{Q}_0 \boldsymbol{x} + \boldsymbol{q}_0^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x}^\top \boldsymbol{Q}_i \boldsymbol{x} + \boldsymbol{q}_i^\top \boldsymbol{x} \leq r_i \quad \forall i \in [m], \tag{5.4}$$

where $\boldsymbol{Q}_0$, $\boldsymbol{Q}_i$,$\boldsymbol{q}_0$ $\boldsymbol{q}_i$, r_i are given problem data. This problem is non-convex, and encompasses binary quadratic optimization [124] and alternating current optimal power flow problems [154]. The fundamental difficulty in Problem (5.4) is the potential non-convexity of the outer product $\boldsymbol{x}\boldsymbol{x}^\top$. However, we can isolate this non-convexity by introducing a rank-one matrix $\boldsymbol{X}$ to model the outer product $\boldsymbol{x}\boldsymbol{x}^\top$. This leads to:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{X}\in S^n} \langle \boldsymbol{Q}_0, \boldsymbol{X}\rangle + \langle \boldsymbol{q}_0, \boldsymbol{x}\rangle \text{ s.t. } \langle \boldsymbol{Q}_i, \boldsymbol{X}\rangle + \langle \boldsymbol{q}_i, \boldsymbol{x}\rangle \leq r_i \ \forall i \in [m], \ \text{Rank} \begin{pmatrix} 1 & \boldsymbol{x}^\top \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} = 1.$$

We have established that QCQOPs are rank constrained problems. Notably however, the converse is also true: rank constrained problems with linear, second-order cone, or semidefinite constraints are QCQOPs. Indeed, the constraint $\text{Rank}(\boldsymbol{X}) \leq k$ is equivalent to requiring that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top : \boldsymbol{U} \in \mathbb{R}^{n\times k}, \boldsymbol{V} \in \mathbb{R}^{m\times k}$, i.e., imposing $m \times n$ non-convex quadratic equalities. As modern solvers such as `Gurobi` can now solve non-convex QCQOPs to global optimality, this QCQOP formulation can be used to solve low-rank problems, although it is not particularly scalable; we expand on this point in this chapter's numerical experiments.

Minimal degree sum-of-squares decomposition of a polynomial

Many central problems in optimization and control can be addressed by optimizing over the space of globally non-negative polynomials. As separating over this space exactly is NP-hard [178] and requires invoking computationally expensive results from real algebraic geometry such as Stengle's Positivstellensatz [see, e.g., 46, Section 3.4.3], non-negative polynomial optimization is typically addressed by taking a safe inner approximation, namely the set of polynomials which are a sum of squares [153, 189]. For the sake of both interpretability and tractability, a desirable attribute is to obtain a polynomial composed of a sum of at most k squares, where k is small. Recalling that a polynomial $p(\boldsymbol{z})$ of degree $2d$ is a sum-of-squares (SOS) if and only if $p(x) = \boldsymbol{z}^\top \boldsymbol{Q}\boldsymbol{z}$, where $\boldsymbol{z} = [1, x_1, \ldots, x_n, x_1x_2, \ldots, x_n^d]$ and $\boldsymbol{Q}$ is a PSD matrix [189], the minimal SOS decomposition of a polynomial is given by:

$$\min_{\boldsymbol{Q}\succeq\boldsymbol{0}} \quad \text{Rank}(\boldsymbol{Q}) \quad \text{s.t.} \quad p(x) = \boldsymbol{z}^\top \boldsymbol{Q}\boldsymbol{z}, \tag{5.5}$$

$$\text{where} \quad \boldsymbol{z} = [1, x_1, \ldots, x_n, x_1x_2, \ldots, x_n^d]. \tag{5.6}$$

This allows us to optimize over the space of *low-complexity* SOS polynomials.

5.1 Background and Literature Review

Our work arises at the intersection of three complementary areas of the low-rank optimization literature: (a) global optimization algorithms for non-convex quadratically constrained problems, (b) the interplay of convex relaxations and their dual side, randomized rounding methods, and (c) heuristics which provide high-quality solutions to non-convex problems in an efficient fashion.

Global optimization techniques

Branch-and-bound A broad class of global optimization algorithms have been proposed for QCQOs, since [174] observed that convex envelopes of non-convex regions supply globally valid lower bounds. This gives rise to a numerical strategy where one recursively partitions the QCQO's feasible region into subregions, constructs convex envelopes for each subregion and uses these envelopes to construct iteratively improving bounds. This approach is known as spatial branch-and-bound; see [156] for a scheme which decomposes a matrix into a sparse matrix plus a low-rank matrix, [151] for a modern implementation in alternating current optimal power flow, and [31] for an exact branch-and-bound approach to low-rank factor analysis.

Branch-and-cut In a complementary direction, several branch-and-cut methods [8, 160] have been proposed for solving non-convex QCQOs, by borrowing decomposition schemes from the mixed-integer nonlinear optimization literature [92]. While often efficient in practice, a common theme in these methods is that the more efficient decomposition schemes used for MINLOs cannot be applied out-of-the-box, because they may fail to converge to a globally optimal solution [see 125, for a counterexample]. As a result, non-convex problems need to be preprocessed in an expensive fashion. This preprocessing step has inhibited the use of global optimization methods for low-rank problems; indeed, we are not aware of any works which apply branch-and-cut techniques to solve low-rank problems to certifiable optimality.

Complementarity In an opposite direction, several authors have proposed applying general nonlinear optimization techniques to address low-rank problems, since Ding et al. [86] observed that a low-rank constraint is equivalent to a complementarity constraint over the positive semidefinite cone, and thus can be addressed by general techniques for mathematical programs with equilibrium constraints [see 163]. Among others, Bai et al. [9] invoked the complementarity observation to design a completely positive reformulation of low-rank SDOs, and Bi et al. [41] developed a multi-stage convex relaxation of the complementarity constraint.

Algebraic By taking an algebraic view of rank constraints, several algebraic geometry techniques have been proposed for addressing low-rank SDOs. Among others, [78] proposed reformulating low-rank constraints as systems of polynomial equations which can be addressed via the sum-of-squares hierarchy [153]. More recently, Naldi [180] proposed a semi-algebraic reformulation of rank-constrained SDOs, which can be optimized over via Gröbner basis computation [75]. Unfortunately, algebraic approaches do not scale well in practice. Indeed, as observed by Recht et al. [198], it seems unlikely that algebraic approaches can solve low-rank SDOs when $n > 10$.

Convex relaxations and rounding methods

Convex relaxations A number of authors have studied convex relaxations of low-rank problems, since [103] observed that the nuclear norm of a matrix is the convex envelope of a rank constraint on the matrices with spectral norm at most M, i.e.,

$$\begin{aligned} &\operatorname{Conv}\Big(\big\{\boldsymbol{X} \in \mathbb{R}^{n\times m} :\ \|\boldsymbol{X}\|_\sigma \leq M, \operatorname{Rank}(\boldsymbol{X}) \leq k\big\}\Big) \\ &= \big\{\boldsymbol{X} \in \mathbb{R}^{n\times m} :\ \|\boldsymbol{X}\|_\sigma \leq M, \|\boldsymbol{X}\|_* \leq kM\big\}. \end{aligned} \tag{5.7}$$

Because the epigraph of a nuclear norm is semidefinite representable [198], this gives rise to semidefinite relaxations which can be computed in polynomial time.

Rounding methods A complementary line of work aims to supply certifiably near-optimal solutions to low-rank problems, by rounding their semidefinite relaxations. Initiated by Goemans and Williamson [124] in the context of binary quadratic optimization, who established that randomly rounding an SDO relaxation supplies a 0.878-approximation, it has evolved into a successful framework for solving rank-one optimization problems; see Nemirovski et al. [183] for a unified approach in the rank-one case. However, this line of work has a key drawback. Namely, existing rounding methods do not address rank-k problems such as matrix completion, due to the analytic difficulty of constructing a rounding mechanism which preserves both feasibility and near-optimality in the rank-k case.

Heuristic methods

Due to the computational difficulty of solving Problem (5.1) to optimality, a variety of heuristic methods have been proposed for solving (5.1), originating with methods for solving low-rank linear matrix inequalities in the optimal control literature [55].

Although slow and somewhat ad-hoc in their original implementations, heuristic methods were moved front-and-center by the works of Fazel [103], Burer and Monteiro [58, 59]. [103] observed that low-rank positive semidefinite matrices lie on the boundary of the PSD cone, and used this observation to justify a "log-det" heuristic, where a rank minimization objective is replaced with the function $\log\det(\boldsymbol{X} + \delta\mathbb{I})$. [58, 59] proposed implicitly modeling a rank constraint $\mathrm{Rank}(\boldsymbol{X}) \leq k$ by applying the non-linear reformulation $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, where $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times k}$ and eliminating $\boldsymbol{X}$, to obtain a problem which is non-convex in $(\boldsymbol{U}, \boldsymbol{V})$.

Although originally solved using augmented Lagrangian techniques, subsequent implementations of the Burer-Monterio heuristic typically used alternating minimization [138], successive over-relaxations [222] (stochastic) gradient descent [234, 197, 214] and manifold methods [52, 53]. This popularity has been driven by the fact that, under particular assumptions, the problem has no spurious local optima [see 40, 121, 52, 72] and the Burer-Monterio approach recovers a globally optimal solution; see Udell et al. [215], Nguyen et al. [184] for reviews of heuristic approaches.

5.2 From Cardinality to Rank: Unifying Perspective

Low rank constraints $\text{Rank}(\boldsymbol{X}) \leq k$ are a natural generalization of cardinality constraints $\|\boldsymbol{x}\|_0 \leq k$ from vectors to matrices. Indeed, if $\boldsymbol{X}$ is a diagonal matrix then $\text{Rank}(\boldsymbol{X}) \leq k$ if and only if $\|\boldsymbol{X}\|_0 \leq k$, and more generally $\text{Rank}(\boldsymbol{X}) \leq k$ if and only if $\|\sigma(\boldsymbol{X})\|_0 \leq k$, where $\sigma(\boldsymbol{X})$ is the vector of singular values of $\boldsymbol{X}$. However, while cardinality and rank constraints are intimately linked, they are addressed using different algorithms. Namely, we can solve cardinality constrained problems with $100,000$s of variables to optimality [27], while low-rank problems are dramatically harder and have not yet been solved to certifiable optimality for $n > 10$ [180].

In our opinion, the difference between the community's understanding of cardinality and rank constraints has arisen because of two algorithmic barriers. The first barrier is that rank constraints belong to a harder complexity class, namely they are as hard to optimize over as deciding whether an arbitrary system of polynomial inequalities admits a solution; see [34]. The second barrier arises because cardinality constraints can be represented using binary variables, while rank constraints cannot [161, Corollary 4.1]. This presents a challenge for researchers, who have developed scalable methods for cardinality constraints by exploiting advances in mixed-integer conic optimization (MICO), but cannot use these advances to address rank constraints. In this section, we question these barriers by characterizing the complexity of low-rank problems and proposing a new framework for modeling rank.

Complexity of Rank-Constrained Optimization

Existing studies of Problem (5.1) typically claim that it is intractable, and support this claim by proving that it is NP-hard, by reduction from an NP-complete problem such as Boolean linear programming [see, e.g., 216, Section 7.3]. In our opinion, this argument needs to be revisited, for two separate reasons. First, NP-hardness is a worst-case analysis statement. In practice, NP-hard problems are often tractable. For instance, sparse regression can usually be solved to certifiable optimality with $100,000$s of features in minutes [27]. Second, there is no evidence that Problem (5.1)

is even in NP. Indeed, Problem (5.1) cannot be represented using mixed-integer convex optimization [161, Corollary 4.1], while all 21 of Karp's NP-complete problems can, and the best known algorithms for Problem (5.1) run in EXPTIME [71, 180].

We now provide a more complete characterization of Problem (5.1)'s complexity than is currently available in the literature. First, we demonstrate that it belongs to a different class than NP. In particular, it is *existential theory of the reals*-hard ($\exists\mathbb{R}$-hard; see Renegar [199] for a general theory), i.e., as hard as any polynomial optimization problem, which implies that, if NP$\subsetneqq \exists\mathbb{R}$, Problem (5.1) is strictly harder than NP-complete problems. Second, we prove that Problem (5.1) is actually in $\exists\mathbb{R}$.

We now demonstrate that Problem (5.1) is existential theory of the reals complete (i.e., $\exists\mathbb{R}$-complete). We begin by reminding the reader of the definition of the $\exists\mathbb{R}$ complexity class [c.f. 208]:

Definition 5.1. *A decision problem belongs to the existential theory of the reals complexity class if it reduces to deciding whether a statement*

$$(\exists x_1, ..., x_n)\, \phi(x_1, ..., x_n)$$

is true or false, where $\phi(\cdot)$ is a quantifier-free Boolean formula involving polynomials equalities and inequalities, e.g., deciding the emptiness of a semi-algebraic set. We say a problem is $\exists\mathbb{R}$-hard if any problem in $\exists\mathbb{R}$ reduces to it.

Note that 3-SAT $\in \exists\mathbb{R}$, so NP$\subseteq \exists\mathbb{R}$, and any statement in $\exists\mathbb{R}$ can be decided in PSPACE [64], so $\exists\mathbb{R} \subseteq$ PSPACE. To establish that Problem (5.1) is $\exists\mathbb{R}$ hard, we require the following proposition, which is essentially a restatement of [208, Theorem 3.1] in the language of optimization.

Proposition 5.1. *Let $G := (V, E)$ be a graph, and $\ell(e)$ be the length of edge e. Then, deciding if G can be embedded in $\mathbb{R}^2$ is $\exists\mathbb{R}$ complete, even if all edges have unit length.*

By reducing Proposition 5.1's planar embedding problem to a Euclidean Distance Embedding problem, we obtain (see [34], for a proof):

Theorem 5.1. *Problem* (5.1) *is $\exists\mathbb{R}$-hard.*

Theorem 5.1 demonstrates that Problem (5.1) is, from a traditional complexity theory perspective, at least as hard as any problem in $\exists\mathbb{R}$. However, its complexity remains unresolved. Indeed, while [62] have observed that Problem (5.1) is in EXPTIME, it seems likely that $\exists\mathbb{R} \subset$ EXPTIME. We now address this, by proving that if $\mathcal{K}$ represents the semidefinite cone then Problem (5.1) is in $\exists\mathbb{R}$, and hence $\exists\mathbb{R}$-complete; note that the examples listed in Chapter 5 can all be rewritten as low-rank SDOs, so this result applies to all aforementioned examples (see [34] for a proof):

Theorem 5.2. *Let $\mathcal{K} = S^n_+$ denote the $n \times n$ positive semidefinite cone. Then, Problem* (5.1) *is in $\exists\mathbb{R}$, and hence $\exists\mathbb{R}$-complete.*

Remark 9. *Since $\exists\mathbb{R} \subseteq$ PSPACE $\subseteq$ EXPTIME, this upper bound improves upon the EXPTIME bound on Problem* (5.1)*'s complexity stated by Recht et al. [198], Candes and Plan [62] among others. Moreover, it seems unlikely to us that this bound can be further improved without settling fundamental questions in complexity theory (e.g. characterizing NP vs. $\exists\mathbb{R}$ vs. PSPACE vs. EXPTIME).*

Projection Matrices for Modeling Rank

As previously discussed, rank constraints can be seen as a generalization to the matrix case of cardinality constraints. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, the cardinality constraint $\|\boldsymbol{x}\|_0 \leq k$ ensures that at most k coordinates of $\boldsymbol{x}$ are non-zero, and can be modeled by introducing a vector of binary variables since

$$\|\boldsymbol{x}\|_0 \leq k \iff \exists \boldsymbol{z} \in \{0,1\}^n : \ \boldsymbol{e}^\top \boldsymbol{z} \leq k, \ \boldsymbol{x} = \boldsymbol{z} \circ \boldsymbol{x}. \tag{5.8}$$

Unfortunately, rank constraints cannot be modeled using mixed-integer convex optimization [161, Corollary 4.1] and therefore MICO techniques cannot be applied "out-of-the-box" to address rank constraints. Therefore, we now propose a new framework to model rank in optimization problems. Instead of a binary vector $\boldsymbol{z}$ to encode the support of $\boldsymbol{x}$, we introduce a projection matrix $\boldsymbol{Y}$ to capture the column space of $\boldsymbol{X}$ and obtain a similar non-linear reformulation.

Definition 5.2. *A matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ is an projection matrix if it satisfies $\boldsymbol{Y}^2 = \boldsymbol{Y}$. Moreover, if $\boldsymbol{Y}$ is symmetric, $\boldsymbol{Y}$ is an orthogonal projection matrix.*

As symmetric matrices, orthogonal projections are diagonalizable and their eigenvalues satisfy $\lambda_i^2 = \lambda_i$, i.e., are binary. As a result, the pseudoinverse of an orthogonal projection $\boldsymbol{Y}$ is $\boldsymbol{Y}$ itself ($\boldsymbol{Y} = \boldsymbol{Y}^\dagger$). In addition, since its eigenvalues are binary, the trace of $\boldsymbol{Y}$ equals the number of non-zero eigenvalues, i.e., $\text{Rank}(\boldsymbol{Y}) = \text{tr}(\boldsymbol{Y})$.

We are now in a position to link projection matrices and rank constraints.

Proposition 5.2. *For any $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, $\text{Rank}(\boldsymbol{X}) \leq k \iff \exists \boldsymbol{Y} \in \mathcal{Y}_n$: $tr(\boldsymbol{Y}) \leq k$, $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, where $\mathcal{Y}_n := \{\boldsymbol{P} \in S^n : \boldsymbol{P}^2 = \boldsymbol{P}\}$ is the set of orthogonal projections.*

Proof. We prove the two implications successively.

- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, with $\boldsymbol{U} \in \mathbb{R}^{n \times k}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$, $\boldsymbol{V} \in \mathbb{R}^{m \times k}$, be a singular value decomposition of $\boldsymbol{X}$ and define $\boldsymbol{Y} = \boldsymbol{U}\left(\boldsymbol{U}^\top\boldsymbol{U}\right)^{-1}\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{U}^\top$. By construction, $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, since $\boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}$. Moreover, $\text{tr}(\boldsymbol{Y}) = \text{rank}(\boldsymbol{Y}) = \text{rank}(\boldsymbol{X}) \leq k$.
- Since $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, $\text{rank}(\boldsymbol{X}) \leq \text{rank}(\boldsymbol{Y}) = \text{tr}(\boldsymbol{Y}) \leq k$. □

Remark 10. *In Proposition 5.2, the rank constraint is expressed via a trace constraint on $\boldsymbol{Y}$, the orthogonal projection onto the image or column space of $\boldsymbol{X}$. Alternatively, one could model the rank constraint via a matrix $\boldsymbol{Y'} \in \mathcal{Y}_m$ such that $tr(\boldsymbol{Y'}) \leq k$ and $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Y'}$. In this case, $\boldsymbol{Y'}$ encodes the projection onto the row space of $\boldsymbol{X}$. In practice, one could introduce both $\boldsymbol{Y}$ and $\boldsymbol{Y'}$ and obtain tighter formulations, at the price of introducing additional notation.*

Proposition 5.2 suggests that projection matrices are to rank constraints what binary variables are to cardinality constraints. Indeed, similarities between the two are evident: binary variables z are idempotent scalars which solve $z^2 = z$, while projection matrices $\boldsymbol{Y}$ are idempotent matrices which solve $\boldsymbol{Y}^2 = \boldsymbol{Y}$. Also, if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are diagonal, Proposition 5.2 recovers cardinality constrained optimization.

Over the past decades, extensive efforts have been devoted to improving the scalability of mixed-integer optimization. We believe that similar achievements can be

obtained for rank constrained problems by adapting techniques from MICO to MPCO. In this direction, Table 5.1 establishes a dictionary linking cardinality and rank, and demonstrates many of the techniques developed for binary convex optimization admit generalizations to MPCO, including the main results from Chapter 2. Note that we have not yet established most of the connections claimed in Table 5.1; this is the focus of the next two sections of the chapter.

Table 5.1: Analogy between mixed-integer and mixed-projection.

Framework	Chapter 2	This chapter
Parsimony concept	cardinality	rank
Non-convex outer set	binaries	projection matrices
Strongly convex regularizer	ℓ_2^2	Frobenius squared
Boundedness regularizer	ℓ_∞	spectral
Non-linear formulation	$\boldsymbol{x} = \boldsymbol{x} \circ \boldsymbol{z};\ \boldsymbol{z} \in \{0,1\}^n$	$\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X},\ \boldsymbol{Y} \in \mathcal{Y}_n$
Big-M formulation	$-M\boldsymbol{z} \leq \boldsymbol{x} \leq M\boldsymbol{z}$	$\begin{pmatrix} M\boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X}^\top & M\mathbb{I} \end{pmatrix} \succeq \boldsymbol{0}$
Perspective formulation	$\begin{pmatrix} \theta_i & x_i \\ x_i & z_i \end{pmatrix} \succeq \boldsymbol{0}$	$\begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0}$
Convex relaxation complexity	linear/second-order cone	semidefinite
Greedy rounding mechanism	coordinate-wise	SVD

5.3 Regularization and a Reformulation

In this section, we prove that (5.10) can be reformulated as a saddle-point mixed-projection problem by leveraging regularization terms analogous to the big-M and ridge regularization techniques from MICO, and derive their semidefinite relaxations, as summarized in Table 5.1.

Throughout this chapter, we let $\mathcal{Y}_n := \{\boldsymbol{P} \in S^n : \boldsymbol{P}^2 = \boldsymbol{P}\}$ denote the set of $n \times n$ orthogonal projection matrices and $\mathcal{Y}_n^k := \{\boldsymbol{P} \in S^n : \boldsymbol{P}^2 = \boldsymbol{P}, \mathrm{tr}(\boldsymbol{P}) \leq k\}$ denote projection matrices with rank at most k. Although $\mathcal{Y}_n$ and $\mathcal{Y}_n^k$ do not commonly appear in the optimization literature, their convex hulls are well-studied, as we now remind the reader, by restating [185, Theorem 3]:

Lemma 5.1. *Let $\mathcal{Y}_n$ denote the $n \times n$ orthogonal projection matrices and $\mathcal{Y}_n^k$ denote the low-rank orthogonal projection matrices. Then,* $\mathrm{Conv}(\mathcal{Y}_n) = \{\boldsymbol{P} : 0 \preceq \boldsymbol{P} \preceq \mathbb{I}\}$ *and* $\mathrm{Conv}(\mathcal{Y}_n^k) = \{\boldsymbol{P} : 0 \preceq \boldsymbol{P} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{P}) \leq k\}$. *Moreover, the extreme points of* $\mathrm{Conv}(\mathcal{Y}_n)$ *are $\mathcal{Y}_n$, the extreme points of* $\mathrm{Conv}(\mathcal{Y}_n^k)$ *are $\mathcal{Y}_n^k$.*

A Regularization Assumption

By invoking Proposition 5.2, we rewrite (5.1) as a mixed-projection conic problem:

$$\min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} \quad \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B},\ \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X},\ \boldsymbol{X} \in \mathcal{K}. \tag{5.9}$$

Observe that Problem (5.9) has a two-stage structure which involves first selecting a low-rank projection matrix $\boldsymbol{Y}$ and second selecting a matrix $\boldsymbol{X}$ under the constraint $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$. Moreover, selecting an optimal $\boldsymbol{X}$ given $\boldsymbol{Y}$ is *easy*, because it involves solving a conic optimization problem under the linear constraint $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, while selecting an optimal $\boldsymbol{Y}$ is *hard*, because $\mathcal{Y}_n^k$ is a non-convex set. Therefore, our modeling framework isolates the hardness of Problem (5.9) in $\mathcal{Y}_n^k$.

To cope with the non-linear constraints $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ in a tractable fashion, we augment the objective function in (5.9) with a regularization term. Namely:

$$\min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} \quad \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \Omega(\boldsymbol{X}) + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B},\ \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X},\ \boldsymbol{X} \in \mathcal{K}, \tag{5.10}$$

where the regularization term $\Omega(\boldsymbol{X})$ satisfies the following assumption:

Assumption 5.1. *In Problem* (5.10)*, the regularization term $\Omega(\boldsymbol{X})$ is one of:*

- *A spectral norm penalty, $\Omega(\boldsymbol{X}) = 0$ if $\|\boldsymbol{X}\|_\sigma \leq M$, $\Omega(\boldsymbol{X}) = +\infty$ otherwise.*
- *A Frobenius norm penalty, $\Omega(\boldsymbol{X}) = \frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2$.*

As we demonstrate in Section 5.3, Assumption 5.1 is crucial for developing efficient low-rank algorithms, for the regularizer drives the convexity (see Theorem 5.3) and smoothness of the problem, and also make computationally cheap to evaluate

subgradients readily accessible (Table 5.2). We remark however that after we wrote [34], upon which this chapter is based, we discovered in [35] that this assumption can be relaxed to $\Omega(\boldsymbol{X}) = \operatorname{tr}(f(\boldsymbol{X}))$ for a matrix-convex function f; we invite the reader to read Chapter 6 for more details on this.

The two regularizers are matrix analogues of the popular big-M constraints (constraints on the ℓ_∞ norm of the continuous variables) and ridge regularization (penalty on the ℓ_2^2 norm) for vectors. In mixed-integer optimization, such regularization terms can efficiently cope with non-linear constraints between continuous and binary variables [33] and motivate our current approach. Practically speaking, regularization can be a natural component of the original problem (5.9), otherwise we advocate for introducing it artificially, for it leads to tractable algorithms with moderate impact on the resulting solution. For instance, if M is large enough so that the optimal solution to Problem (5.9), $\boldsymbol{X}^\star$, satisfies $\|\boldsymbol{X}^\star\|_\sigma \leq M$, Problems (5.10) and (5.9) are equivalent. With the Frobenius norm penalty, the gap between Problem (5.10)'s and (5.9)'s objective is at most $\frac{1}{2\gamma}\|\boldsymbol{X}^\star\|_F^2$, which can certainly be bounded whenever $\operatorname{tr}(\boldsymbol{X})$ is bounded, as often occurs in practice.

For ease of notation, we let

$$g(\boldsymbol{X}) = \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \begin{cases} 0, & \text{if } \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B},\ \boldsymbol{X} \in \mathcal{K}, \\ +\infty, & \text{otherwise,} \end{cases}$$

denote the unregularized second-stage cost for a given $\boldsymbol{X}$. Therefore, Problem (5.10) can be written as:

$$\min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \ f(\boldsymbol{Y}) + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}), \tag{5.11}$$

$$\text{where } f(\boldsymbol{Y}) := \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} g(\boldsymbol{X}) + \Omega(\boldsymbol{X}) \quad \text{s.t.} \quad \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X} \tag{5.12}$$

yields a best choice of $\boldsymbol{X}$ given $\boldsymbol{Y}$. As we establish in this section, this turns out to be a computationally useful reformulation, for f is convex in $\boldsymbol{Y}$ (see Theorem 5.3),

and therefore the non-convexity in the problem has been isolated within the set $\mathcal{Y}_n^k$.

Observe that both regularizers are coercive (i.e., "blow up" to $+\infty$ as $\|\boldsymbol{X}\| \to \infty$), and therefore render all unbounded solutions infeasible and ensure the compactness of the level sets of $\boldsymbol{X} \mapsto g(\boldsymbol{X}) + \Omega(\boldsymbol{X})$. This alleviates two of the major issues with conic duality [17, Theorem 2.4.1]. First, regularization ensures that optimal solutions to conic problems are attained [see 46, Example 2.27, for a regularization-free counterexample]. Second, regularization ensures that infeasibility of a conic system is *certifiable*[1], i.e., there is either a feasible solution or a certificate of infeasibility. In general, such a procedure is not possible because a conic system could be infeasible but asymptotically feasible, i.e.,

$$\nexists \boldsymbol{X} : \boldsymbol{A}\boldsymbol{X} = \boldsymbol{B}, \boldsymbol{X} \in \mathcal{K} \text{ but } \exists \{\boldsymbol{X}_t\}_{t=1}^{\infty} : \boldsymbol{X}_t \in \mathcal{K} \quad \forall t \text{ with } \|\boldsymbol{A}\boldsymbol{X}_t - \boldsymbol{B}\| \to 0.$$

Here, the regularization term ensures that the set of feasible $\boldsymbol{X}$ (with objective at most $\theta_0 \in \mathbb{R}$) is a closed convex compact set. Therefore, $f(\boldsymbol{Y})$ cannot generate an asymptotically feasible problem.

Finally, the two regularization functions in Assumption 5.1 satisfy a non-trivial property which turns out to be crucial in both proving that $f(\boldsymbol{Y})$ is convex and deriving our overall algorithmic strategy (see [34] for a proof):

Lemma 5.2. *Consider a regularization function $\Omega(\boldsymbol{X})$ satisfying Assumption 5.1. There, there exists a Fenchel conjugate $\Omega^\star$ [see, e.g., 54, Chap. 3.3.1] such that, for any projection matrix $\boldsymbol{Y} \in \mathcal{Y}_n$ and any matrix $\boldsymbol{\alpha}$, we have*

$$\min_{\boldsymbol{X}} \{\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X} \rangle\} = \max_{\boldsymbol{V}_{11}, \boldsymbol{V}_{22}} -\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}),$$

and $\Omega^\star$ is linear in $\boldsymbol{Y}$ (see Table 5.2 for its explicit definition).

[1]Unless the conic dual is also infeasible, this case is unimportant for our purposes, because it only arises when the original problem is itself infeasible for any $\boldsymbol{Y}$, which can be checked a priori.

Table 5.2: Regularizers and conjugates, as defined in Lemma 5.2.

$\Omega(\boldsymbol{X})$	$\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$	$\frac{\partial}{\partial Y_{i,j}}\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$
$\begin{cases} 0, & \text{if } \|\boldsymbol{X}\|_\sigma \leq M, \\ +\infty, & \text{o.w.}, \end{cases}$	$\frac{M}{2}\langle \boldsymbol{Y}, \boldsymbol{V}_{11}\rangle + \frac{M}{2}\langle \boldsymbol{I}_m, \boldsymbol{V}_{22}\rangle$ s.t. $\begin{pmatrix} \boldsymbol{V}_{11} & \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^\top & \boldsymbol{V}_{22} \end{pmatrix} \succeq \boldsymbol{0}$,	$\frac{M}{2}V_{11,i,j}$.
$\begin{cases} 0, & \text{if } \|\boldsymbol{X}\|_\sigma \leq M, \\ +\infty, & \text{o.w.}, \end{cases}$	$M\langle \boldsymbol{Y}, \boldsymbol{V}_{11} + \boldsymbol{V}_{22}\rangle$ s.t. $\boldsymbol{\alpha} = \boldsymbol{V}_{11} - \boldsymbol{V}_{22}$, $\boldsymbol{V}_{11}, \boldsymbol{V}_{2,2} \succeq \boldsymbol{0}$,	$M(V_{11} + V_{22})_{i,j}$.
$\frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2$	$\frac{\gamma}{2}\langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\alpha}\rangle$	$\frac{\gamma}{2}\langle \boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j\rangle$

Proof. We start with the Frobenius regularization case, $\Omega(\boldsymbol{X}) = \frac{1}{2\gamma}\|\boldsymbol{X}\|_F$ and

$$\min_{\boldsymbol{X}} \{\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X}\rangle\} = \tfrac{1}{2\gamma}\|\boldsymbol{Y}\boldsymbol{X}\|_F + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X}\rangle.$$

Any solution to the minimization problem satisfies the first-order condition $\frac{1}{\gamma}\boldsymbol{Y}\boldsymbol{X} + \boldsymbol{Y}\boldsymbol{\alpha} = 0$. Hence, since $\boldsymbol{Y}^2 = Y$, $\boldsymbol{X}^\star = -\gamma\boldsymbol{Y}\boldsymbol{\alpha}$ satisfies the first-order condition and the optimal objective value is $-\Omega^\star(\alpha, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}) = -\frac{\gamma}{2}\langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\alpha}\rangle$.

The spectral case is technically more challenging and detailed proofs are deferred to [Section E.C.2 34]. In the rectangular case, one can show that

$$\min_{\boldsymbol{X}} \{\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X}\rangle\} = \max_{\boldsymbol{V}_{11}, \boldsymbol{V}_{22}} -\frac{M}{2}\langle \boldsymbol{Y}, \boldsymbol{V}_{11}\rangle + \frac{M}{2}\langle \boldsymbol{I}_m, \boldsymbol{V}_{22}\rangle \text{ s.t. } \begin{pmatrix} \boldsymbol{V}_{11} & \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^\top & \boldsymbol{V}_{22} \end{pmatrix} \succeq \boldsymbol{0}.$$

In the symmetric case, one can show that

$$\min_{\boldsymbol{X}} \{\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X}\rangle\} = \max_{\boldsymbol{V}_{11}, \boldsymbol{V}_{22} \succeq \boldsymbol{0}} -M\langle \boldsymbol{Y}, \boldsymbol{V}_{11} + \boldsymbol{V}_{22}\rangle \text{ s.t. } \boldsymbol{\alpha} = \boldsymbol{V}_{11} - \boldsymbol{V}_{22}. \square$$

A Saddle-Point Reformulation

We now reformulate Problem (5.10) as a saddle-point problem. This reformulation is significant for two reasons. First, it leverages the nonlinear constraint $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ by introducing a new matrix of variables $\boldsymbol{V} \in \mathbb{R}^{n \times m}$ such that $\boldsymbol{V} = \boldsymbol{Y}\boldsymbol{X}$, giving:

$$f(\boldsymbol{Y}) = \min_{\boldsymbol{V}, \boldsymbol{X}} \{g(\boldsymbol{V}) + \Omega(\boldsymbol{Y}\boldsymbol{X}) : \boldsymbol{V} = \boldsymbol{Y}\boldsymbol{X}\}.$$

a substitution reminiscent of the Douglas-Rachford splitting technique for composite convex optimization problems [90, 94]. Second, it proves the regularizer $\Omega(\boldsymbol{X})$ drives the convexity and smoothness of $f(\boldsymbol{Y})$. To derive the problem's dual, we require:

Assumption 5.2. *For each subproblem* (5.12) *generated by* $f(\boldsymbol{Y})$ *where* $\boldsymbol{Y} \in \mathcal{Y}_n^k$, *either the optimization problem is infeasible, or strong duality holds.*

Assumption 5.2 holds under Slater's condition [54, Section 5.2.3]. By invoking Assumption 5.2, the following theorem reformulates (5.11) as a saddle-point problem:

Theorem 5.3. *Suppose Assumption 5.2 holds and* $\Omega(\cdot)$ *is either the spectral or Frobenius regularizer. Then, the following two optimization problems are equivalent:*

$$f(\boldsymbol{Y}) := \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} \quad g(\boldsymbol{X}) + \Omega(\boldsymbol{X}) \quad \text{s.t.} \quad \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}, \tag{5.13}$$

$$= \max_{\boldsymbol{\alpha}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}} \quad h(\boldsymbol{\alpha}) - \Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}), \tag{5.14}$$

where $h(\boldsymbol{\alpha}) := \max_{\boldsymbol{\Pi} : \boldsymbol{C} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\Pi} \in \mathcal{K}^\star} \langle \boldsymbol{b}, \boldsymbol{\Pi} \rangle$, $\mathcal{K}^\star := \{\boldsymbol{W} : \langle \boldsymbol{W}, \boldsymbol{X} \rangle \geq 0 \quad \forall \boldsymbol{X} \in \mathcal{K}\}$ *denotes the dual cone to* $\mathcal{K}$, *and* $\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$ *is defined in Table 5.2.*

Proof. Let us fix $\boldsymbol{Y} \in \mathcal{Y}_n^k$, and suppose that strong duality holds for the inner minimization problem which defines $f(\boldsymbol{Y})$. To progress, we introduce a matrix $\boldsymbol{V} \in \mathbb{R}^{n \times m}$ such that $\boldsymbol{V} = \boldsymbol{Y}\boldsymbol{X}$ and obtain the relaxation:

$$\min_{\boldsymbol{X}, \boldsymbol{V}} \; g(\boldsymbol{V}) + \Omega(\boldsymbol{Y}\boldsymbol{X}) \quad \text{s.t. } \boldsymbol{V} = \boldsymbol{Y}\boldsymbol{X}. \tag{5.15}$$

Let us verify that this relaxation is a valid substitution, i.e., that Problems (5.13) and (5.15) have the same optimal objective, $f(\boldsymbol{Y})$. If $\boldsymbol{X}$ is feasible for (5.13), then $(\boldsymbol{V} = \boldsymbol{X}, \boldsymbol{X})$ is obviously feasible for (5.15) with same objective value. Similarly, let $(\boldsymbol{V}, \boldsymbol{X})$ be feasible for (5.15). $\boldsymbol{Y}\boldsymbol{V} = \boldsymbol{Y}^2\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X} = \boldsymbol{V}$ since $\boldsymbol{Y}^2 = \boldsymbol{Y}$. Hence, $\boldsymbol{V}$ is feasible for (5.13) with same objective value.

Now, let $\boldsymbol{\alpha}$ denote the dual variables associated with the coupling constraints $\boldsymbol{V} = \boldsymbol{Y}\boldsymbol{X}$. The minimization problem is then equivalent to its dual, which is:

$$f(\boldsymbol{Y}) = \max_{\boldsymbol{\alpha}} \; h(\boldsymbol{\alpha}) + \min_{\boldsymbol{X}} \left[\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X} \rangle\right],$$

where $h(\boldsymbol{\alpha}) := \inf_{\boldsymbol{V}} g(\boldsymbol{V}) - \langle \boldsymbol{V}, \boldsymbol{\alpha} \rangle$ is, up to a sign, the Fenchel conjugate of g. By a standard application of Fenchel duality, it follows that

$$h(\boldsymbol{\alpha}) = \max_{\boldsymbol{\Pi}} \langle \boldsymbol{b}, \boldsymbol{\Pi} \rangle + \begin{cases} 0, & \text{if } \boldsymbol{C} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\Pi} \in \mathcal{K}^\star, \\ +\infty, & \text{otherwise.} \end{cases}$$

Finally, from Lemma 5.2 we have

$$\min_{\boldsymbol{X}} \left\{\Omega(\boldsymbol{Y}\boldsymbol{X}) + \langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{X} \rangle\right\} = \max_{\boldsymbol{V}_{11}, \boldsymbol{V}_{22}} \; -\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}),$$

which concludes the proof. Alternatively, under either penalty, if the inner problem defining $f(\boldsymbol{Y})$ is infeasible, then its dual problem is unbounded by weak duality. □

Remark 11. *In the unregularized case $\Omega(\boldsymbol{X}) = 0$, we can derive the reformulation:*

$$\min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \quad \max_{\boldsymbol{\alpha} \in \mathbb{R}^{n \times m}} \quad h(\boldsymbol{\alpha}) + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}) \; \textit{s.t.} \quad \boldsymbol{Y}\boldsymbol{\alpha} = \boldsymbol{0}. \tag{5.16}$$

Under this lens, regularization of the primal problem is equivalent to a relaxation in the dual formulation: the hard constraint $\boldsymbol{Y}\boldsymbol{\alpha} = \boldsymbol{0}$ is penalized by $-\Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$.

Remark 12. *By Theorem 5.3 and Lemma 5.2, $f(\boldsymbol{Y})$ is convex as the point-wise maximum of functions which are linear in $\boldsymbol{Y}$.*

By Theorem 5.3, when we evaluate $f(\hat{\boldsymbol{Y}})$, one of two alternatives occur. The first is that we have $f(\hat{\boldsymbol{Y}}) < +\infty$ and there is some optimal $(\boldsymbol{\alpha}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$. In this case, we construct the lower approximation

$$f(\boldsymbol{Y}) \geq f(\hat{\boldsymbol{Y}}) + \langle \boldsymbol{H}, \boldsymbol{Y} - \hat{\boldsymbol{Y}} \rangle,$$

where $H_{i,j} = \frac{\partial}{\partial Y_{i,j}} \Omega^\star(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22})$ (see Table 5.2 for closed-form expression of the partial derivatives.The second alternative is that $f(\hat{\boldsymbol{Y}}) = +\infty$, in which case, by the conic duality theorem [see 17, Chapter 2] there exists a $(\boldsymbol{\alpha}, \boldsymbol{\Pi})$ such that

$$\boldsymbol{C} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\Pi} \in \mathcal{K}^\star, \text{ and } \langle \boldsymbol{b}, \boldsymbol{\Pi} \rangle > \langle -\boldsymbol{H}, \hat{\boldsymbol{Y}} \rangle. \tag{5.17}$$

Under this alternative, we can separate $\hat{\boldsymbol{Y}}$ from the set of feasible $\boldsymbol{Y}$'s by imposing the cut $0 \geq \langle \boldsymbol{b}, \boldsymbol{\Pi} \rangle + \langle \boldsymbol{H}, \boldsymbol{Y} \rangle$. Under either alternative, we obtain a globally valid first-order underestimator of the form

$$z f(\boldsymbol{Y}) \geq h + \langle \boldsymbol{H}, \boldsymbol{Y} - \hat{\boldsymbol{Y}} \rangle, \tag{5.18}$$

where z, h are defined as

$$z = \begin{cases} 1, & \text{if } f(\hat{\boldsymbol{Y}}) < +\infty, \\ 0, & \text{if } f(\hat{\boldsymbol{Y}}) = +\infty, \end{cases} \quad \text{and} \quad h = \begin{cases} f(\hat{\boldsymbol{Y}}), & \text{if } f(\hat{\boldsymbol{Y}}) < +\infty, \\ \langle \boldsymbol{b}, \boldsymbol{\Pi} \rangle + \langle \boldsymbol{H}, \hat{\boldsymbol{Y}} \rangle, & \text{if } f(\hat{\boldsymbol{Y}}) = +\infty. \end{cases} \tag{5.19}$$

This observation suggests that a valid numerical strategy for minimizing $f(\boldsymbol{Y})$ is to iteratively minimize and refine a piecewise linear underestimator of $f(\boldsymbol{Y})$ defined by the pointwise supremum of a finite number of underestimators of the form $z f(\boldsymbol{Y}) \geq h + \langle \boldsymbol{H}, \boldsymbol{Y} - \boldsymbol{Y} \rangle$. Indeed, as we will see in Section 5.4, this strategy gives rise to the global optimization algorithm known as outer-approximation.

Smoothness We now demonstrate that $f(\boldsymbol{Y})$ is smooth, in the sense of Lipschitz continuity, under a boundedness assumption on the size of the dual variables, which is a crucial property for ensuring the convergence of our global optimization methods and bounding the quality of our semidefinite relaxation and greedy rounding methods. Formally, the following result follows directly from Theorem 5.3.

Lemma 5.3. *Let $\boldsymbol{Y}, \boldsymbol{Y}' \in \mathrm{Conv}(\mathcal{Y}_n^k)$ be on the convex hull of orthogonal projections. Then*

$$f(\boldsymbol{Y}) - f(\boldsymbol{Y}') \leq \Omega^\star(\boldsymbol{\alpha}^\star(\boldsymbol{Y}), \boldsymbol{Y}' - \boldsymbol{Y}, \boldsymbol{V}_{11}^\star(\boldsymbol{Y}), \boldsymbol{V}_{22}^\star(\boldsymbol{Y})).$$

Moreover, suppose $\boldsymbol{\alpha}^\star(\boldsymbol{Y}), \boldsymbol{V}_{11}^\star(\boldsymbol{Y}), \boldsymbol{V}_{22}^\star(\boldsymbol{Y})$ can be bounded independently from $\boldsymbol{Y}$, i.e., $\|\boldsymbol{\alpha}^\star(\boldsymbol{Y})\|_\sigma \leq L_1$, $\|\boldsymbol{V}_{11}^\star(\boldsymbol{Y})\|_\sigma \leq L_2$, $\|\boldsymbol{V}_{22}^\star(\boldsymbol{Y})\|_\sigma \leq L_2$. Then, under spectral regularization we have

$$f(\boldsymbol{Y}) - f(\boldsymbol{Y}') \leq M\langle \boldsymbol{V}_{11}^\star(\boldsymbol{Y}), \boldsymbol{Y}' - \boldsymbol{Y}\rangle \leq ML_2\|\boldsymbol{Y}' - \boldsymbol{Y}\|_*, \tag{5.20}$$

and under Frobenius regularization we have

$$f(\boldsymbol{Y}) - f(\boldsymbol{Y}') \leq \frac{\gamma}{2}\langle \boldsymbol{\alpha}^{\star\top}(\boldsymbol{Y})\boldsymbol{\alpha}^\star(\boldsymbol{Y}), \boldsymbol{Y}' - \boldsymbol{Y}\rangle \leq \frac{\gamma}{2}L_1^2\|\boldsymbol{Y}' - \boldsymbol{Y}\|_*, \tag{5.21}$$

where the bounds involving L_1, L_2 follow from Holder's inequality[2]*.*

Remark 13. *In the paper this chapter is based upon, we develop disciplined techniques for computing an M such that the constraint $\|\boldsymbol{X}\|_\sigma \leq M$ does not alter the optimal objective [34, Section E.C.5.1]. The same technique, applied to the dual, yields explicit bounds on L_1. Moreover, since there exists an optimal pair $(\boldsymbol{V}_{11}, \boldsymbol{V}_{22})$ which is an explicit functions of an optimal $\boldsymbol{\alpha}$, this translates into explicit bounds on L_2.*

[2]Namely, $|\langle \boldsymbol{X}, \boldsymbol{Y}\rangle| \leq \|\boldsymbol{X}\|_\sigma\|\boldsymbol{Y}\|_*$, since the $\|\cdot\|_\sigma$ and $\|\cdot\|_*$, as the matrix analogs of the ℓ_∞ and ℓ_1 norms, are dual.

Semidefinite Relaxations

To bound (5.11), we invoke Lemma 5.1 to relax the non-convex constraint $\boldsymbol{Y} \in \mathcal{Y}_n^k$ to

$$\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n^k\right) = \{\boldsymbol{Y} \in S^n : \boldsymbol{0} \preceq \boldsymbol{Y} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{Y}) \leq k\}.$$

This yields the saddle-point problem

$$\min_{\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \max_{\boldsymbol{\alpha}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22} \in S^m} h(\boldsymbol{\alpha}) - \Omega^\star\left(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}\right) + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}). \tag{5.22}$$

Problem (5.22) can in turn be reformulated as an SDO. Indeed, under Assumption 5.2, we obtain a semidefinite formulation by taking Problem (5.22)'s dual with respect to $\boldsymbol{\alpha}$. Formally, we have the following results:

Lemma 5.4. *Suppose Assumption 5.2 holds. Then, strong duality holds between:*

$$\min_{\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \max_{\boldsymbol{\alpha} \in \mathbb{R}^{n \times m}} \quad h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\alpha} \rangle + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}), \tag{5.23}$$

$$\min_{\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}, \boldsymbol{\theta} \in S^n} \quad g(\boldsymbol{X}) + \frac{1}{2\gamma}\mathrm{tr}\left(\boldsymbol{\theta}\right) + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}) \ \textit{s.t.} \ \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0}. \tag{5.24}$$

Proof. Let us fix $\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n\right)$. Then, we have that:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\left\langle \boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\alpha} \right\rangle &= \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\left\langle \boldsymbol{\beta}, \boldsymbol{Y}\boldsymbol{\beta} \right\rangle \ \text{s.t.} \ \boldsymbol{\beta} = \boldsymbol{\alpha}, \\
&= \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \min_{\boldsymbol{X}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\left\langle \boldsymbol{\beta}, \boldsymbol{Y}\boldsymbol{\beta} \right\rangle - \left\langle \boldsymbol{X}, \boldsymbol{\beta} - \boldsymbol{\alpha} \right\rangle, \\
&= \min_{\boldsymbol{X}} \underbrace{\max_{\boldsymbol{\alpha}} \left[h(\boldsymbol{\alpha}) + \langle \boldsymbol{X}, \boldsymbol{\alpha} \rangle\right]}_{(-h)^\star(\boldsymbol{X}) = g(\boldsymbol{X})} + \max_{\boldsymbol{\beta}} \left[\frac{-\gamma}{2}\left\langle \boldsymbol{Y}\boldsymbol{\beta}, \boldsymbol{\beta} \right\rangle - \left\langle \boldsymbol{X}, \boldsymbol{\beta} \right\rangle\right].
\end{aligned}$$

Finally, the optimality condition with respect to $\boldsymbol{\beta}$ is $\boldsymbol{Y}\boldsymbol{\beta} = \frac{-1}{\gamma}\boldsymbol{X}$, which implies

$$\max_{\boldsymbol{W}} \left[\frac{1}{2\gamma}\langle \boldsymbol{X}, \boldsymbol{Y}^\dagger \boldsymbol{X} \rangle - \frac{1}{2}\langle \boldsymbol{X}, (\mathbb{I} - \boldsymbol{Y}^\dagger \boldsymbol{Y})\boldsymbol{W} \rangle\right]$$

$$
\begin{aligned}
&= \max_{\boldsymbol{W}} \left[\frac{1}{2\gamma} \langle \boldsymbol{X}, \boldsymbol{Y}^\dagger \boldsymbol{X} \rangle - \frac{1}{2} \langle \boldsymbol{W}, (\mathbb{I} - \boldsymbol{Y}^\dagger \boldsymbol{Y}) \boldsymbol{X} \rangle \right] \\
&= \begin{cases} \dfrac{1}{2\gamma} \left\langle \boldsymbol{X}, \boldsymbol{Y}^\dagger \boldsymbol{X} \right\rangle & \text{if } \boldsymbol{Y} \in \operatorname{Span}(\boldsymbol{X}), \\ +\infty & \text{otherwise.} \end{cases}
\end{aligned}
$$

We therefore conclude that the later term is equal to $\frac{1}{2\gamma} \left\langle \boldsymbol{X}, \boldsymbol{Y}^\dagger \boldsymbol{X} \right\rangle$ whenever the constraint $\boldsymbol{Y}^\dagger \boldsymbol{Y} \boldsymbol{X} = \boldsymbol{X}$ holds. By the generalized Schur complement lemma, this expression is equivalent to introducing a new matrix $\boldsymbol{\theta}$, imposing the term $\frac{1}{2\gamma} \operatorname{tr}(\boldsymbol{\theta})$ and requiring that $\begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0}$. □

Lemma 5.5. *Suppose Assumption 5.2 holds. Then, strong duality holds between:*

$$
\begin{aligned}
\min_{\boldsymbol{Y} \in \operatorname{Conv}\left(\mathcal{Y}_n^k\right)} \max_{\boldsymbol{\alpha} \in S^n, \boldsymbol{V}_{11}, \boldsymbol{V}_{22} \succeq \boldsymbol{0}} \quad & h(\boldsymbol{\alpha}) - M \langle \boldsymbol{Y}, \boldsymbol{V}_{11} + \boldsymbol{V}_{22} \rangle + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}) && (5.25) \\
\text{s.t.} \quad & \boldsymbol{\alpha} = \boldsymbol{V}_{11} - \boldsymbol{V}_{22}, \\
\min_{\boldsymbol{Y} \in \operatorname{Conv}\left(\mathcal{Y}_n^k\right)} \min_{\boldsymbol{X} \in S^n} \quad & g(\boldsymbol{X}) + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}) && (5.26) \\
\text{s.t.} \quad & -M\boldsymbol{Y} \preceq \boldsymbol{X} \preceq M\boldsymbol{Y}.
\end{aligned}
$$

Proof. Let us fix $\boldsymbol{Y} \in \operatorname{Conv}(\mathcal{Y}_n)$. Then, we have that:

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha} \in S^n, \boldsymbol{W}_+, \boldsymbol{W}_- \succeq \boldsymbol{0}} h(\boldsymbol{\alpha}) - M \langle \boldsymbol{Y}, \boldsymbol{W}_+ - \boldsymbol{W}_- \rangle \quad \text{s.t.} \quad \boldsymbol{\alpha} = \boldsymbol{W}_+ - \boldsymbol{W}_- \\
&= \max_{\boldsymbol{\alpha} \in S^n, \boldsymbol{W}_+, \boldsymbol{W}_- \succeq \boldsymbol{0}} \min_{\boldsymbol{X} \in S^n} h(\boldsymbol{\alpha}) - M \langle \boldsymbol{Y}, \boldsymbol{W}_+ - \boldsymbol{W}_- \rangle + \langle \boldsymbol{X}, \boldsymbol{\alpha} - \boldsymbol{W}_+ + \boldsymbol{W}_- \rangle \\
&= \min_{\boldsymbol{X} \in S^n} \max_{\boldsymbol{\alpha} \in S^n, \boldsymbol{W}_+, \boldsymbol{W}_- \succeq \boldsymbol{0}} h(\boldsymbol{\alpha}) - M \langle \boldsymbol{Y}, \boldsymbol{W}_+ - \boldsymbol{W}_- \rangle + \langle \boldsymbol{X}, \boldsymbol{\alpha} - \boldsymbol{W}_+ + \boldsymbol{W}_- \rangle \\
&= \min_{\boldsymbol{X} \in S^n} \max_{\boldsymbol{\alpha} \in S^n} \underbrace{[h(\boldsymbol{\alpha}) + \langle \boldsymbol{X}, \boldsymbol{\alpha} \rangle]}_{(-h)^\star(\boldsymbol{X}) = g(\boldsymbol{X})} + \max_{\boldsymbol{W}_+, \boldsymbol{W}_- \succeq \boldsymbol{0}} \left[-M \langle \boldsymbol{Y}, \boldsymbol{W}_+ - \boldsymbol{W}_- \rangle + \langle \boldsymbol{X}, -\boldsymbol{W}_+ + \boldsymbol{W}_- \rangle \right].
\end{aligned}
$$

Finally, the optimality conditions with respect to $\boldsymbol{W}_+, \boldsymbol{W}_-$ imply

$$
-M\boldsymbol{Y} \preceq \boldsymbol{X} \preceq M\boldsymbol{Y}. \qquad \square
$$

We now offer some remarks on these bi-dual problems:

- We can derive a more general version of Lemma 5.5 without the symmetry assumption on $\boldsymbol{X}$ in much the same manner, via the Schur complement lemma.
- Problem (5.24)'s formulation generalizes the perspective relaxation from vectors to matrices. This suggests that (5.24) is an efficient formulation for addressing rank constraints, as perspective formulations efficiently address cardinality constrained problems with conic quadratic [126] or power cone [4] objectives, indeed, they provide a theoretical basis for scalable algorithms for sparse regression [36, 129], sparse portfolio selection [235, 24] and network design [106] problems among others..

Convex Penalty Interpretations of Relaxations

In this section, we consider instances where rank is penalized in the objective and interpret the convex relaxations as penalty functions, in the tradition of [103, 198].

With a spectral regularizer, the convex relaxation is equivalent to using the popular nuclear norm penalty. However, Zhang et al. [232] show that the nuclear norm cannot encourage low-rank solutions for problems with constraints $\boldsymbol{X} \succeq \mathbf{0}, \operatorname{tr}(\boldsymbol{X}) = k$, such as sparse PCA [80], k-means clustering [191]. In the presence of the Frobenius penalty, Lemma 5.6 exhibits an alternative to the nuclear norm penalty that can encourage low-rank solutions in these situations. Our result generalizes the *reverse Huber penalty* of [193, 89] from cardinality to rank objectives.

Lemma 5.6. *Let Assumption 5.2 hold. Then, the following problems are equivalent:*

$$\min_{\boldsymbol{Y} \in \operatorname{Conv}(\mathcal{Y}_n)} \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}, \boldsymbol{\theta} \in S^n} \quad g(\boldsymbol{X}) + \frac{1}{2\gamma} \operatorname{tr}(\boldsymbol{\theta}) + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}) \ \ s.t. \ \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \mathbf{0}, \quad (5.27)$$

$$\min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}} \quad g(\boldsymbol{X}) + \sum_{i=1}^{n} \min \left(\sqrt{\frac{2\lambda}{\gamma}} \sigma_i(\boldsymbol{X}), \lambda + \frac{\sigma_i(\boldsymbol{X})^2}{2\gamma} \right). \quad (5.28)$$

Remark 14.

$$\textit{Since} \quad \min_{0\le\theta\le 1}\left[\lambda\theta + \frac{t^2}{\theta}\right] = \begin{cases} 2\sqrt{\lambda}|t|, & \textit{if } |t| \le \sqrt{\lambda}, \\ t^2 + \lambda, & \textit{otherwise,} \end{cases}$$

Problems (5.27)-(5.28) *are equivalent to minimizing*

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m},\boldsymbol{\theta}\in\mathbb{R}^n:\ \mathbf{0}\le\boldsymbol{\theta}\le\boldsymbol{e}} \quad g(\boldsymbol{X}) + \sum_{i=1}^{n}\left(\lambda\theta_i + \frac{\sigma_i(\boldsymbol{X})^2}{2\gamma\theta_i}\right), \tag{5.29}$$

which applies the smooth penalty $t \to \lambda\theta + \frac{t^2}{2\gamma\theta} : 0 \le \theta \le 1$ *to model the non-convex cost* $t \to \lambda\|t\|_0 + \frac{t^2}{2\gamma}$ *incurred by each singular value of* $\boldsymbol{X}$*. Indeed, this smooth penalty is precisely the convex envelope of the non-convex cost function [see, e.g., 126]. Compared to other penalties for rank problems [97, 231], this generalized Huber penalty is convex and could be of independent interest to the statistical learning community.*

Proof. Observe that, by the Generalized Schur Complement Lemma, an optimal choice of $\boldsymbol{\theta}$ in Problem (5.27) is $\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{Y}^\dagger\boldsymbol{X}^\top$. Therefore, we can eliminate $\boldsymbol{\theta}$ from Problem (5.27), to obtain the equivalent objective:

$$\min_{\boldsymbol{Y}\in\mathrm{Conv}(\mathcal{Y}_n)} \ \min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} \quad \lambda\cdot\mathrm{tr}(\boldsymbol{Y}) + g(\boldsymbol{X}) + \frac{1}{2\gamma}\langle\boldsymbol{X}\boldsymbol{X}^\top, \boldsymbol{Y}^\dagger\rangle.$$

Moreover, by the rank-nullity theorem [see, e.g., 134, Chapter 0.2.3], we can split the columns of $\boldsymbol{Y}$ into columns in the span of the columns of $\boldsymbol{X}$ and columns orthogonal to the columns of $\boldsymbol{X}$. Since the columns orthogonal to the columns of $\boldsymbol{X}$ do not affect the objective value, it follows that we can write $\boldsymbol{Y}^\dagger = \sum_{i=1}^{n}\frac{1}{\theta_i}\boldsymbol{u}_i\boldsymbol{u}_i^\top$ without loss of optimality, where $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$ is an SVD of $\boldsymbol{X}\boldsymbol{X}^\top$, and $0 \le \theta_i \le 1$ for each θ_i, because $\boldsymbol{Y} \in \mathrm{Conv}(\mathcal{Y}_n)$. Problem (5.27) then becomes:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m},\boldsymbol{\theta}\in\mathbb{R}^n:\ \mathbf{0}\le\boldsymbol{\theta}\le\boldsymbol{e}} \quad g(\boldsymbol{X}) + \sum_{i=1}^{n}\left(\lambda\theta_i + \frac{\sigma_i(\boldsymbol{X})^2}{2\gamma\theta_i}\right).$$

The result then follows because, for any $\lambda > 0$, [c.f. 193, Equation (30)]

$$\min_{0\le\theta\le 1}\left[\lambda\theta+\frac{t^2}{\theta}\right]=\begin{cases}2\sqrt{\lambda}|t|, & \text{if } |t|\le\sqrt{\lambda},\\ t^2+\lambda, & \text{otherwise.}\end{cases}\quad\square$$

Lemma 5.6 proposes an alternative to the nuclear norm penalty for approximately solving low-rank problems. This is significant, as many low-rank problems have constraints $\boldsymbol{X}\succeq\boldsymbol{0},\mathrm{tr}(\boldsymbol{X})=k$ (e.g. sparse PCA [80], k-means clustering [191]), and under these constraints a nuclear norm cannot encourage low-rank solutions [232].

Our next results relate rank minimization problems with a spectral regularizer to the nuclear norm penalty, in both the square symmetric and the rectangular case:

Lemma 5.7. *Suppose Assumption 5.2 holds. Then, the following are equivalent:*

$$\min_{\boldsymbol{Y}\in\mathrm{Conv}(\mathcal{Y}_n)}\ \min_{\boldsymbol{X}\in S^n}\quad g(\boldsymbol{X})+\lambda\cdot\mathrm{tr}(\boldsymbol{Y})\quad\text{s.t.}\quad -M\boldsymbol{Y}\preceq\boldsymbol{X}\preceq M\boldsymbol{Y}, \tag{5.30}$$

$$\min_{\boldsymbol{X}\in S^n}\quad g(\boldsymbol{X})+\frac{\lambda}{M}\|\boldsymbol{X}\|_*\quad\text{s.t.}\quad\|\boldsymbol{X}\|_\sigma\le M. \tag{5.31}$$

Proof. In Problem (5.30), it is not too hard to see that for any $\boldsymbol{X}$ an optimal choice of $\boldsymbol{Y}$ is $\boldsymbol{Y}=\frac{1}{M}\boldsymbol{X}_+ +\frac{1}{M}\boldsymbol{X}_-$, where $\boldsymbol{X}_+,\boldsymbol{X}_-$ are orthogonal positive semidefinite matrices such that $\boldsymbol{X}=\boldsymbol{X}_+-\boldsymbol{X}_-$. The result follows as $\mathrm{tr}(\boldsymbol{X}_++\boldsymbol{X}_-)=\|\boldsymbol{X}\|_*$. $\square$

Lemma 5.8. *Suppose Assumption 5.2 holds. Then, the following are equivalent:*

$$\min_{\boldsymbol{Y}\in\mathrm{Conv}(\mathcal{Y}_n),\boldsymbol{Y}'\in\mathrm{Conv}(\mathcal{Y}_m)}\ \min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}}\quad g(\boldsymbol{X})+\frac{\lambda}{2}\mathrm{tr}(\boldsymbol{Y})+\frac{\lambda}{2}\mathrm{tr}(\boldsymbol{Y}') \tag{5.32}$$

$$s.t.\quad\begin{pmatrix}M\boldsymbol{Y} & \boldsymbol{X}\\ \boldsymbol{X}^\top & M\boldsymbol{Y}'\end{pmatrix}\succeq\boldsymbol{0},$$

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}}\quad g(\boldsymbol{X})+\frac{\lambda}{M}\|\boldsymbol{X}\|_*\quad\text{s.t.}\quad\|\boldsymbol{X}\|_\sigma\le M. \tag{5.33}$$

Proof. In (5.32), for any feasible $\boldsymbol{X}$ we have $\|\boldsymbol{X}\|_\sigma\le M$. It follows that for any $\boldsymbol{X}$ an optimal choice of $\boldsymbol{Y},\boldsymbol{Y}'$ is $\boldsymbol{Y}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$, $\boldsymbol{Y}'=\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is an SVD of $\boldsymbol{X}$. The result follows as $\mathrm{tr}(\boldsymbol{Y})=\mathrm{tr}(\boldsymbol{\Sigma})=\|\boldsymbol{X}\|_*$. $\square$

5.4 Efficient Algorithmic Approaches

In this section, we present an efficient numerical approach to solve Problem (5.1) and its relaxations. The backbone is an outer-approximation strategy, embedded within a non-convex QCQO branch-and-bound procedure to solve the problem exactly. We also propose rounding heuristics to find good feasible solutions, and semidefinite free methods for optimizing over (5.1)'s relaxations.

The primary motivations for developing an outer-approximation procedure and solving mixed-projection problem as saddle-point problems are twofold. First, we are not aware of any solvers which address mixed-projection problems with semidefinite constraints. Instead, a decomposition strategy like outer-approximation can be readily implemented using `Gurobi` (to solve non-convex quadratically constrained master problems) and `Mosek` (to solve conic subproblems). Second, decomposition schemes for mixed-integer semidefinite problems typically outperform one-shot strategies [16], so we expect - and observe in Section 5.7 - a similar comparison for mixed-projection optimization, hence connecting the frameworks in theory (see Table 5.1) and practice.

A Globally Optimal Cutting-Plane Method

The analysis in the previous section reveals that evaluating $f(\boldsymbol{Y})$ yields a globally valid first-order underestimator of $f(\cdot)$. Therefore, a numerically efficient strategy for minimizing $f(\boldsymbol{Y})$ is to iteratively minimize and refine a piecewise linear underestimator of $f(\boldsymbol{Y})$. This strategy is known as outer-approximation (OA), and was originally proposed by Duran and Grossmann [92]. OA iteratively constructs underestimators of the following form at each iterate $t+1$:

$$f_{t+1}(\boldsymbol{Y}) = \max_{1\leq i\leq t} \left\{ f(\boldsymbol{Y}_i) + \langle \boldsymbol{H}_i, \boldsymbol{Y} - \boldsymbol{Y}_i \rangle \right\}. \tag{5.34}$$

By iteratively minimizing $f_{t+1}(\boldsymbol{Y})$ and imposing the resulting cuts when constructing the next underestimator, we obtain a non-decreasing sequence of underestimators

$f_t(\boldsymbol{Y}_t)$ and non-increasing sequence of overestimators $\min_{i\in[t]} f(\boldsymbol{Y}_i)$ which converge to an ϵ-optimal solution within a finite number of iterations; see also Section 5.3 for details on cut generation. Indeed, since $\text{Conv}\left(\mathcal{Y}_n^k\right)$ is a compact set and $f(\cdot)$ is an L-Lipschitz continuous function in $\boldsymbol{Y}$, OA never visits a ball of radius $\frac{\epsilon}{L}$ twice.

We now formalize this procedure in Algorithm 5.1, and state its properties:

Algorithm 5.1 An outer-approximation method for Problem (5.11)

Require: Initial solution $\boldsymbol{Y}_1$
  $t \leftarrow 1$
  repeat
    Compute $\boldsymbol{Y}_{t+1}, \theta_{t+1}$ solution of

$$\min_{\boldsymbol{Y}\in\mathcal{Y}_n^k,\theta} \theta + \lambda \cdot \text{tr}(\boldsymbol{Y}) \quad \text{s.t. } z_i\theta \geq h_i + \langle \boldsymbol{H}_i, \boldsymbol{Y} - \boldsymbol{Y}_i\rangle \quad \forall i \in [t].$$

    Compute $f(\boldsymbol{Y}_{t+1})$, $\boldsymbol{H}_{t+1}$, z_{t+1}, d_{t+1}
  until $f(\boldsymbol{Y}_t) - \theta_t \leq \varepsilon$ **return** $\boldsymbol{Y}_t$

Theorem 5.4. *Suppose Assumptions 5.1-5.2 hold, and that there exists some Lipschitz constant L such that for any feasible $\boldsymbol{Y}, \boldsymbol{Y}' \in \text{Conv}(\mathcal{Y}_n^k)$ we have: $|f(\boldsymbol{Y}) - f(\boldsymbol{Y}')| \leq L\|\boldsymbol{Y} - \boldsymbol{Y}'\|_F$, and for any feasibility cut $\langle \boldsymbol{H}_i, \boldsymbol{Y} - \boldsymbol{Y}_i\rangle + h_i \leq 0$ we have $|\langle \boldsymbol{H}_i, \boldsymbol{Y} - \boldsymbol{Y}'\rangle| \leq L\|\boldsymbol{Y} - \boldsymbol{Y}'\|_F$. Let $\boldsymbol{Y}_t \in \mathcal{Y}_n^k$ be a feasible solution returned by the t^{th} iterate of Algorithm 5.1, where*

$$t \geq \left(\frac{Lk}{\epsilon} + 1\right)^{n^2}.$$

Then, $\boldsymbol{Y}_t$ is an ϵ-optimal and ϵ-feasible solution to Problem (5.10)*. Moreover, suppose that we set $\epsilon \to 0$. Then, any limit point of $\{\boldsymbol{Y}_t\}_{t=1}^{\infty}$ solves* (5.10)*.*

Proof. We detail the ϵ-optimality case; the proof of ϵ-feasibility is identical [179].

Suppose that at some iteration $k > 1$, Algorithm 5.1 has not converged. Then,

$$\theta_k - f(\boldsymbol{Y}_k) < -\epsilon, \qquad \text{and} \quad \theta_k \geq f(\boldsymbol{Y}_i) + \langle \boldsymbol{H}_i, \boldsymbol{Y}_k - \boldsymbol{Y}_i\rangle \ \forall i < k.$$

But $\theta_k \leq f(\boldsymbol{Y}_i)$, since θ_k and $f(\boldsymbol{Y}_i)$ are respectively valid lower and upper bounds on the optimal objective. Therefore, $\langle \boldsymbol{H}_i, \boldsymbol{Y}_k - \boldsymbol{Y}_i\rangle \geq 0$. Putting the two inequalities

together then implies that

$$f(\boldsymbol{Y}_k)-f(\boldsymbol{Y}_i) > \epsilon+\langle H_i, \boldsymbol{Y}_k-\boldsymbol{Y}_i\rangle \geq \epsilon, \text{ or equivalently } \quad \epsilon < f(\boldsymbol{Y}_k)-f(\boldsymbol{Y}_i) \leq L\|\boldsymbol{Y}_i-\boldsymbol{Y}_k\|_F,$$

where the second inequality holds by Lipschitz continuity. Rearranging this inequality implies that $\|\boldsymbol{Y}_i - \boldsymbol{Y}_k\|_F > \frac{\epsilon}{L}$, i.e., Algorithm 5.1 never visits any point within a ball of radius $\frac{\epsilon}{L}$ (with respect to the Frobenius norm) twice. Moreover, by iteration k, Algorithm 5.1 visits k points within non-overlapping balls with combined volume

$$k\frac{\pi^{\frac{n^2}{2}}}{\Gamma(\frac{n^2}{2}+1)}\left(\frac{\epsilon}{L}\right)^{n^2},$$

and these balls are centered at feasible points, i.e., contained within a ball of radius $K + \frac{\epsilon}{L}$ with volume

$$\frac{\pi^{\frac{n^2}{2}}}{\Gamma(\frac{n^2}{2}+1)}\left(K+\frac{\epsilon}{L}\right)^{n^2}.$$

That is, if Algorithm 5.1 has not converged at iteration k, we have: $k < \left(\frac{LK}{\epsilon}+1\right)^{n^2}$, which implies we converge to an ϵ-optimal solution in $k \leq \left(\frac{LK}{\epsilon}+1\right)^{n^2}$ iterations. $\square$

Optimizing over orthogonal projection matrices

To successfully implement Algorithm 5.1, we need to repeatedly solve optimization problems of the form

$$\min_{\boldsymbol{Y}\in\mathcal{Y}_n^k,\theta} \theta + \lambda\cdot\operatorname{tr}(\boldsymbol{Y}) \quad \text{s.t. } z_i\theta \geq h_i + \langle \boldsymbol{H}_i, \boldsymbol{Y}-\boldsymbol{Y}_i\rangle \quad \forall i\in[t], \tag{5.35}$$

which requires a tractable representation of $\mathcal{Y}_n^k$. Fortunately, `Gurobi` 9.0 contains a globally optimal spatial branch-and-bound method for general QCQOs which recursively partitions the feasible region into boxes and invokes the ubiquitous McCormick inequalities to obtain valid upper and lower bounds on each box—see Achterberg and Towle [1] for a discussion of Gurobi's bilinear solver, Belotti et al. [16] for a general theory of spatial branch-and-bound. Therefore, we represent $\boldsymbol{Y}$ by introducing a matrix $\boldsymbol{U} \in \mathbb{R}^{n\times k}$ and requiring that $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}$. This

allows Algorithm 5.1 to be implemented by iteratively solving a sequence of QCQOs and conic optimization problems. Moreover, to decrease the amount of branching required in each iteration of Algorithm 5.1, we impose an outer-approximation of the valid constraint $\boldsymbol{Y} \succeq \boldsymbol{U}\boldsymbol{U}^\top$. Specifically, we strengthen the formulation by imposing second-order cone relaxations of the PSD constraint. First, we require that the 2×2 minors in $\boldsymbol{Y}$ are non-negative , i.e., $Y_{i,j}^2 \leq Y_{i,i}Y_{j,j}\ \forall i, j \in [n]$, as proposed in [2, 23]. Second, we require that the on-diagonal entries of $\boldsymbol{Y} \succeq \boldsymbol{U}\boldsymbol{U}^\top$ are non-negative i.e., $Y_{i,i} \geq \sum_{i=1}^{k} U_{i,t}^2\ \forall i \in [n]$. Finally, we follow Atamtürk and Gomez [7, Proposition 5] in taking a second-order cone approximation of the 2×2 minors in $\boldsymbol{Y} \succeq \boldsymbol{U}\boldsymbol{U}^\top$ i.e., $0 \geq \|\boldsymbol{U}_i \pm \boldsymbol{U}_j\|_2^2 \pm 2Y_{i,j} - Y_{i,i} - Y_{j,j},\ \forall i, j \in [n]$. All told, we have[3]:

$$
\begin{aligned}
\min_{\boldsymbol{Y} \in S^n, \boldsymbol{U} \in \mathbb{R}^{n \times k}, \theta} \quad & \theta + \lambda \cdot \operatorname{tr}(\boldsymbol{Y}) \quad \text{s.t. } z_i\theta \geq h_i + \langle \boldsymbol{H}_i, \boldsymbol{Y} - \boldsymbol{Y}_i \rangle \quad \forall i \in [t], \\
& \boldsymbol{Y} = \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}, Y_{i,i}Y_{j,j} \geq Y_{i,j}^2\ \forall i, j \in [n], \\
& Y_{i,i} \geq \sum_{t=1}^{k} U_{i,t}^2\ \forall i \in [n], \operatorname{tr}(\boldsymbol{Y}) = k, \\
& 0 \geq \|\boldsymbol{U}_i + \boldsymbol{U}_j\|_2^2 - 2Y_{i,j} - Y_{i,i} - Y_{j,j}, \\
& 0 \geq \|\boldsymbol{U}_i - \boldsymbol{U}_j\|_2^2 + 2Y_{i,j} - Y_{i,i} - Y_{j,j} \quad \forall i, j \in [n].
\end{aligned}
\tag{5.36}
$$

Finally, for a given $\boldsymbol{Y}, \boldsymbol{U}$, we strengthen this formulation by imposing second-order cone cuts of the form $\langle \boldsymbol{Y} - \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{u}\boldsymbol{u}^\top \rangle \geq 0$, where $\boldsymbol{u}$ is the most negative eigenvector of $\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{U}^\top$, as proposed by [209].

As described, a linear optimization problem over the set of orthogonal projection matrices is solved at each iteration, hence building a new branch-and-bound tree each time. We refer to this implementation as a "multi-tree" method. Although inefficient if implemented naively, multi-tree methods benefit from gradually tightening the numerical tolerance of the solver as the number of cuts increases.

To improve the efficiency of Algorithm 5.1, one can integrate the entire procedure

[3]It should be noted that this formulation is rather complicated because non-convex QCQO solvers such as Gurobi currently do not model PSD constraints. If they did, we would supplant the second-order cone constraints with $\boldsymbol{Y} \succeq \boldsymbol{U}\boldsymbol{U}^\top$ and thereby obtain a simpler master problem.

within a single branch-and-cut tree using lazy callbacks, as originally proposed in the context of MICO by [195]. Henceforth, we refer to this implementation as a "single-tree" method. However, the benefit from using multi-tree over single-tree is not straightforward for it depends on how the method is engineered. We benchmark both implementations in Section 5.7.

5.5 Lower bounds via Semidefinite Relaxations

To certify optimality, high-quality lower bounds are of interest and can be obtained by relaxing the non-convex constraint $\boldsymbol{Y} \in \mathcal{Y}_n^k$ to $\boldsymbol{Y} \in \operatorname{Conv}\left(\mathcal{Y}_n^k\right)$ to obtain a semidefinite relaxation as discussed in Lemma 5.1. In addition to a valid lower bound on (5.11)'s objective, the optimal solution to the relaxation $\boldsymbol{Y}^\star$ is a natural candidate for a random rounding strategy, for stronger convex relaxations lead to superior random rounding strategies. We explore such strategies in detail in the next section.

The convex relaxation yields the optimization problem (5.22) which can be solved using a cutting-plane method (see Section 5.5), an alternating minimization method (see Section 5.5) or reformulated as an SDO and solved as such. Since Algorithm 5.1 is an outer-approximation scheme, solving the convex relaxation via a cutting-plane method has the additional benefit of producing valid linear lower-approximations of $f(\boldsymbol{Y})$ to initialize Algorithm 5.1 with.

Cutting-plane methods for improving the root node bound

As mentioned previously, Problem (5.22) can be solved by a cutting-plane method such as Kelley's algorithm [see 146], which is a continuous analog of Algorithm 5.1 that solves Problem (5.11) over $\operatorname{Conv}(\mathcal{Y}_n^k)$, rather than $\mathcal{Y}_n^k$. The main benefit of such a cutting-plane method is that the cuts generated are valid for both $\operatorname{Conv}(\mathcal{Y}_n^k)$ and $\mathcal{Y}_n^k$, and therefore can be used to initialize Algorithm 5.1 and ensure that its *initial* lower bound is equal to the semidefinite relaxation. As demonstrated by Fischetti et al. [106] in the context of MICO and facility location problems, this approach often accelerates the convergence of decomposition schemes by orders of magnitude.

Figure 5-1's left panel illustrates the convergence of Kelley's method and the `in-out` method for solving the semidefinite relaxation of a noiseless matrix completion problem[4]. Note that in our plot of the `in-out` method on the continuous relaxation we omit the time required to first solve the SDO relaxation; this is negligible (38.4s) compared to the time required for either approach to solve the relaxation using cutting planes. Observe that the `in-out` method's lower bound is both initially better and converges substantially faster to the optimal solution than Kelley's method. This justifies our use of the `in-out` method over Kelley's method for a stabilizing cut loop in numerical experiments.

Once the relaxation is solved, the generated cuts are used to initialize Algorithm 5.1. Figure 5-1's right panel displays the convergence profile of the lower bound of Algorithm 5.1 initialized with cuts from Kelley's or the `in-out` method (with a limit of 100 cuts). We use a single-tree implementation of Algorithm 5.1[5] and again a noiseless matrix completion setting[6]. We also consider the impact of using the SOC inequalities $Y_{i,j}^2 \leq Y_{i,i}Y_{j,j}$ in the master problem formulation. Using the `in-out` method and imposing the SOC inequalities are both vitally important for obtaining high-quality lower bounds from Algorithm 5.1. Accordingly, we make use of both ingredients in our numerical experiments.

Solving the semidefinite relaxation at scale

In preliminary numerical experiments, we found that modern IPM codes such as `Mosek` 9.0 cannot optimize over the Frobenius/nuclear norm penalties when $n > 200$ on a standard laptop. As real-world low-rank problems are often large-scale, we now explore more scalable alternatives for optimizing over these penalties. As scalable alternatives for the nuclear norm penalty have been studied, we focus on the Frobenius

[4]The data generation process is detailed in Section 5.7. Here, $n = 100$, $p = 0.25$, $r = 1$, $\gamma = \frac{20}{p}$.

[5]We warm-start the upper bound with greedy rounding and the Burer-Monterio local improvement heuristic described in Section 5.6. To mitigate against numerical instability, we opted to be conservative with our parameters, and therefore turned Gurobi's heuristics off, set FuncPieceError and FuncPieceLength to their minimum possible values (10^{-5} and 10^{-6}), set the MIP gap to 1% and the time limit for each solve to one hour.

[6]Here, $n = 10$, $p = 0.25$, $r = 1$, and $\gamma = \frac{5}{p}$.

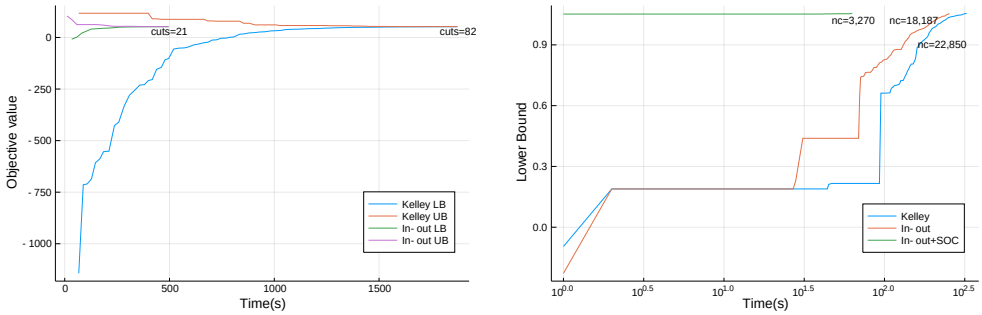


Figure 5-1: Convergence behavior of Kelley's method and the in-out method for solving the semidefinite relaxation of a synthetic matrix completion instance where $n = 100$ (left), and lower bounds generated by a single-tree implementation of Algorithm 5.1 for a synthetic matrix completion instance where $n = 10$ (right).

penalty, and refer to [198] for nuclear norm minimization. We begin our analysis with the following result (proof deferred to [34]):

Lemma 5.9. *Let us fix $\boldsymbol{X}_t$ and consider the following optimization problem:*

$$\min_{\boldsymbol{Y} \in \mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \min_{\boldsymbol{\theta} \in S^n} \quad g(\boldsymbol{X}_t) + \frac{1}{2\gamma}\mathrm{tr}\left(\boldsymbol{\theta}\right) + \lambda \cdot \mathrm{tr}(\boldsymbol{Y}) \quad \text{s.t.} \quad \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X}_t \\ \boldsymbol{X}_t^\top & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0}. \tag{5.37}$$

Then, an optimal choice of $\boldsymbol{\theta}$ is given by $\boldsymbol{\theta}^\star = \boldsymbol{X}_t^\top (\boldsymbol{Y}^\star)^\dagger \boldsymbol{X}_t$, where $\boldsymbol{Y}^\star = \sum_{i=1}^n \rho_i^\star \boldsymbol{u}_i \boldsymbol{u}_i^\top$, $\boldsymbol{X}_t = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is an SVD of $\boldsymbol{X}_t$, and $\boldsymbol{\rho}^\star$ is an optimal solution to the following second order cone problem:

$$\min_{\boldsymbol{\rho} \in [0,1]^n:\ \boldsymbol{e}^\top \boldsymbol{\rho} \leq k} \lambda \cdot \boldsymbol{e}^\top \boldsymbol{\rho} + \sum_{i=1}^n \frac{\sigma(\boldsymbol{X}_t)^2}{2\gamma\rho_i}. \tag{5.38}$$

As optimizing over $\boldsymbol{X}$ for a fixed $\boldsymbol{Y}_t$ is straightforward, Lemma 5.9 suggests a viable approach for optimize over the Frobenius norm penalty is alternating minimization [AM; see 15, for a modern implementation]. By specializing [15]'s implementation of AM to the Frobenius norm penalty, we obtain an efficient numerical strategy for obtaining an optimal solution to (5.24), which we present in Algorithm 5.2; we note that since $\langle \boldsymbol{X}\boldsymbol{X}, \boldsymbol{Y}^\dagger \rangle$ is jointly convex in $\boldsymbol{X}, \boldsymbol{Y}$ (this follows directly from Lemma 5.4), alternating minimization converges to an optimal solution to the semidefinite

relaxation under standard convergence conditions for block coordinate descent techniques for convex programs [see, e.g., 22, Section 3.7] such as the introduction of a proximal term.

We discuss some enhancements to Algorithm 5.2 which improve its convergence.

- Imposing a proximal regularization term in the objective, namely $+\frac{\tau}{2}\|\boldsymbol{X}-\boldsymbol{X}_t\|_F^2$, improves the rate of convergence of the method by stabilizing the iterates; we make use of this in our experiments.
- The method stalls when the eigenvalues of $\boldsymbol{Y}_t$ are near zero (a) due to numerical instability and (b) because $\boldsymbol{Y}_t$ is near the boundary of $\mathrm{Conv}(\mathcal{Y}_n^k)$. Therefore, to accelerate convergence, we require that $\lambda_{\min}(\boldsymbol{Y}) \geq \frac{K}{t}$ at the tth iterate, where $K \approxeq 10^{-2}$. In practice, this introduces very little error.
- We solve for $\boldsymbol{V}^{t+1}$ via the first-order optimality condition using a successive over-relaxation technique, or in rare instances where the linear system solver fails to converge we use `Ipopt` to solve the QP's first-order optimality condition.

Algorithm 5.2 An Accelerated Alternating Minimization Algorithm [c.f. 15]

Require: Initial solution $\boldsymbol{X}_1, \tau_1 \leftarrow 1$
  $t \leftarrow 1, T_{\max}$
  repeat
    Compute $\boldsymbol{W}^{t+1}$ solution of $\mathrm{argmin}_{\boldsymbol{Y}\in\mathrm{Conv}(\mathcal{Y}_n^k)}\ g(\boldsymbol{X}_t) + \frac{1}{2\gamma}\langle \boldsymbol{X}_t\boldsymbol{X}_t^\top, \boldsymbol{Y}^\dagger\rangle$
    Set $\boldsymbol{Y}^{t+1} = \boldsymbol{W}^t + \frac{\tau_{t-1}}{\tau_{t+1}}(\boldsymbol{W}_t - \boldsymbol{W}_{t-1})$
    Compute $\boldsymbol{V}^{t+1}$ solution of $\mathrm{argmin}_{\boldsymbol{X}\in\mathbb{R}^{n\times m}}\ g(\boldsymbol{X}) + \frac{1}{2\gamma}\langle \boldsymbol{X}\boldsymbol{X}^\top, \boldsymbol{Y}_t^\dagger\rangle$
    Set $\boldsymbol{X}^{t+1} = \boldsymbol{V}^t + \frac{\tau_{t-1}}{\tau_{t+1}}(\boldsymbol{V}_t - \boldsymbol{V}_{t-1})$
    Set $\tau_{t+1} = \frac{1+\sqrt{1+4\tau_t^2}}{2}$
    If $t \mod 20 = 0$ compute dual bound at $\boldsymbol{Y}^{t+1}$ via Equation (5.40).
    $t \leftarrow t+1$
  until $t > T_{\max}$ or duality gap $\leq \epsilon$ **return** $\boldsymbol{X}_t, \boldsymbol{Y}_t$

To confirm that Algorithm 5.2 has indeed converged (at least approximately) to an optimal solution, we require a dual certificate. As optimizing over the set of dual variables $\boldsymbol{\alpha}$ for a fixed $\boldsymbol{Y}_t$ does not supply such a bound, we now invoke strong duality to derive a globally valid lower bound. Formally, we have the following result:

Lemma 5.10. *Suppose Assumption 5.2 holds. Then, strong duality holds between:*

$$\min_{\boldsymbol{Y}\in\mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \max_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times m}} \quad h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\sum_{i=1}^n\sum_{j=1}^n Y_{i,j}\langle\boldsymbol{\alpha}_i,\boldsymbol{\alpha}_j\rangle, \tag{5.39}$$

$$\max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^{n\times m},\\ \boldsymbol{U}\succeq\mathbf{0},t\geq 0}} \quad h(\boldsymbol{\alpha}) - \mathrm{tr}(\boldsymbol{U}) - kt \quad \text{s.t.} \quad \boldsymbol{U}+\mathbb{I}t \succeq \frac{\gamma}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top. \tag{5.40}$$

Proof. As Assumption 5.2 holds, we can exchange the minimization and maximization operators in Problem (5.39). Therefore, (5.39) has the same optimal objective as:

$$\max_{\boldsymbol{\alpha}} h(\boldsymbol{\alpha}) - \max_{\boldsymbol{Y}\in\mathrm{Conv}\left(\mathcal{Y}_n^k\right)} \frac{\gamma}{2}\sum_{i=1}^n\sum_{j=1}^n Y_{i,j}\langle\boldsymbol{\alpha}_i,\boldsymbol{\alpha}_j\rangle. \tag{5.41}$$

Therefore, to establish the result, it suffices to show that we obtain Problem (5.40) after taking the dual of Problem (5.41)'s inner problem. This is indeed the case, because $\mathrm{Conv}(\mathcal{Y}_n^k)$ is a convex compact set with non-empty relative interior, and therefore strong duality holds between the following two problems:

$$\begin{aligned}
\max_{\boldsymbol{Y}\succeq\mathbf{0}} \quad & \frac{\gamma}{2}\sum_{i=1}^n\sum_{j=1}^n Y_{i,j}\langle\boldsymbol{\alpha}_i,\boldsymbol{\alpha}_j\rangle && \text{s.t.} \quad \boldsymbol{Y}\preceq\mathbb{I},\ [\boldsymbol{U}]\ \langle\mathbb{I},\boldsymbol{Y}\rangle\leq k,\ [t],\\
\min_{\boldsymbol{U}\succeq\mathbf{0},t\geq 0} \quad & \mathrm{tr}(\boldsymbol{U}) + kt && \text{s.t.} \quad \boldsymbol{U}+\mathbb{I}t\succeq\frac{\gamma}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top. \qquad \square
\end{aligned}$$

Note that Lemma 5.10 supplies a valid dual bound for a given $\boldsymbol{Y}_t$, by fixing $\boldsymbol{Y}_t$, partially maximizing for $\boldsymbol{\alpha}$ in (5.39), and evaluating this $\boldsymbol{\alpha}$'s objective value in (5.40).

Lemma 5.10 demonstrates that Problem (5.1)'s semidefinite relaxation is equivalent to maximizing the dual conjugate $h(\boldsymbol{\alpha})$, minus the k largest eigenvalues of $\frac{\gamma}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top$. Moreover, as proven in the special case of sparse regression by Bertsimas et al. [32], one can show that if the kth and $k+1$th largest eigenvalues of $\boldsymbol{\alpha}\boldsymbol{\alpha}^\top$ in a solution to (5.39) are distinct then Problem (5.39)'s lower bound is tight.

Despite superficial similarities, we should emphasize the difference between Algorithm 5.2 and the Burer-Monterio (BM) heuristic method discussed in the introduction. BM decomposes $\boldsymbol{X}$ into $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}^\top$ and iteratively optimizes over $\boldsymbol{U}$ and $\boldsymbol{V}$. Although the problem is usually convex in $\boldsymbol{U}$ for a fixed $\boldsymbol{V}$ (and vice versa), it is not

jointly convex and BM is only guaranteed to converge to a stationary point. In our setting, we decompose $\boldsymbol{X}$ into $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, leading to semidefinite relaxation that is jointly convex in $(\boldsymbol{Y}, \boldsymbol{X})$. In short, BM returns a stationary solution to the original problem, while Algorithm 5.2 solves its semidefinite relaxation exactly.

5.6 Upper Bounds via Greedy Rounding

We now propose a greedy rounding method for rounding $\boldsymbol{Y}^\star$, an optimal $\boldsymbol{Y}$ in a semidefinite relaxation of Problem (5.10), to obtain near-optimal solutions to Problem (5.10) quickly. Rounding schemes for approximately solving low-rank optimization problems by rounding their SDO relaxations have received a great deal of attention since they were first proposed by Goemans and Williamson [124]. Our analysis is, however, more general than typically conducted when solving low-rank problems, as it involves rounding a projection matrix $\boldsymbol{Y}$, rather than rounding $\boldsymbol{X}$, and therefore generalizes to the rank-k case for $k > 1$, which has historically been challenging.

Observe that for any feasible $\boldsymbol{Y} \in \text{Conv}(\mathcal{Y}_n)$, $0 \leq \lambda_i(\boldsymbol{Y}) \leq 1$ for each eigenvalue of $\boldsymbol{Y}$, and $\boldsymbol{Y}$ is a projection matrix if and only if its eigenvalues are binary. Combining this observation with the Lipschitz continuity of $f(\boldsymbol{Y})$ in $\boldsymbol{Y}$ suggests that high-quality feasible projection matrices can be found in the neighborhood of a solution to the semidefinite relaxation, and a good method for obtaining them is to greedily round the eigenvalues of $\boldsymbol{Y}$. Namely, let $\boldsymbol{Y}^\star$ denote a solution to the semidefinite relaxation (5.22), $\boldsymbol{Y}^\star = \boldsymbol{U}\boldsymbol{\Lambda}^\star\boldsymbol{U}^\top$ be a singular value decomposition of $\boldsymbol{Y}^\star$ such that $\boldsymbol{\Lambda}$ is a diagonal matrix with on-diagonal entries $\Lambda_{i,i}$, and $\boldsymbol{\Lambda}_{greedy}$ be a diagonal matrix obtained from rounding up (to 1) k of the highest diagonal coefficients of $\boldsymbol{\Lambda}^\star$, and rounding down (to 0) the $n-k$ others, with diagonal entries $\Lambda_{i,i} := (\Lambda_{greedy})_{i,i}$. We then let $\boldsymbol{Y}_{greedy} = \boldsymbol{U}\boldsymbol{\Lambda}_{greedy}\boldsymbol{U}^\top$. We now provide guarantees on the quality of the rounding:

Theorem 5.5. *Let $\boldsymbol{Y}^\star$ denote a solution to the relaxation (5.22), $\boldsymbol{Y}^\star = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ be a singular value decomposition of $\boldsymbol{Y}^\star$, $\mathcal{R}$ denote the indices of strictly fractional*

diagonal entries in $\boldsymbol{\Lambda}$*, and* $\boldsymbol{\alpha}^\star(\boldsymbol{Y})$ *denote an optimal choice of* $\boldsymbol{\alpha}$ *for a given* $\boldsymbol{Y}$*, i.e.,*

$$\alpha^\star(\boldsymbol{Y}) \in \arg\max_{\boldsymbol{\alpha}} \left\{ \max_{\boldsymbol{V}_{11}\boldsymbol{V}_{22}} h(\boldsymbol{\alpha}) - \Omega^\star\left(\boldsymbol{\alpha}, \boldsymbol{Y}, \boldsymbol{V}_{11}, \boldsymbol{V}_{22}\right) \right\}.$$

Suppose that for any $\boldsymbol{Y} \in \mathcal{Y}_n^k$*, we have* $\sigma_{\max}(\boldsymbol{\alpha}^\star(\boldsymbol{Y})) \leq L$*. Then, any valid rounding of* $\boldsymbol{Y}^\star$ *which preserves the relaxation's eigenbasis, i.e.,* $\boldsymbol{Y}_{rounded} = \boldsymbol{U}\boldsymbol{\Lambda}_{rounded}\boldsymbol{U}^\top$ *where* $\boldsymbol{Y}^\star = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ *and* $\boldsymbol{\Lambda}_{rounded}$ *is a diagonal matrix with binary diagonal entries* $\Lambda_{i,i}^{rounded}$ *such that* $\operatorname{tr}(\boldsymbol{\Lambda}_{rounded}) \leq k$*, satisfies*

$$f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) \leq \frac{\gamma}{2} L^2 |\mathcal{R}| \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda_{i,i}^\star - \Lambda_{i,i}^{rounded}) \beta_i, \tag{5.42}$$

under the Frobenius penalty and

$$f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) \leq ML |\mathcal{R}| \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda_{i,i}^\star - \Lambda_{i,i}^{rounded}) \beta_i, \tag{5.43}$$

for the spectral penalty. Moreover, let $\boldsymbol{Y}_{greedy} = \boldsymbol{U}\boldsymbol{\Lambda}_{greedy}\boldsymbol{U}^\top$ *be an instance of* $\boldsymbol{Y}_{rounded}$ *obtained by setting* $\Lambda_{i,i} = 1$ *for* k *of the highest diagonal coefficients in* $\boldsymbol{\Lambda}^\star$*. Then, the above bounds imply that* $0 \leq f(\boldsymbol{Y}_{greedy}) - f(\boldsymbol{Y}^\star) \leq \epsilon$*, where* $\epsilon = ML \min(|\mathcal{R}|, n-k)$ *for the spectral penalty and* $\epsilon = \frac{\gamma}{2} \min(|\mathcal{R}|, n-k) L^2$ *for the Frobenius penalty.*

This result calls for multiple remarks:

- When the relaxation gap $f(\boldsymbol{Y}_{\text{greedy}}) - f(\boldsymbol{Y}^\star) = 0$, and the optimal solution to the relaxation, $\boldsymbol{Y}^\star$, is unique, $|\mathcal{R}| = 0$. This justifies retaining $\mathcal{R}$ in the bound, rather than replacing it with n.

- The rounding technique is robust, because it minimizes the worst-case Lipschitz upper bound, under the assumption $\sigma_{\max}(\boldsymbol{\alpha}^\star) \leq L$ (i.e., we have no information about which coordinate[7] has the largest Lipschitz upper bound). For instance,

[7] If we had this information then, as the proof of Theorem 5.5 suggests, we would greedily round to one k of the indices with the largest values of $L_i \Lambda_{i,i}^\star$.

under Frobenius regularization the bound is

$$f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) \leq \frac{\gamma}{2} L^2 |\mathcal{R}| \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}) \beta_i, \tag{5.44}$$

which is minimized over $\boldsymbol{\Lambda}^{rounded} : \mathrm{tr}(\boldsymbol{\Lambda}^{rounded}) \leq k$ by solving:

$$\min_{\boldsymbol{\lambda} \in \mathcal{S}^k_n} \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \frac{\gamma}{2} L^2 |\mathcal{R}| \sum_{i \in \mathcal{R}} (\Lambda^\star_{i,i} - \lambda_i) \beta_i, \tag{5.45}$$

i.e., rounding greedily. Therefore, greedy rounding never performs too badly.

Proof. Since $\mathrm{tr}(\boldsymbol{\Lambda}_{rounded}) \leq k < n$, the vector $(\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i})_i$ has at least one non-negative entry and

$$\max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}) \beta_i = \max \left\{ 0, \max_i (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}) \right\} = \max_i (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}).$$

Consequently, to establish the result, we need only establish the following inequalites:

$$\begin{aligned} f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) &\leq \frac{\gamma}{2} L^2 \max_{\boldsymbol{\beta} \geq 0 : \|\boldsymbol{\beta}\|_\infty \leq 1} \sum_{i \in \mathcal{R}} \left(\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i} \right) \beta_i \\ &\leq \frac{\gamma}{2} L^2 |\mathcal{R}| \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}) \beta_i, \end{aligned}$$

under the Frobenius penalty and

$$\begin{aligned} f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) &\leq ML \max_{\boldsymbol{\beta} \geq 0 : \|\boldsymbol{\beta}\|_\infty \leq 1} \sum_{i \in \mathcal{R}} \left(\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i} \right) \beta_i \\ &\leq ML |\mathcal{R}| \max_{\boldsymbol{\beta} \geq \boldsymbol{0} : \|\boldsymbol{\beta}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (\Lambda^\star_{i,i} - \Lambda^{rounded}_{i,i}) \beta_i, \end{aligned}$$

for the spectral penalty. The second half of both inequalities follows readily from the fact that $\|\boldsymbol{\beta}\|_1 \leq |\mathcal{R}| \|\boldsymbol{\beta}\|_\infty \leq |\mathcal{R}|$ which allows us to replace $\|\boldsymbol{\beta}\|_\infty \leq 1$ with $\|\boldsymbol{\beta}\|_1 \leq |\mathcal{R}|$ and move $|\mathcal{R}|$ outside the bound, so we focus our attention to the first half. After establishing these inequalities, the result follows by observing that $\boldsymbol{Y}_{greedy}$

minimizes the right-hand-side of (5.42)-(5.43) over the projection matrices $\boldsymbol{Y}_{rounded}$.

Under a Frobenius penalty, by Lipschitz continuity, we have

$$\begin{aligned} f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) &\leq \frac{\gamma}{2}\langle \boldsymbol{\alpha}^\star(\boldsymbol{Y})\boldsymbol{\alpha}^\star(\boldsymbol{Y})^\top, \boldsymbol{U}(\boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded})\boldsymbol{U}^\top\rangle \\ &= \frac{\gamma}{2}\langle \boldsymbol{U}^\top\boldsymbol{\alpha}^\star(\boldsymbol{Y})\boldsymbol{\alpha}^\star(\boldsymbol{Y})^\top\boldsymbol{U}, \boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded}\rangle. \end{aligned}$$

Moreover, since $\boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded}$ is a diagonal matrix we need only include the diagonal terms in the inner product. Therefore, since

$$(\boldsymbol{U}^\top\boldsymbol{\alpha}^\star(\boldsymbol{Y})\boldsymbol{\alpha}^\star(\boldsymbol{Y})^\top\boldsymbol{U})_{i,i} = \langle \boldsymbol{\alpha}^\star(\boldsymbol{Y})^\top\boldsymbol{\alpha}^\star(\boldsymbol{Y}), \boldsymbol{U}_i\boldsymbol{U}_i^\top\rangle \leq \lambda_{\max}(\boldsymbol{\alpha}^\star(\boldsymbol{Y})^\top\boldsymbol{\alpha}^\star(\boldsymbol{Y})) \leq L^2,$$

where the inequality holds as $\|\boldsymbol{U}_i\|_2 = 1$, the bound on $f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star)$ holds.

Alternatively, under spectral norm regularization, by Lipschitz continuity we have

$$\begin{aligned} f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star) &\leq M\langle \boldsymbol{V}_{11}^\star(\boldsymbol{Y}) + \boldsymbol{V}_{22}^\star(\boldsymbol{Y}), \boldsymbol{U}(\boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded})\boldsymbol{U}^\top\rangle \\ &= M\langle \boldsymbol{U}^\top(\boldsymbol{V}_{11}^\star(\boldsymbol{Y}) + \boldsymbol{V}_{22}^\star(\boldsymbol{Y}))\boldsymbol{U}, \boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded}\rangle. \end{aligned}$$

Moreover, $\boldsymbol{\Lambda}^\star - \boldsymbol{\Lambda}_{rounded}$ is a diagonal matrix and therefore

$$(\boldsymbol{U}^\top(\boldsymbol{V}_{11}^\star(\boldsymbol{Y}) + \boldsymbol{V}_{22}^\star(\boldsymbol{Y}))\boldsymbol{U})_{i,i} = \langle \boldsymbol{U}_i\boldsymbol{U}_i^\top, \boldsymbol{V}_{11}^\star(\boldsymbol{Y}) + \boldsymbol{V}_{22}^\star(\boldsymbol{Y})\rangle \leq \lambda_{\max}(\boldsymbol{\alpha}^\star(\boldsymbol{Y})) \leq L,$$

where the last inequality follows since $\boldsymbol{V}_{11}, \boldsymbol{V}_{22}$ are orthogonal at optimality, meaning $\boldsymbol{V}_{11}+\boldsymbol{V}_{22}$'s leading eigenvalue equals $\boldsymbol{\alpha}^\star$'s leading singular value. Therefore, the bound on $f(\boldsymbol{Y}_{rounded}) - f(\boldsymbol{Y}^\star)$ holds. □

To improve the greedily rounded solution, we implement a local search strategy which obtains even higher quality warm-starts. Namely, a variant of the popular Burer-Monterio (BM) heuristic [58], which seeks low-rank solutions $\boldsymbol{X}$ by applying a non-linear factorization $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{n\times l}, \boldsymbol{V} \in \mathbb{R}^{m\times k}$ and iteratively optimizing over $\boldsymbol{U}$ for a fixed $\boldsymbol{V}$ (resp. $\boldsymbol{V}$ for a fixed $\boldsymbol{U}$) until convergence to a local optima occurs. This strategy improves our greedily rounded solution because

we initially set $\boldsymbol{U}$ to be the square root of $\boldsymbol{Y}_{greedy}$ and optimize over $\boldsymbol{V}$; recall that if $\boldsymbol{Y}$ is a projection matrix we have $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ for some singular value decomposition $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^\top$.

5.7 Numerical Experiments

In this section, we evaluate the algorithmic strategies derived in the previous section, implemented in `Julia` 1.3 using `JuMP.jl` 0.20.1, `Gurobi` 9.0.1 to solve the non-convex QCQO master problems[8], and Mosek 9.1 to solve the conic subproblems/continuous relaxations. Except where indicated otherwise, all experiments were performed on a Intel Xeon E5—2690 v4 2.6GHz CPU core using 32 GB RAM. To bridge the gap between theory and practice, we have made our code available at `github.com/ryancorywright/MixedProjectionSoftware`.

We evaluate the different ingredients of our numerical strategy on a matrix completion example: First, we solve the semidefinite relaxation by implementing Algorithm 5.2 and demonstrate its increased scalability over `Mosek`'s IPM in Section 5.7. From the solution of the relaxation, our rounding and local search heuristics then provide near-optimal solutions that outperform state-of-the-art heuristic methods, as discussed in Section 5.7. We implement Algorithm 5.1, benchmark its performance and, for the first time, solve low-rank matrix completion to certifiable optimality in Section 5.7. In Section 5.7, we explore the role which regularization plays in our numerical strategy, by showing that increasing the amount of regularization in Problem (5.1) decreases the relative gap, the problem's complexity, and the amount of time required to solve the problem to optimality.

Exploring the Scalability of the Convex Relaxations

In this section, we explore the relative scalability of `Mosek`'s and Algorithm 5.2.

[8]We remark that Gurobi solves the non-convex QCQO master problems by translating them to piecewise linear optimization problems. Since rank constraints are not MICO representable, this introduces some error. To mitigate against this error, we set the Gurobi parameters FuncPieceError and FuncPieceLength to their minimum possible values (10^{-6} and 10^{-5} respectively). Additionally, we set NonConvex to 2, and otherwise use default Gurobi/Mosek parameters.

We consider relaxations of matrix completion. Similarly to [62], we generate two matrices $\boldsymbol{M}_L, \boldsymbol{M}_R \in \mathbb{R}^{n\times r}$ with i.i.d. $\mathcal{N}(0,1)$ entries, and attempt to recover the matrix $\boldsymbol{M} = \boldsymbol{M}_L\boldsymbol{M}_R^\top$ given a proportion p of its observations. Here, we fix $p = 0.25$ and $k = r = 5$, vary n, and set $\gamma = \frac{20}{p}$ where we scale γ proportionally to $1/p$ so that the relative importance of $\|\boldsymbol{X}\|_F^2$ and $\sum_{(i,j)\in\Omega}(X_{i,j} - A_{i,j})^2$ remains constant with p.

We solve the continuous relaxation

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times n}, \boldsymbol{Y}\in\mathrm{Conv}(\mathcal{Y}_n^k), \boldsymbol{\theta}\in S^n} \frac{1}{2\gamma}\mathrm{tr}(\boldsymbol{\theta}) + \sum_{(i,j)\in\Omega}(X_{i,j} - A_{i,j})^2 \text{ s.t. } \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0}. \quad (5.46)$$

Table 5.3 reports the time required by Algorithm 5.2 to obtain a solution with a relative duality gap of 0.1%. To evaluate numerical stability, we also report the relative MSE of the greedily rounded solution; experiments where $n \leq 250$ were run on a standard MacBook pro with 16GB RAM, while larger experiments were run on the previously described cluster with 100GB RAM.

Table 5.3: Scalability of convex relaxations, averaged over 5 matrices. Problem is regularized with Frobenius norm and $\gamma = \frac{20}{p}$. "-" indicates an instance could not be solved with the supplied memory budget.

n	Mosek		Algorithm 5.2		n	Algorithm 5.2	
	Rel. MSE	Time (s)	Rel. MSE	Time (s)		Rel. MSE	Time (s)
50	0.429	2.28	0.438	17.28	350	0.058	6,970
100	0.138	47.20	0.139	79.01	400	0.056	8,096
150	0.082	336.1	0.081	228.7	450	0.055	26,350
200	0.0675	1,906	0.067	841.7	500	0.054	28,920
250	-	-	0.062	1,419	550	0.0536	39,060
300	-	-	0.059	2,897	600	0.0525	38,470

Our results demonstrate the efficiency of Algorithm 5.2: the relative MSE is comparable to `Mosek`'s, but computational time does not explode with n. Since it does not require solving any SDOs and avoids the computational burden of performing the Newton step in an IPM, Algorithm 5.2 scales beyond $n = 600$ ($1,440,000$ decision variables), compared to $n = 200$ for IPMs ($80,000$ decision variables).

Numerical Evaluation of Greedy Rounding

In this section, we compare the greedy rounding method with state-of-the-art heuristic methods, and demonstrate that, by combining greedy rounding with the local search heuristic of [58], our approach outperforms state-of-the-art heuristic methods and therefore should be considered as a viable and efficient warm-start for Algorithm 5.1.

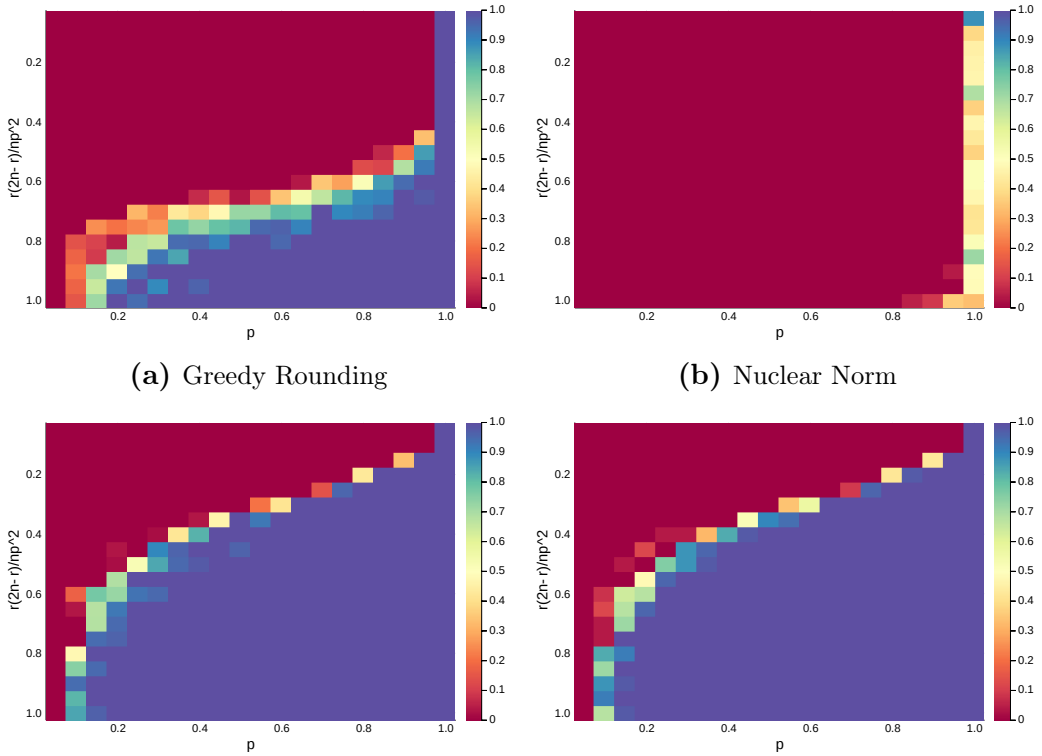
We consider the previous matrix completion problems and assess the ability to recover the low-rank matrix $\boldsymbol{M}$ (up to a relative MSE of 1%), for varying fraction of observed entries p and rank r, with $n = 100$ fixed. Note that, other than the inclusion of a Frobenius regularization term, this is the same experimental setup considered by [63, 198] among others.

We compare the performance of four methods: the greedy rounding method, both with and without the local improvement heuristic from [58], against the local improvement heuristic alone (with a thresholded-SVD initialization point) and the nuclear norm approach. Specifically, the greedy rounding method takes the solution of the previous convex relaxation with $\gamma = \frac{500}{p}$ and rounds its singular values to generate a feasible solution $\boldsymbol{Y}_{greedy}$. For the local improvement heuristic, we solve:

$$\min_{\boldsymbol{X} \in \mathbb{R}^{n \times n}, \boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n \times k}} \frac{1}{2\gamma} \|\boldsymbol{X}\|_2^2 + \sum_{(i,j) \in \Omega} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top,$$

for $\gamma = \frac{500}{p}$ and $k = r$, and iteratively optimize over $\boldsymbol{U}$ and $\boldsymbol{V}$ using `Mosek`. We provide an initial value for $\boldsymbol{U}$ by either taking the first k left-singular vectors of a matrix $\boldsymbol{A}$ where unobserved entries are replaced by 0, or taking the square root of $\boldsymbol{Y}_{greedy}$. For the nuclear norm regularization strategy, since our observations are noiseless, we solve: $\min_{\boldsymbol{X} \in \mathbb{R}^{n \times n}} \|\boldsymbol{X}\|_*$ s.t. $X_{i,j} = A_{i,j} \quad \forall (i,j) \in \Omega$.

Figure 5-2 depicts the proportion of times the matrix was recovered exactly (averaged over 25 samples per tuple of (n, p, r)). As in [63, 198], we vary p between 0 and 1 and consider all possible ranks r such that $r(2n - r) \leq pn^2$. From this set of experiments, we make several observations: First, greedy rounding and the local improvement heuristic outperform nuclear norm minimization both in terms of average relative MSE and amount of data required to recover the matrix. Second, the



(a) Greedy Rounding **(b)** Nuclear Norm

(c) SVD+Local Improvement **(d)** Greedy+Local Improvement

Figure 5-2: Prop. matrices recovered with $\leq 1\%$ relative MSE (higher is better), for different values of p (x-axis) and $r(2n-r)/pn^2 \propto 1/n$ (y-axis), averaged over 25 rank-r matrices.

local improvement heuristic improves upon greedy rounding. In terms of its ability to recover the underlying matrix exactly, it performs equally well with either initialization strategy. However, initialization with the greedy rounding supplies dramatically lower average MSEs in instances where no approach recovers the true matrix exactly. This suggests that initialization strategies for the Burer-Monterio heuristic should be revisited and greedy rounding considered as a viable and more accurate alternative than selecting a random feasible point.

Benchmarking Algorithm 5.1 on Matrix Completion Problems

We now benchmark Algorithm 5.1 on matrix completion problems where $n \in \{10, 20, 30\}$.

We first compare the two different implementations of Algorithm 5.1, single- and multi-tree. In Algorithm 5.1, the lower bounds are warm-started with 200 cuts from

the in-out method, and greedy rounding with local search improvement is used for the upper bounds; if a single-tree instance fails to find a feasible solution (due to numerical instability in `Gurobi`) we return the gap between the warm-start and the semidefinite relaxation. At the tth iteration, we impose a time limit of $10t$ seconds for generating the new cut so as to increase numerical precision as the solver progresses. We also impose a limit of 20 cuts for the multi-tree approach, a time limit of $30,000$s for the single-tree approach, and an optimality gap of 1%[9]. Average runtime, number of nodes, and optimality gap are reported in Table 5.4. Note that the same random instances were solved by all three approaches (by fixing the random seeds), to facilitate a less noisy comparison.

We observe that multi-tree dominates single-tree and Gurobi in terms of runtime and the quality of the solution found, although single-tree occasionally has a smaller gap at termination. Moreover, multi-tree consistently finds high-quality feasible solutions earlier than single tree and accepts our warm-start more consistently, which suggests it may scale better to high-dimensional settings.

Next, we evaluate the performance of the multi-tree implementation of Algorithm 5.1 on a more extensive test-set, including instances where $\text{Rank}(\boldsymbol{M}) > 1$, in Table 5.5. Note that when $r = 1$ we use the same experimental setup (although we impose a time limit of $30t$ seconds, or 7200 seconds if there has been no improvement for two consecutive iterations, a cut limit of 50 cuts when $n > 20$), and when $r > 1$ we increase the time limit per iteration to 300t seconds (or 7200 seconds if there has been no improvement for two consecutive iterations), and allow up to 100 PSD cuts per iteration to be added at the root node via a user cut callback, in order to strengthen the approximation of the PSD constraint $\boldsymbol{Y} \succeq \boldsymbol{0}$. We observe that the problem's complexity increases with the rank, although not too excessively. Moreover, when $r > 1$ the bound gap is actually smaller when $\gamma = \frac{100}{p}$ than when $\gamma = \frac{20}{p}$. We

[9]We report the absolute gap between the better of Gurobi's lower bound and the semidefinite lower bound, compare to the objective value which we evaluate directly; this is sometimes $1 - 2\%$ even when Gurobi reports that it has found an optimal solution, due to numerical instability in Gurobi. Note that we report the absolute, rather than relative, gap since the relative gap depends on the quality of Gurobi's approximation of $\mathcal{Y}_n^k$, which is controlled by the parameter FuncPieceError and cannot be set lower than 10^{-6}; also note that the objective values are on the order of 0.5-5.0 for the problems reported in Table 4.

Table 5.4: Scalability of Algorithm 5.1 for solving rank-1 matrix completion problems to certifiable optimality, averaged over 20 random matrices per row. In multi-tree, Nodes (t) denotes the number of nodes expanded over all trees for the multi-tree implementation.

			Algorithm 5.1 (single-tree)			Algorithm 5.1 (multi-tree)		
n	p	γ	Time(s)	Gap	Cuts	Time(s)	Gap	Cuts
10	0.1	$20/p$	10,310	0.0004	23,460	252.3	0.0019	2.95
10	0.2	$20/p$	19,440	0.0229	19,370	1,672	0.0104	11.0
10	0.3	$20/p$	20,368	0.0433	20,290	2,319	0.0317	15.4
10	0.1	$100/p$	18,580	0.0015	42,200	239.5	0.0003	3.20
10	0.2	$100/p$	27,990	0.0492	31,060	1,269	0.0042	8.40
10	0.3	$100/p$	25,750	0.0434	23,390	2,472	0.0098	19.6
20	0.1	$20/p$	$> 30,000$	0.741	13,070	2,917	0.0166	18.5
20	0.2	$20/p$	$> 30,000$	0.1816	7,008	3,512	0.247	20.0
20	0.3	$20/p$	28,700	0.1066	6,828	3,287	0.253	19.6
20	0.1	$100/p$	$> 30,000$	0.7714	13,799	6,152	0.0072	17.6
20	0.2	$100/p$	$> 30,000$	0.0543	6,395	3,106	0.0903	17.6
20	0.3	$100/p$	29,530	0.0271	6,510	2,910	0.1368	17.0

believe this is because Gurobi cannot represent the SDO constraint $\boldsymbol{Y} \succeq \mathbf{0}$ and its SOC approximation is inexact (even with PSD cuts), and in some cases refining this approximation is actually harder than refining our approximation of $g(\boldsymbol{X})$.

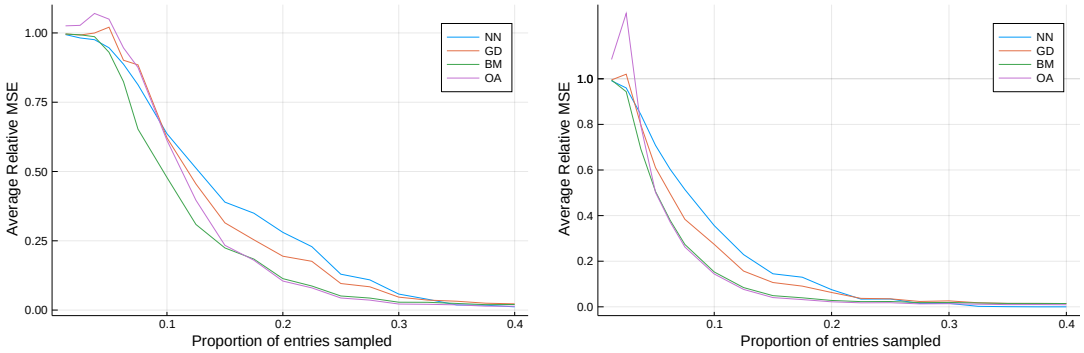
Note that the main bottleneck inhibiting solving matrix completion problems where $n \geq 50$ is `Gurobi`, as the non-convex solver takes increasing amounts of time to process warm-starts (sometimes in the 100s or 1000s of seconds) when n increases. We believe this may be because of the way `Gurobi` translates orthogonal projection matrices to a piecewise linear formulation. Encouragingly, this suggests that our approach may successfully scale to 100×100 matrices as `Gurobi` improves their solver.

Finally, we compare the solution from the exact formulation (5.10) solved using Algorithm 5.1 (multi-tree) with the initial warm-start we proposed and two state-of-the-art heuristics, namely nuclear norm minimization and the Burer-Monterio approach, as in Section 5.7. Here, we take $n \in \{25, 50\}$, $r = 1$, p ranging from 0 to 0.4, and $\gamma = \frac{100}{p}$. Figure 5-3 depicts the average relative MSE over the entire matrix, averaged over 25 random instances per value of p. When $p \geq 0.2$, the exact method supplies an out-of-sample relative MSE around 0.6% lower than Burer-Monterio[10].

[10]Because we ran all methods on the same random instances, this difference is statistically signif-

Table 5.5: Scalability of Algorithm 5.1 (multi-tree) for solving low-rank matrix completion problems to certifiable optimality, averaged over 20 random matrices per row.

			Rank-1				Rank-2				Rank-3			
n	p	γ	Time(s)	Nodes	Gap	Cuts	Time(s)	Nodes	Gap	Cuts	Time(s)	Nodes	Gap	Cuts
10	0.1	$20/p$	182.1	9,755	0.0005	2.56	24,220	35,670	0.0034	5.78	37,780	39,870	0.0071	9.28
10	0.2	$20/p$	3,508	21,060	0.0026	10.8	209,900	108,000	0.0252	35.3	135,260	35,870	0.031	26.2
10	0.3	$20/p$	5,488	30,970	0.0039	13.1	302,200	70,500	0.0866	50.0	302,100	31,870	0.0197	50.0
10	0.1	$100/p$	656.5	28,870	0.0001	2.14	676.1	25,493	0.0009	1.83	842.7	20,700	0.0024	1.79
10	0.2	$100/p$	1,107	10,010	0.0009	4.29	2,065	42,490	0.0019	5.61	57,530	36,910	0.0124	10.7
10	0.3	$100/p$	3,364	48,730	0.0022	6.30	272,300	33,150	0.0195	44.7	249,700	35,530	0.0499	42.2
20	0.1	$20/p$	2,017	4,756	0.0061	8.20	253,900	8,030	0.0279	42.7	255,400	3,015	0.0309	43.2
20	0.2	$20/p$	6,369	6,636	0.0136	15.0	298,700	3,342	0.549	50.0	295,500	236.5	0.879	50.0
20	0.3	$20/p$	6,687	4,187	0.0082	18.4	296,500	3,175	1.123	50.0	291,100	41.35	2.147	50.0
20	0.1	$100/p$	1,266	8,792	0.0087	8.35	211,700	6,860	0.0073	34.24	171,900	2,350	0.0131	29.8
20	0.2	$100/p$	1,220	2,710	0.0104	7.80	302,800	2,426	0.123	50.0	298,800	221.4	0.123	50.0
20	0.3	$100/p$	1,272	1,837	0.0064	3.14	299,000	2,518	0.264	50.0	293,500	43.0	0.659	50.0
30	0.1	$20/p$	300,300	2,735	0.0905	50.0	304,300	164.0	0.790	50.0	303,100	1.10	0.365	50.0
30	0.2	$20/p$	298,700	1,511	0.136	50.0	301,700	9.62	3.105	50.0	302,600	1.00	5.581	50.0
30	0.3	$20/p$	183,800	1,743	0.0476	36.9	303,000	1.63	5.232	50.0	305,000	0.70	14.60	50.0
30	0.1	$100/p$	305,600	2,262	0.0273	50.0	302,800	97.40	0.0973	50.0	305,000	1.90	0.0967	50.0
30	0.2	$100/p$	246,300	3,285	0.0315	43.6	304,300	6.17	0.697	50.0	302,600	1.00	1.419	50.0
30	0.3	$100/p$	25,970	11,020	0.0089	17.1	304,000	1.00	0.923	50.0	304,700	1.00	3.221	50.0



(a) $n = 25$ **(b)** $n = 50$

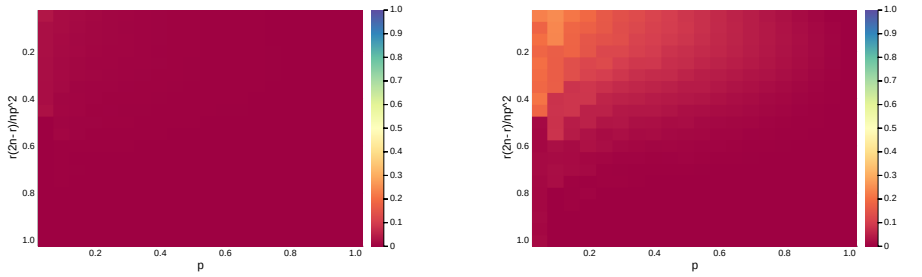
Figure 5-3: Average relative MSE for nuclear norm (NN), greedy rounding (GD), Burer-Monterio (BM), and outer-approximation (OA) when imputing a rank-1 $n \times n$ matrix. All results are averaged over 25 matrices.

Impact of Regularization on Problem Complexity

We now examine the impact of the regularization term $\frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2$ on the problem complexity, as captured by the relative in-sample duality gap between the semidefinite relaxation and the objective value of the greedy solution with a BM local improvement heuristic. We generate the problem data in the same manner as the previous experiment, and display results for four values of γ in Figure 5-4. Observe that as γ increases, both the duality gap and the problem's complexity increase. This observation confirms similar results on the impact of regularization in mixed-integer conic optimization problems [24, 33, c.f.]. Additionally, when $\gamma = \frac{500}{p}$ in Figure 5-4(d), the region where the in-sample duality gap is zero corresponds to exactly recovering the underlying matrix with high probability, while a strictly positive duality gap corresponds to instances with partial recovery only (see Figure 5-2). This suggests a deep connection between relaxation tightness and statistical recovery.

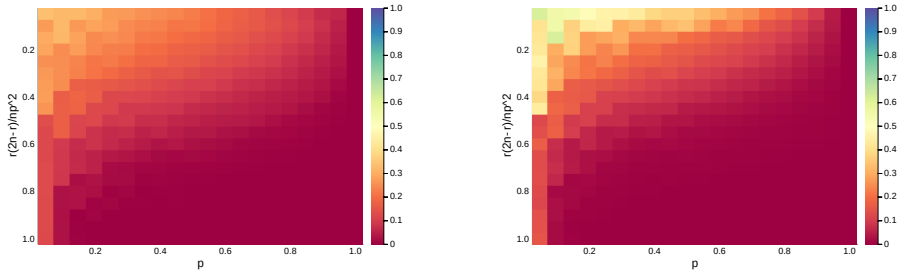
While the relative in-sample semidefinite relaxation gap is a theoretical measure of problem difficulty, it does not indicate how fast Algorithm 5.1 converged in practice. In this direction, we solve the 20 synthetic matrix completion problems considered in Table 5.4 where $n \in \{10, 20\}$, $r = 1$, $p \in \{0.2, 0.3\}$ for 20 different values of $\gamma \in$

icant, with a p-value of 2×10^{-51} (resp. 2×10^{-129}) that the relative MSE is lower for the exact method when $n = 25$ (resp. $n = 50$).



(a) Relative in-sample gap ($\gamma = \frac{0.5}{p}$)

(b) Relative in-sample gap ($\gamma = \frac{5}{p}$)



(c) Relative in-sample gap ($\gamma = \frac{50}{p}$)

(d) Relative in-sample gap ($\gamma = \frac{500}{p}$)

Figure 5-4: Average relative in-sample bound gap (%), averaged over 25 rank-r matrices.

$[10^0, 10^4]$ (distributed uniformly on a log-scale), and compare the relative in-sample semidefinite gap (greedily rounded solution vs. semidefinite bound) with Algorithm 5.1's runtimes in Figure 5-5, for the single-tree (left panel) and multi-tree (right panel) implementation. Results are averaged over 20 random synthetic instances per value of γ. We observe that the relaxation gap does correlate with runtime for single-tree. Yet, the relationship between the relaxation gap and runtime is less straightforward for multi-tree, as it depends on how Gurobi balances cut generation and node expansion, and the conditioning of the problem.

The regularizer γ also impact the bias term $\frac{1}{2\gamma}\|\boldsymbol{X}\|_F^2$ added to the objective function, hence the suboptimality of the solution. To further illustrate the impact of the regularizer γ on solve times and the trade-off between tractability and sub-optimality, Figure 5-6 reports the average runtime and MSE for the previously solved instances, as a function of γ. Figure 5-6 illustrates how γ balances tractability (runtime, top row) and optimality of the solution (MSE, bottom row). Also, single-tree (left panel) is one order of magnitude slower than multi-tree (right panel), and is also more nu-

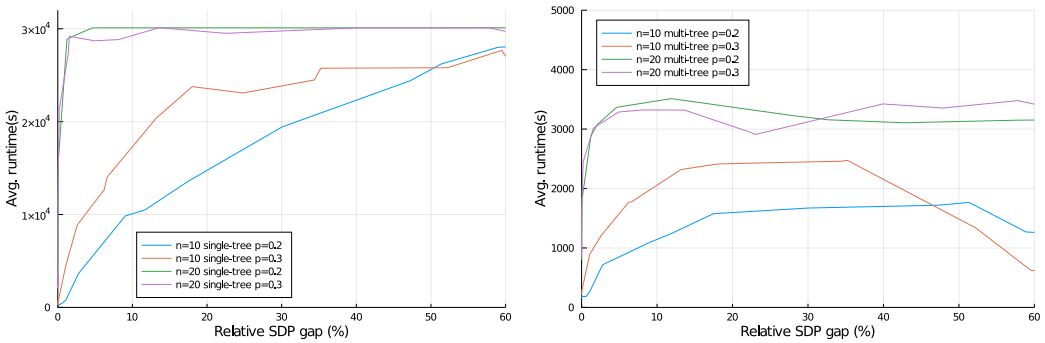


Figure 5-5: Average runtime against relative semidefinite relaxation gap for Algorithm 5.1 single-tree (left) and multi-tree (right) over 20 synthetic matrix completion instances per data point, where $p \in \{0.2, 0.3\}$, $r = 1$, $n \in \{10, 20\}$.

merically instable when γ increases, largely because of the difficulty of combining a non-convex master problem and lazy constraint callbacks (which imposes many cuts, without processing the implications of these cuts as quickly). Echoing our findings in the previous section, this suggests that, while in MICO single-tree typically outperforms multi-tree, at the current state of technology multi-tree should be considered as a viable and potentially more efficient alternative for matrix completion problems which have non-convex master problems. However, as the algorithmic implementations of non-convex QCQOP solvers mature, this finding should be revisited.

Algorithm 5.1 on Coordinate Recovery Problems

We now benchmark the performance of Algorithm 5.1 on anchor-free synthetic coordinate recovery problems, as previously studied by [45, 164] among others.

Specifically, we sample n coordinates $\boldsymbol{x}_i$ uniformly over $[-0.5, 0.5]^k$ for $k \in \{2, 3\}$, and attempt to recover a noisy Gram matrix $\boldsymbol{G} \in S^n_+$ of the $\boldsymbol{x}_i$'s, given a subset of observations of the underlying matrix. Similarly to Biswas and Ye [45], we supply the distance between the points $D_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 + z$, where $z \sim \mathcal{N}(0, 0.01)$, if and only if the radio range between the two points is such that $D_{i,j} \leq d^2_{radio}$. Note that we solve these problems in precisely the same fashion as the largest matrix completion problems solved in the previous section (multi-tree, with a limit of 50 cut passes etc.)

Formally, in order to account for noise in the observed entries, we solve the fol-

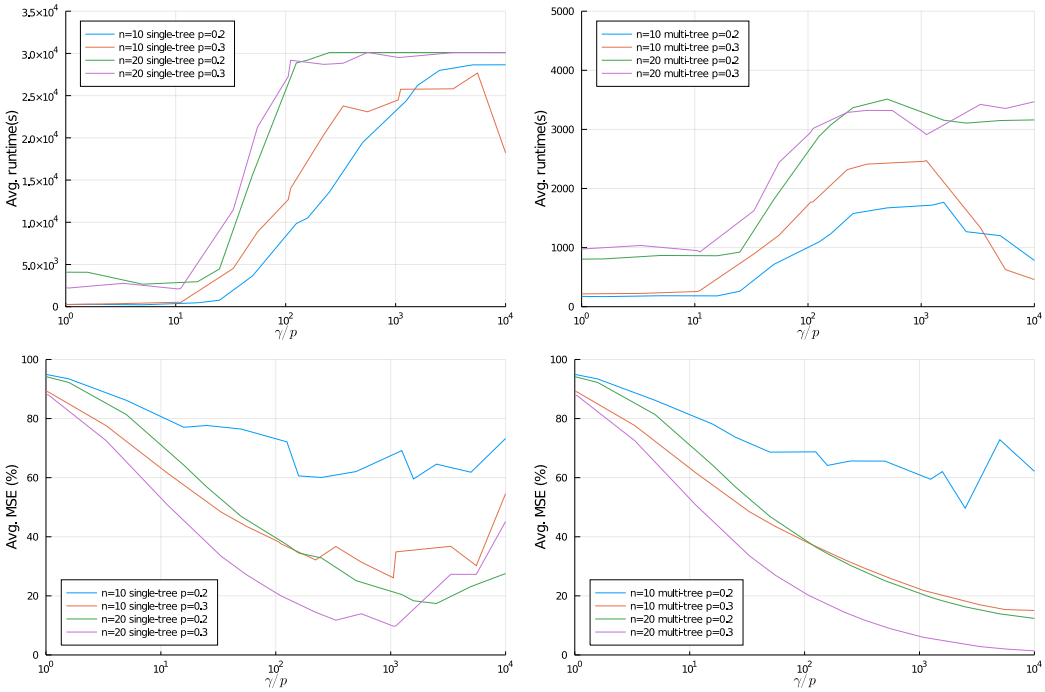


Figure 5-6: Average runtime (top) and MSE (bottom) vs. γ for Algorithm 5.1 single-tree (left) and multi-tree (right) implementations over 20 synthetic matrix completion instances where $p \in \{0.2, 0.3\}$, $r = 1$ and $n \in \{10, 20\}$. The same random seeds were used to generate random matrices completed by single-tree and multi-tree.

lowing problem:

$$\begin{aligned} \min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\boldsymbol{G} \in S_+^n} \quad & \frac{1}{2\gamma}\|\boldsymbol{G}\|_F^2 + \operatorname{tr}(\boldsymbol{G}) + \lambda \cdot \|\boldsymbol{\xi}\|_1 \\ \text{s.t.} \quad & G_{i,i} + G_{j,j} - 2G_{i,j} + \xi_{i,j} = D_{i,j} \ \forall (i,j) \in \Omega, \ \boldsymbol{G} = \boldsymbol{Y}\boldsymbol{G}, \end{aligned}$$

where $\lambda > 0$ is a penalty term which encourages robustness, and the Frobenius norm objective likewise encourages robustness against noise in $\boldsymbol{G}$. The performance of Algorithm 5.1 (multi-tree) on various synthetic instances is reported in Table 5.6, for γ, n, d_{radio}, k varying.

We observe that the problem's complexity increases with the rank and with the dimensionality of the Gram matrix, although not too excessively. Indeed, Algorithm 5.1 can solve coordinate recovery problems with tens of data points to certifiable optimality in hours.

Table 5.6: Scalability of Algorithm 5.1 (multi-tree) for solving sensor location problems to certifiable optimality, averaged over 20 random instances per row. A "-" denotes an instance that cannot be solved within the time budget, because Gurobi fails to accept our warm-start and cannot find a feasible solution. We let $\lambda = n^2$ for all instances.

			Rank-2				Rank-3			
n	d_{radio}	γ	Time(s)	Nodes	Gap	Cuts	Time(s)	Nodes	Gap	Cuts
10	0.1	$1/p$	135.3	6,926	0.0001	1.00	45.14	0.02	0.0000	1.00
10	0.2	$1/p$	3,189	5,249	0.0024	11.5	216.8	7,819	0.0022	1.00
10	0.1	$100/p$	76.2	1,155	0.0000	1.00	140.6	950	0.0000	1.00
10	0.2	$100/p$	480.6	0.05	0.0001	21.7	92.6	139	0.0000	1.14
20	0.1	$1/p$	3,475	4,548	0.0007	13.0	3,090	9,740	0.0001	1.00
20	0.2	$1/p$	73,000	0.50	0.0149	50.0	7,173	5,313	0.0038	1.20
20	0.1	$100/p$	1,878	0.00	0.0000	3.91	64.9	0.00	0.0000	1.07
20	0.2	$100/p$	67,530	0.20	0.0044	50.0	55.7	0.00	0.0002	1.00

Summary of Findings from Numerical Experiments

Our main findings from the numerical experiments are as follows:

- Algorithm 5.2 successfully solves convex relaxations of low-rank problems where n = 100s, in a faster and more scalable fashion than state-of-the-art interior point codes such as `Mosek`.
- Increasing the amount of regularization in a low-rank problem by decreasing γ decreases the duality gap between a low-rank problem with Frobenius or spectral norm problem, and its convex relaxation. Therefore, increasing the amount of regularization makes the problem easier in a practical sense (although not necessarily in a complexity-theoretic sense).
- Algorithm 5.1 scales to solve problems where n is in the tens, i.e., hundreds or thousands of decision variables, in hours. Moreover, the main bottleneck inhibiting solving problems where n is in the hundreds or thousands is that we solve our master problems using Gurobi, a QCQO solver which translates the orthogonal projection matrix constraint into many piecewise linear constraints. This suggests that a custom branch-and-bound solver which explicitly models orthogonal projection matrices constitutes a promising area for future work.

5.8 Conclusion

In this chapter, we introduced Mixed-Projection Conic Optimization, a new framework for modeling rank constrained optimization problems that, for the first time, solves low-rank problems to certifiable optimality at moderate problem sizes. We also provided a characterization of the complexity of rank constraints, and proposed new convex relaxations and rounding methods that lead to viable and more accurate solutions than those obtained via existing techniques such as the log-det or nuclear norm heuristics. Inspired by the collective successes achieved in mixed-integer optimization, we hope that MPCO constitutes an exciting new research direction for the optimization community. For instance, we believe that custom branch-and-bound solvers that explicitly model orthogonal projection matrices could further enhance the scalability of the MPCO framework.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

A New Perspective on Low-Rank Optimization

Over the past decade, a considerable amount of attention has been devoted to low-rank optimization, resulting in theoretically and practically efficient algorithms for problems as disparate as matrix completion, reduced rank regression, or computer vision. In spite of this progress, almost no equivalent progress has been made on developing strong lower bounds for low-rank problems. Accordingly, this chapter proposes a procedure for obtaining strong lower bounds.

We consider the following low-rank optimization problem:

$$\begin{aligned} \min_{\boldsymbol{X} \in \mathcal{S}^n_+} \quad & \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \Omega(\boldsymbol{X}) + \mu \cdot \mathrm{Rank}(\boldsymbol{X}) \\ \text{s.t.} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle = b_i \ \forall i \in [m], \ \boldsymbol{X} \in \mathcal{K}, \ \mathrm{Rank}(\boldsymbol{X}) \leq k, \end{aligned} \tag{6.1}$$

where $\boldsymbol{C}, \boldsymbol{A}_1, \ldots \boldsymbol{A}_m \in \mathcal{S}^n$ are $n \times n$ symmetric matrices, $b_1, \ldots b_m \in \mathbb{R}$ are scalars, $[n]$ denotes the set of running indices $\{1, ..., n\}$, $\mathcal{S}^n_+$ denotes the $n \times n$ positive semidefinite cone, and $\mu \in \mathbb{R}_+, k \in \mathbb{N}$ are parameters which controls the complexity of $\boldsymbol{X}$ by respectively penalizing and constraining its rank. The set $\mathcal{K}$ is a proper—i.e., closed, convex, solid and pointed—cone [c.f. 54, Section 2.4.1], and $\Omega(\boldsymbol{X}) = \mathrm{tr}(f(\boldsymbol{X}))$ for some matrix convex function f; see definitions and assumptions in Chapter 6.3.

For problems with logical constraints, strong relaxations can be obtained by formulating them as mixed-integer optimization (MIO) problems and applying the perspective reformulation technique [see 110, 126]. In this chapter, we develop a matrix analog of the perspective reformulation technique to obtain strong yet computationally tractable relaxations of low-rank optimization problems of the form (6.1).

Motivating Example

In this section, we illustrate the implications of our results on a statistical learning example. To emphasize the analogy with the perspective reformulation technique in MIO, we first consider the best subset selection problem and review its perspective relaxations. We then consider a reduced-rank regression problem – the rank-analog of best subset selection – and provide new relaxations that naturally arise from our Matrix Perspective Reformulation Technique (MPRT).

Best Subset Selection: Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and a response vector $\boldsymbol{y} \in \mathbb{R}^p$, the $\ell_0 - \ell_2$ regularized best subset selection problem is to solve:

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \quad \frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \frac{1}{2\gamma}\|\boldsymbol{w}\|_2^2 + \mu\|\boldsymbol{w}\|_0, \tag{6.2}$$

where $\mu, \gamma > 0$ are parameters which control $\boldsymbol{w}$'s sparsity, sensitivity to noise.

Early attempts at solving (6.2) exactly relied upon weak implicit or big-M formulations of logical constraints which supply low-quality relaxations and do not scale well [see 129, for a discussion]. However, very similar algorithms now solve these problems to certifiable optimality with millions of features. The key ingredient in modernizing these (previously inefficient) algorithms was invoking the perspective reformulation technique—a technique for obtaining high-quality convex relaxations of non-convex sets—first stated in Stubbs [212] PhD thesis [see also 213, 67] and popularized by Frangioni and Gentile [110], Aktürk et al. [4], Günlük and Linderoth [126].

Relaxation via the Perspective Reformulation Technique: By applying the perspective reformulation technique [110, 4, 126] to the term $\mu\|\boldsymbol{w}\|_0 + \frac{1}{2\gamma}\|\boldsymbol{w}\|_2^2$, we obtain the following reformulation:

$$\min_{\boldsymbol{w},\boldsymbol{\rho}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2 + \frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\rho} + \mu\cdot\boldsymbol{e}^\top\boldsymbol{z} \text{ s.t. } z_i\rho_i \geq w_i^2 \ \forall i\in[p]. \tag{6.3}$$

Interestingly, this formulation can be represented using second-order cones [126, 193] and optimized over efficiently using projected subgradient descent [36]. Moreover, it reliably supplies near-exact relaxations for most practically relevant cases of best subset selection [193, 36]. In instances where it is not already tight, one can apply a refinement of the perspective reformulation technique to the term $\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2$ and thereby obtain the following (tighter yet more expensive) relaxation [89]:

$$\begin{aligned}\min_{\boldsymbol{w}\in\mathbb{R}^p,\boldsymbol{z}\in[0,1]^p,\boldsymbol{W}\in S_+^p} \quad & \frac{1}{2n}\|\boldsymbol{y}\|_2^2 - \frac{1}{n}\langle\boldsymbol{y},\boldsymbol{X}\boldsymbol{w}\rangle + \frac{1}{2}\langle\boldsymbol{W},\frac{1}{\gamma}\mathbb{I}+\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\rangle + \mu\boldsymbol{e}^\top\boldsymbol{z} \\ \text{s.t.} \quad & \boldsymbol{W}\succeq\boldsymbol{w}\boldsymbol{w}^\top,\ z_iW_{i,i}\geq w_i^2\ \forall i\in[p].\end{aligned} \tag{6.4}$$

Recently, a class of even tighter relaxations were developed by [7, 128, 116]. As they were developed by considering multiple binary variables simultaneously and therefore do not generalize readily to the low-rank case, we do not discuss them here.

Reduced Rank Regression: Given m observations of a response vector $\boldsymbol{Y}_j\in\mathbb{R}^n$ and a predictor $\boldsymbol{X}_j\in\mathbb{R}^p$, an important problem in high-dimensional statistics is to recover a low-complexity model which relates $\boldsymbol{X},\boldsymbol{Y}$. A popular choice for doing so is to assume that $\boldsymbol{X},\boldsymbol{Y}$ are related via $\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{E}$, where $\boldsymbol{\beta}\in\mathbb{R}^{p\times n}$ is a coefficient matrix which we assume to be low-rank, $\boldsymbol{E}$ is a matrix of noise and we require that the rank of $\boldsymbol{\beta}$ is small in order that the linear model is parsimonious [181]. Introducing Frobenius regularization gives rise to the problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{p\times n}} \quad \frac{1}{2m}\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}\|_F^2 + \frac{1}{2\gamma}\|\boldsymbol{\beta}\|_F^2 + \mu\cdot\mathrm{Rank}(\boldsymbol{\beta}), \tag{6.5}$$

where $\gamma,\mu>0$ control the robustness to noise and the complexity of the estimator,

and we normalize the OLS loss by dividing by m, the number of observations.

Existing attempts at solving this problem generally involve replacing the low-rank term with a nuclear norm term [181], which succeeds under some strong assumptions on the problem data but not in general. However, in Chapter 5, we proposed a new framework to model rank constraints, using orthogonal projection matrices which satisfy $\boldsymbol{Y}^2 = \boldsymbol{Y}$ instead of binary variables which satisfy $z^2 = z$. By building on this idea, in this chapter we propose a generalization of the perspective function to matrix-valued functions with positive semidefinite arguments and develop a matrix analog of the perspective reformulation technique from MIO which uses projection matrices instead of binary variables.

Relaxations via the Matrix Perspective Reformulation Technique: By applying the matrix perspective reformulation technique (Theorem 6.1) to the term $\frac{1}{2\gamma}\|\boldsymbol{\beta}\|_F^2 + \mu \cdot \text{Rank}(\boldsymbol{\beta})$, we will prove that the following problem is a valid—and numerically high-quality—relaxation of (6.5):

$$\begin{aligned} \min_{\boldsymbol{\beta}\in\mathbb{R}^{p\times n}, \boldsymbol{W}\in\mathcal{S}_+^n, \boldsymbol{\theta}\in S_+^p} \quad & \frac{1}{2m}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_F^2 + \frac{1}{2\gamma}\text{tr}(\boldsymbol{\theta}) + \mu \cdot \text{tr}(\boldsymbol{W}) \\ \text{s.t.} \quad & \boldsymbol{W} \preceq \mathbb{I}, \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^\top & \boldsymbol{W} \end{pmatrix} \succeq \boldsymbol{0}. \end{aligned} \tag{6.6}$$

The analogy between problems (6.2)-(6.5) and their relaxations (6.3)-(6.6) is striking. The goal of the present chapter is to develop the corresponding theory to support and derive the relaxation (6.6). Interestingly, the main argument that led [89] to the improved relaxation (6.4) for (6.2) can be extended to reduced-rank regression. Combined with our MPRT, it leads to the relaxation:

$$\begin{aligned} \min_{\substack{\boldsymbol{\theta}\in\mathcal{S}_+^n, \boldsymbol{\beta}\in\mathbb{R}^{p\times n}, \\ \boldsymbol{B}\in\mathcal{S}_+^n, \boldsymbol{W}\in\mathcal{S}_+^n}} \quad & \frac{1}{2m}\|\boldsymbol{Y}\|_F^2 - \frac{1}{m}\langle \boldsymbol{Y}, \boldsymbol{X}\boldsymbol{\beta}\rangle + \frac{1}{2}\langle \boldsymbol{B}, \frac{1}{\gamma}\mathbb{I} + \frac{1}{m}\boldsymbol{X}^\top\boldsymbol{X}\rangle + \mu \cdot \text{tr}(\boldsymbol{W}) \\ \text{s.t.} \quad & \begin{pmatrix} \boldsymbol{B} & \boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{W} \end{pmatrix} \succeq \boldsymbol{0}, \boldsymbol{W} \preceq \mathbb{I}. \end{aligned} \tag{6.7}$$

It is not too hard to see that this is a valid semidefinite relaxation: if $\boldsymbol{W}$ is a rank-k projection matrix then, by the Schur complement lemma [see 55, Equation 2.41], $\boldsymbol{\beta} = \boldsymbol{\beta}\boldsymbol{W}$, and thus the rank of $\boldsymbol{\beta}$ is at most k. Moreover, if we let $\boldsymbol{B} = \boldsymbol{\beta}\boldsymbol{\beta}^\top$ in a solution, we recover a low-rank solution to the original problem. Actually, as we show in Section 6.4, a similar technique can be applied to any instance of Problem (6.1), for which the applications beyond matrix regression are legion.

6.1 Literature Review

Three classes of approaches have been proposed for solving Problem (6.1): (a) heuristics, which prioritize computational efficiency and obtain typically high-quality solutions to low-rank problems efficiently but without optimality guarantees [see 184, for a review]; (b) relax-and-round approaches, which balance computational efficiency and accuracy concerns by relaxing the rank constraint and rounding a solution to the relaxation to obtain a provably near-optimal low-rank matrix [34, Section 1.2.2]; and (c) exact approaches, which prioritize accuracy over computational efficiency and solve Problem (6.1) exactly in exponential time [34, Section 1.2.1].

Of the three classes of approaches, heuristics currently dominate the literature, because their superior runtime and memory usage allows them to address larger-scale problems. However, recent advances in algorithmic theory and computational power have drastically improved the scalability of exact and approximate methods, to the point where they can now solve moderately sized problems which are relevant in practice [34]. Moreover, relaxations of strong exact formulations often give rise to very efficient heuristics (via tight relaxations of the exact formulation) which outperform existing heuristics. This suggests that heuristic approaches may not maintain their dominance going forward, and motivates the exploration of tight yet affordable relaxations of low-rank problems.

6.2 Background on Perspective Functions

In this section, we review perspective functions and their interplay with tight formulations of logically constrained problems. This prepares the ground for and motivates our study of matrix perspective functions and their interplay with tight formulations of low-rank problems. Many of our subsequent results can be viewed as (nontrivial) generalizations of the results in this section, since a rank constraint is a cardinality constraint on the singular values.

Preliminaries

Consider a proper closed convex function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is a convex subset of $\mathbb{R}^n$. The perspective function of f is commonly defined for any $\boldsymbol{x} \in \mathbb{R}^n$ and any $t > 0$ as $(\boldsymbol{x}, t) \mapsto tf(\boldsymbol{x}/t)$. Its closure is defined by continuity for $t = 0$ and is equal to [c.f. 131, Proposition IV.2.2.2]:

$$g_f(\boldsymbol{x}, t) = \begin{cases} tf(\boldsymbol{x}/t) & \text{if } t > 0, \boldsymbol{x}/t \in \mathcal{X}, \\ 0 & \text{if } t = 0, \boldsymbol{x} = 0, \\ f_\infty(\boldsymbol{x}) & \text{if } t = 0, \boldsymbol{x} \neq 0, \\ +\infty & \text{otherwise,} \end{cases}$$

where f_∞ is the recession function of f, as originally stated in [204, p. 67]:

$$f_\infty(\boldsymbol{x}) = \lim_{t \to 0} tf\left(\boldsymbol{x}_0 - \boldsymbol{x} + \frac{\boldsymbol{x}}{t}\right) = \lim_{t \to +\infty} \frac{f(\boldsymbol{x}_0 + t\boldsymbol{x}) - f(\boldsymbol{x}_0)}{t},$$

for any $\boldsymbol{x}_0$ in the domain of f.

The perspective function was first investigated by Rockafellar [204], who made the important observation that f is convex in $\boldsymbol{x}$ if and only if g_f is convex in $(\boldsymbol{x}, t)$. Among other properties, we have that, for any $t > 0$, $(\boldsymbol{x}, t, s) \in \operatorname{epi}(g_f)$ if and only if $(\boldsymbol{x}/t, s/t) \in \operatorname{epi}(f)$ [131, Proposition IV.2.2.1]. We refer to the review by Combettes [74] for further properties of perspective functions.

Throughout this work, we refer to g_f as the *perspective function* of f –although it technically is the closure of the perspective. We also consider a family of convex functions f which satisfy:

Assumption 6.1. *The function $f : \mathcal{X} \to \mathbb{R}$ is proper, closed, and convex. $\mathbf{0} \in \mathcal{X}$ and for any $\boldsymbol{x} \neq \mathbf{0}$, $f_\infty(\boldsymbol{x}) = +\infty$.*

The condition $f_\infty(\boldsymbol{x}) = +\infty, \forall \boldsymbol{x} \neq \mathbf{0}$ means that, asymptotically, f increases to infinity faster than any affine function. In particular, it is satisfied if the domain of f is bounded or if f is strictly convex. Under Assumption 6.1, the definition of the perspective function of f simplifies to

$$g_f(\boldsymbol{x}, t) = \begin{cases} tf(\boldsymbol{x}/t) & \text{if } t > 0, \\ 0 & \text{if } t = 0, \boldsymbol{x} = 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{6.8}$$

The Perspective Reformulation Technique

A number of authors have observed that optimization problems over binary and continuous variables admit tight reformulations involving perspective functions of appropriate substructures of an MIO, since Ceria and Soares [67], building upon the work of Rockafellar [204, Theorem 9.8], derived the convex hull of a disjunction of convex constraints. To motivate our study of the matrix perspective function in the sequel, we now demonstrate that a class of logically-constrained problems admit reformulations in terms of perspective functions. We remark that this development bears resemblance to other works on perspective reformulations including [33, 128, 116].

Consider a logically-constrained problem of the form

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{6.9}$$

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\boldsymbol{c} \in \mathbb{R}^n$ is a cost vector, $f(\cdot)$ is a generic convex function which possibly models convex constraints $\boldsymbol{x} \in \mathcal{X}$ for a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ implicitly—by

requiring that $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$, and $\Omega(\cdot)$ is a regularization function which satisfies the following assumption:

Assumption 6.2. $\Omega(\boldsymbol{x}) = \sum_{i\in[n]} \Omega_i(x_i)$, *where each* Ω_i *satisfies Assumption 6.1.*

Since z_i is binary, imposing the logical constraint "$x_i = 0$ if $z_i = 0$" plus the term $\Omega_i(x_i)$ in the objective is equivalent to $g_\Omega(x_i, z_i) + (1 - z_i)\Omega_i(0)$ in the objective, where g_{Ω_i} is the perspective function of Ω_i, and thus Problem (6.9) is equivalent to:

$$\min_{\boldsymbol{z}\in\mathcal{Z},\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{x}) + \sum_{i=1}^{n} \Big(g_{\Omega_i}(x_i, z_i) + (1 - z_i)\Omega_i(0) \Big). \tag{6.10}$$

Notably, while Problems (6.9)-(6.10) have the same feasible regions, (6.10) often has substantially stronger relaxations, as frequently noted in the perspective reformulation literature [110, 126, 106, 33].

For completeness, we provide a formal proof of equivalence between (6.9)-(6.10); note that a related (although dual, and weaker as it requires $\Omega(\mathbf{0}) = \mathbf{0}$) result can be found in [33, Thm. 2.5]:

Lemma 6.1. *Suppose* (6.9) *attains a finite optimal value. Then,* (6.10) *attains the same value.*

Proof. It suffices to establish that the following equality holds:

$$g_{\Omega_i}(x_i, z_i) + (1 - z_i)\Omega_i(0) = \Omega_i(x_i) + \begin{cases} 0 & \text{if } x_i = 0 \text{ or } z_i = 1, \\ +\infty & \text{otherwise.} \end{cases}$$

Indeed, this equality shows that any feasible solution to one problem is a feasible solution to the other with equal cost. We prove this by considering the cases where $z_i = 0$, $z_i = 1$ separately.

- Suppose $z_i = 1$. Then, $g_{\Omega_i}(x_i, z_i) = z_i\Omega_i(x_i/z_i) = \Omega_i(x_i)$ and $x_i = z_i \cdot x_i$, so the result holds.

- Suppose $z_i = 0$. If $x_i = 0$ we have $g_{\Omega_i}(0,0) + \Omega_i(0) = \Omega_i(0)$, and moreover the right-hand-side of the equality is certainly $\Omega_i(0)$. Alternatively, if $x_i \neq 0$ then both sides equal $+\infty$. □

In Table 6.1, we present examples of penalties Ω for which Assumption 6.1 holds and the perspective reformulation technique is applicable.

We remind the reader that the exponential cone is [c.f. 70]:

$$\mathcal{K}_{\text{exp}} = \{\boldsymbol{x} \in \mathbb{R}^3 : x_1 \geq x_2 \exp(x_2/x_3), x_2 > 0\} \cup \{(x_1, 0, x_3) : x_1 \geq 0, x_3 \leq 0\},$$

while the power cone is defined for any $\alpha \in (0,1)$ as:

$$\mathcal{K}^{\alpha}_{\text{pow}} = \{\boldsymbol{x} \in \mathbb{R}^3 : x_1^{\alpha} x_2^{1-\alpha} \geq |x_3|\}.$$

Table 6.1: Convex substructures which frequently arise in MIOs and their perspective reformulations.

Penalty	$\Omega(x)$	$g_\Omega(x,z)$
Big-M	$\begin{cases} 0 & \text{if } \lvert x\rvert \leq M, \\ +\infty & \text{otherwise} \end{cases}$	$\begin{cases} 0 & \text{if } \lvert x\rvert \leq Mz \\ +\infty & \text{otherwise} \end{cases}$
Ridge	$\frac{1}{2\gamma}x^2$	$\begin{cases} x^2/2\gamma z & \text{if } z > 0 \\ 0 & \text{if } x = z = 0 \\ +\infty & \text{otherwise} \end{cases}$
Ridge + Big-M	$\frac{1}{2\gamma}x^2 + \begin{cases} 0 & \text{if } \lvert x\rvert \leq M \\ +\infty & \text{otherwise} \end{cases}$	$\begin{cases} x^2/2\gamma z & \text{if } z > 0, \lvert x\rvert \leq Mz \\ 0 & \text{if } x = z = 0 \\ +\infty & \text{otherwise} \end{cases}$
Power	$\lvert x\rvert^p$, $p > 1$	$\begin{cases} \lvert x\rvert^p z^{1-p} & \text{if } z > 0, \\ 0 & \text{if } x = z = 0 \\ +\infty & \text{otherwise} \end{cases}$
Logarithm+Big-M	$-\log(x+\epsilon) : 0 \leq x \leq M$	$\begin{cases} -z\log(x/z+\epsilon) & \text{if } x \leq Mz \\ 0 & \text{if } x = z = 0 \\ +\infty & \text{otherwise} \end{cases}$
Entropy	$x \log x$	$\begin{cases} x\log(x/z) & \text{if } x, z > 0 \\ 0 & \text{if } x = z = 0 \\ +\infty & \text{otherwise} \end{cases}$

Perspective Cuts

Another computationally useful application of the perspective reformulation technique has been to derive a class of cutting-planes for MIOs with logical constraints [110]. To motivate our generalization of these cuts to low-rank problems, we now briefly summarize their main result. Consider the following problem:

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{x}\in\mathbb{R}^n} \; \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{x}) + \sum_{i=1}^{n} \Omega_i(x_i) \text{ s.t. } \boldsymbol{A}^i x_i \leq b_i z_i \; \forall i \in [n], \tag{6.11}$$

where $\{x_i : \boldsymbol{A}^i x_i \leq 0\} = \{0\}$, which implies the set of feasible $\boldsymbol{x}$ is bounded, $\Omega_i(x_i)$ is a closed convex function, we take $\Omega_i(0) = 0$ as in [110] for simplicity, and $f(\boldsymbol{x})$ is a convex function. Then, letting ρ_i model the epigraph of $\Omega_i(x_i) + c_i z_i$ and s_i be a subgradient of Ω_i at $\bar{x}_i$, i.e., $s_i \in \partial\Omega_i(\bar{x}_i)$, we have the following result [110, 126]:

Proposition 6.1. *The following cut*

$$\rho_i \geq (c_i + \Omega_i(\bar{x}_i))z_i + s_i(x_i - \bar{x}_i z_i) \tag{6.12}$$

is valid for the equivalent MINLO:

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{x},\boldsymbol{\rho}\in\mathbb{R}^n} \; f(\boldsymbol{x}) + \sum_{i=1}^{n} \rho_i \; s.t. \; \boldsymbol{A}^i x_i \leq b_i z_i \; \forall i \in [n], \; \rho_i \geq \Omega_i(x_i) + c_i z_i \; \forall i \in [n].$$

Remark 15. *In the special case where $\Omega_i(x_i) = x_i^2$, the cut reduces to:*

$$\rho_i \geq 2x_i\bar{x}_i - \bar{x}_i^2 z_i + c_i z_i \quad \forall \bar{x}_i. \tag{6.13}$$

The class of cutting planes defined in Proposition 6.1 are commonly referred to as perspective cuts, because they define a linear lower approximation of the perspective of $\Omega_i(x_i)$, $g_{\Omega_i}(x_i, z_i)$. Consequently, Proposition 6.1 implies that a perspective reformulation of (6.11) is equivalent to adding all (infinitely many) perspective cuts (6.12). This may be helpful where the problem is nonlinear, as a sequence of linear MIOs can be easier to solve than one nonlinear MIO [see 113, for a comparison].

6.3 A Matrix Perspective and Applications

In this section, we generalize the perspective function from vectors to matrices, and invoke the matrix perspective function to propose a new technique for generating strong yet efficient relaxations of a diverse family of low-rank problems, which we call the matrix perspective reformulation technique (MPRT).

A Matrix Perspective Function

To generalize the ideas from the previous section to low-rank constraints, we require a more expressive transform than the perspective transform, which introduces a single (scalar) additional degree of freedom and cannot control the eigenvalues of a matrix. Therefore, we invoke a generalization from quantum mechanics—the matrix perspective function defined in [93, 95], building upon the work of [96]; see also [170, 171, 172, 77] for a related generalization of perspective functions to functionals.

Definition 6.1. *For a matrix-valued function $f : \mathcal{X} \to \mathcal{S}^n_+$ where $\mathcal{X} \subseteq \mathcal{S}^n$ is a convex set, the matrix perspective function of f, g_f, is defined as*

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{Y}^{\frac{1}{2}} f\left(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}\right) \boldsymbol{Y}^{\frac{1}{2}} & \text{if } \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} \in \mathcal{X}, \boldsymbol{Y} \succ \boldsymbol{0}, \\ \infty & \text{otherwise.} \end{cases}$$

Remark 16. *If $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute and f is analytic, then Definition 6.1 simplifies into $\boldsymbol{Y} f\left(\boldsymbol{Y}^{-1} \boldsymbol{X}\right)$, which is the analog of the usual definition of the perspective function originally stated in [96]. Definition 6.1, however, generalizes this definition to the case where $\boldsymbol{X}$ and $\boldsymbol{Y}$ do not commute by ensuring that $\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}$ is nonetheless symmetric, in a manner reminiscent of the development of interior point methods [see, e.g., 5]. In particular, if $\boldsymbol{Y}$ is a projection matrix such that $\boldsymbol{X} = \boldsymbol{Y} \boldsymbol{X}$–as occurs for the exact formulations of the low-rank problems we consider in this paper–then it is safe to assume that $\boldsymbol{X}, \boldsymbol{Y}$ commute. However, when $\boldsymbol{Y}$ is not a projection matrix, this cannot be assumed in general.*

Properties of the Matrix Perspective Function

The matrix perspective function generalizes the definition of the perspective transformation to matrix-valued functions and satisfies analogous properties:

Proposition 6.2. *Let f be a matrix-valued function and g_f its matrix perspective function. Then:*

(a) f is matrix convex, i.e.,

$$tf(\boldsymbol{X}) + (1-t)f(\boldsymbol{W}) \succeq f(t\boldsymbol{X} + (1-t)\boldsymbol{W}) \quad \forall \boldsymbol{X}, \boldsymbol{W} \in \mathcal{S}^n,\ t \in [0,1], \tag{6.14}$$

if and only if g_f is matrix convex in $(\boldsymbol{X}, \boldsymbol{Y})$.

(b) g_f is a positive homogeneous function, i.e., for any μ>0 we have

$$g_f(\mu\boldsymbol{X}, \mu\boldsymbol{Y}) = \mu g_f(\boldsymbol{X}, \boldsymbol{Y}). \tag{6.15}$$

(c) Let $\boldsymbol{Y}$ be a positive definite matrix. Then, letting the epigraph of f be

$$\operatorname{epi}(f) := \{(\boldsymbol{X}, \boldsymbol{\theta}) : \boldsymbol{X} \in \operatorname{dom}(f), f(\boldsymbol{X}) \preceq \boldsymbol{\theta}\}, \tag{6.16}$$

we have $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) \in \operatorname{epi}(g_f)$ if and only if $(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}, \boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{\theta}\boldsymbol{Y}^{-\frac{1}{2}}) \in \operatorname{epi}(f)$.

Proof. We prove the claims successively:

(a) This is precisely the main result of Ebadian et al. [93, Theorem 2.2].

(b) For $\mu > 0$, $g_f(\mu\boldsymbol{X}, \mu\boldsymbol{Y}) = \mu\boldsymbol{Y}^{\frac{1}{2}} f\left((\mu\boldsymbol{Y})^{-\frac{1}{2}}\mu\boldsymbol{X}(\mu\boldsymbol{Y})^{-\frac{1}{2}}\right)\boldsymbol{Y}^{\frac{1}{2}} = \mu g_f(\boldsymbol{X}, \boldsymbol{Y})$.

(c) By generalizing the main result in [54, Chapter 3.2.6], for any $\boldsymbol{Y} \succ \boldsymbol{0}$ we have

$$\begin{aligned}
(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) \in \operatorname{epi}(g_f) \iff & \quad \boldsymbol{Y}^{\frac{1}{2}} f(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}})\boldsymbol{Y}^{\frac{1}{2}} \preceq \boldsymbol{\theta}, \\
\iff & \quad f(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}) \preceq \boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{\theta}\boldsymbol{Y}^{-\frac{1}{2}}, \\
\iff & \quad (\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}, \boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{\theta}\boldsymbol{Y}^{-\frac{1}{2}}) \in \operatorname{epi}(f). \qquad \square
\end{aligned}$$

We now specialize our attention to matrix-valued functions defined by a scalar convex function, as suggested at the start of this chapter.

Matrix Perspectives of Operator Functions

From any function $\omega : \mathbb{R} \to \mathbb{R}$, we define its extension to the set of symmetric matrices, $f_\omega : \mathcal{S}^n \to \mathcal{S}^n$ as

$$f_\omega(\boldsymbol{X}) = \boldsymbol{U} \operatorname{Diag}(\omega(\lambda_1^x), \ldots, \omega(\lambda_n^x))\boldsymbol{U}^\top, \tag{6.17}$$

where $\boldsymbol{X} = \boldsymbol{U} \operatorname{Diag}(\lambda_1^x, \ldots, \lambda_n^x)\boldsymbol{U}^\top$ is an eigendecomposition of $\boldsymbol{X}$. Functions of this form are called *operator functions* [see 39, for a general theory]. In particular, one can show that the trace of operator functions is invariant under an orthogonal rotation, i.e., $\operatorname{tr}(f_\omega(\boldsymbol{X})) = \operatorname{tr}(f_\omega(\boldsymbol{U}^\top \boldsymbol{X}\boldsymbol{U}))$ for any orthogonal rotation $\boldsymbol{U}$. Also, if ω is analytical, then f_ω is also analytical with the same Taylor expansion.

In our analysis, we will use the following bound on $\boldsymbol{v}^\top f_\omega(\boldsymbol{A})\boldsymbol{v}$ when ω is convex:

Lemma 6.2. *Consider a convex function $\omega : \mathbb{R} \to \mathbb{R}$ and a symmetric matrix $\boldsymbol{A} \in \mathcal{S}^n$. Consider a unit vector $\boldsymbol{v}$. Then,*

$$\boldsymbol{v}^\top f_\omega(\boldsymbol{A})\boldsymbol{v} \geq \omega\left(\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}\right).$$

Proof. Consider a spectral decomposition of $\boldsymbol{A}$, $\boldsymbol{A} = \sum_{i=1}^n \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$. Then, $f_\omega(\boldsymbol{A}) = \sum_{i=1}^n \omega(\lambda_i)\boldsymbol{u}_i\boldsymbol{u}_i^\top$ and

$$\boldsymbol{v}^\top f_\omega(\boldsymbol{A})\boldsymbol{v} = \sum_{i=1}^n \omega(\lambda_i)\boldsymbol{v}^\top \boldsymbol{u}_i\boldsymbol{u}_i^\top \boldsymbol{v} \geq \omega\left(\sum_{i=1}^n \lambda_i \boldsymbol{v}^\top \boldsymbol{u}_i\boldsymbol{u}_i^\top \boldsymbol{v}\right) = \omega\left(\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}\right),$$

where the inequality comes from the convexity of ω since $\boldsymbol{v}^\top \boldsymbol{u}_i\boldsymbol{u}_i^\top \boldsymbol{v} = (\boldsymbol{u}_i^\top \boldsymbol{v})^2 \geq 0$ and $\sum_{i=1}^n \boldsymbol{v}^\top \boldsymbol{u}_i\boldsymbol{u}_i^\top \boldsymbol{v} = \boldsymbol{v}^\top \left(\sum_{i=1}^n \boldsymbol{u}_i\boldsymbol{u}_i^\top\right)\boldsymbol{v} = \|\boldsymbol{v}\|^2 = 1$. □

Central to our analysis is that we can explicitly characterize the closure of the matrix perspective of f_ω under some assumptions on ω, i.e., define by continuity $g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})$ for rank-deficient matrices $\boldsymbol{Y}$:

Proposition 6.3. *Consider a function $\omega : \mathbb{R} \to \mathbb{R}$ satisfying Assumption 6.1. Then, the closure of the matrix perspective of f_ω is, for any $\boldsymbol{X} \in \mathcal{S}^n$, $\boldsymbol{Y} \in \mathcal{S}^n_+$,*

$$g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{Y}^{\frac{1}{2}} f_\omega(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}) \boldsymbol{Y}^{\frac{1}{2}} & \text{if } \mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y}), \boldsymbol{Y} \succeq \boldsymbol{0}, \\ \infty & \text{otherwise,} \end{cases}$$

where $\boldsymbol{Y}^{-\frac{1}{2}}$ denotes the pseudo-inverse of the square root of $\boldsymbol{Y}$.

Remark 17. *Note that in the expression of g_{f_ω} above, the matrix $\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}$ is unambiguously defined if and only if $\mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y})$ (otherwise, its value depends on how we define the pseudo-inverse of $\boldsymbol{Y}^{\frac{1}{2}}$ outside of its range). Accordingly, in the remainder of the paper, we omit the condition $\mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y})$ whenever the analytic expression for g_{f_ω} explicitly involves $\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}$.*

Proof. Fix $\boldsymbol{X} \in \mathcal{S}^n$. For $\boldsymbol{Y} \succ \boldsymbol{0}$, the perspective of f_ω is well-defined according to Definition 6.1. Now, consider an arbitrary $\boldsymbol{Y} \succeq \boldsymbol{0}$ and define $\boldsymbol{P}$ as the orthogonal projection onto the kernel of $\boldsymbol{Y}$, which is orthogonal to $\mathrm{Span}(\boldsymbol{Y})$. Then, $\boldsymbol{Y}_\varepsilon := \boldsymbol{Y} + \varepsilon \boldsymbol{P}$ for $\varepsilon > 0$ is invertible. The closure of the matrix perspective of f_ω is defined by continuity as the limit of $\boldsymbol{M}_\varepsilon := \boldsymbol{Y}_\varepsilon^{\frac{1}{2}} f_\omega \left(\boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \right) \boldsymbol{Y}_\varepsilon^{\frac{1}{2}}$ for $\varepsilon \to 0$.

Since the ranges of $\boldsymbol{Y}$ and $\boldsymbol{P}$ are orthogonal ($\boldsymbol{Y}\boldsymbol{P} = \boldsymbol{P}\boldsymbol{Y} = \boldsymbol{0}$), we have $\boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} = \boldsymbol{Y}^{-\frac{1}{2}} + \varepsilon^{-\frac{1}{2}} \boldsymbol{P}$, and

$$\boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} = \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} + \varepsilon^{-\frac{1}{2}} \boldsymbol{P} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} + \varepsilon^{-\frac{1}{2}} \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{P} + \varepsilon^{-1} \boldsymbol{P} \boldsymbol{X} \boldsymbol{P}.$$

Note that $\lim_{\varepsilon \to 0} \boldsymbol{Y}_\varepsilon^{\frac{1}{2}} = \boldsymbol{Y}^{\frac{1}{2}}$ but $\lim_{\varepsilon \to 0} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \neq \boldsymbol{Y}^{-\frac{1}{2}}$. We now distinguish two cases.

Case 1: If $\mathrm{span}(\boldsymbol{X}) \subseteq \mathrm{span}(\boldsymbol{Y})$, $\boldsymbol{X}\boldsymbol{P} = \boldsymbol{P}\boldsymbol{X} = \boldsymbol{0}$ so

$$\begin{aligned} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} &= \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}, \\ \boldsymbol{M}_\varepsilon = \boldsymbol{Y}_\varepsilon^{\frac{1}{2}} f_\omega \left(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} \right) \boldsymbol{Y}_\varepsilon^{\frac{1}{2}} &\quad \to_{\varepsilon \to 0} \boldsymbol{Y}^{\frac{1}{2}} f_\omega \left(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} \right) \boldsymbol{Y}^{\frac{1}{2}}. \end{aligned}$$

Case 2: If $\mathrm{span}(\boldsymbol{X}) \not\subseteq \mathrm{span}(\boldsymbol{Y})$, consider an orthonormal basis of $\mathbb{R}^n$ such that $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ is an eigenbasis of $\mathrm{Span}(\boldsymbol{Y})$ (with respective eigenvalues $\lambda_1^y, \ldots \lambda_k^y$) and

$\boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_n$ is a basis of $\mathrm{Span}(\boldsymbol{Y})^\perp = \mathrm{Ker}(\boldsymbol{Y})$. By assumption, $k < n$ and there exists $j > k$ such that $\boldsymbol{u}_j^\top \boldsymbol{X} \boldsymbol{u}_j \neq 0$. Without loss of generality, we shall assume $\boldsymbol{u}_n^\top \boldsymbol{X} \boldsymbol{u}_n \neq 0$. We show that the matrix $\boldsymbol{M}_\varepsilon$ goes to infinity as $\varepsilon \to 0$ by showing that $\boldsymbol{u}_n^\top \boldsymbol{M}_\varepsilon \boldsymbol{u}_n$ diverges.

Since $\boldsymbol{Y}_\varepsilon^{\pm\frac{1}{2}} \boldsymbol{u}_n = \varepsilon^{\pm\frac{1}{2}} \boldsymbol{u}_n$, we have

$$\boldsymbol{u}_n^\top \boldsymbol{M}_\varepsilon \boldsymbol{u}_n = \varepsilon \; \boldsymbol{u}_n^\top f_\omega \left(\boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \right) \boldsymbol{u}_n \;\; \geq \varepsilon \; \omega \left(\boldsymbol{u}_n^\top \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{u}_n \right) = \varepsilon \; \omega \left(\varepsilon^{-1} \boldsymbol{u}_n^\top \boldsymbol{X} \boldsymbol{u}_n \right),$$

where the inequality follows by convexity of ω and Lemma 6.2. By Assumption 6.1,

$$\lim_{\varepsilon \to 0} \varepsilon \omega \left(\varepsilon^{-1} \boldsymbol{u}_n^\top \boldsymbol{X} \boldsymbol{u}_n \right) = \omega_\infty (\boldsymbol{u}_n^\top \boldsymbol{X} \boldsymbol{u}_n) = +\infty,$$

because $\boldsymbol{u}_n^\top \boldsymbol{X} \boldsymbol{u}_n \neq 0$ and ω is coercive. □

We now provide a simple extension of Proposition 6.3 that will prove useful later.

Corollary 6.1. *Consider a function $\omega : \mathbb{R} \to \mathbb{R}$ satisfying Assumption 6.1 and denote its associated operator function f_ω. Consider a closed set $\mathcal{X} \subseteq \mathcal{S}^n$ and define*

$$f(\boldsymbol{X}) = \begin{cases} f_\omega(\boldsymbol{X}) & \text{if } \boldsymbol{X} \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, the closure of the matrix perspective of f is, for any $\boldsymbol{X} \in \mathcal{S}^n$, $\boldsymbol{Y} \in \mathcal{S}_+^n$,

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{Y}^{\frac{1}{2}} f_\omega(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}) \boldsymbol{Y}^{\frac{1}{2}} & \text{if } \boldsymbol{Y} \succeq \boldsymbol{0}, \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} \in \mathcal{X}, \\ \infty & \text{otherwise,} \end{cases}$$

where $\boldsymbol{Y}^{-\frac{1}{2}}$ denotes the pseudo-inverse of the square root of $\boldsymbol{Y}$.

Proof. Fix $\boldsymbol{X} \in \mathcal{S}^n$ and $\boldsymbol{Y} \in \mathcal{S}_+^n$. From Proposition 6.3, we know that $g_f(\boldsymbol{X}, \boldsymbol{Y}) = +\infty$ if $\mathrm{Span}(\boldsymbol{X}) \not\subseteq \mathrm{Span}(\boldsymbol{Y})$. Let us assume that $\mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y})$. Following the same construction as in the proof of Proposition 6.3, we obtain a sequence $\boldsymbol{Y}_\varepsilon$ that converges to $\boldsymbol{Y}$ as $\varepsilon \to 0$ and such that $\boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}_\varepsilon^{-\frac{1}{2}} = \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}$. □

To gain intuition on how the matrix perspective function transforms $\boldsymbol{X}$ and $\boldsymbol{Y}$, we now provide an interesting connection between the matrix perspective of f_ω and the perspective of ω in the case where $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute.

Proposition 6.4. *Consider two matrices $\boldsymbol{X} \in \mathcal{S}^n, \boldsymbol{Y} \in \mathcal{S}_+^n$ that commute and such that $\mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y})$. Hence, there exists an orthogonal matrix $\boldsymbol{U}$ which jointly diagonalizes $\boldsymbol{X}$ and $\boldsymbol{Y}$. Let $\lambda_1^x, \dots, \lambda_n^x$ and $\lambda_1^y, \dots, \lambda_n^y$ denote the eigenvalues of $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively, ordered according to this basis $\boldsymbol{U}$. Consider an operator function f_ω with ω satisfying Assumption 6.1. Then, we have that:*

$$g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{U}\,\mathrm{Diag}\left(g_\omega(\lambda_1^x, \lambda_1^y), \dots, g_\omega(\lambda_n^x, \lambda_n^y)\right)\boldsymbol{U}^\top$$

Proof. By simultaneously diagonalizing $\boldsymbol{X}$ and $\boldsymbol{Y}$, we get

$$\begin{aligned}
\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}} &= \boldsymbol{U}\mathrm{Diag}\left(\lambda_1^x/\lambda_1^y, \dots, \lambda_n^x/\lambda_n^y\right)\boldsymbol{U}^\top, \\
f_\omega\left(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\right) &= \boldsymbol{U}\mathrm{Diag}\left(\omega(\lambda_1^x/\lambda_1^y), \dots, \omega(\lambda_n^x/\lambda_n^y)\right)\boldsymbol{U}^\top, \\
\boldsymbol{Y}^{\frac{1}{2}} f_\omega\left(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\right)\boldsymbol{Y}^{\frac{1}{2}} &= \boldsymbol{U}\mathrm{Diag}\left(\lambda_1^y\omega(\lambda_1^x/\lambda_1^y), \dots, \lambda_n^y\omega(\lambda_n^x/\lambda_n^y)\right)\boldsymbol{U}^\top. \qquad \square
\end{aligned}$$

Note that if $\boldsymbol{Y}$ is a projection matrix such that $\mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y})$ then we necessarily have that $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Y}$ and the assumptions of Proposition 6.4 hold.

In contrast with Proposition 6.4, in the general case where $\boldsymbol{X}$ and $\boldsymbol{Y}$ do not commute, we cannot simultaneously diagonalize them and connect g_{f_ω} with g_ω. However, we can still project $\boldsymbol{Y}$ onto the space of matrices that commute with $\boldsymbol{X}$ and obtain the following result when g_{f_ω} is matrix convex:

Lemma 6.3. *Let $\boldsymbol{X} \in \mathcal{S}^n$ and $\boldsymbol{Y} \in \mathcal{S}_+^n$ be matrices, and define $\mathcal{X} := \{\boldsymbol{M} \,:\, \boldsymbol{M}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{M}\}$ as the set of matrices which commute with $\boldsymbol{X}$. For any matrix $\boldsymbol{M}$, denote $\boldsymbol{M}_{|\mathcal{X}}$ the orthogonal projection of $\boldsymbol{M}$ onto $\mathcal{X}$. Then, since $\boldsymbol{M} \mapsto \boldsymbol{M}_{|\mathcal{X}}$ is a projection operator, we have that*

$$\boldsymbol{Y}_{|\mathcal{X}} \in \mathcal{S}_+^n, \quad \textit{and} \quad \mathrm{tr}\left(\boldsymbol{Y}_{|\mathcal{X}}\right) = \mathrm{tr}\left(\boldsymbol{Y}\right).$$

Moreover, if $\boldsymbol{Y} \mapsto g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})$ *is matrix convex, then we have*

$$\operatorname{tr}\left[g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y}_{|\mathcal{X}})\right] \leq \operatorname{tr}\left[g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})\right].$$

Proof. First, let us observe that $\mathcal{X}$ is a closed subset of $\mathcal{S}^n$, contains the identity, and is closed under multiplication and transposition, also know as a Von Neumann subalgebra [see 65, Section 4 for a detailed treatment of projections onto subalgebras]. The orthogonal projection of a semidefinite matrix onto $\mathcal{X}$ is also semidefinite and has the same trace [65, Theorem. 4.13], so $\operatorname{tr}\left(\boldsymbol{Y}_{|\mathcal{X}}\right) = \operatorname{tr}\left(\boldsymbol{Y}\right)$. Furthermore, since $\boldsymbol{Y} \mapsto g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})$ is matrix convex, Carlen [65, Theorem 4.16] yields

$$g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y}_{|\mathcal{X}}) \preceq g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})_{|\mathcal{X}}.$$

Taking the trace on both sides and using that $\operatorname{tr}\left(g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})_{|\mathcal{X}}\right) = \operatorname{tr}\left(g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})\right)$ concludes the proof. □

In other words, taking the projection of $\boldsymbol{Y}$ onto the commutant of $\boldsymbol{X}$ is a trace preserving operation that can only reduce the value of $\operatorname{tr}\left(g_{f_\omega}(\boldsymbol{X}, \cdot)\right)$. In this paper, we invoke the projection onto $\mathcal{X}$ (a non-convex set) for theoretical purposes, not computational ones. So we are not interested in how to compute $\boldsymbol{Y}_{|\mathcal{X}}$ in practice. Note that, according to Proposition 6.2(a), Lemma 6.3 holds if f_ω is matrix convex.

6.4 The Matrix Perspective Reformulation Technique

Definition 6.1 and Proposition 6.2 supply the necessary language to lay out our Matrix Perspective Reformulation Technique (MPRT). Therefore, we now state the technique; details regarding its implementation become clearer throughout the chapter.

Let us revisit Problem (6.1), and assume that the term $\Omega(\boldsymbol{X})$ satisfies:

Assumption 6.3. $\Omega(\boldsymbol{X}) = \operatorname{tr}\left(f_\omega(\boldsymbol{X})\right)$*, where* ω *is a function satisfying Assumption 6.1 and whose associated operator function,* f_ω*, is matrix convex.*

Assumption 6.3 implies that the regularizer can be rewritten as operating on the eigenvalues of $\boldsymbol{X}$, $\lambda_i(\boldsymbol{X})$, directly: $\Omega(\boldsymbol{X}) = \sum_{i\in[n]} \omega(\lambda_i(\boldsymbol{X}))$. As we discuss in the next section, a broad class of functions satisfy this property. For ease of notation, we refer to f_ω as f in the remainder of the paper (and accordingly denote by g_f its matrix perspective function).

After letting a projection matrix $\boldsymbol{Y}$ model the rank of $\boldsymbol{X}$—as per Chapter 5—Problem (6.1) admits the mixed-projection reformulation:

$$\begin{aligned} \min_{\boldsymbol{Y}\in\mathcal{Y}_n^k} \min_{\boldsymbol{X}\in\mathcal{S}_+^n} \quad & \langle \boldsymbol{C}, \boldsymbol{X}\rangle + \mu \cdot \mathrm{tr}(\boldsymbol{Y}) + \mathrm{tr}(f(\boldsymbol{X})) \\ \text{s.t.} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X}\rangle = b_i \quad \forall i \in [m],\ \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X},\ \boldsymbol{X} \in \mathcal{K}, \end{aligned} \tag{6.18}$$

where $\boldsymbol{Y} \in \mathcal{Y}_n^k$ is the set of $n \times n$ orthogonal projection matrices with trace k:

$$\mathcal{Y}_n^k := \left\{ \boldsymbol{Y} \in \mathcal{S}_+^n : \boldsymbol{Y}^2 = \boldsymbol{Y},\ \mathrm{tr}(\boldsymbol{Y}) \leq k \right\}.$$

Note that for $k \in \mathbb{N}$, the convex hull of $\mathcal{Y}_n^k$ is given by

$$\mathrm{Conv}(\mathcal{Y}_n^k) = \{\boldsymbol{Y} \in S_+^n : \boldsymbol{Y} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{Y}) \leq k\},$$

which is a well-studied object in its own right [185, 186].

Since $\boldsymbol{Y}$ is an orthogonal projection matrix, imposing the nonlinear constraint $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ plus the term $\Omega(\boldsymbol{X}) = \mathrm{tr}(f(\boldsymbol{X}))$ in the objective is equivalent to imposing $\mathrm{tr}(g_f(\boldsymbol{X}, \boldsymbol{Y})) + (n - \mathrm{tr}(\boldsymbol{Y}))\omega(0)$, where g_f is the matrix perspective of f, and thus Problem (6.18) is equivalent to:

$$\begin{aligned} \min_{\boldsymbol{Y}\in\mathcal{Y}_n^k} \min_{\boldsymbol{X}\in\mathcal{S}_+^n} \quad & \langle \boldsymbol{C}, \boldsymbol{X}\rangle + \mu \cdot \mathrm{tr}(\boldsymbol{Y}) + \mathrm{tr}(g_f(\boldsymbol{X}, \boldsymbol{Y})) + (n - \mathrm{tr}(\boldsymbol{Y}))\omega(0) \\ \text{s.t.} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X}\rangle = b_i \quad \forall i \in [m],\ \boldsymbol{X} \in \mathcal{K}, \end{aligned} \tag{6.19}$$

Let us formally state and verify the equivalence between (6.18)-(6.19) via:

Theorem 6.1. *Problems* (6.18)-(6.19) *attain the same optimal objective value.*

Proof. It suffices to show that for any feasible solution to (6.18) we can construct a feasible solution to (6.19) with an equal or lower cost, and vice versa:

- Let $(\boldsymbol{X}, \boldsymbol{Y})$ be a feasible solution to (6.18). Since $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X} \in \mathcal{S}^n$, $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute. Hence, by Proposition 6.4, we have:

$$\operatorname{tr}\left(g_f(\boldsymbol{X}, \boldsymbol{Y})\right) = \sum_{i\in[n]} g_\omega\left(\lambda_i^x, \lambda_i^y\right) = \sum_{i\in[n]} 1\{\lambda_i^y > 0\}\omega(\lambda_i^x),$$

  where $1\{\lambda_i^y > 0\}$ is an indicator function which denotes whether the ith eigenvalue of $\boldsymbol{Y}$ (which is either 0 or 1) is strictly positive. Moreover, since $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$, $\lambda_i^y = 0 \implies \lambda_i^x = 0$ and

$$\begin{aligned}\operatorname{tr}\left(f(\boldsymbol{X})\right) &= \sum_{i\in[n]} \omega(\lambda_i^x) = \operatorname{tr}\left(g_f(\boldsymbol{X}, \boldsymbol{Y})\right) + \sum_{i\in[n]} 1\{\lambda_i^y = 0\}\omega(0) \\ &= \operatorname{tr}\left(g_f(\boldsymbol{X}, \boldsymbol{Y})\right) + (n - \operatorname{tr}(\boldsymbol{Y}))\omega(0). \end{aligned} \tag{6.20}$$

  This establishes that $(\boldsymbol{X}, \boldsymbol{Y})$ is feasible in (6.19) with the same cost.

- Let $(\boldsymbol{X}, \boldsymbol{Y})$ be a feasible solution to (6.19). Then, it follows that $\boldsymbol{X} \in \operatorname{Span}(\boldsymbol{Y})$, which implies that $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ since $\boldsymbol{Y}$ is a projection matrix. Therefore, (6.20) holds, which establishes that $(\boldsymbol{X}, \boldsymbol{Y})$ is feasible in (6.18) with the same cost. □

Remark 18. *Note that, based on the proof of Theorem 6.1, we could replace $g_f(\boldsymbol{X}, \boldsymbol{Y})$ in (6.19) by any function $\tilde{g}(\boldsymbol{X}, \boldsymbol{Y})$ such that $g_f(\boldsymbol{X}, \boldsymbol{Y}) = \tilde{g}(\boldsymbol{X}, \boldsymbol{Y})$ for $\boldsymbol{X}, \boldsymbol{Y}$ that commute, with no impact on the objective value. However, it might impact tractability if $\tilde{g}(\boldsymbol{X}, \boldsymbol{Y})$ is not convex in $(\boldsymbol{X}, \boldsymbol{Y})$.*

Remark 19. *Under Assumption 6.3, the regularization term $\Omega(\boldsymbol{X})$ penalizes all eigenvalues of $f_\omega(\boldsymbol{X})$ equally. The MPRT can be extended to a wider class of regularization functions that penalize the largest eigenvalues more heavily, at the price of additional notation. For brevity, we lay out this extension in Section 6.10.*

Theorem 6.1 only uses the fact that f is an operator function with ω satisfying Assumption 6.1, not the fact that f is matrix convex. In other words, (6.19) is always

an equivalent reformulation of (6.18). An interesting question is to identify the set of necessary conditions for the objective of (6.19) to be convex in $(\boldsymbol{X}, \boldsymbol{Y})$–$f$ being matrix convex is clearly sufficient. The objective in (6.19) is convex only as long as $\mathrm{tr}\,(g_f)$ is. Interestingly, this is not equivalent to the convexity of $\mathrm{tr}(f)$. See the next section for a counter-example. It is, however, an open question whether a weaker notion than matrix convexity could ensure the joint convexity of $\mathrm{tr}(g_f)$.

Non joint convexity of trace matrix perspective of cube

In this section, we demonstrate by counterexample that if ω is a convex and continuous function then, even though the trace of its matrix extension, $\mathrm{tr}(f_\omega)$, is convex [c.f. 65, Theorem 2.10], the trace of its matrix perspective need not be convex.

Specifically, let us consider $\omega(x) = x^3$. In this case, ω is convex on $\mathbb{R}_+$, f_ω is not matrix convex, but $\mathrm{tr}(f_\omega)$ is matrix convex. We have that

$$\mathrm{tr}(g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y})) = \mathrm{tr}\left(\boldsymbol{X}\boldsymbol{Y}^\dagger \boldsymbol{X}\boldsymbol{Y}^\dagger \boldsymbol{X}\right)$$

for $\boldsymbol{X} \in \mathrm{Span}(\boldsymbol{Y}), \boldsymbol{X}, \boldsymbol{Y} \in \mathcal{S}_+^n$. Let us now consider

$$\boldsymbol{Y}_1 = \begin{pmatrix} 0.160378 & 0.343004 \\ 0.343004 & 0.764592 \end{pmatrix}, \quad \boldsymbol{Y}_2 = \begin{pmatrix} 0.0859208 & 0.181976 \\ 0.181976 & 0.52666 \end{pmatrix},$$
$$\boldsymbol{X}_1 = \begin{pmatrix} 0.242865 & 0.543321 \\ 0.543321 & 1.26604 \end{pmatrix}, \quad \boldsymbol{X}_2 = \begin{pmatrix} 0.0595215 & 0.241702 \\ 0.241702 & 1.0596 \end{pmatrix}.$$

Then, some elementary algebra reveals that

$$\mathrm{tr}\left[g_{f_\omega}\left(\tfrac{1}{2}\boldsymbol{X}_1 + \tfrac{1}{2}\boldsymbol{X}_2, \tfrac{1}{2}\boldsymbol{Y}_1 + \tfrac{1}{2}\boldsymbol{Y}_2\right)\right] = 6.248327,$$
$$\tfrac{1}{2}\mathrm{tr}\left[g_{f_\omega}\left(\boldsymbol{X}_1, \boldsymbol{Y}_1\right)\right] + \tfrac{1}{2}\mathrm{tr}\left[g_{f_\omega}\left(\boldsymbol{X}_2, \boldsymbol{Y}_2\right)\right] = 6.23977,$$

which verifies that $\mathrm{tr}(g_{f_\omega}(\boldsymbol{X}, \boldsymbol{Y}))$ is not midpoint convex in $(\boldsymbol{X}, \boldsymbol{Y})$.

6.5 Convex Hulls of Low-Rank Sets and the MPRT

We now show that, for a general class of low-rank sets, applying the MPRT is equivalent to taking the convex hull of the set. This is significant, because we are not aware of any general-purpose techniques for taking convex hulls of low-rank sets. Formally, we have the following result:

Theorem 6.2. *Consider an operator function f satisfying Assumption 6.3. Let*

$$\mathcal{T} = \{\boldsymbol{X} \in \mathcal{S}^n : \operatorname{tr}(f(\boldsymbol{X})) + \mu \cdot \operatorname{Rank}(\boldsymbol{X}) \leq t, \operatorname{Rank}(\boldsymbol{X}) \leq k\} \tag{6.21}$$

be a set where t, k are fixed. Then, an extended formulation of the convex hull is:

$$\begin{aligned}\mathcal{T}^c = \Big\{ &(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^n \times \operatorname{Conv}(\mathcal{Y}_n^k) : \\ &\operatorname{tr}(g_f(\boldsymbol{X}, \boldsymbol{Y})) + \mu \cdot \operatorname{tr}(\boldsymbol{Y}) + (n - \operatorname{tr}(\boldsymbol{Y}))\omega(0) \leq t \Big\}.\end{aligned} \tag{6.22}$$

Where $\operatorname{Conv}(\mathcal{Y}_n^k) = \{\boldsymbol{Y} \in \mathcal{S}_+^n : \boldsymbol{Y} \preceq \mathbb{I}, \operatorname{tr}(\boldsymbol{Y}) \leq k\}$ is the convex hull of trace-k projection matrices, and g_f is the matrix perspective function of f.

Remark 20. *Since linear optimization problems over convex sets admit extremal optima, Theorem 6.2 demonstrates that unconstrained low-rank problems with spectral objectives can be recast as linear semidefinite problems, where the rank constraint is dropped without loss of optimality. This suggests that work on hidden convexity in low-rank optimization, i.e., deriving conditions under which low-rank linear optimization problems admit exact relaxations where the rank constraint is omitted [see, e.g., 190, 220], could be extended to incorporate spectral functions.*

Proof. We prove the two directions sequentially:

- $\operatorname{Conv}(\mathcal{T}) \subseteq \mathcal{T}^c$: let $\boldsymbol{X} \in \mathcal{T}$. Then, since the rank of $\boldsymbol{X}$ is at most k, there exists some $\boldsymbol{Y} \in \mathcal{Y}_n^k$ such that $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ and $\operatorname{tr}(\boldsymbol{Y}) = \operatorname{Rank}(\boldsymbol{X})$. Moreover, by the same argument as in the proof of Theorem 6.1, it follows that (6.20)

holds and $\mathrm{tr}(g_f(\boldsymbol{X},\boldsymbol{Y}))+\mu\cdot\mathrm{tr}(\boldsymbol{Y})+(n-\mathrm{tr}(\boldsymbol{Y}))\omega(0)\leq t$, which confirms that $(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{T}^c$. Since $\mathcal{T}^c$ is a convex set, we therefore have $\mathrm{Conv}\,(\mathcal{T})\subseteq\mathcal{T}^c$.

- $\mathcal{T}^c\subseteq\mathrm{Conv}\,(\mathcal{T})$: let $(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{T}^c$. Our proof uses Proposition 6.4, which requires $\boldsymbol{X}$ and $\boldsymbol{Y}$ to commute. Let $\mathcal{X}$ denote the set of matrices that commute with $\boldsymbol{X}$: $\mathcal{X}:=\{\boldsymbol{M}\ :\ \boldsymbol{X}\boldsymbol{M}=\boldsymbol{M}\boldsymbol{X}\}$. Denote $\boldsymbol{Y}_{|\mathcal{X}}$ the projection of $\boldsymbol{Y}$ onto $\mathcal{X}$. By Lemma 6.3, we have that $\boldsymbol{Y}_{|\mathcal{X}}\in\mathrm{Conv}(\mathcal{Y}_n^k)$, and $\mathrm{tr}\left(g_f(\boldsymbol{X},\boldsymbol{Y}_{|\mathcal{X}})\right)\leq\mathrm{tr}\left(g_f(\boldsymbol{X},\boldsymbol{Y})\right)<\infty$ so $(\boldsymbol{X},\boldsymbol{Y}_{|\mathcal{X}})\in\mathcal{T}^c$ as well. Hence, without loss of generality, by renaming $\boldsymbol{Y}\leftarrow\boldsymbol{Y}_{|\mathcal{X}}$, we can assume that $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute. Then, it follows from Proposition 6.4 that the vectors of eigenvalues of $\boldsymbol{X}$ and $\boldsymbol{Y}$ (ordered according to a shared eigenbasis $\boldsymbol{U}$), $(\boldsymbol{\lambda}(\boldsymbol{X}),\boldsymbol{\lambda}(\boldsymbol{Y}))$ belong to the set

$$\left\{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{R}^n\times[0,1]^n:\sum_i y_i\leq k,\sum_{i=1}^n y_i\omega\left(\tfrac{x_i}{y_i}\right)+\mu\sum_i y_i+(n-\sum_i y_i)\omega(0)\leq t\right\},$$

which, by [126, Lemma 6], is the convex hull of

$$\mathcal{U}^c:=\Bigg\{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{R}^n\times\{0,1\}^n:\sum_i y_i\leq k,\sum_{i=1}^n\omega\left(x_i\right)+\mu\sum_i y_i\leq t,\\ x_i=0\text{ if }y_i=0\ \forall i\in[n]\Bigg\}.$$

Let us decompose $(\boldsymbol{\lambda}(\boldsymbol{X}),\boldsymbol{\lambda}(\boldsymbol{Y}))$ into $\boldsymbol{\lambda}(\boldsymbol{X})=\sum_k\alpha_k\boldsymbol{x}^{(k)}$, $\lambda(\boldsymbol{Y})=\sum_k\alpha_k\boldsymbol{y}^{(k)}$, with $\alpha_k\geq 0$, $\sum_k\alpha_k=1$, and $(\boldsymbol{x}^{(k)},\boldsymbol{y}^{(k)})\in\mathcal{U}^c$. By definition,

$$\boldsymbol{T}^{(k)}:=\boldsymbol{U}\mathrm{Diag}(\boldsymbol{x}^{(k)})\boldsymbol{U}^\top\in\mathcal{T}$$

and $\boldsymbol{X}=\sum_k\alpha_k\boldsymbol{T}^{(k)}$. Therefore, we have that $\boldsymbol{X}\in\mathrm{Conv}(\mathcal{T})$, as required. □

6.6 Examples of the Matrix Perspective Function

Theorem 6.2 demonstrates that, for spectral functions under low-rank constraints, taking the matrix perspective is equivalent to taking the convex hull. To highlight the utility of Theorems 6.1-6.2, we therefore supply the perspective functions of some

spectral regularization functions which frequently arise in the low-rank matrix literature, and summarize them in Table 6.2. We also discuss how these functions and their perspectives can be efficiently optimized over. Note that all functions introduced in this section are either matrix convex or the trace of a matrix convex function, and thus supply valid convex relaxations when used as regularizers for the MPRT.

Spectral constraint: Let $\omega(x) = 0$ if $|x| \leq M$, $+\infty$ otherwise. Then,

$$f(\boldsymbol{X}) = \begin{cases} \boldsymbol{0} & \text{if} \quad \|\boldsymbol{X}\|_\sigma \leq M, \\ +\infty & \text{otherwise,} \end{cases}$$

for $\boldsymbol{X} \in \mathcal{S}^n$, where $\|\cdot\|_\sigma$ denotes the spectral norm, i.e., the largest eigenvalue in absolute magnitude of $\boldsymbol{X}$. Observe that the condition $\|\boldsymbol{X}\|_\sigma \leq M$ can be expressed via semidefinite constraints$-M\mathbb{I} \preceq \boldsymbol{X} \preceq M\mathbb{I}$. The perspective g_f is

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} 0 & \text{if} \quad -M\boldsymbol{Y} \preceq \boldsymbol{X} \preceq M\boldsymbol{Y}, \\ +\infty & \text{otherwise.} \end{cases}$$

If $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute, $g_f(\boldsymbol{X}, \boldsymbol{Y})$ requires that $|\lambda_j(\boldsymbol{X})| \leq M\lambda_j(\boldsymbol{Y})$ $\forall j \in [n]$–the spectral analog of a big-M constraint. This constraint can be modeled using two semidefinite cones, and thus handled by semidefinite solvers.

Convex quadratic: For $\omega(x) = x^2$, $f(\boldsymbol{X}) = \boldsymbol{X}^\top\boldsymbol{X}$. Then, the perspective g_f is

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{X}^\top\boldsymbol{Y}^\dagger\boldsymbol{X} & \text{if } \boldsymbol{Y} \succeq \boldsymbol{0}, \\ +\infty & \text{otherwise.} \end{cases}$$

Observe that this function's epigraph is semidefinite-representable. Indeed, by the Schur complement lemma, minimizing the trace of $g_f(\boldsymbol{X}, \boldsymbol{Y})$ is equivalent to solving

$$\min_{\boldsymbol{\theta}\in\mathcal{S}^n,\boldsymbol{Y}\in\mathcal{S}^n,\boldsymbol{X}\in\mathcal{S}^n} \quad \mathrm{tr}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \mathbf{0}.$$

Interestingly, this perspective function allows us to rewrite the rank-k SVD

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} \quad \|\boldsymbol{X}-\boldsymbol{A}\|_F^2 : \ \mathrm{Rank}(\boldsymbol{X}) \leq k$$

as a linear optimization problem over the set of orthogonal projection matrices, which implies that the orthogonal projection constraint can be relaxed to its convex hull without loss of optimality. This is significant, because while rank-k SVD is commonly thought of as a non-convex problem which "surprisingly" admits a closed-form solution, the MPRT shows that it actually admits an *exact* convex reformulation:

$$\min_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\theta}} \quad \frac{1}{2}\mathrm{tr}(\boldsymbol{\theta}) - \langle \boldsymbol{A}, \boldsymbol{X}\rangle + \frac{1}{2}\|\boldsymbol{A}\|_F^2 \text{ s.t. } \boldsymbol{Y} \preceq \mathbb{I}, \ \mathrm{tr}(\boldsymbol{Y}) \leq k, \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \succeq \mathbf{0}.$$

Note that we extended our results for symmetric matrices to rectangular matrices $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ without justification. We rigorously derive this extension for $f(\boldsymbol{X}) = \boldsymbol{X}^\top\boldsymbol{X}$ in Chapter 6.11, and defer the general case to future research.

Spectral plus convex quadratic: Let

$$f(\boldsymbol{X}) = \begin{cases} \boldsymbol{X}^2 & \text{if} \quad \|\boldsymbol{X}\|_\sigma \leq M, \\ +\infty & \text{otherwise,} \end{cases}$$

for $\boldsymbol{X} \in \mathcal{S}^n$. Then, the perspective function g_f is

$$g_f(\boldsymbol{X},\boldsymbol{Y}) = \begin{cases} \boldsymbol{X}^\top\boldsymbol{Y}^\dagger\boldsymbol{X} & \text{if} \quad -M\boldsymbol{Y} \preceq \boldsymbol{X} \preceq M\boldsymbol{Y}, \\ +\infty & \text{otherwise.} \end{cases}$$

This is the spectral analog of combining a big-M and a ridge penalty.

Convex quadratic over completely positive cone: Consider the problem

$$\min_{\boldsymbol{X}\in\mathcal{S}^n} \ \boldsymbol{X}^\top\boldsymbol{X} \text{ s.t. } \boldsymbol{X}\in\mathcal{C}_+^n,$$

where $\mathcal{C}_+^n = \{\boldsymbol{X} : \boldsymbol{X} = \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{U}\in\mathbb{R}_+^{n\times n}\} \subseteq \mathcal{S}_+^n$ denotes the completely positive cone. Then, by denoting $f(\boldsymbol{X}) = \boldsymbol{X}^\top\boldsymbol{X}$ and g_f its perspective function we obtain a valid relaxation by minimizing $\operatorname{tr}(g_f)$, which, by the Schur complement lemma [see 55, Equation 2.41], can be reformulated as

$$\min_{\boldsymbol{\theta}\in\mathcal{S}^n,\boldsymbol{Y}\in\mathcal{S}^n,\boldsymbol{X}\in\mathcal{S}^n} \quad \operatorname{tr}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{Y} \end{pmatrix} \in \mathcal{S}_+^{2n}, \boldsymbol{X}\in\mathcal{C}_+^n.$$

Unfortunately, this formulation cannot be tractably optimized over, since separating over the completely positive cone is NP-hard. However, by relaxing the completely positive cone to the doubly non-negative cone—$\mathcal{S}_+^n \cap \mathbb{R}_+^{n\times n}$—we obtain a tractable and near-exact relaxation. Indeed, as we shall see in our numerical experiments, combining this relaxation with a state-of-the-art heuristic supplies certifiably near-optimal solutions in both theory and practice.

Power: Let $f(\boldsymbol{X}) = \boldsymbol{X}^\alpha$ for $\alpha\in[0,1]$, $\boldsymbol{X}\in\mathcal{S}_+^n$. The matrix perspective is:

$$g_f(\boldsymbol{X},\boldsymbol{Y}) = \begin{cases} \boldsymbol{Y}^{\frac{1-\alpha}{2}}\boldsymbol{X}^\alpha\boldsymbol{Y}^{\frac{1-\alpha}{2}} & \text{if } \boldsymbol{Y}^{\frac{-1}{2}}\boldsymbol{X}\boldsymbol{Y}^{\frac{-1}{2}} \in \mathcal{S}_+^n, \boldsymbol{Y}\succeq\boldsymbol{0}, \\ +\infty & \text{otherwise.} \end{cases}$$

Remark 21 (Matrix Power Cone)**.** *This function's epigraph, the matrix power cone:*

$$\mathcal{K}_{mat}^{pow,\alpha} = \{(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3)\in\mathcal{S}_+^n\times\mathcal{S}_+^n\times\mathcal{S}^n : \boldsymbol{X}_2^{\frac{1-\alpha}{2}}\boldsymbol{X}_1^\alpha\boldsymbol{X}_2^{\frac{1-\alpha}{2}} \succeq \boldsymbol{X}_{3,+} + \boldsymbol{X}_{3,-}\}$$

is a closed convex cone which is semidefinite representable for any rational p [100]. Consequently, it is a tractable object which successfully models the matrix power function (and its perspective) and we shall make repeated use of it when we apply the MPRT to several important low-rank problems in Section 6.6.

Logarithm: Let $f(\boldsymbol{X}) = -\log(\boldsymbol{X})$ be the matrix logarithm. We have

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} -\boldsymbol{Y}^{\frac{1}{2}} \log\left(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}\right) \boldsymbol{Y}^{\frac{1}{2}} & \text{if } \boldsymbol{X}, \boldsymbol{Y} \succ \boldsymbol{0}, \\ +\infty & \text{otherwise.} \end{cases}$$

Observe that when $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute, $g_f(\boldsymbol{X}, \boldsymbol{Y})$ can be rewritten as $\boldsymbol{Y}(\log(\boldsymbol{Y}) - \log(\boldsymbol{X}))$, which is the quantum relative entropy function [see 101, for a general theory]. We remark that the domain of $\log(\boldsymbol{X})$ requires that $\boldsymbol{X}$ is full-rank, which at a first glance makes the use of this function problematic for low-rank optimization. Accordingly, we consider the $\epsilon-$logarithm function, i.e., $\log_\epsilon(\boldsymbol{X}) = \log(\boldsymbol{X} + \epsilon \mathbb{I})$ for $\epsilon > 0$, as advocated by Fazel et al. [104] in a different context.

Observe that $\operatorname{tr}(\log(\boldsymbol{X})) = \log\det(\boldsymbol{X})$ while $\operatorname{tr}(g_f) = \operatorname{tr}(\boldsymbol{X}(\log(\boldsymbol{X}) - \log(\boldsymbol{Y}))$. Thus, the matrix logarithm and its trace verify the concavity of the logdet function—which has numerous applications in low-rank problems [104] and interior point methods [200] among others—while the perspective of the matrix logarithm provides an elementary proof of the convexity of the quantum relative entropy: a task for which perspective-free proofs are technically demanding [96].

Von Neumann entropy: Let $f(\boldsymbol{X}) = \boldsymbol{X}\log(\boldsymbol{X})$ denote the von Neumann quantum entropy of a density matrix $\boldsymbol{X}$. Then, its perspective function is $g_f(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}} \log(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}) \boldsymbol{Y}^{\frac{1}{2}}$. When $\boldsymbol{X}$ and $\boldsymbol{Y}$ commute, this perspective can be equivalently written as

$$g_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{X}^{\frac{1}{2}} \log(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}) \boldsymbol{X}^{\frac{1}{2}} & \text{if } \boldsymbol{X}, \boldsymbol{Y} \succ \boldsymbol{0}, \\ +\infty & \text{otherwise.} \end{cases}$$

which is referred to as the Umegaski relative entropy or the matrix Kullback-Leibler divergence in the literature.

Remark 22 (Quantum relative entropy cone)**.** *Note the epigraph of g_f, namely,*

$$\mathcal{K}_{mat}^{op,\ rel} = \Big\{ (\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) \in \mathcal{S}^n \times \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n : \\ \boldsymbol{X}_1 \succeq -\boldsymbol{X}_2^{\frac{1}{2}} \log(\boldsymbol{X}_2^{-\frac{1}{2}} \boldsymbol{X}_3 \boldsymbol{X}_2^{-\frac{1}{2}}) \boldsymbol{X}_2^{\frac{1}{2}} \Big\},$$

is a closed convex cone which can be approximated using semidefinite cones and optimized over using either the `Matlab` *package* `CVXQuad` *(see [101]), or optimized over directly using an interior point method for asymmetric cones [145]*[1]*. Consequently, this is a tractable object which models the matrix log and Von Neumann entropy.*

Finally, Table 6.2 relates the matrix perspectives with their scalar analogs.

Table 6.2: Analogy between perspectives of scalars and perspectives of matrix convex functions.

	Perspective of function		Matrix perspective of function	
Type	$f(x) : \mathbb{R} \to \mathbb{R}$	$g_f(\boldsymbol{x}, t)$	f	g_f
Quadratic	x^2	x^2/t	$\boldsymbol{X}^\top \boldsymbol{X}$	$\boldsymbol{X}^\top \boldsymbol{Y}^\dagger \boldsymbol{X}$
Power	$-x^\alpha : 0 < \alpha < 1$	$-x^\alpha t^{1-\alpha}$	$-\boldsymbol{X}^\alpha$	$-\boldsymbol{Y}^{\frac{1-\alpha}{2}} \boldsymbol{X}^\alpha \boldsymbol{Y}^{\frac{1-\alpha}{2}}$
Log	$-\log(x)$	$-t\log(\frac{x}{t})$	$\log(\boldsymbol{X})$	$-\boldsymbol{Y}^{\frac{1}{2}} \log\left(\boldsymbol{X}^{-\frac{1}{2}} \boldsymbol{Y} \boldsymbol{X}^{-\frac{1}{2}}\right) \boldsymbol{Y}^{\frac{1}{2}}$
Entropy	$x\log(x)$	$x\log(\frac{x}{t})$	$\boldsymbol{X}\log(\boldsymbol{X})$	$\boldsymbol{X}^{\frac{1}{2}} \log(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}}) \boldsymbol{X}^{\frac{1}{2}}$

Matrix Perspective Cuts

We now generalize the perspective cuts of [110, 126] from vectors to matrices and cardinality to rank constraints. Let us reconsider the previously defined mixed-projection optimization problem:

$$\min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\boldsymbol{X} \in \mathcal{S}_+^n} \quad \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \mu \cdot \mathrm{tr}(\boldsymbol{Y}) + \mathrm{tr}(f(\boldsymbol{X})) \\ \text{s.t. } \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle = b_i \quad \forall i \in [m],\ \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}, \boldsymbol{X} \in \mathcal{K},$$

[1]Specifically, if we are interested in quantum relative entropy problems where we minimize the trace of $\boldsymbol{X}_1$, as occurs in the context of the MPRT, we may achieve this using the domain-driven solver developed by [145]. However, we are not aware of any IPMs which can currently optimize over the full quantum relative entropy cone.

where similarly to [110] we assume that $f(\mathbf{0}) = \mathbf{0}$ to simplify the cut derivation procedure. Letting $\boldsymbol{\theta}$ model the epigraph of f via $\boldsymbol{\theta} \succeq f(\boldsymbol{X})$ and $\boldsymbol{S}$ be a subgradient:

$$\boldsymbol{\theta} \succeq f(\bar{\boldsymbol{X}})\boldsymbol{Y} + \boldsymbol{S}^\top(\boldsymbol{X} - \bar{\boldsymbol{X}}\boldsymbol{Y}), \tag{6.23}$$

which if $f(\boldsymbol{X}) = \boldsymbol{X}^2$ —as discussed previously—reduces to

$$\boldsymbol{\theta}^i \succeq \bar{\boldsymbol{X}}(2\boldsymbol{X} - \bar{\boldsymbol{X}}\boldsymbol{Y}),$$

which is precisely the analog of perspective cuts in the vector case. Note however that these cuts require semidefinite constraints to impose, which suggests they may not be as practically useful. For instance, Chapter 5's outer-approximation scheme for low-rank problems has a non-convex QCQOP master problem, which can only be currently solved using `Gurobi`, while `Gurobi` does not support semidefinite constraints.

We remark however that the inner product of Equation (6.23) with an arbitrary PSD matrix supplies a valid linear inequality. Two interesting cases arise when we take the inner product of the cut with a rank-one matrix or the identity matrix.

Taking an inner product with the identity matrix supplies the inequality:

$$\operatorname{tr}(\boldsymbol{\theta}) \geq \langle f(\bar{\boldsymbol{X}}), \boldsymbol{Y}\rangle + \langle \boldsymbol{S}, \boldsymbol{X} - \bar{\boldsymbol{X}}\boldsymbol{Y}\rangle \quad \forall \boldsymbol{Y} \in \mathcal{Y}_n^k. \tag{6.24}$$

Moreover, by analogy to Chapter 2, if we "project out" the $\boldsymbol{X}$ variables by decomposing the problem into a master problem in $\boldsymbol{Y}$ and subproblems in $\boldsymbol{X}$ then this cut becomes the cut derived in Chapter 5.

Alternatively, taking the inner product with a rank-one matrix $\boldsymbol{b}\boldsymbol{b}^\top$ gives:

$$\boldsymbol{b}^\top\boldsymbol{\theta}\boldsymbol{b} \geq \boldsymbol{b}^\top \left(f(\bar{\boldsymbol{X}})\boldsymbol{Y} + \boldsymbol{S}^\top(\boldsymbol{X} - \bar{\boldsymbol{X}}\boldsymbol{Y})\right)\boldsymbol{b}.$$

The analysis in this section suggests that applying a perspective cut decomposition scheme out-of-the-box may be impractical, but leaves the door open to adaptations of the scheme which account for the projection matrix structure.

6.7 Examples and Perspective Relaxations

In this section, we apply the MRPT to several important low-rank problems, in addition to the previously discussed reduced-rank regression problem. We also recall Theorem 6.2 to demonstrate that applying the MPRT to spectral functions which feature in these problems actually gives the convex hull of substructures.

Matrix Completion Revisited

Given a sample $(A_{i,j} : (i,j) \in \mathcal{I} \subseteq [n] \times [n])$ of a matrix $\boldsymbol{A} \in \mathcal{S}^n_+$, the matrix completion problem is to reconstruct the entire matrix, by assuming $\boldsymbol{A}$ is approximately low-rank [63]. Letting $\mu, \gamma > 0$ be penalty multipliers, this problem admits the formulation:

$$\min_{\boldsymbol{X} \in \mathcal{S}^n_+} \quad \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 + \frac{1}{2\gamma} \|\boldsymbol{X}\|_F^2 + \mu \cdot \mathrm{Rank}(\boldsymbol{X}). \tag{6.25}$$

Applying the MPRT to the $\|\boldsymbol{X}\|_F^2 = \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X})$ term demonstrates that this problem is equivalent to the mixed-projection problem:

$$\begin{aligned} \min_{\boldsymbol{X}, \boldsymbol{\theta} \in \mathcal{S}^n_+, \boldsymbol{Y} \in \mathcal{Y}^n_n} \quad & \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 + \frac{1}{2\gamma} \mathrm{tr}(\boldsymbol{\theta}) + \mu \cdot \mathrm{tr}(\boldsymbol{Y}) \\ \text{s.t.} \quad & \begin{pmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X} & \boldsymbol{\theta} \end{pmatrix} \succeq \boldsymbol{0}, \end{aligned}$$

and relaxing $\boldsymbol{Y} \in \mathcal{Y}^n_n$ to $\boldsymbol{Y} \in \mathrm{Conv}(\mathcal{Y}^n_n) = \{\boldsymbol{Y} \in \mathcal{S}^n : \boldsymbol{0} \preceq \boldsymbol{Y} \preceq \mathbb{I}\}$ supplies a valid relaxation. We now argue that this relaxation is often high-quality, by demonstrating that the MPRT supplies the convex envelope of $t \geq \frac{1}{2\gamma} \|\boldsymbol{X}\|_F^2 + \mu \cdot \mathrm{Rank}(\boldsymbol{X})$, via the following corollary to Theorem 6.2:

Corollary 6.2.

$$\textit{Let} \quad \mathcal{S} = \left\{(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\theta}) \in \mathcal{Y}^k_n \times \mathcal{S}^n_+ \times \mathcal{S}^n : \boldsymbol{\theta} \succeq \boldsymbol{X}^\top \boldsymbol{X}, u\boldsymbol{Y} \succeq \boldsymbol{X} \succeq \ell \boldsymbol{Y}\right\}$$

be a set where $\ell, u \in \mathbb{R}_+$. *Then, this set's convex hull is given by:*

$$\mathcal{S}^c = \Big\{(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\theta}) \in \mathcal{S}_+^n \times \mathcal{S}_+^n \times \mathcal{S}^n : \\ \boldsymbol{Y} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{Y}) \leq k, u\boldsymbol{Y} \succeq \boldsymbol{X} \succeq \ell \boldsymbol{Y}, \begin{pmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{\theta} \end{pmatrix} \succeq \mathbf{0}\Big\}.$$

Tensor Completion

A central problem in machine learning is to reconstruct a d-tensor $\boldsymbol{\mathcal{X}}$ given a subsample of its entries $(A_{i_1,\ldots i_d} : (i_1, \ldots i_d) \in \mathcal{I} \subseteq [n_1] \times [n_2] \times \ldots \times [n_d])$, by assuming that the tensor is low-rank. Since even evaluating the rank of a tensor is NP-hard [152], a popular approach for solving this problem is to minimize the reconstruction error while constraining the ranks of different unfoldings of the tensor [see, e.g., 119]. After imposing Frobenius norm regularization and letting $\|\cdot\|_{HS} = \sqrt{\sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} X_{i_1,\ldots,i_d}^2}$ denote the (second-order cone representable) Hilbert-Schmidt norm of a tensor, this leads to optimization problems of the form:

$$\begin{aligned} \min_{\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}} \quad & \sum_{(i_1,\ldots i_d) \in \mathcal{I}} (\mathcal{A}_{i_1,\ldots i_d} - \mathcal{X}_{i_1,\ldots i_d})^2 + \sum_{i=1}^{n} \|\mathcal{X}_{(i)}\|_F^2 \\ \text{s.t.} \quad & \mathrm{Rank}(\mathcal{X}_{(i)}) \leq k \quad \forall i \in [n]. \end{aligned} \tag{6.26}$$

Similarly to low-rank matrix completion, it is tempting to apply the MRPT to model the $\boldsymbol{X}_{(i)}^\top \boldsymbol{X}_{(i)}$ term for each mode-n unfolding. We now demonstrate this supplies a tight approximation of the convex hull of the sum of the regularizers, via the following lemma (proof omitted, follows in the spirit of [126, Lemma 4]):

Lemma 6.4.

$$\textit{Let} \quad \mathcal{Q} = \Big\{(\rho, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) : \\ \rho \geq \sum_{i=1}^{m} q_i \mathrm{tr}(\boldsymbol{\theta}_i), (\boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{\theta}_i) \in \mathcal{S}^i \ \forall i \in [m]\Big\}$$

be a set where $l_i, u_i, q_i \in \mathbb{R}^n_+ \ \forall i \in [m]$*, and* $\mathcal{S}_i$ *is a set of the same form as* $\mathcal{S}$*, but* l, u *are replaced by* l_i, u_i*. Then, an extended formulation of the convex hull is given by:*

$$\mathcal{Q}^c = \Bigg\{ (\rho, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) : \\ \rho \geq \sum_{i=1}^m q_i \text{tr}(\boldsymbol{\theta}_i), (\boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{\theta}_i) \in \mathcal{S}_i^c \ \forall i \in [m] \Bigg\}.$$

Lemma 6.4 suggests that the MPRT may improve algorithms which aim to recover tensors of low slice rank. For instance, in low-rank tensor problems where (6.26) admits multiple local solutions, solving the convex relaxation coming from $\mathcal{Q}^c$ and greedily rounding may give a high-quality initial point for an alternating minimization method such as the method of [99], and indeed allow such a strategy to return better solutions than if it were initialized at a random point.

Note however that Lemma 6.4 does not necessarily give the convex hull of the sum of the regularizers, since the regularization terms involve different slices of the same tensor and thus interact; see also [206] for a related proof that the tensor trace norm does not give the convex envelope of the sum of ranks of slices.

Low-Rank Factor Analysis

An important problem in statistics, psychometrics and economics is to decompose a covariance matrix $\boldsymbol{\Sigma} \in \mathcal{S}^n_+$ into a low-rank matrix $\boldsymbol{X} \in \mathcal{S}^n_+$ plus a diagonal matrix $\boldsymbol{\Phi} \in \mathcal{S}^n_+$, as explored by [31] and references therein. This corresponds to solving:

$$\min_{\boldsymbol{X}, \boldsymbol{\Phi} \in \mathcal{S}^n_+} \ \|\boldsymbol{\Sigma} - \boldsymbol{\Phi} - \boldsymbol{X}\|_q^q \text{ s.t. } \text{Rank}(\boldsymbol{X}) \leq k, \ \Phi_{i,j} = 0, \forall i, j : i \neq j, \ \|\boldsymbol{X}\|_\sigma \leq M \quad (6.27)$$

where $q \geq 1$, $\|\boldsymbol{X}\|_q = (\sum_{i=1}^n \lambda_i(\boldsymbol{X})^q)^{\frac{1}{q}}$ denotes the matrix q norm, and we constrain the spectral norm of $\boldsymbol{X}$ via a big-M constraint for the sake of tractability.

This problem's objective involves minimizing $\text{tr}\,(\boldsymbol{\Sigma} - \boldsymbol{\Phi} - \boldsymbol{X})^q$, and it is not immediately obvious how to either apply the technique in the presence of the $\boldsymbol{\Phi}$ variables or alternatively seperate out the $\boldsymbol{\Phi}$ term and apply the MPRT to an appropriate

($\mathbf{\Phi}$-free) substructure. To proceed, let us therefore first consider its scalar analog, obtaining the convex closure of the following set:

$$\mathcal{T} = \{(x, y, z, t) \in \mathbb{R} \times \mathbb{R} \times \{0, 1\} \times \mathbb{R}^+ : t \geq |x + y - d|^q, |x| \leq M, x = zx\},$$

where $d \in \mathbb{R}$ and $q \geq 1$ are fixed constants, and we require that $|x| \leq M$ for the sake of tractability. We obtain the convex closure via the following proposition:

Proposition 6.5. *The convex closure of the set $\mathcal{T}$, $\mathcal{T}^c$, is given by:*

$$\mathcal{T}^c = \Bigg\{(x, y, z, t) \in \mathbb{R} \times \mathbb{R} \times [0, 1] \times \mathbb{R}^+ : \exists \beta \geq 0 : \\ t \geq \frac{|y - \beta - d(1 - z)|^q}{(1 - z)^{q-1}} + \frac{|x + \beta - dz|^q}{z^{q-1}},\ |x| \leq Mz\Bigg\}.$$

Remark 23. *To check that this set is indeed a valid convex relaxation, observe that if $z = 0$ then $x = 0$ and $x = -\beta \implies \beta = 0$ and $t \geq |y - d|^q$, while if $z = 1$ then $y = \beta$ and $t \geq |x + y - d|^q$.*

Proof. We use the proof technique laid out in [128, Section 3.1], namely writing $\mathcal{T}$ as the disjunction of two convex sets driven by whether z is active and applying Fourier-Motzkin elimination. That is, we have $\mathcal{T} = \mathcal{T}^1 \cup \mathcal{T}^2$ where:

$$\mathcal{T}^1 = \{(0, y_1, 0, t_1) : t_1 \geq |y_1 - d|^q\},$$
$$\mathcal{T}^2 = \{(x_2, y_2, 1, t_2) : t_2 \geq |x_2 - y_2 - d|^q, |x_2| \leq M\}.$$

Moreover, a point (x, y, z, t) is in the convex hull $\mathcal{T}^c$ if and only if it can be written as a convex combination of points in $\mathcal{T}^1, \mathcal{T}^2$. Letting λ_1, λ_2 denote the weight of points in this system, we then have that $(x, y, z, t) \in \mathcal{T}^c$ if and only if the following system

admits a solution:

$$\begin{aligned}
&\lambda_1 + \lambda_2 = 1,\\
&x = \lambda_2 x_2,\\
&y = \lambda_1 y_1 + \lambda_2 y_2,\\
&t = \lambda_1 t_1 + \lambda_2 t_2,\\
&z = \lambda_2,\\
&t_1 \geq |y_1 - d|^q,\\
&t_2 \geq |x_2 + y_2 - d|^q,\\
&\lambda_1, \lambda_2 \geq 0,\\
&|x_2| \leq M.
\end{aligned} \tag{6.28}$$

For ease of computation, we now eliminate variables. First, one can substitute t_1, t_2 for their lower bounds in the definition of t and replace λ_2 with z to obtain

$$\begin{aligned}
&\lambda_1 + z = 1,\\
&x = z x_2,\\
&y = \lambda_1 y_1 + z y_2,\\
&t \geq \lambda_1 |y_1 - d|^q + z|x_2 + y_2 - d|^q,\\
&\lambda_1, z \geq 0,\\
&|x_2| \leq M.
\end{aligned} \tag{6.29}$$

Next, we substitute x/z for x_2 and $(y - zy_2)/\lambda_1$ for y_1 to obtain

$$\begin{aligned}
&\lambda_1 + z = 1,\ \lambda_1, z \geq 0,\ |x| \leq Mz\\
&t \geq \frac{1}{\lambda_1^{q-1}}|y - y_2 z - d(1-z)|^q + \frac{1}{z^{q-1}}|x + y_2 z - dz|^q.
\end{aligned} \tag{6.30}$$

Finally, we let zy_2 be the free variable β and set $\lambda_1 = 1 - z$ to obtain the required convex set. □

Observe that $\mathcal{T}^c$ can be modeled using two power cones and one inequality.

Proposition 6.5 suggests that we can obtain high-quality convex relaxations for

low-rank factor analysis problems via a judicious use of the matrix power cone. Namely, introduce an epigraph matrix $\boldsymbol{\theta}$ to model the eigenvalues of $(\boldsymbol{\Sigma} - \boldsymbol{\Phi} - \boldsymbol{X})^q$ and an orthogonal projection matrix $\boldsymbol{Y}_2$ to model the span of $\boldsymbol{X}$. This then leads to the following matrix power cone representable relaxation:

$$\begin{aligned}
\min_{\boldsymbol{X},\boldsymbol{\Phi},\boldsymbol{\theta},\boldsymbol{Y}_1,\boldsymbol{Y}_2 \in \mathcal{S}^n_+, \boldsymbol{\beta} \in \mathcal{S}^n} \quad & \operatorname{tr}(\boldsymbol{\theta}) \\
\text{s.t.} \quad & \boldsymbol{\theta} \succeq \boldsymbol{Y}_1^{\frac{1-q}{2}} (\boldsymbol{Y}_1^{\frac{1}{2}} \boldsymbol{\Sigma} \boldsymbol{Y}_1^{\frac{1}{2}} - \boldsymbol{\beta} - \boldsymbol{\Phi}) \boldsymbol{Y}_1^{\frac{1-q}{2}} \\
& \quad + \boldsymbol{Y}_2^{\frac{1-q}{2}} (\boldsymbol{Y}_2^{\frac{1}{2}} \boldsymbol{\Sigma} \boldsymbol{Y}_2^{\frac{1}{2}} + \boldsymbol{\beta} - \boldsymbol{X}) \boldsymbol{Y}_2^{\frac{1-q}{2}}, \\
& \boldsymbol{Y}_1 + \boldsymbol{Y}_2 = \mathbb{I}, \operatorname{tr}(\boldsymbol{Y}) \leq k, \Phi_{i,j} = 0, \forall i, j \in [n] : i \neq j, \\
& \boldsymbol{\Phi} \preceq \boldsymbol{X}, \boldsymbol{X} \preceq M \boldsymbol{Y}_2, -\boldsymbol{X} \preceq M \boldsymbol{Y}_2.
\end{aligned}$$

Optimal Experimental Design

Letting $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ where $m \geq n$ be a matrix of linear measurements of the form $y_i = \boldsymbol{a}_i^\top \boldsymbol{\beta} + \epsilon_i$ from an experimental setting, the D-optimal experimental design problem (a.k.a. the sensor selection problem) is to pick $k \leq m$ of these experiments in order to make the most accurate estimate of $\boldsymbol{\beta}$ possible, by solving [see 142, 210, for a modern approach]:

$$\max_{\boldsymbol{z} \in \{0,1\}^n : \boldsymbol{e}^\top \boldsymbol{z} \leq k} \quad \log\det_\epsilon \left(\sum_{i \in [n]} z_i \boldsymbol{a}_i \boldsymbol{a}_i^\top \right), \tag{6.31}$$

where we define $\log\det_\epsilon(\boldsymbol{X}) = \sum_{i=1}^n \log(\lambda_i(\boldsymbol{X}) + \epsilon)$ for $\epsilon > 0$ to be the pseudo log-determinant of a rank-deficient PSD matrix, which can be thought of as imposing an uninformative prior of importance ϵ on the experimental design process. Since $\log\det(\boldsymbol{X}) = \operatorname{tr}(\log(\boldsymbol{X}))$ and $\log\det(\boldsymbol{X} + \epsilon\mathbb{I}) = \log\det_\epsilon(\boldsymbol{X})$ for $\boldsymbol{X} \succeq \boldsymbol{0}$ and $\epsilon > 0$, a valid convex relaxation is given by:

$$\max_{\boldsymbol{z} \in [0,1]^n, \boldsymbol{\theta} \in \mathcal{S}^n_+} \quad \operatorname{tr}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \log\left(\boldsymbol{A} \operatorname{Diag}(\boldsymbol{z}) \boldsymbol{A}^\top + \epsilon\mathbb{I}\right) \succeq \boldsymbol{\theta},$$

which can be modeled using the quantum entropy cone: $(-\boldsymbol{\theta}, \mathbb{I}, \boldsymbol{A}\mathrm{Diag}(\boldsymbol{z})\boldsymbol{A}^\top + \epsilon\mathbb{I}) \in \mathcal{K}_{\mathrm{mat}}^{\mathrm{rel,\ op}}$. This is equivalent to perhaps the most common relaxation of D-optimal design, as proposed by Boyd and Vandenberghe [54, Eqn. 7.2.6]. By formulating in terms of the quantum relative entropy cone, the identity term suggests this relaxation leaves something "on the table".

In this direction, let us apply the MPRT. Observe that $\boldsymbol{X} := \sum_{i\in[n]} z_i \boldsymbol{a}_i \boldsymbol{a}_i^\top$ is a rank-k matrix and thus at an optimal solution to the original problem there is some orthogonal projection matrix $\boldsymbol{Y}$ such that $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$. Therefore, we can take the perspective function of $f(\boldsymbol{X}) = \log(\boldsymbol{X} + \epsilon\mathbb{I})$, and thereby obtain the following valid—and potentially much tighter when $k < n$—convex relaxation:

$$\begin{aligned} \max_{\boldsymbol{z}\in[0,1]^n, \boldsymbol{\theta}, \boldsymbol{Y}\in\mathcal{S}_+^n} \quad & \mathrm{tr}(\boldsymbol{\theta}) + (n - \mathrm{tr}(\boldsymbol{Y}))\log(\epsilon) \\ \text{s.t.} \quad & \boldsymbol{Y}^{\frac{1}{2}} \log\left(\boldsymbol{Y}^{-\frac{1}{2}}(\boldsymbol{A}\mathrm{Diag}(\boldsymbol{z})\boldsymbol{A}^\top + \epsilon\boldsymbol{Y})\boldsymbol{Y}^{-\frac{1}{2}}\right)\boldsymbol{Y}^{\frac{1}{2}} \succeq \boldsymbol{\theta}, \\ & \boldsymbol{Y} \preceq \mathbb{I}, \mathrm{tr}(\boldsymbol{Y}) \leq k, \end{aligned} \tag{6.32}$$

which can be modeled via the quantum relative entropy cone:

$$(-\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{A}\mathrm{Diag}(\boldsymbol{z})\boldsymbol{A}^\top + \epsilon\boldsymbol{Y}) \in \mathcal{K}_{\mathrm{mat}}^{\mathrm{rel,\ op}}.$$

We now argue that this relaxation is high-quality, by demonstrating that the MPRT supplies the convex envelope of $t \geq -\log\det_\epsilon(\boldsymbol{X})$ under a low-rank constraint, via the following corollary to Theorem 6.2:

Corollary 6.3.

$$\textit{Let} \quad \mathcal{S} = \left\{\boldsymbol{X} \in \mathcal{S}_+^n : t \geq -\log\det_\epsilon(\boldsymbol{X}), \mathrm{Rank}(\boldsymbol{X}) \leq k\right\}$$

be a set where ϵ, k, t are fixed. Then, this set's convex hull is:

$$\mathcal{S}^c = \Bigg\{(\boldsymbol{Y}, \boldsymbol{X}) \in \mathcal{S}_+^n \times \mathcal{S}_+^n : \boldsymbol{0} \preceq \boldsymbol{Y} \preceq \mathbb{I}, \operatorname{tr}(\boldsymbol{Y}) \leq k, \\ t \geq -\operatorname{tr}(\boldsymbol{Y}^{\frac{1}{2}} \log_\epsilon(\boldsymbol{X}^{\frac{1}{2}} \boldsymbol{Y}^\dagger \boldsymbol{X}^{\frac{1}{2}}) - (n - \operatorname{tr}(\boldsymbol{Y})) \log(\epsilon)\Bigg\}.$$

Remark 24. *Observe that* (6.32)*'s relaxation is not useful in the over-determined regime where $k \geq n$, since setting $\boldsymbol{Y} = \mathbb{I}$ recovers* (6.31)*'s Boolean relaxation, which is considerably cheaper to optimize over. Accordingly, we only consider the underdetermined regime in our experiments.*

Non-Negative Matrix Optimization

Many important problems in combinatorial optimization, statistics and computer vision [see, e.g., 57] reduce to optimizing over the space of low-rank matrices with non-negative factors. An important special case is when we would like to find the low-rank completely positive matrix $\boldsymbol{X}$ which best approximates (in a least-squares sense) a given matrix $\boldsymbol{A} \in \mathcal{S}_+^n$, i.e., perform non-negative principal component analysis. Formally, we have the problem:

$$\min_{\boldsymbol{X} \in \mathcal{C}_+^n : \operatorname{Rank}(\boldsymbol{X}) \leq k} \quad \|\boldsymbol{X} - \boldsymbol{A}\|_F^2, \tag{6.33}$$

where $\mathcal{C}_+^n := \{\boldsymbol{U}\boldsymbol{U}^\top : \boldsymbol{U} \in \mathbb{R}_+^{n \times n}\}$ is the cone of completely positive matrices.

Applying the MPRT to the strongly convex $\frac{1}{2}\|\boldsymbol{X}\|_F^2$ term in the objective therefore yields the following completely positive program:

$$\min_{\boldsymbol{X} \in \mathcal{C}_+^n, \boldsymbol{Y}, \boldsymbol{\theta} \in \mathcal{S}^n} \quad \frac{1}{2}\operatorname{tr}(\boldsymbol{\theta}) - \langle \boldsymbol{X}, \boldsymbol{A} \rangle + \frac{1}{2}\|\boldsymbol{A}\|_F^2 \tag{6.34}$$

$$\text{s.t. } \boldsymbol{Y} \preceq \mathbb{I}, \ \operatorname{tr}(\boldsymbol{Y}) \leq k, \ \begin{pmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{\theta} \end{pmatrix} \in S_+^{2n}. \tag{6.35}$$

Interestingly, since (6.34)'s reformulation has a linear objective, some extreme point in its relaxation is optimal, which means we can relax the requirement that $\boldsymbol{Y}$ is

a projection matrix without loss of optimality and the computational complexity of the problem is entirely concentrated in the completely positive cone. Unfortunately however, completely positive optimization itself is intractable. Nonetheless, it can be approximated by replacing the completely positive cone with the doubly non-negative cone, $\mathcal{S}^n_+ \cap \mathbb{R}^{n\times n}_+$. Namely, we instead solve

$$\min_{\boldsymbol{X}\in\mathcal{S}^n_+\cap\mathbb{R}^{n\times n}_+,\boldsymbol{Y},\boldsymbol{\theta}\in\mathcal{S}^n} \frac{1}{2}\mathrm{tr}(\boldsymbol{\theta}) - \langle \boldsymbol{X}, \boldsymbol{A}\rangle + \frac{1}{2}\|\boldsymbol{A}\|_F^2 \tag{6.36}$$

$$\text{s.t. } \begin{pmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{\theta} \end{pmatrix} \in S^{2n}_+,\ \boldsymbol{Y} \preceq \mathbb{I},\ \mathrm{tr}(\boldsymbol{Y}) \leq k. \tag{6.37}$$

Remark 25. *If $\boldsymbol{X} = \boldsymbol{D}\boldsymbol{\Pi}$ is a monomial matrix, i.e., decomposable as the product of a diagonal matrix $\boldsymbol{D}$ and a permutation matrix $\boldsymbol{\Pi}$, as occurs in binary optimization problems such as k-means clustering problems among others [c.f. 191], then it follows that $(\boldsymbol{X}^\top\boldsymbol{X})^\dagger \geq \boldsymbol{0}$ [see 194] and thus $\boldsymbol{Y} := \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^\dagger\boldsymbol{X}^\top$ is elementwise non-negative. In this case, the doubly non-negative relaxation* (6.36) *should be strengthened by requiring that $\boldsymbol{Y} \geq \boldsymbol{0}$.*

6.8 Numerical Results

In this section, we evaluate the algorithmic strategies derived in the previous section, implemented in `Julia` 1.5 using `JuMP.jl` 0.21.6 and `Mosek` 9.1 to solve the conic problems considered here. Except where indicated otherwise, all experiments were performed on a Intel Xeon E5—2690 v4 2.6GHz CPU core using 32 GB RAM. To bridge the gap between theory and practice, we have made our code available at `github.com/ryancorywright/MatrixPerspectiveSoftware`.

Reduced Rank Regression

In this section, we compare our convex relaxations for reduced rank regression developed in the introduction and laid out in (6.6)-(6.7)—which we refer to as "Persp" and "DCL" respectively—against the nuclear norm estimator proposed by [181] ("NN"),

who solve

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{p\times n}} \quad \frac{1}{2m}\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}\|_F^2+\frac{1}{2\gamma}\|\boldsymbol{\beta}\|_F^2+\mu\|\boldsymbol{\beta}\|_*. \tag{6.38}$$

Similarly to [181], we attempt to recover rank$-k_{true}$ estimators $\boldsymbol{\beta}_{\text{true}}=\boldsymbol{U}\boldsymbol{V}^\top$, where each entry of $\boldsymbol{U}\in\mathbb{R}^{p\times k_{true}},\boldsymbol{V}\in\mathbb{R}^{n\times k_{true}}$ is i.i.d. standard Gaussian $\mathcal{N}(0,1)$, the matrix $\boldsymbol{X}\in\mathbb{R}^{m\times p}$ contains i.i.d. standard Gaussian $\mathcal{N}(0,1)$ entries, $\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{E}$, and $E_{i,j}\sim\mathcal{N}(0,\sigma)$ injects a small amount of i.i.d. noise. We set $n=p=50,k=10$, $\gamma=10^6$, $\sigma=0.05$ and vary m. To ensure a fair comparison, we cross-validate μ for both of our relaxations and [181]'s approach so as to minimize the MSE on a validation set. For each m, we evaluate 20 different values of μ which are distributed uniformly in logspace between 10^{-4} and 10^4 across 50 random instances for our convex relaxations and report on 100 different random instances with the "best" μ for each method and each p.

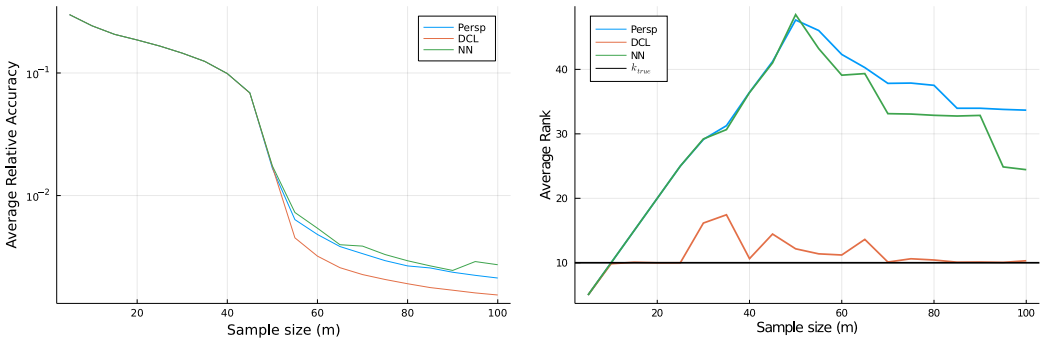
Rank recovery and statistical accuracy: Figures 6-1a-6-1c report the relative accuracy ($\|\boldsymbol{\beta}_{\text{est}}-\boldsymbol{\beta}_{\text{true}}\|_F/\|\boldsymbol{\beta}_{\text{true}}\|_F$), the rank (i.e., number of singular values of $\boldsymbol{\beta}_{\text{est}}$ which exceed 10^{-4}), and the out-of-sample MSE[2] $\|\boldsymbol{X}_{\text{new}}\boldsymbol{\beta}_{\text{est}}-\boldsymbol{y}_{\text{new}}\|_F^2$ (normalized by the out-of-sample MSE of the ground truth $\|\boldsymbol{X}_{\text{new}}\boldsymbol{\beta}_{\text{true}}-\boldsymbol{y}_{\text{new}}\|_F^2$). Results are averaged over 100 random instances per value of m. We observe that—even though we did not supply the true rank of the optimal solution in our formulation—Problem (6.7)'s relaxation returns solutions of the correct rank ($k_{true}=10$) and better MSE/accuracy, while our more "naive" perspective relaxation (6.6) and the nuclear norm approach (6.38) return solutions of a higher rank and lower accuracy. This suggests that (6.7)'s formulation should be considered as a more accurate estimator for reduced rank problems, and empirically confirms that the MPRT can lead to significant improvements in statistical accuracy.

[2]Evaluated on $m=1000$ new observations of $\boldsymbol{X}_j,\boldsymbol{Y}_k$ generated from the same distribution.

Scalability w.r.t. m: Figure 6-1d reports the average time for `Mosek` to converge[3] to an optimal solution (over 100 random instances per m). Surprisingly, although (6.7) is a stronger relaxation than (6.6), it is one to two orders of magnitude faster than (6.6) and (6.38)'s formulations. The relative scalability of (6.7)'s formulation as m—the number of observation— increases can be explained by the fact that (6.7) considers a inner product of the Gram matrix $\boldsymbol{X}^\top \boldsymbol{X}$ with a semidefinite matrix $\boldsymbol{B}$ (the size of which does not vary with m) while Problems (6.6)-(6.38) have a quadratic inner product $\langle \boldsymbol{\beta}\boldsymbol{\beta}^\top, \boldsymbol{X}^\top \boldsymbol{X} \rangle$ which must be modeled using a rotated second-order cone constraint (the size of which depends on m), since modern conic solvers such as `Mosek` do not allow quadratic objective terms and semidefinite constraints to be simultaneously present (if they did, all three formulations would scale similarly).

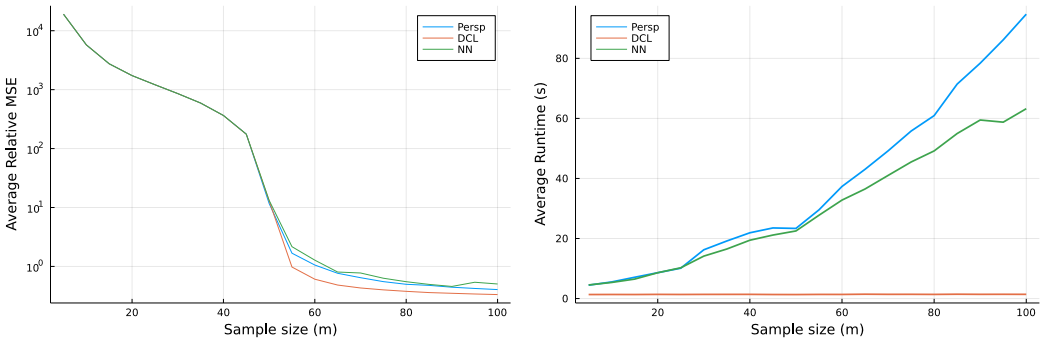
Scalability w.r.t p: Next, we evaluate the scalability of all three approaches in terms of their solve times and peak memory usage (measured using the `slurm` command `MaxRSS`), as $n = p$ increases. Fig. 6-2 depicts the average time to converge to an optimal solution (a) and peak memory consumption (b) by each method as we vary $n = p$ with $m = n$, $k = 10$, $\gamma = 10^6$, each μ fixed to the average cross-validated value found in the previous experiment, a peak memory budget of 120GB, a runtime budget of 12 hours, and otherwise the same experimental setup as previously (averaged over 20 random instances per n). We observe (6.7)'s relaxation is dramatically more scalable than the other two approaches considered, and can solve problems of nearly twice the size (4 times as many variables), and solves problems of a similar size in substantially less time and with substantially less peak memory consumption (40s vs. 1000s when $n = 100$). All in all, the proposed relaxation (6.7) seems to be the best method of the three considered.

[3]We model the convex quadratic $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Y}\|_F^2$ using a rotated second order cone for formulations (6.6) and (6.38) (the quadratic term doesn't appear directly in (6.7)), model the nuclear norm term in (6.38) by introducing matrices $\boldsymbol{U}, \boldsymbol{V}$ such that $\begin{pmatrix} \boldsymbol{U} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^\top & \boldsymbol{V} \end{pmatrix} \succeq \boldsymbol{0}$ and minimizing $\mathrm{tr}(\boldsymbol{U}) + \mathrm{tr}(\boldsymbol{V})$, use default Mosek parameters for all approaches.



(a) Accuracy

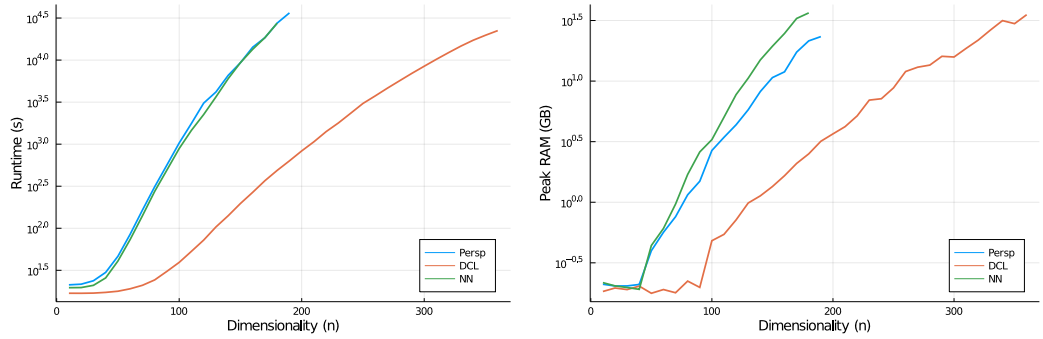
(b) Rank



(c) Relative MSE

(d) Runtime

Figure 6-1: Comparative performance, as the number of samples m increases, of formulations (6.6) (Persp, in blue), (6.7) (DCL, in orange) and (6.38) (NN, in green), averaged over 100 synthetic reduced rank regression instances where $n = p = 50$, $k_{true} = 10$. The hyperparameter μ was first cross-validated for all approaches separately.



(a) Runtime

(b) Peak Memory

Figure 6-2: Average time to compute an optimal solution (left panel) and peak memory usage (right panel) vs. dimensionality $n = p$ for Problems (6.6) (Persp, in blue), (6.7) (DCL. in orange) and (6.38) (NN, in green) over 20 synthetic reduced rank regression instances where $k_{true} = 10$.

Non-Negative Matrix Factorization

In this section, we benchmark the quality of our dual bound for non-negative matrix factorization laid out in Section 6.7 by using the non-linear reformulation strategy proposed by [58] (alternating least squares or ALS) to obtain upper bounds. Namely, we obtain upper bounds by solving for local minima of the problem

$$\min_{\boldsymbol{U}\in\mathbb{R}_+^{n\times k}} \quad \|\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{A}\|_F^2. \tag{6.39}$$

In our implementation of ALS, we obtain a local minimum by introducing a dummy variable $\boldsymbol{V}$ which equals $\boldsymbol{U}$ at optimality and alternating between solving the following two problems

$$\boldsymbol{U}_{t+1} = \arg\min_{\boldsymbol{U}\in\mathbb{R}_+^{n\times k}} \quad \|\boldsymbol{U}\boldsymbol{V}_t^\top - \boldsymbol{A}\|_F^2 + \rho_t\|\boldsymbol{U} - \boldsymbol{V}_t\|_F^2, \tag{6.40}$$

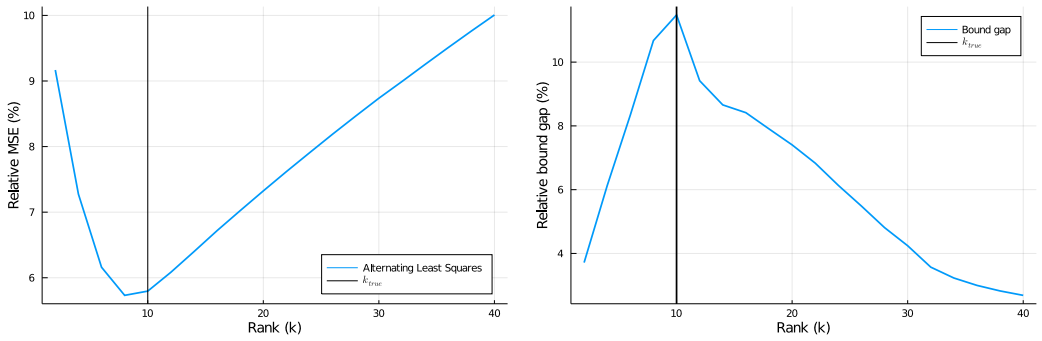
$$\boldsymbol{V}_{t+1} = \arg\min_{\boldsymbol{V}\in\mathbb{R}_+^{n\times k}} \quad \|\boldsymbol{U}_t\boldsymbol{V}^\top - \boldsymbol{A}\|_F^2 + \rho_t\|\boldsymbol{U}_t - \boldsymbol{V}\|_F^2, \tag{6.41}$$

where we set $\rho_t = \min(10^{-4}\times 2^{t-1}, 10^5)$ at the tth iteration in order that the final matrix is positive semidefinite, as advocated in [22, Section 5.2.3] (we cap ρ_t to avoid numerical instability). We iterate over solving these two problems from a random initialization point $\boldsymbol{V}_0$—where each $V_{0,i,j}$ is i.i.d. standard uniform—until either the objective between iterations does not change by 10^{-4} or we exceed the maximum number of allowable iterations, which we set to 100.

To generate problem instances, we let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^\top + \boldsymbol{E}$ where $\boldsymbol{U}\in\mathbb{R}^{n\times k_{true}}$, each $U_{i,j}$ is uniform on $[0,1]$, $E_{i,j}\sim\mathcal{N}(0, 0.0125k_{true})$, and set $A_{i,j}=0$ if $A_{i,j}<0$. We set $n=50, k_{true}=10$. We use the ALS heuristic to compute a feasible solution $\boldsymbol{X}$ and an upper-bound on the problem's objective value. By comparing it with the lower bound derived from our MPRT, we can assess the sub-optimality of the heuristic solution, which previously lacked optimality guarantees.

Figure 6-3 depicts the average in-sample MSE of the heuristic ($\|\boldsymbol{X}-\boldsymbol{A}\|_F/\|\boldsymbol{A}\|_F$) and the relative bound gap—(UB-LB)/UB— as we vary the target rank, averaged

over 100 random synthetic instances. We observe that the method is most accurate and has the lowest MSE when k is set to $k_{true} = 10$, which confirms that the method can recover solutions of the correct rank. In addition, by combining the solution from OLS with our lower-bound, we can compute a duality gap and assert that the heuristic solution is $0\% - 3\%$-optimal, with the gap peaking at $k = k_{true}$ and stabilizing as $k \to n$. This echoes similar findings in k-means clustering and alternating current optimal power flow problems, where the SDO relaxation need not be near-tight in theory but nonetheless is nearly exact in practice [191, 154]. Further, this suggests our convex relaxation may be a powerful weapon for providing gaps for heuristics for non-negative matrix factorization, and particularly detecting when they are performing well or can be further improved.



(a) Relative MSE **(b)** Bound gap

Figure 6-3: Average relative MSE and duality gap vs. target rank k using the ALS heuristic (UB) and the MPRT relaxation (LB). Results are averaged over 100 synthetic completely positive matrix factorization instances where $n = 50$, $k_{true} = 10$.

Figure 6-4 reports the time needed to compute both the upper bound and a lower bound solution as we vary the target rank.

Optimal Experimental Design

In this section, we benchmark our dual bound for D-optimal experimental design (6.32) against the convex relaxation (6.31) and a greedy submodular maximization approach, in terms of both bound quality and the ability of all three approaches

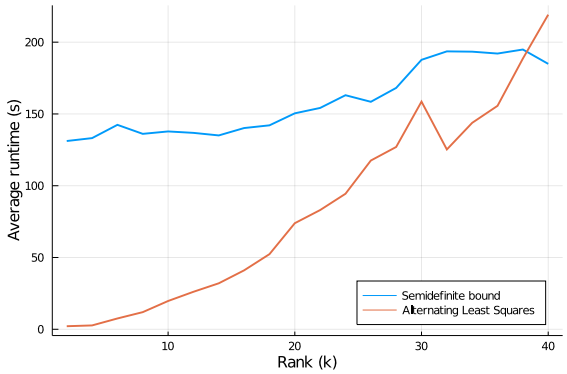


Figure 6-4: Computational time to compute a feasible solution (ALS) and solve the relaxation (Semidefinite bound) vs. target rank k, averaged over 100 synthetic completely positive matrix factorization instances where $n = 50$, $k_{true} = 10$.

to generate high-quality feasible solutions. We round both relaxations to generate feasible solutions greedily, by setting the k largest z_i's in a continuous relaxation to 1, while for the submodular maximization approach we iteratively set the jth index of $\boldsymbol{z}$ to 1, where $\mathcal{S}$ is initially an empty set and we iteratively take

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{j\} : j \in \arg\max_{i \in [n] \setminus \mathcal{S}} \left\{ \log\det_{\epsilon} \left(\sum_{l \in \mathcal{S}} z_l \boldsymbol{a}_l \boldsymbol{a}_l^\top + \boldsymbol{a}_i \boldsymbol{a}_i^\top \right) \right\}.$$

Interestingly, the greedy rounding approach enjoys rigorous approximation guarantees [see 142, 210], while the submodular maximization approach also enjoys strong guarantees [see 182].

We benchmark all methods in terms of their performance on synthetic D-optimal experimental design problems, where we let $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ be a matrix with i.i.d. $\mathcal{N}(0, \frac{1}{\sqrt{n}})$ entries. We set $n = 20, m = 10, \epsilon = 10^{-6}$ and vary $k < m$ over 20 random instances. Table 6.3 depicts the average relative bound gap, objective values, and runtimes for all 3 methods (we use the lower bound from (6.31)'s relaxation to compute the submodular bound gap). Note that all results for this experiment were generated on a standard Macbook pro laptop with a 2.9GHZ 6-core Intel i9 CPU using 16GB DDR4 RAM, CVX version 1.22, Matlab R2021a, and `Mosek` 9.1. Moreover, we optimize over (6.32)'s relaxation using the `CVXQuad` package developed by [101].

Table 6.3: Average runtime in seconds and relative bound gap per approach, over 20 random instances where $n = 10, m = 20$.

	Problem (6.31)+round		Submodular		Problem (6.32)+round	
k	Time(s)	Gap (%)	Time(s)	Gap (%)	Time(s)	Gap (%)
1	0.52	88.8	0.00	88.9	347.0	0.00
2	0.63	93.7	0.00	93.7	338.5	0.01
3	0.59	97.1	0.00	97.0	320.8	0.06
4	0.63	100.2	0.00	100.2	338.7	0.18
5	0.53	103.8	0.00	103.9	331.1	0.37
6	0.53	109.0	0.00	109.0	287.5	1.40
7	0.55	117.7	0.00	117.7	255.1	2.39
8	0.60	136.9	0.00	138.5	236.1	5.25
9	0.54	260.9	0.00	287.5	235.9	28.43

Relaxation quality: We observe that (6.32)'s relaxation is dramatically stronger than (6.31), offering bound gaps on the order of $0\% - 3\%$ when $k \leq 7$, rather than gaps of 90% or more. This confirms the efficacy of the MPRT, and demonstrates the value of taking low-rank constraints into account when designing convex relaxations, even when not obviously present.

Scalability: We observe that (6.32)'s relaxation is around two orders of magnitude slower than the other proposed approaches, largely because semidefinite approximations of quantum relative entropy are expensive, but is still tractable for moderate sizes. We believe that the relaxation would scale significantly better if it were optimized over using an interior point method for non-symmetric cones [see, e.g., 211, 145], or an alternating minimization approach [see 102]. As such, (6.32)'s relaxation is potentially useful at moderate problem sizes with off-the-shelf software, or at larger problem sizes with problem-specific techniques such as alternating minimization.

6.9 Conclusion

In this chapter, we introduced the Matrix Perspective Reformulation Technique, or MPRT, a new technique for deriving tractable and often high-quality relaxations of a wide variety of low-rank problems. We also invoked the technique to derive the convex hulls of some frequently-studied low-rank sets, and provided examples where

the technique proves useful in practice. This is significant and potentially useful to the community, because substantial progress on producing tractable upper bounds for low-rank problems has been made over the past decade, but until now almost no progress on tractable lower bounds has followed.

6.10 Appendix: Generalizing MPRT to Functions

We now demonstrate the MPRT can be extended to incorporate a different separability of eigenvalues assumption, at the price of (a possibly significant amount of) additional notations. For any symmetric matrix $\boldsymbol{X}$, let us denote $\lambda_i^{\downarrow}(\boldsymbol{X})$ the ith largest eigenvalue of $\boldsymbol{X}$. Before proceeding any further, we recall the following result, due to [17, Example 18.c], which provides a semidefinite representation of the sum of the k largest eigenvalues:

Lemma 6.5. *Let $S_k(\boldsymbol{X}) := \sum_{i=1}^{k} \lambda_i^{\downarrow}(\boldsymbol{X})$ denote the sum of the k largest eigenvalues of a symmetric matrix $\boldsymbol{X} \in \mathcal{S}^n$. Then, the epigraph of S_k, $S_k(\boldsymbol{X}) \leq t_k$, admits the following semidefinite representation:*

$$t_k \geq k s_k + \operatorname{tr}(\boldsymbol{Z}_k), \;\; \boldsymbol{Z}_k + s_k \mathbb{I} \succeq \boldsymbol{X}, \boldsymbol{Z}_k \succeq \boldsymbol{0}.$$

Based on this result, we can relax the assumption that the penalty term $\Omega(\boldsymbol{X})$ corresponds to the trace of an operator function. Instead, we can assume:

Assumption 6.4. *$\Omega(\boldsymbol{X}) = \sum_{i \in [n]} p_i \lambda_i^{\downarrow}(f_\omega(\boldsymbol{X}))$, where $p_1 \geq \ldots \geq p_n \geq 0$ and where ω is a function satisfying Assumption 6.1 and whose associated operator function, f_ω, is matrix convex.*

This assumption is particularly suitable for Markov Chain problems [see, e.g., 54, Chapter 4.6], where we are interested in controlling the behaviour of the largest eigenvalue (which always equals 1) plus the second largest eigenvalue of a matrix. However, it might appear to be challenging to model, since, e.g., $\lambda_2^{\downarrow}(\boldsymbol{X})$ is a non-convex function. By applying a telescoping sum argument reminiscent of the one in

[17, Prop. 4.2.1], namely

$$\Omega(\boldsymbol{X}) = \sum_{i=1}^{n} p_i \lambda_i^{\downarrow}(f(\boldsymbol{X})) = \sum_{i=1}^{n} (p_i - p_{i+1}) S_i(f(\boldsymbol{X}))$$

with the convention $p_{n+1} = 0$, Lemma 6.5 allows us to rewrite low-rank problems where $\Omega(\boldsymbol{X})$ satisfies Assumption 6.4 in the form:

$$\begin{aligned} \min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\substack{\boldsymbol{X} \in \mathcal{S}_+^n, \\ \boldsymbol{Z}_i \in \mathcal{S}_+^n, s_i, t_i \in \mathbb{R}_+ \ \forall i \in [n]}} & \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \mu \cdot \operatorname{tr}(\boldsymbol{Y}) + \sum_{i=1}^{n} (p_i - p_{i+1}) t_i \\ \text{s.t.} \quad & \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle = b_i \ \forall i \in [m], \ \boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}, \ \boldsymbol{X} \in \mathcal{K}, \\ & t_i \geq i s_i + \operatorname{tr}(\boldsymbol{Z}_i), \ \boldsymbol{Z}_i + s_i \mathbb{I} \succeq f(\boldsymbol{X}), \boldsymbol{Z}_i \succeq \boldsymbol{0} \ \forall i \in [n], \end{aligned} \tag{6.42}$$

where t_i models the sum of the i largest eigenvalues of $f(\boldsymbol{X})$. Applying the MPRT then yields the following extension to Theorem 6.1:

Proposition 6.6. *Suppose Problem* (6.42) *attains a finite optimal value. Then, the following problem attains the same value:*

$$\begin{aligned} \min_{\boldsymbol{Y} \in \mathcal{Y}_n^k} \min_{\substack{\boldsymbol{X} \in \mathcal{S}_+^n, \\ \boldsymbol{Z}_i \in \mathcal{S}_+^n, s_i, t_i \in \mathbb{R}_+ \ \forall i \in [n]}} & \langle \boldsymbol{C}, \boldsymbol{X} \rangle + \mu \cdot \operatorname{tr}(\boldsymbol{Y}) + \sum_{i=1}^{n} (p_i - p_{i+1}) t_i \\ s.t. \quad & \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle = b_i \ \forall i \in [m], \ \boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y}^{-\frac{1}{2}} \in \mathcal{K}, \\ & t_i \geq i s_i + i - \operatorname{tr}(\boldsymbol{Y}) + \operatorname{tr}(\boldsymbol{Z}_i) \ \forall i \in [n], \\ & \boldsymbol{Z}_i + s_i \mathbb{I} \succeq g_f(\boldsymbol{X}, \boldsymbol{Y}) + \omega(0)(\mathbb{I} - \boldsymbol{Y}), \boldsymbol{Z}_i \succeq \boldsymbol{0} \ \forall i \in [n]. \end{aligned} \tag{6.43}$$

The proof of this reformulation is almost identical to the proof of Theorem 6.1, after observing that (6.20) holds not only for the traces but for the matrices directly, i.e., if $\boldsymbol{X}$ and $\boldsymbol{Y} \in \mathcal{Y}_n^k$ commute, we have

$$f(\boldsymbol{X}) = g_f(\boldsymbol{X}, \boldsymbol{Y}) + \omega(0)(\mathbb{I} - \boldsymbol{Y}).$$

Problem (6.43) involves n times as many variables as (6.18) and therefore supplies

substantially less tractable relaxations. Nonetheless, it could be useful in specific instances. In the aforementioned Markov Chain mixing problem, $p_i - p_{i+1} = 0 \; \forall i \geq k$ with $k = 2$, so we omit the variables which model the eigenvalues larger than 2 .

6.11 Appendix: Extension to the Rectangular Case

In this section, we extend the MPRT to the case where $\boldsymbol{X}$ is a generic $n \times m$ matrix and $f(\boldsymbol{X})$ is the convex quadratic penalty $f(\boldsymbol{X}) = \boldsymbol{X}^\top \boldsymbol{X}$. In this case, $\operatorname{tr}(f(\boldsymbol{X})) = \|\boldsymbol{X}\|_F^2$ is the squared Frobenius norm of $\boldsymbol{X}$.

First, observe that $f : \mathbb{R}^{n \times m} \to \mathcal{S}_+^m$. Alternatively, one could have considered $g(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{X}^\top \in \mathcal{S}_+^n$ and obtain the same penalty, i.e., $\operatorname{tr}(f(\boldsymbol{X})) = \operatorname{tr}(g(\boldsymbol{X}))$. In other words, one can arbitrarily choose whether f preserves the row or the column space of $\boldsymbol{X}$. By the Schur complement lemma, the epigraph is semidefinite representable via

$$\operatorname{epi}(f) := \left\{ (\boldsymbol{X}, \boldsymbol{\theta}) \in \mathbb{R}^{n \times m} \times \mathcal{S}_+^m \; : \; \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X}^\top \\ \boldsymbol{X} & \mathbb{I} \end{pmatrix} \succeq \boldsymbol{0} \right\},$$

so f is matrix convex.

In the symmetric case, we considered the matrix perspective of f at $(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{Y} \succeq \boldsymbol{0}$ is a matrix controlling the range of $\boldsymbol{X}$. When $\boldsymbol{X}$ is no longer symmetric, it is natural to consider a matrix perspective which involves two projection matrices, one of which models the row space and one which models the column space, as proposed in our prior work [34]. More precisely, for $\boldsymbol{Y}, \boldsymbol{Z} \succ \boldsymbol{0}$ we define a perspective of f as

$$g_f(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \boldsymbol{Z}^{\frac{1}{2}} f(\boldsymbol{Y}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Z}^{-\frac{1}{2}}) \boldsymbol{Z}^{\frac{1}{2}}. \tag{6.44}$$

For $f(\boldsymbol{X}) = \boldsymbol{X}^\top \boldsymbol{X}$, this function actually does not depend on $\boldsymbol{Z}$. Hence, we consider

$$\tilde{g}_f(\boldsymbol{X}, \boldsymbol{Y}) = g_f(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \boldsymbol{X}^\top \boldsymbol{Y}^{-1} \boldsymbol{X}.$$

Extending this function to positive semidefinite $\boldsymbol{Y}$ using the same proof technique as

in Proposition 6.3, we then obtain

$$\tilde{g}_f(\boldsymbol{X}, \boldsymbol{Y}) = \begin{cases} \boldsymbol{X}^\top \boldsymbol{Y}^\dagger \boldsymbol{X} & \text{if } \boldsymbol{Y} \succeq \boldsymbol{0},\ \mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Span}(\boldsymbol{Y}), \\ \infty & \text{otherwise.} \end{cases}$$

Proof. Fix $\boldsymbol{X} \in \mathcal{S}^n$ and $\boldsymbol{Y} \succeq \boldsymbol{0}$. As in the proof of Proposition 6.3 denote $\boldsymbol{P}$ the orthogonal projection onto the kernel of $\boldsymbol{Y}$, and define $\boldsymbol{Y}_\varepsilon := \boldsymbol{Y} + \varepsilon \boldsymbol{P}$ for $\varepsilon > 0$. Hence,

$$\boldsymbol{X}^\top \boldsymbol{Y}_\varepsilon^{-1} \boldsymbol{X} = \boldsymbol{X}^\top \boldsymbol{Y}^\dagger \boldsymbol{X} + \varepsilon^{-1} \boldsymbol{X}^\top \boldsymbol{P} \boldsymbol{X}.$$

The right-hand side admits a finite limit if and only if

$$\boldsymbol{X}^\top \boldsymbol{P} \boldsymbol{X} = \boldsymbol{0} \iff \mathrm{Span}(\boldsymbol{X}) \subseteq \mathrm{Ker}(\boldsymbol{P}) = \mathrm{Span}(\boldsymbol{Y}). \qquad \square$$

Furthermore, using the Schur complement lemma as in [34], one can show that $\tilde{g}_f$ is SDP-representable:

$$\mathrm{epi}(\tilde{g}_f) = \left\{ (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) \in \mathbb{R}^{n \times m} \times \mathcal{S}_+^n \times \mathcal{S}^m \ : \ \begin{pmatrix} \boldsymbol{\theta} & \boldsymbol{X}^\top \\ \boldsymbol{X} & \boldsymbol{Y} \end{pmatrix} \succeq \boldsymbol{0} \right\},$$

and hence matrix convex.

Finally, we can easily check that Theorem 6.1 still holds in the symmetric case because (6.20) –which simplifies to $\mathrm{tr}(f(\boldsymbol{X})) = \mathrm{tr}(\tilde{g}_f(\boldsymbol{X}))$ in this case– holds for any $\boldsymbol{Y} \in \mathcal{Y}_n^k$ such that $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$.

Chapter 7

Conclusion and Extensions

In this thesis, we adopted a different perspective on logical and rank constraints, by treating both constraints as purely algebraic ones. Namely, logical constraints are nonlinear constraints of the form $x = z \circ x$ for x continuous and z binary, while rank constraints, $\text{Rank}(\boldsymbol{X}) \leq k$, are a nonlinear constraint of the form $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{X}$ intersected with a linear constraint $\text{tr}(\boldsymbol{Y}) \leq k$ for an orthogonal projection matrix $\boldsymbol{Y}$. By doing so, we built a bridge between mixed-integer and low-rank optimization, and demonstrated that although both types of constraints are typically addressed by different research communities using different algorithms, they are actually two different facets of the same unified story. Moreover, we demonstrated that algorithms which have been used for mixed-integer optimization problems for nigh on 50 years can actually be used to solve low-rank problems more accurately and faster than via state-of-the-art heuristic methods.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] T. Achterberg and E. Towle. Gurobi webinar: Non-convex quadratic optimization. https://www.gurobi.com/resource/non-convex-quadratic-optimization/, 2020. Accessed: 2021-02-09.

[2] A. A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

[3] A. A. Ahmadi, S. Dash, and G. Hall. Optimization over structured subsets of positive semidefinite matrices via column generation. *Discrete Optimization*, 24:129–151, 2017.

[4] M. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Operations Research Letters*, 37(3):187–191, 2009.

[5] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1): 13–51, 1995.

[6] A. A. Amini and M. J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2454–2458. IEEE, 2008.

[7] A. Atamtürk and A. Gomez. Rank-one convexification for sparse regression. *arXiv preprint arXiv:1901.10334*, 2019.

[8] C. Audet, P. Hansen, B. Jaumard, and G. Savard. A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 87(1):131–152, 2000.

[9] L. Bai, J. E. Mitchell, and J.-S. Pang. On conic QPCCs, conic QCQPs and completely positive programs. *Mathematical Programming*, 159(1):109–136, 2016.

[10] B. Barak, J. A. Kelner, and D. Steurer. Rounding sum-of-squares relaxations. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 31–40, 2014.

[11] G. Barker and D. Carlson. Cones of diagonally dominant matrices. *Pacific Journal of Mathematics*, 57(1):15–32, 1975.

[12] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[13] J. E. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[14] N. Beaumont. An algorithm for disjunctive programs. *European Journal of Operational Research*, 48(3):362–371, 1990.

[15] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[16] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1, 2013.

[17] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization*, volume 2. SIAM, 2001.

[18] A. Ben-Tal and A. Nemirovski. On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty. *SIAM Journal on Optimization*, 12(3):811–833, 2002.

[19] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[20] L. Berk and D. Bertsimas. Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation*, 11:381–420, 2019.

[21] Q. Berthet and P. Rigollet. Optimal detection of sparse principal components in high dimension. *Annals of Statistics*, 41(4):1780–1815, 2013.

[22] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, 3rd edition, 2016.

[23] D. Bertsimas and R. Cory-Wright. On polyhedral and second-order cone decompositions of semidefinite optimization problems. *Operations Research Letters*, 48(1):78–85, 2020.

[24] D. Bertsimas and R. Cory-Wright. A scalable algorithm for sparse portfolio selection. *INFORMS Journal on Computing, Articles in Advance*, 2022.

[25] D. Bertsimas and R. Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization & Applications*, 43(1):1–22, 2009.

[26] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[27] D. Bertsimas and B. van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *Annals of Statistics*, 48(1):300–323, 2020.

[28] D. Bertsimas and R. Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.

[29] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. In *Handbook of Combinatorial Optimization*, pages 1473–1491. Springer, 1998.

[30] D. Bertsimas, C. Darnell, and R. Soucy. Portfolio construction through mixed-integer programming at Grantham, Mayo, van Otterloo and company. *Interfaces*, 29(1):49–66, 1999.

[31] D. Bertsimas, M. S. Copenhaver, and R. Mazumder. Certifiably optimal low rank factor analysis. *Journal of Machine Learning Research*, 18(1):907–959, 2017.

[32] D. Bertsimas, J. Pauphilet, and B. Van Parys. Sparse regression: Scalable algorithms and empirical performance. *Statistical Science*, 35(4):555–578, 2020.

[33] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization*, 31(3):2340–2367, 2021.

[34] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. Mixed-projection conic optimization: A new paradigm for modeling rank constraints. *Operations Research, articles in advance*, 2021.

[35] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. A new perspective on low-rank optimization. *arXiv preprint arXiv:2105.05947*, 2021.

[36] D. Bertsimas, J. Pauphilet, and B. Van Parys. Sparse classification: a scalable discrete optimization perspective. *Machine Learning*, 110(11):3177–3209, 2021.

[37] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. Solving large-scale sparse PCA to certifiable (near) optimality. *Journal of Machine Learning Research*, 23(13):1–35, 2022.

[38] D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *Annals of Statistics*, pages 813–852, 2016.

[39] R. Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[40] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. *arXiv preprint arXiv:1605.07221*, 2016.

[41] S. Bi, S. Pan, and D. Sun. A multi-stage convex relaxation approach to noisy structured low-rank matrix recovery. *Mathematical Programming Computation*, 12(4):569–602, 2020.

[42] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140, 1996.

[43] D. Bienstock. Eigenvalue techniques for proving bounds for convex objective, nonconvex programs. *Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, 6080:29–42, 2010.

[44] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.

[45] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

[46] G. Blekherman, P. A. Parrilo, and R. R. Thomas. *Semidefinite optimization and convex algebraic geometry*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012.

[47] E. G. Boman, D. Chen, O. Parekh, and S. Toledo. On factor width and symmetric h-matrices. *Linear Algebra and its Applications*, 405:239–248, 2005.

[48] P. Bonami and M. A. Lejeune. An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations Research*, 57(3):650–670, 2009.

[49] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

[50] B. Borchers and J. E. Mitchell. A computational comparison of branch and bound and outer approximation algorithms for 0–1 mixed integer nonlinear programs. *Computers & Operations Research*, 24(8):699–701, 1997.

[51] M. S. Bostanabad, J. Gouveia, and T. K. Pong. Inner approximating the completely positive cone via the cone of scaled diagonally dominant matrices. *arXiv preprint arXiv:1807.00379*, 2018.

[52] N. Boumal, V. Voroninski, and A. Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

[53] N. Boumal, V. Voroninski, and A. S. Bandeira. Deterministic guarantees for burer-monteiro factorizations of smooth semidefinite programs. *Communications on Pure and Applied Mathematics*, 73(3):581–608, 2020.

[54] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[55] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM Philadelphia, PA, 1994.

[56] A. Brauer. Limits for the characteristic roots of a matrix iv. *Duke Mathematical Journal*, 19:75–91, 1952.

[57] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009.

[58] S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[59] S. Burer and R. D. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[60] S. Burer, K. M. Anstreicher, and M. Dür. The difference between 5×5 doubly nonnegative and completely positive matrices. *Linear Algebra and its Applications*, 431(9):1539–1552, 2009.

[61] E. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[62] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[63] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.

[64] J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–467, 1988.

[65] E. Carlen. Trace inequalities and quantum entropy: an introductory course. *Entropy and the quantum*, 529:73–140, 2010.

[66] M. Carrasco and N. Noumon. Optimal portfolio selection using regularization. Technical report, University of Montreal, 2011.

[67] S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86(3):595–614, 1999.

[68] F. Cesarone, A. Scozzari, and F. Tardella. Efficient algorithms for mean-variance portfolio optimization with hard real-world constraints. *Giornale*

dell'Istituto Italiano degli Attuari, 72:37–56, 2009.

[69] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[70] R. Chares. *Cones and interior-point algorithms for structured convex optimization involving powers and exponentials*. PhD thesis, UCL-Université Catholique de Louvain, 2009.

[71] A. L. Chistov and D. Y. Grigoriev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *International Symposium on Mathematical Foundations of Computer Science, volume 176 of Lecture Notes in Computer Science*, pages 17–31. Springer Verlag, 1984.

[72] D. Cifuentes. Burer-monteiro guarantees for general semidefinite programs. *arXiv preprint arXiv:1904.07147*, 2019.

[73] C. Coey, M. Lubin, and J. P. Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation*, pages 1–45, 2020.

[74] P. L. Combettes. Perspective functions: Properties, constructions, and examples. *Set-Valued and Variational Analysis*, 26(2):247–264, 2018.

[75] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[76] X. Cui, X. Zheng, S. Zhu, and X. Sun. Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *Journal on Global Optimization*, 56(4):1409–1423, 2013.

[77] B. Dacorogna and P. Maréchal. The role of perspective functions in convexity, polyconvexity, rank-one convexity and separate convexity. *Journal of Convex Analysis*, 15(2):271–284, 2008.

[78] A. d'Aspremont. A semidefinite representation for some minimum cardinality problems. In *42nd IEEE International Conference on Decision and Control*, volume 5, pages 4985–4990. IEEE, 2003.

[79] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Advances in Neural Information Processing Systems*, pages 41–48, 2005.

[80] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49 (3):434–448, 2007.

[81] A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[82] A. d'Aspremont, F. Bach, and L. El Ghaoui. Approximation bounds for sparse principal component analysis. *Mathematical Programming*, 148(1-2):89–110, 2014.

[83] V. DeMiguel, L. Garlappi, and R. Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Rev. Financ. Stud.*, 22(5):1915–1953, 2009.

[84] S. S. Dey, R. Mazumder, and G. Wang. A convex integer programming approach for optimal sparse PCA. *arXiv preprint arXiv:1810.09062*, 2018.

[85] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*, volume 15. Springer, 2009.

[86] C. Ding, D. Sun, and J. Y. Jane. First order optimality conditions for mathematical programs with semidefinite cone complementarity constraints. *Mathematical Programming*, 147(1):539–579, 2014.

[87] H. Dong. Relaxing nonconvex quadratic functions by multiple adaptive diagonal perturbations. *SIAM Journal on Optimization*, 26(3):1962–1985, 2016.

[88] H. Dong and K. Anstreicher. Separating doubly nonnegative and completely positive matrices. *Mathematical Programming*, 137(1-2):131–153, 2013.

[89] H. Dong, K. Chen, and J. Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. *arXiv preprint arXiv:1510.06083*, 2015.

[90] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956.

[91] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[92] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3): 307–339, 1986.

[93] A. Ebadian, I. Nikoufar, and M. E. Gordji. Perspectives of matrix convex functions. *Proceedings of the National Academy of Sciences*, 108(18):7313–7314, 2011.

[94] J. Eckstein and D. P. Bertsekas. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

[95] E. Effros and F. Hansen. Non-commutative perspectives. *Annals of Functional Analysis*, 5(2):74–79, 2014.

[96] E. G. Effros. A matrix convexity approach to some celebrated quantum inequalities. *Proceedings of the National Academy of Sciences*, 106(4):1006–1008, 2009.

[97] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456): 1348–1360, 2001.

[98] J. Faraut and A. Koranyi. Analysis on symmetric cones. *Oxford, UK*, 1994.

[99] V. F. Farias and A. A. Li. Learning preferences with side information. *Management Science*, 65(7):3131–3149, 2019.

[100] H. Fawzi and J. Saunderson. Lieb's concavity theorem, matrix geometric means, and semidefinite optimization. *Linear Algebra and its Applications*, 513:240–263, 2017.

[101] H. Fawzi, J. Saunderson, and P. A. Parrilo. Semidefinite approximations of the matrix logarithm. *Foundations of Computational Mathematics*, 19(2):259–296, 2019.

[102] L. Faybusovich and C. Zhou. Self-concordance and matrix monotonicity with applications to quantum entanglement problems. *Applied Mathematics and Computation*, 375:125071, 2020.

[103] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[104] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 2156–2162. IEEE, 2003.

[105] M. Fischetti, I. Ljubić, and M. Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569, 2016.

[106] M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

[107] R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349, 1994.

[108] R. Fletcher and S. Leyffer. Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM Journal on Optimization*, 8(2):604–616, 1998.

[109] R. Fortet. Applications de l'algebre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4(14):17–26, 1960.

[110] A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.

[111] A. Frangioni and C. Gentile. Solving nonlinear single-unit commitment problems with ramping constraints. *Operations Research*, 54(4):767–775, 2006.

[112] A. Frangioni and C. Gentile. SDP diagonalizations and perspective cuts for a class of nonseparable MIQP. *Operations Research Letters*, 35(2):181–185, 2007.

[113] A. Frangioni and C. Gentile. A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes. *Operations Research Letters*, 37(3):206–210, 2009.

[114] A. Frangioni, F. Furini, and C. Gentile. Approximated perspective relaxations: a project and lift approach. *Computational Optimization and Applications*, 63 (3):705–735, 2016.

[115] A. Frangioni, F. Furini, and C. Gentile. Improving the approximated projected perspective reformulation by dual information. *Operations Research Letters*, 45 (5):519–524, 2017.

[116] A. Frangioni, C. Gentile, and J. Hungerford. Decompositions of semidefinite matrices and the perspective reformulation of nonseparable quadratic programs. *Mathematics of Operations Research*, 45(1):15–33, 2020.

[117] T. Gally and M. E. Pfetsch. Computing restricted isometry constants via mixed-integer semidefinite programming. *preprint, submitted*, 2016.

[118] T. Gally, M. E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods & Software*, 33(3):594–632, 2018.

[119] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[120] J. Gao and D. Li. Optimal cardinality constrained portfolio selection. *Opera-*

tions Research, 61(3):745–761, 2013.

[121] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. *arXiv preprint arXiv:1605.07272*, 2016.

[122] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[123] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[124] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[125] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization & Engineering*, 3(3):227–252, 2002.

[126] O. Günlük and J. Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming*, 124(1-2): 183–205, 2010.

[127] O. Günlük and J. Linderoth. Perspective reformulation and applications. In *Mixed Integer Nonlinear Programming*, pages 61–89. Springer, 2012.

[128] S. Han, A. Gómez, and A. Atamtürk. 2x2 convexifications for convex quadratic optimization with indicator variables. *arXiv preprint arXiv:2004.07448*, 2020.

[129] H. Hazimeh, R. Mazumder, and A. Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, pages 1–42, 2021.

[130] M. Hein and T. Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In *Advances in Neural Information Processing Systems*, pages 847–855, 2010.

[131] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 2013.

[132] K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a Lagrangean heuristic and branch-and-bound. *Operations Research*, 46 (2):247–259, 1998.

[133] K. Holmberg, M. Rönnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113(3):544–559, 1999.

[134] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, New York, 1985.

[135] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[136] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[137] N. L. Jacob. A limited-diversification portfolio selection model for the small investor. *The Journal of Finance*, 29(3):847–856, 1974.

[138] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

[139] J. Jeffers. Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 16(3): 225–236, 1967.

[140] I. T. Jolliffe. Rotation of principal components: choice of normalization constraints. *Journal of Applied Statistics*, 22(1):29–35, 1995.

[141] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the Lasso. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[142] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2008.

[143] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(2), 2010.

[144] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[145] M. Karimi and L. Tunçel. Domain-driven solver (DDS): a MATLAB-based software package for convex optimization problems in domain-driven form. *arXiv preprint arXiv:1908.03075*, 2019.

[146] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[147] S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15(3-4): 201–224, 2001.

[148] E. Klotz and A. M. Newman. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18(1-2):18–32, 2013.

[149] K. Kobayashi and Y. Takano. A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems. *Computational Optimization and Applications*, 75(2):493–513, 2020.

[150] K. Kobayashi, Y. Takano, and K. Nakata. Bilevel cutting-plane algorithm for cardinality-constrained mean-cvar portfolio optimization. *Journal of Global Optimization*, 81(2):493–528, 2021.

[151] B. Kocuk, S. S. Dey, and X. A. Sun. Strong SOCP relaxations for the optimal power flow problem. *Operations Research*, 64(6):1177–1196, 2016.

[152] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[153] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[154] J. Lavaei and S. H. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, 2011.

[155] O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *J. Port. Manag.*, 30(4):110–119, 2004.

[156] J. Lee and B. Zou. Optimal rank-sparsity decomposition. *Journal of Global Optimization*, 60(2):307–315, 2014.

[157] B. Li and M. J. Tsatsomeros. Doubly diagonally dominant matrices. *Linear*

Algebra and Its Applications, 261(1-3):221–235, 1997.

[158] Y. Li and W. Xie. Exact and approximation algorithms for sparse PCA. *arXiv preprint arXiv:2008.12438*, 2020.

[159] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[160] J. Linderoth. A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103(2):251–282, 2005.

[161] M. Lubin, J. P. Vielma, and I. Zadik. Mixed-integer convex representability. *Mathematics of Operations Research, to appear*, 2020.

[162] M. Lubin, I. Zadik, and J. P. Vielma. Mixed-integer convex representability. *Mathematics of Operations Research, to appear*, 2020.

[163] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

[164] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3): 20–34, 2010.

[165] R. Luss and A. d'Aspremont. Clustering and feature selection using sparse principal component analysis. *Optimization & Engineering*, 11(1):145–157, 2010.

[166] R. Luss and M. Teboulle. Conditional gradient algorithms for rank-one matrix approximations with a sparsity constraint. *SIAM Review*, 55(1):65–98, 2013.

[167] M. Magdon-Ismail. NP-hardness and inapproximability of sparse PCA. *Information Processing Letters*, 126:35–38, 2017.

[168] T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3): 464–484, 1981.

[169] A. Majumdar, G. Hall, and A. A. Ahmadi. Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:331–360, 2020.

[170] P. Maréchal. On the convexity of the multiplicative potential and penalty functions and related topics. *Mathematical Programming*, 89(3):505–516, 2001.

[171] P. Maréchal. On a functional operation generating convex functions, part 1: duality. *Journal of Optimization Theory and Applications*, 126(1):175–189, 2005.

[172] P. Maréchal. On a functional operation generating convex functions, part 2: algebraic properties. *Journal of Optimization Theory and Applications*, 126(2): 357–366, 2005.

[173] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[174] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[175] L. Miolane. Phase transitions in spiked matrix estimation: information-theoretic analysis. *arXiv preprint arXiv:1806.04343*, 2018.

[176] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Advances in Neural Information Processing*

Systems, pages 915–922, 2006.

[177] Mosek. The mosek optimization toolbox for matlab manual, 2015.

[178] K. G. Murty and S. N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

[179] A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381–406, 2009.

[180] S. Naldi. Solving rank-constrained semidefinite programs in exact arithmetic. *Journal of Symbolic Computation*, 85:206–223, 2018.

[181] S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, pages 1069–1097, 2011.

[182] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1): 265–294, 1978.

[183] A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Mathematical Programming*, 86(3):463–473, 1999.

[184] L. T. Nguyen, J. Kim, and B. Shim. Low-rank matrix completion: A contemporary survey. *IEEE Access*, 7:94215–94237, 2019.

[185] M. L. Overton and R. S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 13(1): 41–45, 1992.

[186] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Mathematical Programming*, 62(1-3):321–357, 1993.

[187] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1): 60–100, 1991.

[188] N. Papadakos. Practical enhancements to the Magnanti–Wong method. *Operations Research Letters*, 36(4):444–449, 2008.

[189] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[190] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2): 339–358, 1998.

[191] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, 2007.

[192] A. F. Perold. Large-scale portfolio optimization. *Management Science*, 30(10): 1143–1160, 1984.

[193] M. Pilanci, M. J. Wainwright, and L. El Ghaoui. Sparse learning via boolean relaxations. *Mathematical Programming*, 151(1):63–87, 2015.

[194] R. Plemmons and R. Cline. The generalized inverse of a nonnegative matrix. *Proceedings of the American Mathematical Society*, pages 46–50, 1972.

[195] I. Quesada and I. E. Grossmann. An LP/NLP based branch and bound al-

gorithm for convex MINLP optimization problems. *Computers & Chemical Engineering*, 16(10-11):937–947, 1992.

[196] P. Raghavan and C. D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[197] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.

[198] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3): 471–501, 2010.

[199] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. parts i–iii. *Journal of Symbolic Computation*, 13(3):255–352, 1992.

[200] J. Renegar. *A mathematical view of interior-point methods in convex optimization*, volume 3. Society for Industrial and Applied Mathematics, 2001.

[201] M. B. Richman. Rotation of principal components. *Journal of Climatology*, 6 (3):293–335, 1986.

[202] P. Richtárik, M. Jahani, S. D. Ahipaşaoğlu, and M. Takáč. Alternating maximization: unifying framework for 8 sparse pca formulations and efficient parallel codes. *Optimization and Engineering*, 22(3):1493–1519, 2021.

[203] P. Rigollet and J.-C. Hütter. High dimensional statistics. *Lecture notes for course 18S997*, 2015.

[204] R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[205] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[206] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. *arXiv preprint arXiv:1307.4653*, 2013.

[207] J. A. Ryan and J. M. Ulrich. quantmod: Quantitative financial modelling framework. *R package*, 2018.

[208] M. Schaefer. Realizability of graphs and linkages. In J. Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 461–482. Springer New York, New York, NY, 2013.

[209] H. D. Sherali and B. M. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22(1-4):233–261, 2002.

[210] M. Singh and W. Xie. Approximation algorithms for D-optimal design. *Mathematics of Operations Research*, 45:1193–1620, 2020.

[211] A. Skajaa and Y. Ye. A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. *Mathematical Programming*, 150(2):391–422, 2015.

[212] R. A. Stubbs. *Branch-and-cut methods for mixed 0-1 convex programming*. PhD thesis, Northwestern University, 1996.

[213] R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86(3):515–532, 1999.

[214] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht. Low-

rank solutions of linear matrix equations via procrustes flow. In *International Conference on Machine Learning*, pages 964–973. PMLR, 2016.

[215] M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

[216] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38 (1):49–95, 1996.

[217] J. P. Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015.

[218] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[219] V. Vu and J. Lei. Minimax rates of estimation for sparse PCA in high dimensions. In *Artificial intelligence and statistics*, pages 1278–1286, 2012.

[220] A. L. Wang and F. Kılınç-Karzan. On the tightness of SDP relaxations of QCQPs. *Optimization Online*, 2019.

[221] T. Wang, Q. Berthet, and R. J. Samworth. Statistical and computational trade-offs in estimation of sparse principal components. *Annals of Statistics*, 44(5): 1896–1930, 2016.

[222] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[223] A. Wiegele. Biq mac library—a collection of max-cut and quadratic 0-1 programming instances of medium size. Technical report, Alpen-Adria-Universität Klagenfurt, Austria, 2007.

[224] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[225] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming*. Springer Science & Business Media, 2012.

[226] W. Xie and X. Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30:3359—3386, 2020.

[227] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[228] X. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14:899–925, 2013.

[229] G. Zakeri, D. Craigie, A. Philpott, and M. Todd. Optimization of demand response through peak shaving. *Operations Research Letters*, 42(1):97–101, 2014.

[230] R. Zass and A. Shashua. Non-negative sparse PCA. In *Advances in Neural Information Processing Systems*, pages 1561–1568, 2007.

[231] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[232] H. Zhang, Z. Lin, and C. Zhang. A counterexample for the validity of using nuclear norm as a convex surrogate of rank. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 226–241.

Springer, 2013.

[233] Y. Zhang, A. d'Aspremont, and L. El Ghaoui. Sparse PCA: Convex relaxations, algorithms and applications. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 915–940. Springer, 2012.

[234] Q. Zheng and J. Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. *arXiv preprint arXiv:1506.06081*, 2015.

[235] X. Zheng, X. Sun, and D. Li. Improving the performance of MIQP solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS Journal on Computing*, 26(4): 690–703, 2014.

[236] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.