On Computing Sparse Generalized Inverses and Sparse-Inverse/Low-Rank Decompositions

by

Victor K. Fuentes

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2019

Doctoral Committee:

Professor Jon Lee, Chair
Assistant Professor Laura Balzano
Professor Marina Epelman
Professor Marcia Fampa

Victor K. Fuentes
vicfuen@umich.edu
ORCID iD: 0000-0002-9874-554X

Dedication

I dedicate my dissertation to Betty S. Keesey and Maureen Keesey Fuentes.

Acknowledgments

There are many people that have earned my gratitude and appreciation for their contributions to my time in graduate school. More specifically, I would like to extend thanks to a number of groups of people, without whom this thesis (and graduate experience) would not have been in the slightest bit possible.

First, I am indebted to my advisor, Jon Lee, for providing an experience unlike any else that has changed me for the better. On an academic level, Jon provided an environment with which to explore what it means to be researcher, one which allowed me the opportunity to both develop my understanding of the vastness that is the field of optimization and experience firsthand the difficulties that come with the territory. On a personal level, I found (and continue to find) myself fortunate to have a person such as Jon to serve as inspiration during my time here, providing a sounding board on how to configure (and then re-configure) the delicate balance of being a graduate student, a researcher, and your own person (as well as how those are not necessarily mutually exclusive). Given the holes I dug for myself, and the efforts needed to recover from them, I must attribute much of what I have accomplished and who I have become during my time here to Jon. A variant on a sentence I perhaps write too much in my emails: *As always, thank you for your time and patience Jon, I truly appreciate all that you have done for me.*

Besides my advisor, I would like to thank the rest of my defense committee members (Laura Balzano, Marina Epelman, Marcia Fampa) for agreeing to be a part of this process.

It would be remiss if I were to omit acknowledgement for my colleagues within the IOE department. From the 2014 cohort to my officemates to

the countless number of incredible individuals with whom I have had the honor of crossing paths with in class, in the hallways or the commons, I have been the most fortunate of beneficiaries in experiencing your passion and knowledge firsthand. Thank you all for adding so much to this experience.

Thank you to the University of Michigan, in particular the Rackham Graduate School, for the opportunity to pursue my doctorate through the Rackham Merit Fellowship (RMF). Additional thanks to Jon Lee for providing funding to cover the summers and academic years not covered by RMF.

Last but not least, I would like to express my sincerest gratitude to the following people without whom I would not be where I am today: my mother Maureen Fuentes, my late grandmother Betty Keesey, my brother Roland Fuentes, the late Ken Bystrom (former highschool math teacher), and my former undergraduate instructors/mentors Rick West, Duane Kouba, and Jesús De Loera.

Table of Contents

Dedication ii

Acknowledgments iii

List of Figures viii

List of Tables ix

Abstract x

Chapter 1. Introduction 1
- 1.1 Definitions . . . 1
- 1.2 Chapter overviews . . . 2

Chapter 2. Sparse Pseudoinverses via LP relaxation 4
- 2.1 Pseudoinverses . . . 5
- 2.2 Sparse left and right pseudoinverses . . . 6
- 2.3 Sparse generalized inverses based on the Moore-Penrose properties . . . 8
- 2.4 Computational experiments . . . 9
- 2.5 Conclusions and ongoing work . . . 16

Chapter 3. Computationally dealing with the non-linear property of the Moore-Penrose Pseudoinverse 18
- 3.1 Toward computationally dealing with (P2) . . . 19
- 3.2 A lighter version of (P1) . . . 22
- 3.3 Quadratic tightening of (P1) . . . 23

3.4 Penalizing "big" $\mathcal{H}_{ij}$ with quadratic penalties 24
3.5 Quadratic PSD-cuts 25
3.6 Pre-computed quadratic PSD-cuts 28
3.7 2-by-2 cuts 28
3.8 Reformulation Linearization Technique (RLT) 29
3.9 Non-symmetric lifting 32
3.10 Modeling our problem 37
3.11 Diving heuristic 42
3.11.1 Weighted branching-point analysis and selection . . 45
3.11.2 Impact of diving heuristic on combinations of (P3) and (P4) 53
3.11.3 Changes in solution norm ratios 57
3.12 Conclusions and further questions 61

Chapter 4. Sparse-inverse/low-rank Decomposition via Woodbury 63
4.1 Introduction 63
4.1.1 Convex approximation 65
4.1.2 Generating test problems via the recovery theory . 66
4.2 Sparse-inverse/low-rank decomposition 68
4.2.1 An algorithmic approach via the Woodbury identity 68
4.2.2 Generating test problems without a recovery theory 69
4.3 Computational experiments 70
4.4 Conclusions 73

Chapter 5. Computational Techniques for Sparse-Inverse/Low-Rank Decomposition 74
5.1 SDP Approach 74
5.2 A convex relaxation 76
5.2.1 An SDP relaxation 76
5.2.2 The dual SDP 82
5.2.3 Disjunctive programming 84
5.3 Remarks 92

Appendix 93

Bibliography **98**

List of Figures

2.1 Absolute value function $|x|$, Indicator function for $x \neq 0$ . . . 6
2.2 Average least-squares ratio vs. rank 15
2.3 Average 2-norm ratio vs. rank 16

3.1 Impact on $||H||_1$ using the midpoint of $[\alpha, \beta]$ 47
3.2 Impact on (P2) viol. of $H134$ using the midpoint of $[\alpha, \beta]$ 48
3.3 Diving impact on $||H||_1$ using $t_{MPP} \in [\alpha, \beta]$ 49
3.4 Impact on (P2) viol. of $H134$ using $t_{MPP} \in [\alpha, \beta]$ 50
3.5 Tradeoff between $||H||_1$ and (P2) violation of $H134$ 51
3.6 Comparison of 25/75 weighting using MPP and MID . . . 53
3.7 Obj. val. vs. (P2) violation of $H1$ 55
3.8 Obj. val. vs. (P2) violation of $H13$ 56
3.9 Obj. val. vs. (P2) violation of $H14$ 57
3.10 2-norm solution ratios: $H13$ vs. MPP 60
3.11 Least-squares norm solution ratios: $H14$ vs. MPP 61

4.1 $f(\bar{\tau})$ vs $\bar{\tau}$ $(n = 75)$ 71
4.2 $n = 75,\ k = 15$ 72

List of Tables

2.1 Sparsity vs quality ($m = n = 40$) 13
2.2 Sparsity vs quality ($m = n = 40$) 14

Abstract

Pseudoinverses are ubiquitous tools for handling over- and under-determined systems of equations. For computational efficiency, sparse pseudoinverses are desirable. Recently, sparse left and right pseudoinverses were introduced, using 1-norm minimization and linear programming. We introduce several new sparse generalized inverses by using 1-norm minimization on a subset of the linear Moore-Penrose properties, again leading to linear programming. Computationally, we demonstrate the usefulness of our approach in the context of application to least-squares problems and minimum 2-norm problems.

One of the Moore-Penrose properties is nonlinear (in fact, quadratic), and so developing an effective convex relaxation for it is nontrivial. We develop a variety of methods for this, in particular a nonsymmetric lifting which is more efficient than the usual symmetric lifting that is normally applied to non-convex quadratic equations. In this context, we develop a novel and computationally effective "diving procedure" to find a path of solutions trading off sparsity against the nice properties of the Moore-Penrose pseudoinverse.

Next, we consider the well-known low-rank/sparse decomposition problem

$$\min \left\{ \bar{\tau} \|A\|_0 + (1 - \bar{\tau}) \mathrm{rank}(B) \ : \ A + B = \bar{C} \right\},$$

where $\bar{C}$ is an $m \times n$ input matrix, $0 < \bar{\tau} < 1$, $\|\cdot\|_0$ counts the number of nonzeros, and A and B are matrix variables. This is a central problem in the area of statistical model selection, where the sparse matrix can correspond to a Gaussian graphical model, and the low-rank matrix can capture the effect of latent, unobserved random variables. There is a well-known recovery theory for this problem, based on a well-studied convex relaxation, and we use it to devise test instances for the low-rank/sparse-inverse problem. The low-rank/sparse-inverse decomposition problem can be related to that of identifying a sparse "precision matrix". We use the Woodbury matrix identity to construct an algorithmic procedure for this problem, based on a procedure used in the ordinary low-rank/sparse decomposition setting. This gives us a computationally effective method for generating test instances for this type of problem, without a supporting recovery theory.

Finally, we present an SDP formulation of the low-rank/sparse-inverse decomposition problem. We further consider a relaxation to deal with the nonconvex (quadratic) constraints, describing our extension to the original formulation, providing primal and dual SDP relaxations, and examine some relevant properties. In dealing with the nonconvex constraint, we propose the construction of disjunctive cuts, describing how these cuts can be generated as well as considerations for tuning them.

Chapter 1

Introduction

1.1 Definitions

We briefly set some notation.

- I_n (respectively $\mathbf{0}_n$) denotes an $n \times n$ identity (all zero) matrix.
- $\vec{0}_n \in \mathbb{R}^n$ denotes a zero vector.
- e denotes an all-one vector.
- For an $M \in \mathbb{R}^{m \times n}$, $\|M\|_0$ denotes the number of nonzeros in M,
- $\text{rank}(M)$ denotes the rank of M,
- $\|M\|_1 := \sum_{i,j} |m_{ij}|$ denotes the matrix 1-norm of M, and
- $\|M\|_* := \sum_{i=1}^{\min\{m,n\}} \sigma_i(M)$ denotes the nuclear norm of M, where $\sigma_1(M) \geq \sigma_2(M) \geq ... \geq \sigma_{\min\{m,n\}}(M) \geq 0$ are the singular values of M.
- If M is square, then we denote the trace of M by $\text{tr}(M) := \sum_{i=1}^{m} m_{ii}$.

- For M, $N \in \mathbb{R}^{m\times n}$, the matrix inner product is $\langle M, N\rangle := \text{tr}(M'N) = \sum_{i,j} m_{ij}n_{ij}$.

1.2 Chapter overviews

In Chapter 2 we define the notion of pseudoinverse and generalized inverse (§2.1), as well as establish motivation for sparsity in this context (§2.2). In §2.3 we delve into developing tractable sparse generalized inverses based on different characterizations of the properties of the Moore-Penrose pseudoinverse (MPP), while also proving some useful results regarding these solutions under different column/row rank and Moore-Penrose (M-P) property assumptions. In §2.4 computational experiments are designed and run to explore inducing sparsity in solutions that satisfy various combinations of the linear M-P properties, along with some relational metrics to compare the quality of the solution with that of the true MPP.

In Chapter 3 we propose a collection of different methods to computationally deal with the nonlinear (nonconvex) M-P property (P2). §3.3 - §3.7 cover some of the ideas considered, although they were not explored as extensively as the techniques in later sections. Of the many methods we describe, we focus much of our energy in §3.8 expanding on the use of Reformulation Linearization Technique (RLT) and in §3.9 applying nonsymmetric lifting equations/inequalities to approximate (P2). We provide an overview of how to model our problem in §3.10. In our exploration of the efficacy of these techniques, we examine in §3.11 the tradeoff between generating a sparse pseudoinverse (or generalized inverse) and a solution that satisfies (P2) via a diving heuristic that iteratively strengthens our relaxation by refining the bounds of our variables. We describe the premise and setup of this diving heuristic, as well as explore computationally the

impact of different weighted combinations in the branching point selection process.

In Chapter 4 we shift from the discussion of sparse generalized inverses, where in §4.1 we begin by considering a variant of the traditional sparse/low-rank matrix decomposition problem, focusing on sparse-inverse/low-rank decomposition. In §4.1.1 we discuss how to generate test problems for the traditional decomposition problem using the recovery theory dealing with convex relaxations, and describe the difficulties in describing a similar recovery theory in the sparse-inverse/low-rank case, which results in a nonconvex relaxation. In §4.2 we utilize the Woodbury matrix identity to establish a correspondence between the convex and non-convex relaxations and introduce an algorithmic approach to generating test problems without an explicit recovery theory. In §4.3 we further provide some computational results for our heuristic and comment on its efficacy in generating test problems that exhibit "significant" recovery in §4.4.

In Chapter 5 (§5.1) we present a modeling approach for the sparse-inverse/low-rank decomposition problem, as well provide an in-depth description of a possible SDP relaxation. In §5.2, with the hopes of dealing with the non-convexity that arises from our relaxation and subsequent reformulation, we propose a means to construct a disjunctive cutting plane that tightens our original relaxation and describe how to generate valid linear inequalities that iteratively strengthen the relaxation.

Chapter 2

Sparse Pseudoinverses via LP relaxation

This chapter is based on [FFL16b]

Introduction

Pseudoinverses are a central tool in matrix algebra and its applications. Sparse optimization is concerned with finding sparse solutions of optimization problems, often for computational efficiency in the use of the output of the optimization. There is usually a tradeoff between an ideal dense solution and a less-ideal sparse solution, and sparse optimization is often focused on tractable methods for striking a good balance. Recently, sparse optimization has been used to calculate tractable sparse left and right pseudoinverses, via linear programming. We extend this theme to derive several other tractable sparse pseudoinverses, employing linear and convex relaxations.

In §2.1, we give a very brief overview of pseudoinverses, and in §2.2, we

describe some prior work on sparse left and right pseudoinverses. In §2.3, we present new sparse pseudoinverses based on tractable convex relaxations of the Moore-Penrose properties. In §2.4, we present preliminary computational results. Finally, in §2.5, we make brief conclusions and describe our ongoing work.

2.1 Pseudoinverses

When a real matrix $A \in \mathbb{R}^{m \times n}$ is not square or not invertible, we consider pseudoinverses of A (see [RM71] for a wealth of information on this topic). For example, there is the well-known *Drazin inverse* for square and even non-square matrices (see [CG80]) and the *generalized Bott-Duffin inverse* (see [Che90]).

The most well-known pseudoinverse of all is the *Moore-Penrose (M-P) pseudoinverse*, independently discovered by E.H. Moore and R. Penrose. If $A = U\Sigma V'$ is the real singular value decomposition of A (see [GV96], for example), then the M-P pseudoinverse (MPP) of A can be defined as $A^+ := V\Sigma^+U'$, where Σ^+ has the shape of the transpose of the diagonal matrix Σ, and is derived from Σ by taking reciprocals of the non-zero (diagonal) elements of Σ (i.e., the non-zero singular values of A). The M-P pseudoinverse is calculated, via its connection with the real singular value decomposition, by the `Matlab` function `pinv`.

The MPP, a central object in matrix theory, has many concrete uses. For example, we can use it to solve least-squares and minimum 2-norm problems, as well as, together with a norm, to define condition numbers of matrices.

2.2 Sparse left and right pseudoinverses

It is well known that in the context of seeking a sparse solution in a convex set, a surrogate for minimizing the sparsity is to minimize the 1-norm. In fact, if the components of the solution have absolute value no more than unity, a minimum 1-norm solution has 1-norm no greater than the number of nonzeros in the sparsest solution. This is due to the fact that the absolute value function is an underestimator of the indicatior function for $x \neq 0$ on the domain $[-1, 1]$.

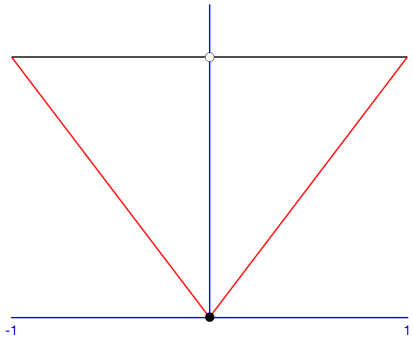


Figure 2.1: Absolute value function $|x|$, Indicator function for $x \neq 0$

With this in mind, [DKV13] defines sparse left and right pseudoinverses in a natural and tractable manner (also see [DG17]). For an "overdetermined case", a *sparse left pseudoinverse* can be defined via the convex formulation

$$\min \{\|H\|_1 \ : \ HA = I_n\}\,. \tag{$\mathcal{O}$}$$

For an "underdetermined case", a *sparse right pseudoinverse* can be defined

via the convex formulation

$$\min \{\|H\|_1 \ : \ AH = I_m\} \,. \tag{$\mathcal{U}$}$$

These definitions emphasize sparsity, while in some sense putting a rather mild emphasis on the aspect of being a pseudoinverse. We do note that if the columns of A are linearly independent, then the M-P pseudoinverse is precisely $(A'A)^{-1}A'$, which is a left inverse of A. Therefore, if A has full column rank, then the MPP is a feasible H for $(\mathcal{O})$. Conversely, if A does not have full column rank, then $(\mathcal{O})$ has no feasible solution, and so there is no sparse left inverse in such a case. On the other hand, if the rows of A are linearly independent, then the M-P pseudoinverse is precisely $A'(AA')^{-1}$, which is a right inverse of A. Therefore, if A has full row rank, then the MPP is a feasible H for $(\mathcal{U})$. Conversely, if A does not have full row rank, then $(\mathcal{U})$ has no feasible solution, and so there is no sparse right inverse in such a case.

These sparse pseudoinverses are easy to calculate, by linear programming:

$$\min \left\{ \sum_{ij \in m \times n} \ t_{ij} \ : \ t_{ij} \geq h_{ij}, \ t_{ij} \geq -h_{ij}, \ \forall ij \in m \times n; \ HA = I_n \right\} \tag{$LP_{\mathcal{O}}$}$$

for the sparse left pseudoinverse, and

$$\min \left\{ \sum_{ij \in m \times n} \ t_{ij} \ : \ t_{ij} \geq h_{ij}, \ t_{ij} \geq -h_{ij}, \ \forall ij \in m \times n; \ AH = I_m \right\} \tag{$LP_{\mathcal{U}}$}$$

for the sparse right pseudoinverse. In fact, the ($LP_{\mathcal{O}}$) decomposes row-wise for H, and ($LP_{\mathcal{U}}$) decomposes column-wise for H, so calculating these sparse pseudoinverses can be made very efficient at large scale. These sparse pseudoinverses also have nice mathematical properties (see [DKV13], [DG17]).

2.3 Sparse generalized inverses based on the Moore-Penrose properties

We seek to define different tractable sparse pseudoinverses, based on the the following nice characterization of the MPP.

Theorem 1. *For $A \in R^{m\times n}$, the MPP A^+ is the unique $H \in \mathbb{R}^{n\times m}$ satisfying:*

$$AHA = A \tag{P1}$$
$$HAH = H \tag{P2}$$
$$(AH)' = AH \tag{P3}$$
$$(HA)' = HA \tag{P4}$$

If we consider properties (P1) - (P4) that characterize the M-P pseudoinverse, we can observe that properties (P1), (P3) and (P4) are all linear in H, and the only non-linearity is property (P2), which is quadratic in H. Another important point to observe is that without property (P1), H could be the all-zero matrix and satisfy properties (P2), (P3) and (P4). Whenever property (P1) holds, H is called a *generalized inverse*. So, in the simplest approach, we can consider minimizing $\|H\|_1$ subject to property (P1) and *any subset* of the properties (P3) and (P4). In this manner, we get several (four) new sparse generalized inverses which can all be calculated by linear programming.

We made some tests of our ideas, using CVX/Matlab (see [GB15], [GB08]). Before describing our experimental setup, we observe the following results.

Proposition 2. *Given $A \in \mathbb{R}^{m\times n}$ and generalized inverse $H \in \mathbb{R}^{n\times m}$:*

(i) *If A has full column rank and H satisfies (P1), then H is a left inverse of A, and H satisfies (P2) and (P4).*

(ii) *If A has full row rank and H satisfies (P1), then H is a right inverse of A, and H satisfies (P2) and (P3).*

Proof. Suppose that A has full column rank and H satisfies (P1) and (P3).

(i) Because A has full column rank, via elementary row operations, we can reduce P1 to

$$\begin{bmatrix} I \\ M \end{bmatrix} HA = \begin{bmatrix} I \\ M \end{bmatrix}.$$

This implies that $HA = I$, that is H is a left inverse of A. Multiplying on the right by H, we immediately have $HAH = H$, that is (P2). Additionally, $HA = I$ immediately implies (P4).

(ii) The proof is similar.

□

Corollary 3. *If A has full column rank and H satisfies (P1) and (P3), then $H = A^+$. If A has full row rank and H satisfies (P1) and (P4), then $H = A^+$.*

2.4 Computational experiments

Due to the results in §2.3, we decided to focus our experiments on matrices A with rank less than $\min\{m, n\}$, testing some of our ideas using CVX/Matlab (see [GB15], [GB08]). We generated random dense $n \times n$ rank-r matrices A of the form $A = UV$, where each U and V' are $n \times r$, with $n = 40$,

and five instance for each $r = 4, 8, 16, \ldots, 36$. The entries in U and V were iid uniform $(-1, 1)$. We construct our problem instances in this manner so that we have a structured way to control the rank, as well as the magnitude of the entries of the matrices A and its pseudoinverse A^+. We then scaled each A by a multiplicative factor of 0.01, which had the effect of making $A+$ fully dense to an entry-wise zero-tolerance of 0.1. In computing various sparse generalized inverses, we used a zero-tolerance of 10^{-5}. We measured sparsity of a sparse pseudoinverse as the number of its nonzero components divided by n^2. We measured quality of a sparse generalized inverse H, relative to the MPP A^+ in two ways:

- *least-squares ratio* ('lsr'): $\|AHb - b\|_2 / \|AA^+b - b\|_2$, with arbitrarily $b := \vec{1}_m$. (Note that $x := A^+b$ always minimizes $\|Ax - b\|_2$.)
- 2*-norm ratio* ('2nr'): $\|HA\vec{1}_n\|_2 / \|A^+A\vec{1}_n\|_2$. (Note that $x := HA\vec{1}_n$ is always a solution to $Ax = A\vec{1}_n$, whenever H satisfies (P1), and one that minimizes $\|x\|_2$ is given by $x := A^+A\vec{1}_n$.)

Proposition 4. *If H satisfies (P1) and (P3), then $AH = AA^+$.*

Proof.

$$\begin{aligned} AHA &= AA^+A \qquad \text{(by (P1))} \\ H'A'A &= (A^+)'A'A \qquad \text{(by (P3))} \\ A'AH &= A'AA^+ \\ (A^+)'A'AH &= (A^+)'A'AA^+ \\ AH &= AA^+, \end{aligned}$$

the last equation following directly from a well-known property of A^+. □

Corollary 5. *If H satisfies (P1) and (P3), then $x := Hb$ (and of course A^+b) solves* $\min\{\|Ax - b\|_2 \ : \ x \in \mathbb{R}^n\}$.

Similarly, we have the following two results:

Proposition 6. *If H satisfies (P1) and (P4), then $HA = A^+A$.*

Proof.

$$\begin{aligned} AHA &= AA^+A \qquad \text{(by (P1))} \\ A'H'A &= A'(A^+)'A \qquad \text{(by (P4))} \\ A'HA &= A'A^+A \\ (A^+)'A'HA &= (A^+)'A'A^+A \\ HA &= A^+A, \end{aligned}$$

the last equation following directly from a property of A^+. □

Corollary 7. *If H satisfies (P1) and (P4), and b is in the column space of A, then Hb (and of course A^+b) solves* $\min\{\|x\|_2 \ : \ Ax = b, \ x \in \mathbb{R}^n\}$.

So in the situations covered by Corollaries 5 and 7, we can seek and use sparser pseudoinverses than A^+. Our computational results are summarized in Table 2.1 and Table 2.2, where '1nr' (1-norm ratio) is simply $\|H\|_1/\|A^+\|_1$, and 'sr' (sparsity ratio) is simply $\|H\|_0/\|A^+\|_0$. Note that the entries of 1 reflect the results above; in particular the results of Corollary 5 are reflected in the column 'lsr' (defined earlier)for solutions where (P1) + (P3) are enforced, while the results of Corollary 7 are reflected in the column '2nr' (defined earlier) for solutions where (P1) + (P4). When enforcing (P1) + (P3) + (P4), the results of Corollaries 5 and 7 are reflected by entries of 1 in 'lsr' and '2nr'. We observe that sparsity can be gained versus the MPP,

often with a modest decrease in quality of the generalized inverse, and we can observe some trends as the rank varies.

For sake the of clarity and succinct notation, let us define $H1$, $H13$, $H14$, and $H134$ to denote solutions H that satisfy (P1), (P1) + (P3), (P1) + (P4), and (P1) + P3) + (P4), respectively.

As illustrated in Figure 2.2, as the rank of A increases we see that when enforcing (P1) or (P1) + (P4) there is a noticeable decrease in the quality of H as a least-squares minimizers (illustrated as an increase in the least-squares ratio). However, when enforcing (P1) + (P3) or (P1) + (P3) + (P4), we observe that the sparse solutions H generated are consistent with the MPP as least-squares minimizers (follows from Corollary 5).

Similarly in Figure 2.3, when we enforce (P1) or (P1) + (P3) there is some observable variability in the quality of H as a 2-norm minimizer, resulting in a sparser, but non-optimal solution to the 2-norm problem. Furthermore, when enforcing a combination of either (P1) + (P4) or (P1) + (P3) + (P4), the resulting sparse solution H is consistent with that of the MPP, as they both serve as minimizers of the 2-norm problem (follows from Corollary 7).

Table 2.1: Sparsity vs quality ($m = n = 40$)

		P1				P1+P3				P1+P4				P1+P3+P4			
r	$\|A^+\|_1$	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr
4	586	0.44	0.01	1.07	2.27	0.60	0.10	1	1.43	0.64	0.10	1.02	1	0.75	0.19	1	1
4	465	0.46	0.01	1.07	1.82	0.63	0.10	1	1.43	0.63	0.10	1.01	1	0.77	0.19	1	1
4	500	0.44	0.01	1.08	1.82	0.62	0.10	1	1.46	0.62	0.10	1.01	1	0.76	0.19	1	1
4	503	0.41	0.01	1.28	2.00	0.62	0.10	1	1.31	0.62	0.10	1.06	1	0.75	0.19	1	1
4	511	0.45	0.01	1.10	2.36	0.63	0.10	1	1.55	0.64	0.10	1.09	1	0.78	0.19	1	1
8	855	0.53	0.04	1.17	1.63	0.69	0.20	1	1.28	0.68	0.20	1.05	1	0.80	0.36	1	1
8	851	0.53	0.04	1.22	1.60	0.69	0.20	1	1.33	0.69	0.20	1.07	1	0.80	0.36	1	1
8	841	0.53	0.04	1.25	1.70	0.69	0.20	1	1.34	0.69	0.20	1.07	1	0.80	0.36	1	1
8	761	0.52	0.04	1.05	1.70	0.69	0.20	1	1.32	0.68	0.20	1.09	1	0.81	0.36	1	1
8	864	0.52	0.04	1.09	1.40	0.69	0.20	1	1.21	0.68	0.20	1.04	1	0.80	0.36	1	1
12	1150	0.60	0.09	1.26	1.68	0.74	0.30	1	1.26	0.75	0.30	1.12	1	0.86	0.51	1	1
12	1198	0.59	0.09	1.20	1.70	0.75	0.30	1	1.25	0.75	0.30	1.05	1	0.85	0.51	1	1
12	1236	0.59	0.09	1.10	1.28	0.75	0.30	1	1.17	0.75	0.30	1.24	1	0.86	0.51	1	1
12	1134	0.60	0.09	1.38	1.43	0.75	0.30	1	1.19	0.74	0.30	1.09	1	0.85	0.51	1	1
12	1135	0.60	0.09	1.20	1.44	0.75	0.30	1	1.21	0.75	0.30	1.14	1	0.85	0.51	1	1
16	1643	0.67	0.16	1.36	1.85	0.79	0.40	1	1.30	0.80	0.40	1.17	1	0.90	0.64	1	1
16	1421	0.65	0.16	1.20	1.61	0.79	0.40	1	1.29	0.79	0.40	1.31	1	0.90	0.64	1	1
16	1518	0.65	0.16	1.33	1.38	0.79	0.40	1	1.20	0.80	0.40	1.30	1	0.89	0.64	1	1
16	1512	0.66	0.16	1.45	1.68	0.80	0.40	1	1.34	0.79	0.40	1.16	1	0.89	0.64	1	1
16	1539	0.65	0.16	1.18	1.25	0.79	0.40	1	1.19	0.79	0.40	1.29	1	0.89	0.64	1	1
20	2147	0.72	0.25	1.51	1.33	0.84	0.50	1	1.15	0.84	0.50	1.42	1	0.94	0.75	1	1
20	2111	0.72	0.25	1.81	1.44	0.83	0.50	1	1.35	0.84	0.50	1.48	1	0.93	0.75	1	1
20	2148	0.71	0.25	2.08	1.49	0.84	0.50	1	1.32	0.83	0.50	1.45	1	0.93	0.75	1	1
20	2061	0.72	0.25	1.50	1.49	0.84	0.50	1	1.35	0.84	0.50	1.31	1	0.93	0.75	1	1
20	2283	0.72	0.25	1.61	1.47	0.83	0.50	1	1.47	0.84	0.50	1.20	1	0.94	0.75	1	1

'1nr' (1-norm ratio) is $\|H\|_1/\|A^+\|_1$.

'sr' (sparsity ratio) is $\|H\|_0/\|A^+\|_0$.

'lsr' (least-squares ratio) is $\|A(H\vec{1}_m) - \vec{1}_m\|_2/\|A(A^+\vec{1}_m) - \vec{1}_m\|_2$.

'2nr' (2-norm ratio) is $\|HA\vec{1}_n\|_2/\|A^+A\vec{1}_n\|_2$.

Table 2.2: Sparsity vs quality ($m = n = 40$)

		P1				P1+P3				P1+P4				P1+P3+P4			
r	$\|A^+\|_1$	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr	1nr	sr	lsr	2nr
24	2865	0.77	0.36	1.86	1.24	0.87	0.60	1	1.18	0.87	0.60	1.51	1	0.96	0.84	1	1
24	3228	0.78	0.36	2.17	1.34	0.87	0.60	1	1.37	0.88	0.60	1.90	1	0.96	0.84	1	1
24	2884	0.77	0.36	2.27	1.72	0.87	0.60	1	1.32	0.87	0.60	1.55	1	0.96	0.84	1	1
24	2853	0.78	0.36	1.50	1.66	0.88	0.60	1	1.50	0.87	0.60	1.53	1	0.96	0.84	1	1
24	2944	0.78	0.36	1.72	1.48	0.87	0.60	1	1.64	0.88	0.60	1.64	1	0.96	0.84	1	1
28	4359	0.82	0.49	1.69	1.65	0.90	0.70	1	1.63	0.91	0.70	1.89	1	0.98	0.91	1	1
28	4268	0.83	0.49	2.27	1.98	0.91	0.70	1	1.79	0.91	0.70	2.08	1	0.98	0.91	1	1
28	4069	0.83	0.49	2.35	1.51	0.91	0.70	1	1.43	0.91	0.70	2.25	1	0.98	0.91	1	1
28	3993	0.83	0.49	2.30	1.58	0.90	0.70	1	1.27	0.91	0.70	2.19	1	0.97	0.91	1	1
28	4387	0.83	0.49	2.54	1.78	0.91	0.70	1	1.34	0.91	0.70	2.76	1	0.98	0.91	1	1
32	6988	0.88	0.64	4.08	1.60	0.94	0.80	1	1.81	0.94	0.80	3.54	1	0.99	0.96	1	1
32	6493	0.89	0.64	3.00	1.75	0.94	0.80	1	1.79	0.94	0.80	2.35	1	0.99	0.96	1	1
32	11445	0.89	0.64	4.50	4.82	0.94	0.80	1	2.58	0.94	0.80	7.18	1	0.99	0.96	1	1
32	8279	0.89	0.64	5.08	2.72	0.95	0.80	1	2.31	0.94	0.80	3.39	1	0.99	0.96	1	1
32	5069	0.89	0.64	2.14	1.90	0.95	0.80	1	1.74	0.94	0.80	2.26	1	0.99	0.96	1	1
36	18532	0.94	0.81	11.16	2.88	0.97	0.90	1	1.85	0.97	0.90	9.80	1	1.00	0.99	1	1
36	16646	0.94	0.81	10.91	2.53	0.97	0.90	1	3.04	0.97	0.90	8.07	1	1.00	0.99	1	1
36	11216	0.95	0.81	4.56	1.50	0.97	0.90	1	1.60	0.97	0.90	4.93	1	1.00	0.99	1	1
36	10299	0.95	0.81	6.12	1.45	0.98	0.90	1	2.14	0.97	0.90	5.37	1	1.00	0.99	1	1
36	11605	0.94	0.81	5.70	1.56	0.97	0.90	1	2.17	0.98	0.90	5.65	1	1.00	0.99	1	1

'1nr' (1-norm ratio) is $\|H\|_1/\|A^+\|_1$.

'sr' (sparsity ratio) is $\|H\|_0/\|A^+\|_0$.

'lsr' (least-squares ratio) is $\|A(H\vec{1}_m) - \vec{1}_m\|_2/\|A(A^+\vec{1}_m) - \vec{1}_m\|_2$.

'2nr' (2-norm ratio) is $\|HA\vec{1}_n\|_2/\|A^+A\vec{1}_n\|_2$.

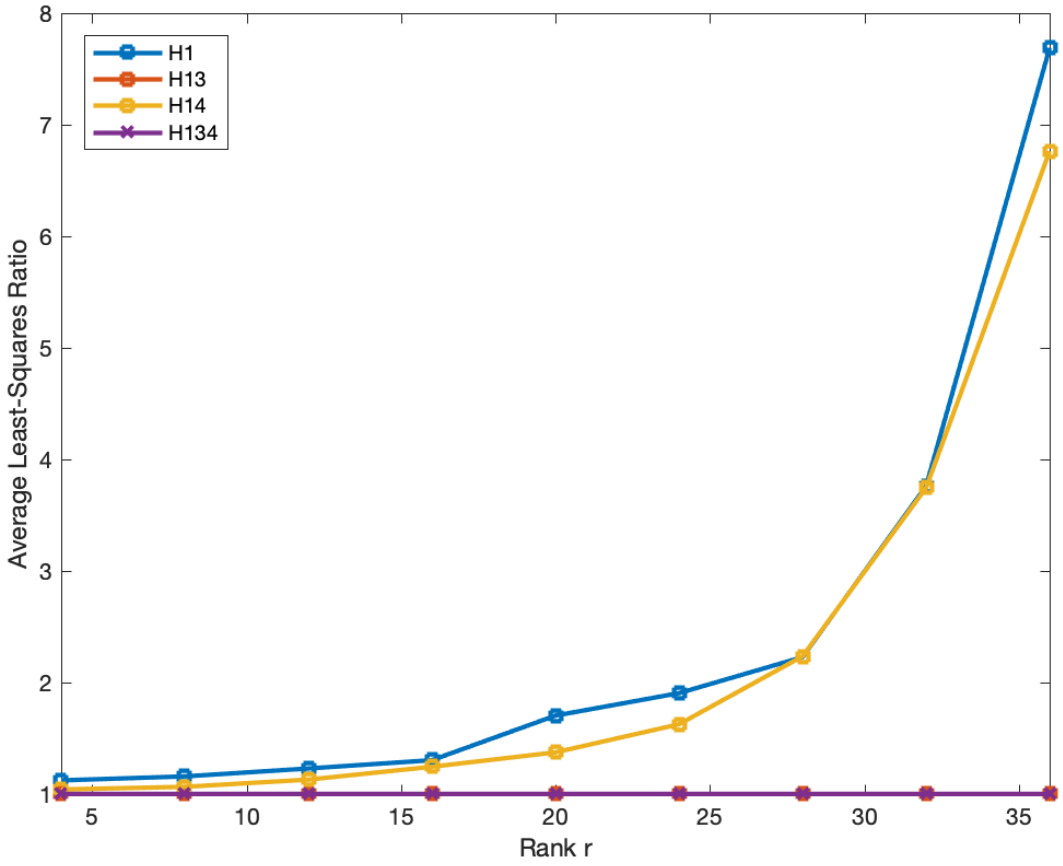


Figure 2.2: Average least-squares ratio vs. rank

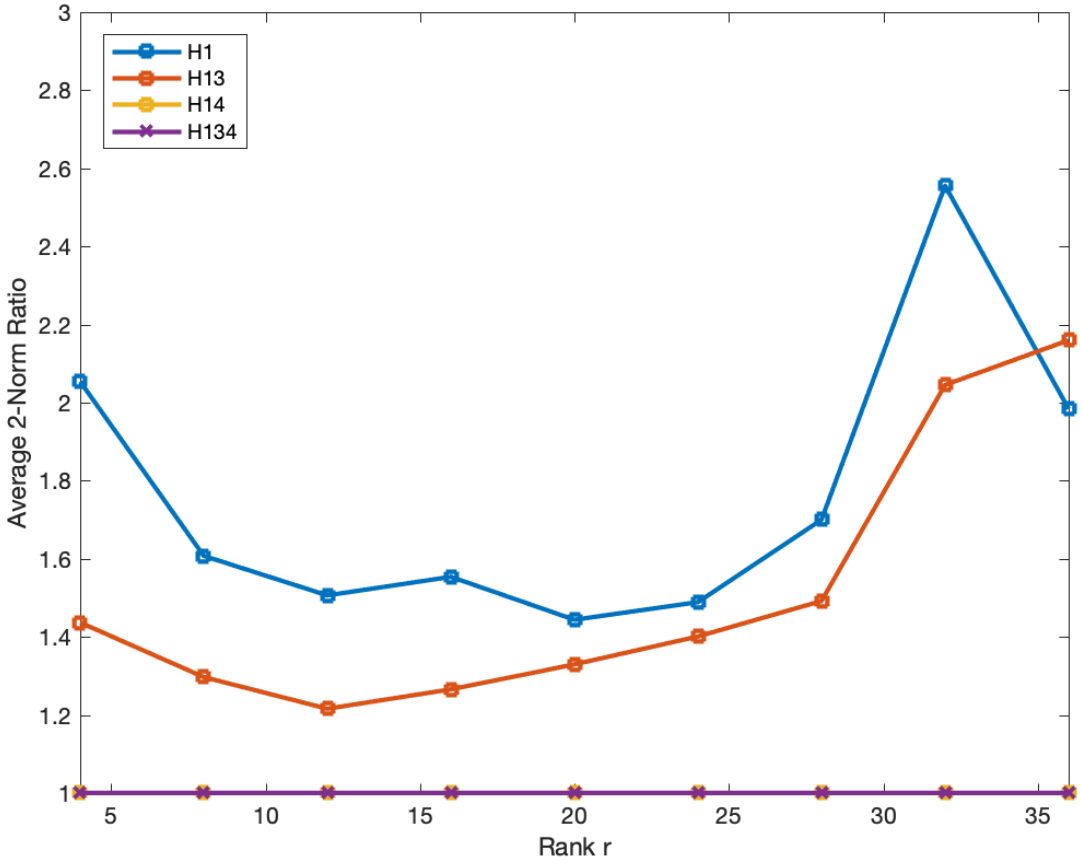


Figure 2.3: Average 2-norm ratio vs. rank

2.5 Conclusions and ongoing work

We have introduced four tractable generalized inverses based on using 1-norm minimization to induce sparsity and making linear-programming relaxations based on the M-P properties. It remains to be seen if any of these

new generalized inverses will be found to be valuable in practice. There is a natural tradeoff between sparsity and closeness to the M-P properties, and where one wants to be on this spectrum may well be application dependent.

We note that (P2) is a nonlinear property and it is nontrivial to handle it in a useful way by convex relaxation. We address this in Chapter 3 via a variety of computational methods.

Another idea that we are exploring is to develop update algorithms for sparse generalized inverses. The Sherman-Morrison-Woodbury formula gives us a convenient way to update a matrix inverse of A after a low-rank modification, where by extending that formula, A^+ can be updated efficiently (see [Mey73] and [Rie92]). It is an interesting challenge to see if we can take advantage of a sparse generalized inverse (or pseudoinverse) of A in calculating a sparse generalized inverse (or pseudoinverse) of a low-rank modification of A.

Chapter 3

Computationally dealing with the non-linear property of the Moore-Penrose Pseudoinverse

This chapter is motivated by work from [FL18b], [FL18a], and based on work from [FFL19].

Introduction

As we saw in the previous chapter, we can construct various sparse generalized inverses via combinations of linear properties of the Moore-Penrose pseudoinverse (MPP). Although solutions were demonstrably sparser than the MPP, especially in the low-rank settings, the question remains what the cost would be to enforce (P2). Any solution of (P1) has rank at least that of A. But a solution of (P1) and (P2) has rank equal to that of A. Thus it may be that a solution of (P1) is sparser than the MPP, but with a rank greater than or equal to that of the MPP, which can reflect how well

the solution satisfies (P2). So, instead of searching for a solution that satisfies (P2) exactly, we may instead explore approaches that, starting with an initial sparse generalized inverse, approximates (P2) and provides a sequence of solutions that illustrates the impact on the solution sparsity as the approximations are strengthened.

In the search for ways to approximate (P2), we consider a variety of frameworks for formulating the property in a convex setting, with particular focus on linear and convex quadratic lifting techniques. Once we establish the model formulations, we establish the quality of the solutions generated by utilizing performance measures 'lsr' and '2nr' (as seen in Chapter 2); while in exploring the tradeoff of sparsity and violation of (P2), we measure sparsity using the matrix 1-norm and violation using the Frobenius norm. We present some preliminary computational results (and illustrations) using these techniques and discuss some further questions born from these experiments.

3.1 Toward computationally dealing with (P2)

The Moore-Penrose (M-P) property (P2) is not convex in H. So we cannot incorporate it in convex relaxations. But, (P2) is quadratic in H, and there are standard approaches for handling nonconvex quadratics to consider.

We can view property (P2) as

$$h_{i\cdot}Ah_{\cdot j} = h_{ij}, \tag{3.1}$$

for all $ij \in m \times n$. So, we have mn quadratic equations to enforce, which

we can be expressed as

$$\frac{1}{2}\left(h_{i\cdot}, h'_{\cdot j}\right)\begin{bmatrix}\mathbf{0}_m & A\\ A' & \mathbf{0}_n\end{bmatrix}\begin{pmatrix}h'_{i\cdot}\\ h_{\cdot j}\end{pmatrix} = h_{ij}, \tag{3.2}$$

for all $ij \in m \times n$. We can view these quadratic equations (3.2) as

$$\frac{1}{2}\left\langle Q, \begin{pmatrix}h'_{i\cdot}\\ h_{\cdot j}\end{pmatrix}\left(h_{i\cdot}, h'_{\cdot j}\right)\right\rangle = h_{ij},$$

for all $ij \in m \times n$, where

$$Q := \begin{bmatrix}\mathbf{0}_m & A\\ A' & \mathbf{0}_n\end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)},$$

and $\langle \cdot, \cdot \rangle$ denotes element-wise dot-product.

Now, we lift the variables to matrix space, defining matrix variables

$$\mathcal{H}_{ij} := \begin{pmatrix}h'_{i\cdot}\\ h_{\cdot j}\end{pmatrix}\left(h_{i\cdot}, h'_{\cdot j}\right) \in \mathbb{R}^{(m+n)\times(m+n)},$$

for all $ij \in m \times n$. So, we can see (3.2) as the *linear* equations

$$\frac{1}{2}\left\langle Q, \mathcal{H}_{ij}\right\rangle = h_{ij}, \tag{3.3}$$

for all $ij \in m \times n$, together with the *non-convex* equations

$$\mathcal{H}_{ij} - \begin{pmatrix}h'_{i\cdot}\\ h_{\cdot j}\end{pmatrix}\left(h_{i\cdot}, h'_{\cdot j}\right) = \mathbf{0}_{m+n}, \tag{3.4}$$

for all $ij \in m \times n$. Next, we relax the equations (3.4) via the *convex semi-*

definiteness constraints:

$$\mathcal{H}_{ij} - \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} (h_{i\cdot}, h'_{\cdot j}) \succeq \mathbf{0}_{m+n}, \tag{3.5}$$

for all $ij \in m \times n$. So we can relax the M-P property (P2) as (3.3) and (3.5), for all $ij \in m \times n$.

To put (3.5) into a standard form for semi-definite programming, we create variables vectors $x_{ij} \in \mathbb{R}^{m+n}$, and we have linear equations

$$x_{ij} = \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix}. \tag{3.6}$$

Next, for all $ij \in m \times n$, we introduce symmetric positive semi-definite matrix variables $Z_{ij} \in \mathbb{R}^{(m+n+1)\times(m+n+1)}$, interpreting the entries as follows:

$$Z_{ij} = \begin{bmatrix} x^{(0)}_{ij} & x'_{ij} \\ x_{ij} & \mathcal{H}_{ij} \end{bmatrix}. \tag{3.7}$$

Then the linear equation

$$x^{(0)}_{ij} = 1 \tag{3.8}$$

and Z_{ij} positive semi-definite precisely enforce (3.5).

Finally, we re-cast (3.3) as

$$\frac{1}{2} \langle \bar{Q}, Z_{ij} \rangle = h_{ij}, \tag{3.9}$$

where

$$\bar{Q} := \begin{bmatrix} 0 & \vec{0}'_{m+n} \\ \vec{0}_{m+n} & Q \end{bmatrix} \in \mathbb{R}^{(m+n+1)\times(m+n+1)}. \tag{3.10}$$

In principle, we can consider minimizing $\|H\|_1$ subject to property (P1) and *any subset* of (P3), (P4), and (3.3)+(3.5) for all $ij \in m \times n$ (though we would reformulate (3.3)+(3.5) as above, so it is in a convenient form for semi-definite programming solvers used in `CVX`). In doing so, we get further new sparse generalized inverses which are all tractable via semi-definite programming.

Of course all of these relaxations have feasible solutions, because the Moore-Penrose pseudoinverse (MPP) A^+ always gives a feasible solution. An important issue is whether there is a strictly feasible solution — the *Slater condition(/constraint qualification)* — as that is sufficient for strong duality to hold and affects the convergence of algorithms (e.g., see [BV04]). Even if the Slater condition does not hold, there is a facial-reduction algorithm that can induce the Slater condition to hold on an appropriate face of the feasible region (see [Pat13]).

3.2 A lighter version of (P1)

While linear-optimization solvers are well equipped for handling redundant linear equations, there is some evidence that quadratic and SDP solvers are not so well equipped. So, it is useful to handle such redundancies at the modeling level[1]. M-P property (P1) involves mn equations $AHA = A$. Considering the singular value decomposition $A = U\Sigma V'$, if A has rank r, then we can take U, Σ and V to be $m \times r$, $r \times r$, and $n \times r$, respectively. Then we can reduce (P1) to the r^2 ($\leq mn$) linear equations

$$(\Sigma V')\, H\, (U\Sigma) \;=\; \Sigma. \tag{3.11}$$

[1] for example, per Erling Andersen with regard to the conic solver MOSEK

3.3 Quadratic tightening of (P1)

We can view (P1) as

$$\sum_{k=1}^{n}\sum_{l=1}^{m} a_{ik}a_{\ell j}h_{k\ell} = a_{ij},$$

for $i = 1, \dots m$, $j = 1, \dots, n$. For $p = 1, \dots, n$ and $q = 1, \dots m$, we can multiply by h_{pq} to get

$$\sum_{k=1}^{n}\sum_{l=1}^{m} a_{ik}a_{\ell j}h_{k\ell}h_{pq} = a_{ij}h_{pq}. \tag{3.12}$$

Next, it is easy to see that

$$h_{k\ell}h_{pq} = \left(\mathcal{H}_{kq}\right)_{m+p,\ell} \text{ (and symmetrically)}$$

and so we arrive at the valid linear equations

$$\sum_{k=1}^{n}\sum_{l=1}^{m} a_{ik}a_{\ell j}\left(\mathcal{H}_{kq}\right)_{m+p,\ell} = a_{ij}h_{pq}. \tag{3.13}$$

We note that we could apply this same idea to (3.11), the lighter version of (P1).

3.4 Penalizing "big" $\mathcal{H}_{ij}$ with quadratic penalties

The *convex semi-definiteness* constraints (3.5):

$$\mathcal{H}_{ij} - \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} \left(h_{i\cdot}, h'_{\cdot j}\right) \succeq \mathbf{0}_{m+n}$$

are not useful by themselves because for any choice of H, they are satisfied by simply choosing $\mathcal{H}_{ij}$ to be "big enough in the positive semi-definite sense" (e.g., choose $\mathcal{H}_{ij} := \lambda I$, where λ is the greatest eigenvalue of

$$\begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} \left(h_{i\cdot}, h'_{\cdot j}\right).$$

Because $\mathcal{H}_{ij}$ appears nowhere else, relaxing (3.4) to (3.5) is like throwing (P2) out entirely. However, we can bring it back into play by replacing our objective min $\|H\|_1$ with

$$\min \ \tau\|H\|_1 + (1-\tau)\sum_{i,j} \operatorname{tr}\left(\mathcal{H}_{ij}\right), \tag{3.14}$$

Because of (3.5), we can add the further restriction that each diagonal entry of

$$\mathcal{H}_{ij} - \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} \left(h_{i\cdot}, h'_{\cdot j}\right)$$

should be non-negative. That is, for all i and j, we have the following convex constraints:

$$\left(\mathcal{H}_{ij}\right)_{\ell\ell} - h_{i\ell}^2 \geq 0, \text{ for } \ell = 1, \ldots, m, \tag{3.15}$$

and

$$\left(\mathcal{H}_{ij}\right)_{m+\ell,m+\ell} - h_{\ell j}^2 \geq 0, \text{ for } \ell = 1, \ldots, n, \tag{3.16}$$

Note that it is really necessary to do something like (3.15)-(3.16), or else (3.14) will be unbounded.

Of course we have to find a suitable value for τ. It may well be advisable to dynamically alter τ, starting with very small τ, and then gradually increasing τ (emphasizing sparsity of H) as we come closer to satisfying (3.5).

3.5 Quadratic PSD-cuts

One idea for getting away from having to impose semi-definiteness constraints is to outer approximate them. We follow such an approach from the literature (see [SBL10a], for example).

For any $v \in \mathbb{R}^{m+n}$, by (3.5), we have the valid inequality

$$v' \left(\mathcal{H}_{ij} - \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} \left(h_{i\cdot}, h'_{\cdot j} \right) \right) v \geq 0.$$

Equivalently, we have the *quadratic PSD-cut*

$$\left\langle vv', \mathcal{H}_{ij} \right\rangle - \left(\left(h_{i\cdot}, h'_{\cdot j} \right) v \right)^2 \geq 0, \tag{3.17}$$

which we can easily see are convex quadratics. By letting v be an eigenvector of

$$\hat{\mathcal{H}}_{ij} - \begin{pmatrix} \hat{h}'_{i\cdot} \\ \hat{h}_{\cdot j} \end{pmatrix} \left(\hat{h}_{i\cdot}, \hat{h}'_{\cdot j} \right)$$

corresponding to a negative eigenvalue, the corresponding quadratic PSD-

cut, which we now refer to as an *eigen-cut*, will be violated by $\widehat{H}_{ij}$, $\widehat{h}_{i\cdot}$, $\widehat{h}_{\cdot j}$.

Schur complements

Using the above, we can see how to get violated quadratic PSD-cuts from negative eigenvalues of appropriate matrices. For convenience, let

$$X := \mathcal{H}_{ij},$$

and let

$$x := \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix}.$$

We think of $\widehat{X}$ and $\widehat{x}$ as fixed at this point — a solution of a relaxation. And then if v is an eigenvector of $\widehat{X} - \widehat{x}\widehat{x}'$ with negative eigenvalue, then $v' (X - xx') v \geq 0$ is a valid quadratic PSD-cut that is violated by $\widehat{X}$, $\widehat{x}$.

Now we can also consider

$$\widehat{Z} := \left(\begin{array}{c|c} 1 & \widehat{x}' \\ \hline \widehat{x} & \widehat{X} \end{array}\right).$$

It is appealing to consider the matrix variable Z because PSD-cuts for Z are *linear* (rather than convex quadratic).

It is helpful to also consider

$$\widehat{W} := \left(\begin{array}{c|c} 0 & \vec{0}' \\ \hline \vec{0} & \widehat{X} - \widehat{x}\widehat{x}' \end{array}\right).$$

Note that bordering $\widehat{X} - \widehat{x}\widehat{x}'$ with zeros includes an additional zero eigenvalue into the set of eigenvalues of $\widehat{X} - \widehat{x}\widehat{x}'$.

Let $\lambda_n \geq \cdots \geq \lambda_2 \geq \lambda_1$ be the decreasingly ordered list of eigenvalues of $\hat{Z}$, and let $\mu_n \geq \mu_{n-1} \geq \cdots \geq \mu_2 \geq \mu_1$ be the decreasingly ordered list of eigenvalues of $\hat{W}$. Then, there is a nice interlacing principle here (see [Zha05, Theorem 2.1]):

$$\lambda_n \geq \mu_n \geq \lambda_{n-1} \geq \mu_{n-1} \geq \cdots \geq \lambda_2 \geq \mu_2 \geq \lambda_1.$$

Note that μ_1 does not appear here. We can conclude:

1. if $\hat{W}$ (or $\hat{X} - \hat{x}\hat{x}'$) has k negative eigenvalues then $\hat{Z}$ has either k or $k-1$ negative eigenvalues;
2. if $\hat{Z}$ has k negative eigenvalues then $\hat{W}$ (or $\hat{X} - \hat{x}\hat{x}'$) has k or $k+1$ negative eigenvalues.

This *might* suggest that, ignoring the time to solve relaxations, it could be better to work with $\hat{X} - \hat{x}\hat{x}'$ rather than $\hat{Z}$. Certainly when $k = 0$ we can see that if $\hat{Z}$ has $k = 0$ negative eigenvalues then $\hat{W}$ has 0 negative eigenvalues. But we can say more. Consider the following definition.

Definition 8. *The inertia of an $n \times n$ Hermitian matrix Q is the ordered triple $In(Q) := (p(Q), q(Q), z(Q))$, in which $p(Q)$, $q(Q)$, and $z(Q)$ are the numbers of the positive, negative, and zero eigenvalues of Q, respectively (including multiplicities) (see [Zha05, Section 1.3]).*

We have

$$\hat{V} := \left(\begin{array}{c|c} 1 & \vec{0}' \\ \hline \vec{0} & \hat{X} - \hat{x}\hat{x}' \end{array}\right) = \left(\begin{array}{c|c} 1 & \vec{0}' \\ \hline -\hat{x} & I \end{array}\right) \quad \hat{Z} \quad \left(\begin{array}{c|c} 1 & -\hat{x}' \\ \hline \vec{0} & I \end{array}\right).$$

So based on [Zha05, Theorem 1.5], In($\hat{V}$)=In($\hat{Z}$). Therefore $\hat{V}$ and $\hat{Z}$ have the same number of negative eigenvalues. Also, based on [Zha05, Theorem 1.6], we have

$$\mathrm{In}(\hat{Z}) = \mathrm{In}([1]) + \mathrm{In}(\hat{X} - \hat{x}\hat{x}').$$

Therefore, $\hat{Z}$ and $\hat{X} - \hat{x}\hat{x}'$ have the same number of negative eigenvalues.

3.6 Pre-computed quadratic PSD-cuts

We have seen very slow convergence with eigen-cuts, and we have found it useful to pre-compute some quadratic PSD-cuts. A useful set of quadratic PSD-cuts appears to come from choosing v to be all choices of non-zero vectors with only ± 1 as non-zeros and at most two such non-zeros (see [AH17], for example). Because there is no need to include the negative of any utilized v, the number of such v (which would give us cuts for each i and j) is $m + n + \binom{m+n}{2}$.

3.7 2-by-2 cuts

Besides quadratic PSD-cuts, which enforce that the diagonal entries of

$$\mathcal{H}_{ij} - \begin{pmatrix} h'_{i\cdot} \\ h_{\cdot j} \end{pmatrix} (h_{i\cdot}, h'_{\cdot j})$$

be non-negative, we can additionally seek to enforce that the 2-by-2 principle submatrices of the $\mathcal{H}_{ij}$ be positive semi-definite (see [KK03]). Letting

$$M := \begin{pmatrix} a & c \\ c & b \end{pmatrix}$$

denote a principle submatrix of $\mathcal{H}_{ij}$, we can easily see that $M \succeq \mathbf{0}$ is equivalent to

$$ab - c^2 \geq 0,\ a \geq 0,\ b \geq 0. \tag{3.18}$$

This solution set of (3.18) may not appear to be convex, but it is — being just a *rotated second-order cone*.

3.8 Reformulation Linearization Technique (RLT)

The Reformulation-Linearization-Technique (RLT) (see [SA99]) is a method that generates strengthened linear programming relaxations. We can take any pairs of inequalities $\alpha' x \geq \beta$ and $\gamma' x \geq \delta$ and consider the valid inequality $(\alpha' x - \beta)(\gamma' x - \delta) \geq 0$. Expanding this we have $\sum_i \sum_j \alpha_i \gamma_j x_i x_j - (\delta\alpha' + \beta\gamma')\, x + \beta\delta \geq 0$, which we then linearize by replacing $x_i x_j$ by a new variable y_{ij}.

In our context, we assume that we can put box constraints on the h_{ij}, say

$$\lambda_{ij} \leq h_{ij} \leq \mu_{ij}, \tag{3.19}$$

for $ij \in n \times m$, and then applying RLT to them.

Now, consider for all $pq \in n \times m$ and $k\ell \in n \times m$, the valid inequalities derived from the box constraints:

$$\begin{aligned}(h_{pq} - \lambda_{pq})(h_{k\ell} - \lambda_{k\ell}) &\geq 0,\\ (h_{pq} - \lambda_{pq})(\mu_{k\ell} - h_{k\ell}) &\geq 0,\\ (\mu_{pq} - h_{pq})(\mu_{k\ell} - h_{k\ell}) &\geq 0.\end{aligned}$$

Equivalently, we have the following inequalities.

For all $pq \in n \times m$ and $k\ell \in n \times m$:

$$\begin{aligned} &h_{pq}h_{k\ell} - \lambda_{k\ell}h_{pq} - \lambda_{pq}h_{k\ell} + \lambda_{pq}\lambda_{k\ell} \geq 0, \\ &-h_{pq}h_{k\ell} + \mu_{k\ell}h_{pq} + \lambda_{pq}h_{k\ell} - \lambda_{pq}\mu_{k\ell} \geq 0, \\ &h_{pq}h_{k\ell} - \mu_{k\ell}h_{pq} - \mu_{pq}h_{k\ell} + \mu_{pq}\mu_{k\ell} \geq 0. \end{aligned} \tag{3.20}$$

For all $j = 1, \dots m$ and $k\ell \in n \times m$:

$$\begin{aligned} &h_{kj}h_{k\ell} - \lambda_{k\ell}h_{kj} - \lambda_{kj}h_{k\ell} + \lambda_{kj}\lambda_{k\ell} \geq 0, \\ &-h_{kj}h_{k\ell} + \mu_{k\ell}h_{kj} + \lambda_{kj}h_{k\ell} - \lambda_{kj}\mu_{k\ell} \geq 0, \\ &h_{kj}h_{k\ell} - \mu_{k\ell}h_{kj} - \mu_{kj}h_{k\ell} + \mu_{kj}\mu_{k\ell} \geq 0. \end{aligned}$$

For all $i = 1, \dots n$ and $pq \in n \times m$:

$$\begin{aligned} &h_{pq}h_{iq} - \lambda_{iq}h_{pq} - \lambda_{pq}h_{iq} + \lambda_{pq}\lambda_{iq} \geq 0, \\ &-h_{pq}h_{iq} + \mu_{iq}h_{pq} + \lambda_{pq}h_{iq} - \lambda_{pq}\mu_{iq} \geq 0, \\ &h_{pq}h_{iq} - \mu_{iq}h_{pq} - \mu_{pq}h_{iq} + \mu_{pq}\mu_{iq} \geq 0. \end{aligned}$$

From (3.4), we have the following identities for all $k = 1, \dots n$, $q = 1, \dots, m$.

For $p = 1, \dots n$, $\ell = 1, \dots, m$:

$$h_{pq}h_{k\ell} = \left(\mathcal{H}_{kq}\right)_{m+p,\ell} \text{ (and symmetrically),}$$

and

$$h_{k\ell}h_{pq} = \left(\mathcal{H}_{p\ell}\right)_{m+k,q} \text{ (and symmetrically).}$$

For $j = 1, \dots m$, $\ell = 1, \dots, m$:

$$h_{kj}h_{k\ell} = \left(\mathcal{H}_{kq}\right)_{j,\ell} \text{ (and symmetrically).}$$

For $p = 1, \dots n$, $i = 1, \dots, n$:

$$h_{pq}h_{iq} = \left(\mathcal{H}_{kq}\right)_{m+p,m+i} \text{ (and symmetrically).}$$

So we can linearize the valid inequalities above in the lifted space, as:

For all $p = 1, \dots, n$, $q = 1, \dots m$, $k = 1, \dots, n$, $\ell = 1, \dots, m$:

$$\begin{aligned}
&\left(\mathcal{H}_{kq}\right)_{m+p,\ell} - \lambda_{k\ell}h_{pq} - \lambda_{pq}h_{k\ell} + \lambda_{pq}\lambda_{k\ell} \geq 0,\\
&-\left(\mathcal{H}_{kq}\right)_{m+p,\ell} + \mu_{k\ell}h_{pq} + \lambda_{pq}h_{k\ell} - \lambda_{pq}\mu_{k\ell} \geq 0,\\
&\left(\mathcal{H}_{kq}\right)_{m+p,\ell} - \mu_{k\ell}h_{pq} - \mu_{pq}h_{k\ell} + \mu_{pq}\mu_{k\ell} \geq 0,
\end{aligned}$$

and

$$\begin{aligned}
&\left(\mathcal{H}_{p\ell}\right)_{m+k,q} - \lambda_{k\ell}h_{pq} - \lambda_{pq}h_{k\ell} + \lambda_{pq}\lambda_{k\ell} \geq 0,\\
&-\left(\mathcal{H}_{p\ell}\right)_{m+k,q} + \mu_{k\ell}h_{pq} + \lambda_{pq}h_{k\ell} - \lambda_{pq}\mu_{k\ell} \geq 0,\\
&\left(\mathcal{H}_{p\ell}\right)_{m+k,q} - \mu_{k\ell}h_{pq} - \mu_{pq}h_{k\ell} + \mu_{pq}\mu_{k\ell} \geq 0.
\end{aligned}$$

For all $q = 1, \dots, m$, $j = 1, \dots, m$, $k = 1, \dots, n$, $\ell = 1, \dots, m$:

$$\begin{aligned}
&\left(\mathcal{H}_{kq}\right)_{j,\ell} - \lambda_{k\ell}h_{kj} - \lambda_{kj}h_{k\ell} + \lambda_{kj}\lambda_{k\ell} \geq 0,\\
&-\left(\mathcal{H}_{kq}\right)_{j,\ell} + \mu_{k\ell}h_{kj} + \lambda_{kj}h_{k\ell} - \lambda_{kj}\mu_{k\ell} \geq 0,\\
&\left(\mathcal{H}_{kq}\right)_{j,\ell} - \mu_{k\ell}h_{kj} - \mu_{kj}h_{k\ell} + \mu_{kj}\mu_{k\ell} \geq 0.
\end{aligned}$$

Note that from the first equations on this last group, we have for all $j = 1, \dots, m$:

$$\begin{aligned}
\left(\mathcal{H}_{kq}\right)_{j,j} &\geq \lambda_{kj}h_{kj} + \lambda_{kj}h_{kj} - \lambda_{kj}\lambda_{kj}\\
&\geq 2\lambda_{kj}\lambda_{kj} - \lambda_{kj}\lambda_{kj} = \lambda_{kj}^2 \geq 0.
\end{aligned}$$

For all $k = 1, \dots, n$, $i = 1, \dots, n$, $p = 1, \dots, n$, $q = 1, \dots, n$:

$$\begin{aligned}
&\left(\mathcal{H}_{kq}\right)_{m+p,m+i} - \lambda_{iq}h_{pq} - \lambda_{pq}h_{iq} + \lambda_{pq}\lambda_{iq} \geq 0,\\
&-\left(\mathcal{H}_{kq}\right)_{m+p,m+i} + \mu_{iq}h_{pq} + \lambda_{pq}h_{iq} - \lambda_{pq}\mu_{iq} \geq 0,\\
&\left(\mathcal{H}_{kq}\right)_{m+p,m+i} - \mu_{iq}h_{pq} - \mu_{pq}h_{iq} + \mu_{pq}\mu_{iq} \geq 0.
\end{aligned}$$

Note that from the first equations on this last group, we have for all $i = 1, \dots, n$:

$$\begin{aligned}
\left(\mathcal{H}_{kq}\right)_{m+i,m+i} &\geq \lambda_{iq}h_{iq} + \lambda_{iq}h_{iq} - \lambda_{iq}\lambda_{iq}\\
&\geq 2\lambda_{iq}\lambda_{iq} - \lambda_{iq}\lambda_{iq} = \lambda_{iq}^2 \geq 0.
\end{aligned}$$

3.9 Non-symmetric lifting

In previous sections we have considered modeling property (P2) by using nm symmetric matrix variables $\mathcal{H}_{ij} \in \mathbb{R}^{(m+n)\times(m+n)}$ in the constraints (3.3) and (3.4). The motivation for this formulation is to relax (P2), or more specifically, to relax (3.4) using semi-definite programming. Although semi-definite relaxations are mathematically appealing and lead to interesting results, lifting to the nm symmetric matrix variables $\mathcal{H}_{ij} \in \mathbb{R}^{(m+n)\times(m+n)}$ is rather heavy.

Aiming to avoid the computational/numerical difficulty introduced with this heavy lifting, we alternatively consider modeling (P2) by the nm *non-symmetric* quadratic equations (3.1), and investigate possible convex relaxations of these equations. We seek to avoid re-casting each quadratic equation (3.1) as a symmetric quadratic equation (3.2). To do this, we work with matrices in $\mathbb{R}^{m\times n}$ rather than in $\mathbb{R}^{(m+n)\times(m+n)}$ so we are still lifting, but in a lighter manner. In the heavier situation that we previously considered, for each of the mn symmetric matrix variables in $\mathbb{R}^{(m+n)\times(m+n)}$,

there are $\binom{m+n}{2}$ independent scalar variables. In the approach that we now suggest to investigate, for each of the mn non-symmetric matrix variables in $\mathbb{R}^{m\times n}$, there are mn scalar variables. For $m = n$, this is a savings of about half the number of scalar variables of the lifting.

General non-symmetric quadratic forms

Our approach may well have other applications when one has general non-symmetric quadratic, so we present it more generally.

Consider a general quadratic form $f(x, y) := x'Ry$, with $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $R \in \mathbb{R}^{m\times n}$. Though we write x and y as if they are disjoint vectors of variables, it can well be that the scalar variables that x and y each comprise overlap. In fact, this will be the case for our application to (3.1), where $h_{i\cdot}$ and $h_{\cdot j}$ overlap on the variable h_{ij}.

We assume that we have or can derive reasonable box constraints on x and y: $\hat{\lambda}_x \leq x \leq \hat{\mu}_x$ and $\hat{\lambda}_y \leq y \leq \hat{\mu}_y$.

We can see our general quadratic form as

$$f(x, y) = x'Ry = \langle R, xy' \rangle \,.$$

Now, we can lift to non-symmetric matrix space by defining $W := xy' \in \mathbb{R}^{m\times n}$. So we model $f(x, y)$ as

$$f(x, y) = \langle R, W \rangle,$$

and we focus on relaxing

$$W - xy' = \mathbf{0}_{m\times n}. \tag{3.21}$$

Emulating Saxena et al.

Suppose that we have solved a relaxation and have values $\hat{W}$, $\hat{x}$, $\hat{y}$, violating (3.21). We can consider the associated SVD:

$$U'(\hat{W} - \hat{x}\hat{y}')V = \Sigma,$$

where a violation of (3.21) means that there is at least one non-zero singular value σ. So, for the associated columns $u \in U$ and $v \in V$, we have

$$u'(\hat{W} - \hat{x}\hat{y}')v = \sigma \neq 0.$$

This motivates looking at the violated valid equation

$$\langle W, uv' \rangle - (u'x)(v'y) = 0. \tag{3.22}$$

To emphasize, note that in (3.22) the variables are W, x, and y, while u and v are fixed. The non-linearity in (3.22) is only in the single product of $u'x$ with $v'y$. To deal with it, we can induce separability by defining

$$\begin{aligned} t_1 &:= (u'x + v'y)/2, \\ t_2 &:= (u'x - v'y)/2. \end{aligned}$$

So,

$$\begin{aligned} u'x &= t_1 + t_2, \\ v'y &= t_1 - t_2, \end{aligned}$$

and then we have

$$(u'x)(v'y) = t_1^2 - t_2^2.$$

In this manner, we may replace (3.22) with

$$\langle W, uv' \rangle - t_1^2 + t_2^2 \leq 0, \tag{3.23}$$

$$-\langle W, uv' \rangle + t_1^2 - t_2^2 \leq 0. \tag{3.24}$$

Then we can treat the quadratic terms of (3.23-3.24) via the technique of Saxena et al. [SBL10a]. That is, (i) we either leave the convex $+t_i^2$ terms as is or possibly linearize via lower-bounding tangents, and (ii) we make secant inequalities and disjunctive cuts on the concave $-t_i^2$ terms, which requires first calculating lower and upper bounds on the t_i. Note that we can either derive bounds on the t_i from the box constraints on x and y, or we can get potentially better bounds by solving further (convex) optimization problems.

Note that if we *simultaneously* treat the two concave terms ($-t_i^2$) via the disjunctive technique of Saxena et al., we are led to a 4-way disjunction.

McCormick instead of Saxena et al.

Another possible way of relaxing (3.22) is to apply a McCormick convexification. Let

$$\begin{cases} s := \langle W, uv' \rangle \ , \\ p_1 := u'x \ , \\ p_2 := v'y \ . \end{cases} \tag{3.25}$$

We first calculate bounds $[a_i, b_i]$, for p_i $(i = 1, 2)$. Then we carry out the associated McCormick relaxation of $s = p_1 p_2$:

$$s \leq b_2 p_1 + a_1 p_2 - a_1 b_2 \tag{I.1}$$
$$s \leq a_2 p_1 + b_1 p_2 - a_2 b_1 \tag{I.2}$$
$$s \geq a_2 p_1 + a_1 p_2 - a_1 a_2 \tag{I.3}$$
$$s \geq b_2 p_1 + b_1 p_2 - b_1 b_2 \tag{I.4}$$

Substituting back in (3.25), we obtain

$$\langle W, uv' \rangle \leq b_2 u' x + a_1 v' y - a_1 b_2 \tag{I.1'}$$
$$\langle W, uv' \rangle \leq a_2 u' x + b_1 v' y - a_2 b_1 \tag{I.2'}$$
$$\langle W, uv' \rangle \geq a_2 u' x + a_1 v' y - a_1 a_2 \tag{I.3'}$$
$$\langle W, uv' \rangle \geq b_2 u' x + b_1 v' y - b_1 b_2, \tag{I.4'}$$

and we can hope that these are violated by $\hat{W}$, $\hat{x}$, $\hat{y}$.

Backing up a bit to compare with Saxena et al., here we are relaxing $s = p_1 p_2$. If $p_1 = p_2 =: p$ (the Saxena et al. case), then we have $s = p^2$, whereupon we can distinguish the two "sides":

$$s \geq p^2 \qquad \text{(convex)}$$
$$s \leq p^2 \qquad \text{(concave)}$$

Then Saxena et al. use (i) the convex side directly (or a linearization of it), and (ii) disjunctive programming on the concave side.

The question now begs, can we take (I.1–I.4) and do disjunctive programming in some nice way? It is convenient to work with box domains, so we could pick η_i in $[a_i, b_i]$, for $i = 1, 2$. Then we get four boxes, by pairing one

of

$$[a_1, \eta_1],\ [\eta_1, b_1],$$

and one of

$$[a_2, \eta_2],\ [\eta_2, b_2].$$

For each box, we get a new McCormick convexification (in the spirit of I.1–I.4). And so, as in the technique of the previous subsection, we have a 4-way disjunction to base a disjunctive cut upon.

3.10 Modeling our problem

As stated, we can write (P2) as the following nm non-symmetric quadratic equations

$$h_{i\cdot} A h_{\cdot j} = h_{ij},$$

which can also be expressed as

$$\left\langle A, h'_{i\cdot} h'_{\cdot j} \right\rangle = h_{ij},$$

for all $ij \in n \times m$.

We lift to non-symmetric matrix space, defining the matrix variables

$$\mathcal{K}_{ij} := h'_{i\cdot} h'_{\cdot j} \in \mathbb{R}^{m \times n},$$

for all $ij \in n \times m$.

Property (P2) can then be modeled by the linear equations

$$\left\langle A, \mathcal{K}_{ij} \right\rangle = h_{ij}, \tag{3.26}$$

together with the non-convex equations

$$\mathcal{K}_{ij} - h'_{i\cdot}h'_{\cdot j} = \mathbf{0}_{m\times n}, \tag{3.27}$$

for all $ij \in n \times m$.

For all $ij \in n \times m$ and $k\ell \in m \times n$, we now have

$$\left(\mathcal{K}_{ij}\right)_{k,\ell} = h_{ik}h_{\ell j},$$

and we relax these non-convex equations with the McCormick/RLT inequalities derived from the box constraints on the h_{ij} (3.19) and the valid inequalities (3.20).

Linearizing (3.20), we now obtain:

$$\begin{aligned}
&\left(\mathcal{K}_{ij}\right)_{k,\ell} - \lambda_{\ell j}h_{ik} - \lambda_{ik}h_{\ell j} + \lambda_{ik}\lambda_{\ell j} \geq 0,\\
&-\left(\mathcal{K}_{ij}\right)_{k,\ell} + \mu_{\ell j}h_{ik} + \lambda_{ik}h_{\ell j} - \lambda_{ik}\mu_{\ell j} \geq 0,\\
&\left(\mathcal{K}_{ij}\right)_{k,\ell} - \mu_{\ell j}h_{ik} - \mu_{ik}h_{\ell j} + \mu_{ik}\mu_{\ell j} \geq 0,
\end{aligned}$$

for all $ij \in n \times m$ and $k\ell \in m \times n$.

We can also consider applying the equations for the quadratic tightening of (P1); that is, (3.12):

$$\sum_{i=1}^{n}\sum_{k=1}^{m} a_{pi}a_{kq}h_{ik}h_{\ell j} = a_{pq}h_{\ell j},$$

for $\ell j \in n \times m$ and $pq \in m \times n$. Linearizing them, we obtain:

$$\sum_{i=1}^{n}\sum_{k=1}^{m} a_{pi}a_{kq}(\mathcal{K}_{ij})_{k,l} = a_{pq}h_{\ell j}. \tag{3.28}$$

We further consider the valid inequalities (3.23) and (3.24). Following the discussion in §3.9, let $u^{ij} \in \mathbb{R}^m$ and $v^{ij} \in \mathbb{R}^n$ be the vectors such that

$$u^{ij'}(\hat{\mathcal{K}}_{ij} - \hat{h}'_{i\cdot}\hat{h}'_{\cdot j})v^{ij} \neq 0.$$

for given values $\hat{\mathcal{K}}_{ij}$, $\hat{h}_{i\cdot}$, $\hat{h}_{\cdot j}$, for all $ij \in n \times m$.

Considering this notation, the valid inequalities (3.23) and (3.24) are now

$$\begin{aligned} \left\langle \mathcal{K}_{ij}, u^{ij}v^{ij'} \right\rangle + w_{1ij} + t_{2ij}^2 \leq 0, \\ -\left\langle \mathcal{K}_{ij}, u^{ij}v^{ij'} \right\rangle + t_{1ij}^2 + w_{2ij} \leq 0, \end{aligned}$$

where the concave terms $-t_{pij}^2$ have been replaced with the linear terms $+w_{pij}$, for $p = 1, 2$. Assuming lower and upper bounds on t_{pij} ($\alpha_{pij} \leq t_{pij} \leq \beta_{pij}$), the new variables w_{pij} are then constrained to satisfy the secant inequalities

$$-\left((t_{pij} - \alpha_{pij}) \frac{\beta_{pij}^2 - \alpha_{pij}^2}{\beta_{pij} - \alpha_{pij}} + \alpha_{pij}^2 \right) \leq w_{pij} \ .$$

Interval bounds $[\alpha_{pij}, \beta_{pij}]$ on t_{pij} can be directly derived from the bounds on h_{ij} ($\lambda_{ij} \leq h_{ij} \leq \mu_{ij}$). As

$$\begin{aligned} t_{1ij} &= (u^{ij'}h'_{i\cdot} + v^{ij'}h_{\cdot j})/2, \\ t_{2ij} &= (u^{ij'}h'_{i\cdot} - v^{ij'}h_{\cdot j})/2, \end{aligned}$$

we have

$$\alpha_{1ij} = \frac{1}{2}\left(\sum_{\ell=1}^{m}\left(\min\{(u^{ij})_\ell\lambda_{i\ell},(u^{ij})_\ell\mu_{i\ell}\}\right)+\sum_{\ell=1}^{n}\left(\min\{(v^{ij})_\ell\lambda_{\ell j},(v^{ij})_\ell\mu_{\ell j}\}\right)\right),$$

$$\beta_{1ij} = \frac{1}{2}\left(\sum_{\ell=1}^{m}\left(\max\{(u^{ij})_\ell\lambda_{i\ell},(u^{ij})_\ell\mu_{i\ell}\}\right)+\sum_{\ell=1}^{n}\left(\max\{(v^{ij})_\ell\lambda_{\ell j},(v^{ij})_\ell\mu_{\ell j}\}\right)\right),$$

$$\alpha_{2ij} = \frac{1}{2}\left(\sum_{\ell=1}^{m}\left(\min\{(u^{ij})_\ell\lambda_{i\ell},(u^{ij})_\ell\mu_{i\ell}\}\right)-\sum_{\ell=1}^{n}\left(\max\{(v^{ij})_\ell\lambda_{\ell j},(v^{ij})_\ell\mu_{\ell j}\}\right)\right),$$

$$\beta_{2ij} = \frac{1}{2}\left(\sum_{\ell=1}^{m}\left(\max\{(u^{ij})_\ell\lambda_{i\ell},(u^{ij})_\ell\mu_{i\ell}\}\right)-\sum_{\ell=1}^{n}\left(\min\{(v^{ij})_\ell\lambda_{\ell j},(v^{ij})_\ell\mu_{\ell j}\}\right)\right).$$

Though, we could also seek to tighten these bounds by casting and solving appropriate optimization problems.

We note that, as we discussed in §3.9, the vectors u^{ij} and v^{ij} can be obtained from the columns of the matrices U^{ij} and V^{ij} in the SVD:

$$U^{ij'}(\hat{\mathcal{K}}_{ij} - \hat{h}'_{i\cdot}\hat{h}'_{\cdot j})V^{ij} = \Sigma^{ij}.$$

Also, as discussed in §3.6, it might be beneficial to pre-compute some of these vectors, before finding cuts iteratively via SVD, although this is not something we decided upon exploring further.

The quadratic model derived for our problem is as follows:

$$\min \ \|H\|_1 \ ,$$

Linear equations on H:

$$AHA = A \quad \text{(P1)}, \ \textit{or the lighter version:} \ (\Sigma V')\, H\, (U\Sigma) \ = \ \Sigma \ ,$$

$$(AH)' = AH \qquad \text{(P3) (optional)} \ ,$$

$$(HA)' = HA \qquad \text{(P4) (optional)} \ ,$$

Lifting equations:

$$\left\langle A, \mathcal{K}_{ij} \right\rangle = h_{ij} \ , \quad \forall ij \in n \times m \ ,$$

$$\sum_{i=1}^{n} \sum_{k=1}^{m} a_{pi} a_{kq} (\mathcal{K}_{ij})_{k,l} = a_{pq} h_{\ell j} \quad \forall pq \in m \times n \ , \quad \forall \ell j \in n \times m \ ,$$

McCormick lifting inequalities:

$$\lambda_{ij} \le h_{ij} \le \mu_{ij} \ , \quad \forall ij \in n \times m \ ,$$

$$\left(\mathcal{K}_{ij}\right)_{k,\ell} - \lambda_{\ell j} h_{ik} - \lambda_{ik} h_{\ell j} + \lambda_{ik} \lambda_{\ell j} \ge 0 \ , \forall k\ell \in m \times n \ , \quad \forall ij \in n \times m \ ,$$

$$-\left(\mathcal{K}_{ij}\right)_{k,\ell} + \mu_{\ell j} h_{ik} + \lambda_{ik} h_{\ell j} - \lambda_{ik} \mu_{\ell j} \ge 0 \ , \forall k\ell \in m \times n \ , \ \ \forall ij \in n \times m \ ,$$

$$\left(\mathcal{K}_{ij}\right)_{k,\ell} - \mu_{\ell j} h_{ik} - \mu_{ik} h_{\ell j} + \mu_{ik} \mu_{\ell j} \ge 0 \ , \forall k\ell \in m \times n \ , \ \ \forall ij \in n \times m \ ,$$

Quadratic lifting inequalities; for various choices of u^{ij} and v^{ij}:

(note that the α_{pij} and β_{pij} depend on u^{ij} and v^{ij})

$$t_{1ij} := (u^{ij'} h'_{i\cdot} + v^{ij'} h_{\cdot j})/2, \quad \text{[substitute below]} \quad \forall ij \in n \times m \ ,$$

$$t_{2ij} := (u^{ij'} h'_{i\cdot} - v^{ij'} h_{\cdot j})/2, \quad \text{[substitute below]} \quad \forall ij \in n \times m \ ,$$

$$\left\langle \mathcal{K}_{ij}, u^{ij} v^{ij'} \right\rangle + w_{1ij} + t_{2ij}^2 \le 0 \ , \quad \text{[convex quadratic]} \quad \forall ij \in n \times m \ ,$$

$$-\left\langle \mathcal{K}_{ij}, u^{ij} v^{ij'} \right\rangle + t_{1ij}^2 + w_{2ij} \le 0 \ , \quad \text{[convex quadratic]} \quad \forall ij \in n \times m \ ,$$

$$-\left((t_{pij} - \alpha_{pij}) \, \frac{\beta_{pij}^2 - \alpha_{pij}^2}{\beta_{pij} - \alpha_{pij}} + \alpha_{pij}^2 \right) \le w_{pij} \ , \quad \text{[secant]} \quad \text{for } p = 1, 2 \ , \quad \forall ij \in n \times m \ ,$$

$$\alpha_{pij} \le t_{pij} \le \beta_{pij} \ , \quad \text{for } p = 1, 2 \ , \quad \forall ij \in n \times m \ .$$

To reduce to a model fully linear model, we can replace the convex quadratic terms $+t_{pij}^2$ with lower-bounding linearizations. That is, we can replace $+t_{pij}^2$ with

$$\eta_{pij}^2 + 2\eta_{pij}(t_{pij} - \eta_{pij}),$$

at one or more values $\eta_{pij} \in [\alpha_{pij}, \beta_{pij}]$ in the interval domain of t_{pij}. More specifically, we can consider substitutions of the form:

$$+t_{1ij}^2 \;\leftarrow\; \eta_{pij}^2 + 2\eta_{1ij}\left((u^{ij'}h'_{i\cdot} + v^{ij'}h_{\cdot j})/2 - \eta_{1ij}\right),$$

and

$$+t_{2ij}^2 \;\leftarrow\; \eta_{2ij}^2 + 2\eta_{2ij}\left((u^{ij'}h'_{i\cdot} - v^{ij'}h_{\cdot j})/2 - \eta_{2ij}\right).$$

In this manner, we can work with a linear rather than quadratic model.

3.11 Diving heuristic

Material from this section is to appear in [FFL19].

The mathematical-programming models that we have been considering in this chapter are rather heavy, and it is not practical to include all of the inequalities that we have introduced. Moreover, it may not even be desirable to include all of the inequalities. The inequalities relax (P2), but we probably do not want to fully enforce (P2). Instead, we understand that there is a trade-off to be made between sparsity, as measured by $\|H\|_1$, and satisfaction of (P2), as measured say by $\|H - HAH\|_{\mathrm{F}}$. In this section we propose a "diving" procedure for progressively enforcing (P2) while heuristically narrowing the domain of our feasible region.

Diving is well known as a key primal heuristic for mixed-integer linear op-

timization, in the context of branch-and-bound (see, for example, [Ber06], [Ber08], [Ach07], [DRL05], [EN07]). The idea is easy to implement within a mixed-integer linear-optimization solver that already has the infrastructure to carry out branch-and-bound. Iteratively, via a sequence of continuous relaxations, variables that should be integer in feasible solutions are heuristically fixed to integer values. This is a bit akin to "reduced-cost fixing" (for mixed-integer linear-optimization), where variables are fixed to bounds in a provably correct manner. Diving heuristics employ special (heuristic) branching rules, with the aim of tending towards (primal) feasibility and not towards a balanced subdivision of the problem (as many branching rules seek to do). That these heuristics "quickly go down" the branch-and-bound tree (in the sense of depth-first search) gives us the term *diving*. The heuristic is so important in the context of mixed-integer linear-optimization solvers that most of them, as a default, do a sequence of dives at the beginning of the solution process, so as to quickly obtain a good feasible solution which is very important for limiting the branching exploration. Applying this type of idea in continuous non-convex global optimization appears to be a fairly recent idea (see [GKL17]).

Our diving heuristic is closely related to this idea, but there is an important difference. Diving in the context of global optimization is aimed at hoping to get lucky and branch directly toward what will turn out to be a globally-optimal solution. Our context is different, our "target" that we will aim at is the MPP A^+. But our goal is not to get there; rather, our goal is find good solutions along the way that trade off sparsity against satisfaction of (P2).

We consider a diving procedure that iteratively increases the enforcement of property (P2), while heuristically localizing our search, inevitably showing its impact on the sparsity (approximately measured by $\|H\|_1$) of a

computed generalized inverse H.

The procedure is initialized with the solution of problem $\mathcal{P}$, where we minimize $\|H\|_1$ subject to: (P1), and any subset of (P3) and (P4), the lifting equations $\left\langle A, \mathcal{K}_{ij} \right\rangle = h_{ij}$, for all $ij \in n \times m$, and the McCormick lifting inequalities. We denote the solution of $\mathcal{P}$ by $(\hat{H}, \hat{\mathcal{K}})$. We define bounds for h_{ij} ($\lambda_{ij} \leq h_{ij} \leq \mu_{ij}$), such that $[\lambda_{ij}, \mu_{ij}]$ is the smallest interval that contains $\hat{h}_{ij}$ and A^+_{ij}. By including the current $\hat{H}$ in the box, we hope to remain localized to a region where there is a somewhat-sparse solution. By including the MPP A^+ in the box, we guaranteed that at every step we will have a feasible solution to our domain-restricted relaxation.

Next, for a fixed number of iterations, we consider the last solution ($\hat{H}$, $\hat{\mathcal{K}}$) of $\mathcal{P}$, and we append to $\mathcal{P}$ the following inequalities, for all ij such that $\hat{\mathcal{K}}_{ij} - \hat{h}'_{i\cdot}\hat{h}'_{\cdot j} \neq \mathbf{0}_{m \times n}$.

$$\begin{aligned}
&\left\langle \mathcal{K}_{ij}, u^{ij} v^{ij\prime} \right\rangle + w_{1ij} + t^2_{2ij} \leq 0, \\
&- \left(\left(t_{1ij} - \alpha_{1ij}\right) \frac{\beta^2_{1ij} - \alpha^2_{1ij}}{\beta_{1ij} - \alpha_{1ij}} + \alpha^2_{1ij} \right) \leq w_{1ij} \ . \\
&\alpha_{1ij} \leq t_{1ij} \leq \beta_{1ij},
\end{aligned}$$

where $u^{ij} \in \mathbb{R}^m$ and $v^{ij} \in \mathbb{R}^n$ are respectively, left- and right-singular vectors of $\hat{\mathcal{K}}_{ij} - \hat{h}'_{i\cdot}\hat{h}'_{\cdot j}$, corresponding to its largest singular value. This amounts to iteratively tightening violated non-convex quadratic equations via secant inequalities.

Finally, we execute our "diving procedure", where at each iteration, we select $ij \in n \times m$, and cut the interval $[\alpha_{1ij}, \beta_{1ij}]$ where a variable t_{1ij} varies into two parts. Between the two, the new interval is selected to contain A^+_{ij}. The branching point can be the midpoint of the interval, the current value of $\hat{h}_{ij}$, or a weighted combination of both. We select ij at each iteration

corresponding to the non-convex inequality

$$u^{ij'}(\mathcal{K}_{ij} - h'_{i\cdot}h'_{\cdot j})v^{ij} \leq 0,$$

that is most violated by the last solution computed for $\mathcal{P}$. We note that by reducing the size of the interval $[\alpha_{1ij}, \beta_{1ij}]$, we reduce the size of the interval where the secant of $-t_{1ij}^2$ is defined, leading to a new secant that better approximates the concave function, and therefore, strengthening the relaxation of (P2) on the new interval. The stopping criterion for the diving procedure is a given maximum violation ϵ for (P2), i.e., the algorithm stops when $\|HAH - H\|_{\mathrm{F}} \leq \epsilon$.

3.11.1 Weighted branching-point analysis and selection

In an effort to explore the potential of selecting different branching points, we tested a variety of weighted combinations to compare the impact of that selection on the rate of convergence to the MPP when (P1), and any subset of (P3) and (P4), are enforced. For a given solution H we measured the impact of the choice of weighted combination through observation in the change of objective function $||H||_1$ and the enforcement of property (P2). Recalling the notation from Chapter 1, let us again define $H134$ to denote a solution H that satisfies M-P properties (P1) + (P3) + (P4), with $H1$, $H13$, and $H14$ denoting solutions that satisfy (P1), (P1) + (P3), and (P1) + (P4), respectively. In this section all experiments conducted assume H (and $\hat{H}$) correspond to $H134$ if not written explicitly.

The data matrix $A \in \mathbb{R}^{10\times 5}$ was generated for this experiment with $\mathrm{rank}(A) = 3$. The entries of A are generated iteratively using vectors $r_k := -1+2x_m$ and $s_k = 1+2x_n$, with $x_m \in \mathbb{R}^m$, $x_n \in \mathbb{R}^n$, and $k = 1,...,\mathrm{rank}(A)$. The entries of x_m, x_n are drawn from the standard uniform distribution on

the open interval (0,1).

The goal of this experiment is two-fold: (1) to decide upon which weighted combination would be most effective in illustrating the tradeoff between the sparsity of a solution (approximated by $||H||_1$) and the violation (enforcement) of property (P2), and (2) whether to use the midpoint of $[\alpha, \beta]$ or the point $t_{MPP} \in [\alpha, \beta]$ using the definition of the intermediary variables t_{pij} in combination with the relaxation solution $\hat{H}$, would provide a "better" branching point selection for our proposed diving heuristic.

In the following two figures, we illustrate the results of five different weighted combinations of the midpoint (MID) of $[\alpha, \beta]$ and previous relaxation solution $\hat{H}$, shown as weighted pairs 25/75, 50/50, 75/25, 90/10, and 100/0. The midpoint of the interval $[\alpha_{1ij}, \beta_{1ij}]$ and corresponding t_{1ij} $:= (u^{ij'}\hat{h}'_i + v^{ij'}\hat{h}_j)/2$, are both derived from the relaxation solution $\hat{H}$ from the previous dive iterate. Figure 3.1 provides a comparison in the rates of increase of the objective value $||H||_1$, while Figure 3.2 provides a comparison in the rate of reduction of the violation of (P2).

Figures 3.3 and 3.4 illustrate the results from testing a similar collection of weighted combinations, but instead, for selected interval $[\alpha_{1ij}, \beta_{ij}]$, we use the corresponding $t_{1ij} := (u^{ij'}\hat{h}'_i + v^{ij'}\hat{h}_j)/2$ and $t_{MPP} := (u^{ij'}(A^+)'_i + v^{ij'}(A^+)_j)/2$, with t_{MPP} derived from the always feasible MPP, denoted as A^+.

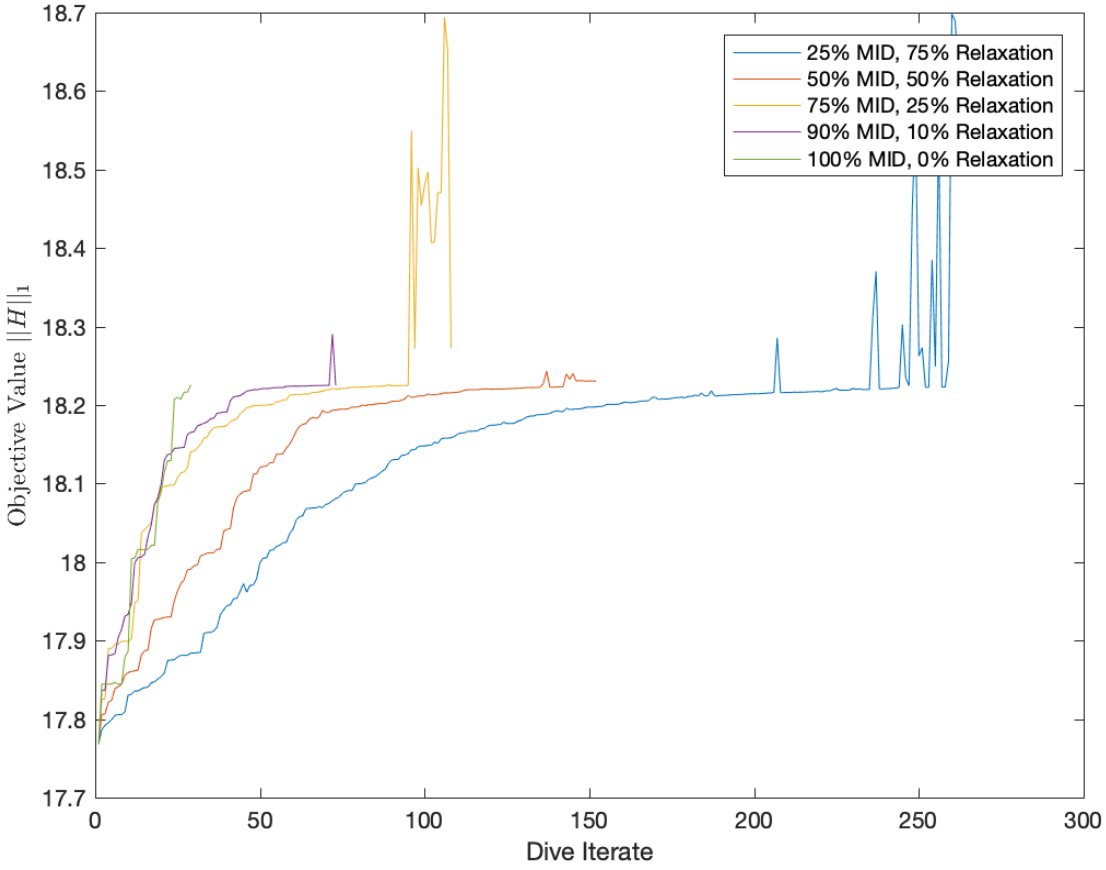


Figure 3.1: Impact on $||H||_1$ using the midpoint of $[\alpha, \beta]$

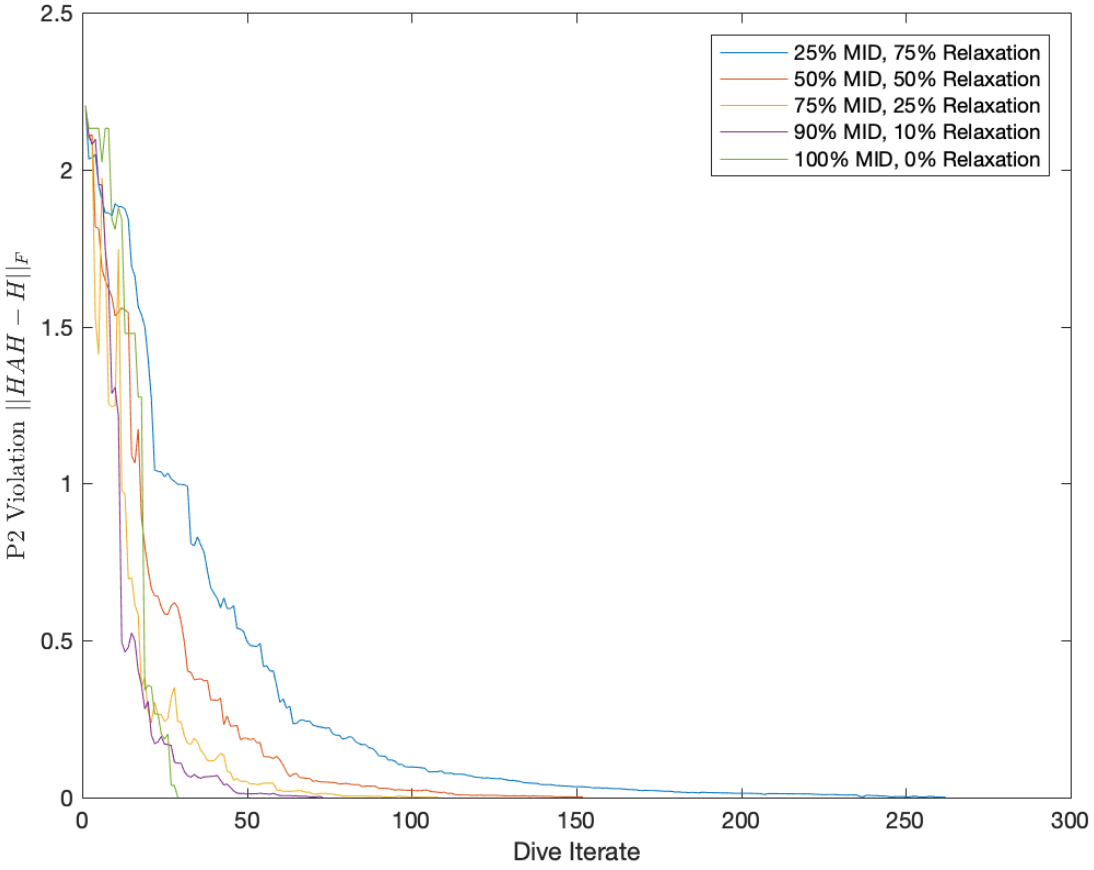


Figure 3.2: Impact on (P2) viol. of $H134$ using the midpoint of $[\alpha, \beta]$

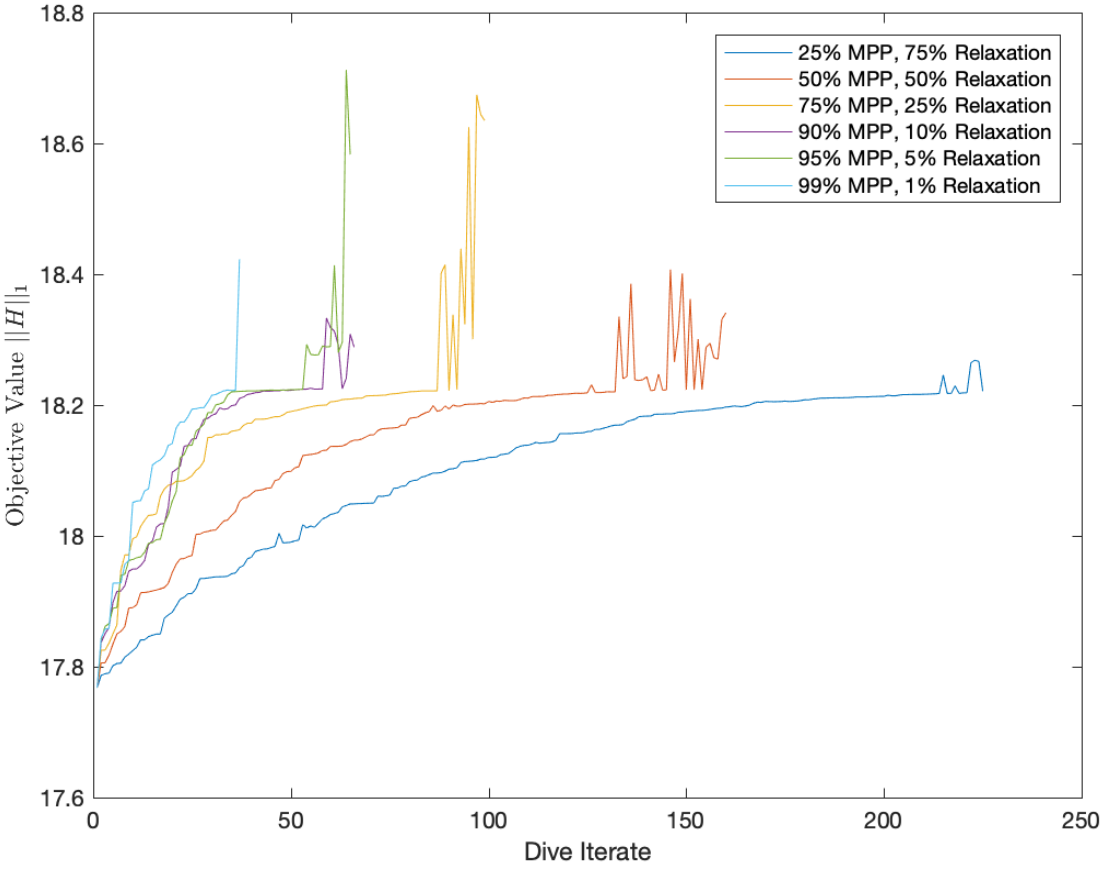


Figure 3.3: Diving impact on $||H||_1$ using $t_{MPP} \in [\alpha, \beta]$

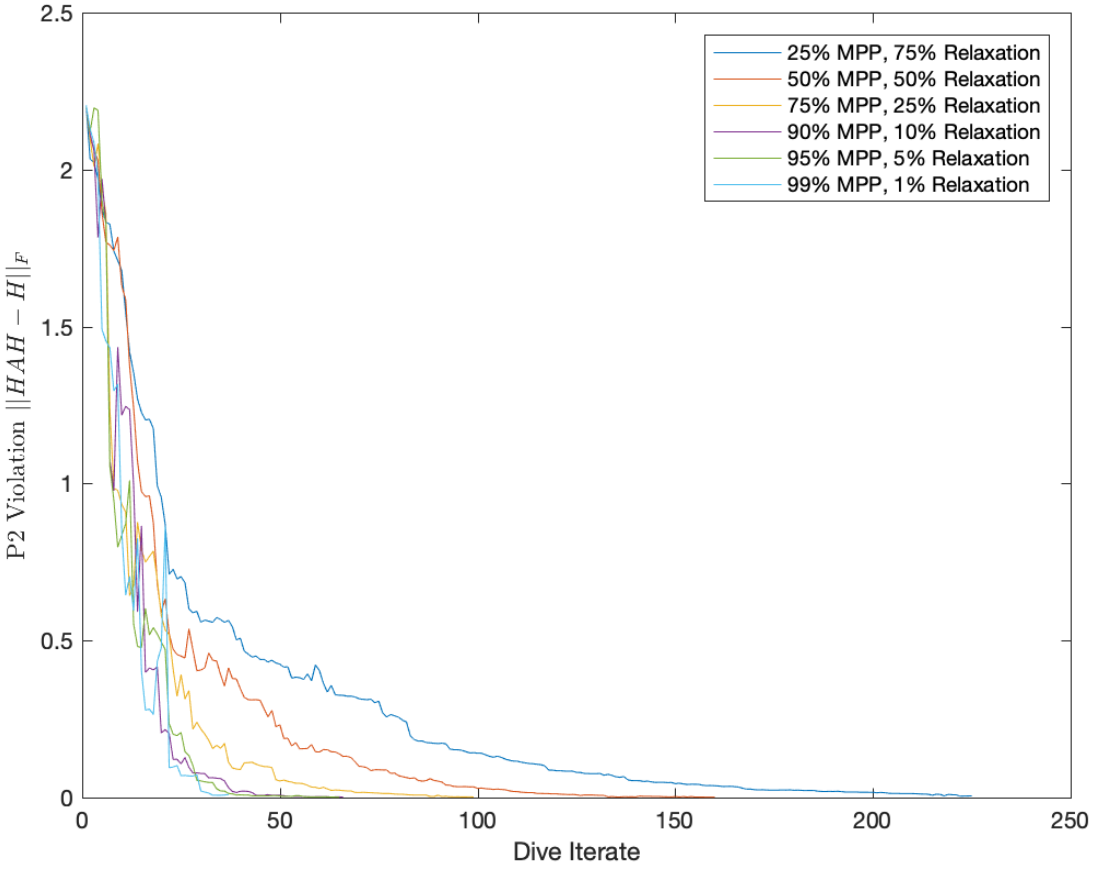


Figure 3.4: Impact on (P2) viol. of $H134$ using $t_{MPP} \in [\alpha, \beta]$

Figure 3.5 illustrates the relationship between the change in the violation of (P2) and the change in $||H||_1$ via a scatterplot representation of the diving heuristic for various weighted combinations used in the branching point selection process.

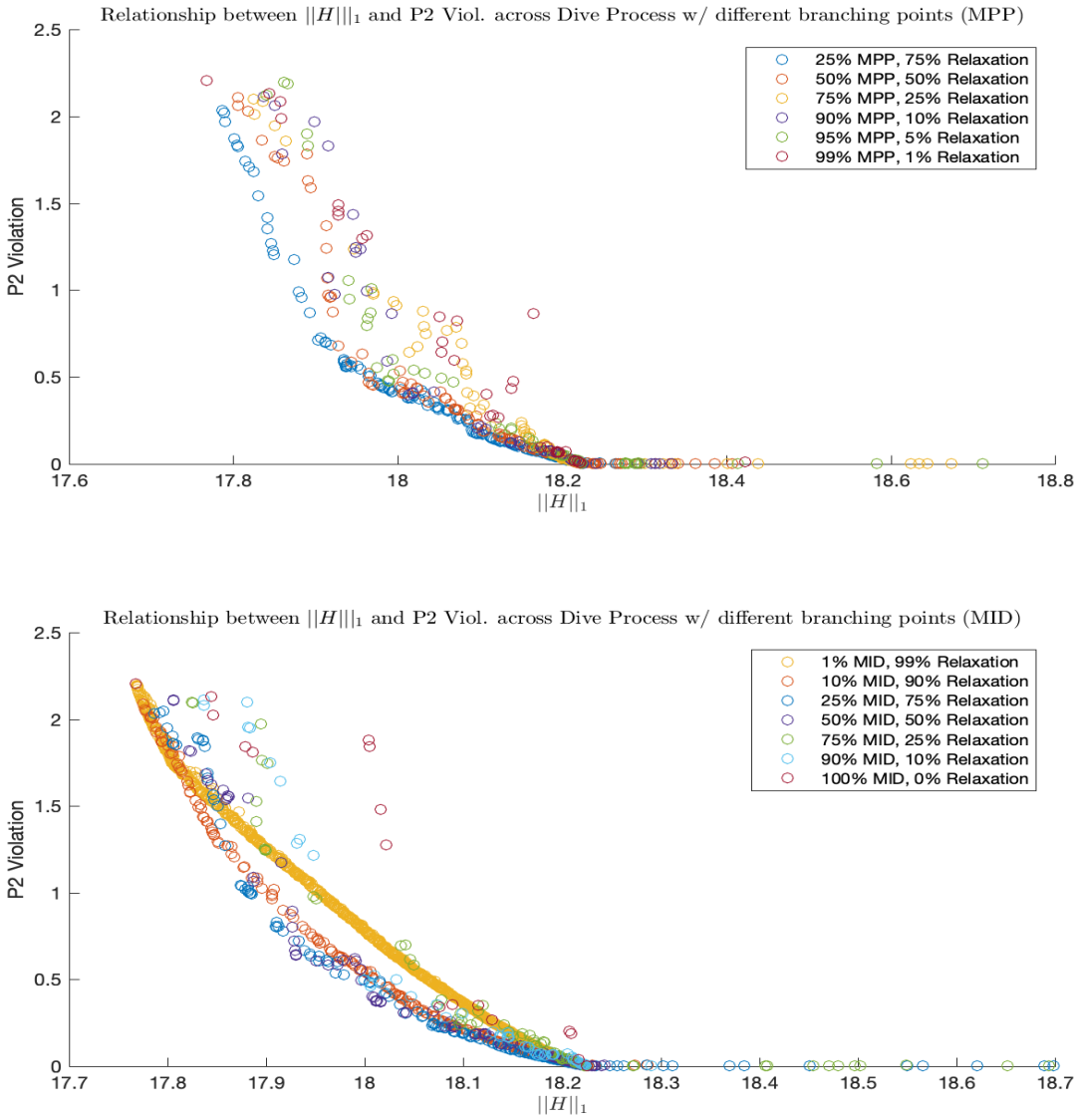


Figure 3.5: Tradeoff between $||H||_1$ and (P2) violation of $H134$

With regards to our first goal, from the selection of weighted combinations we found that using a 25/75 weighting (25% midpoint/MPP, 75% relaxation) roughly defines the pareto frontier with regards to the tradeoff in the satisfaction of (P2) and objective value $||H||_1$, as can be seen in Figure 3.5. In the few extended weighted combinations we tested using the midpoint, 10/90 and 1/99, although these weighted pairs offer a more comprehensive visualization of relationship between $||H||_1$ and violation of (P2), they come at a much larger cost with regards to computation time.

Interestingly, 25/75 is the default weighting for the spatial branch-and-bound software ANTIGONE, and a similar 30/70 for BARON (see [SL18]). Of course, it should be noted that they are doing actual spatial branch-and-bound software, while our diving heuristic is only inspired by the idea of spatial branch-and-bound software.

In hopes of addressing our second goal, Figure 3.6 presents a comparison of the impact in choosing the midpoint versus the MPP when using the 25/75 weighted combination. Both options generate similar tradeoffs, with the use of the midpoint resulting in a slightly larger increase in $||H||_1$ with similar enforcement of property (P2). Given the similarity, we are unable to definitively say that the selection of one is better than the other given our goal to find a branching point weighted combination that best illustrates the tradeoff between solution sparsity and (P2) enforcement. Given these findings, in the following sections we make an arbitrary designation to use the midpoint rather than the MPP.

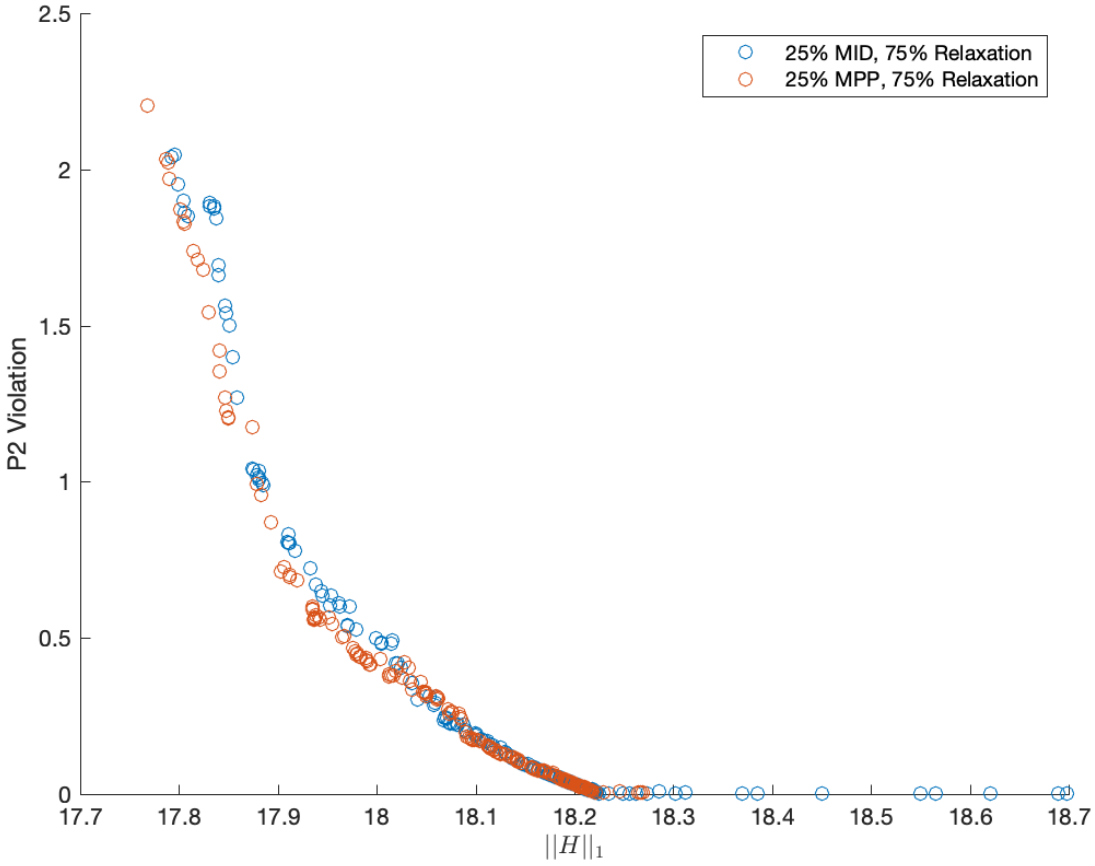


Figure 3.6: Comparison of 25/75 weighting using MPP and MID

3.11.2 Impact of diving heuristic on combinations of (P3) and (P4)

Given that the initial experiment explores the impact of the diving heuristic when enforcing all of the linear Moore-Penrose properties (P1), (P3), and (P4), we consider the tradeoffs between $||H||_1$ and the violation of (P2) when

enforcing subsets of the full set of linear M-P properties. In particular we seek to explore how the iterative strengthening of the relaxation of (P2) via the diving heuristic may change the tradeoff when starting with solutions that satisfy (P1) and a proper subset of (P3) and (P4).

For each of these tests we use the same data matrix A, with the only difference being in which of the linear M-P properties are enforced (along with the inclusion of box constraints and the McCormick lifting inequalities).

As a means of quick comparison, Figures 3.7, 3.8, and 3.9 use weighted combinations 25/75 and 50/50 for the branching points (as well as using the midpoint). When only enforcing (P1), we see that 25/75 weighting provides a thorough visualization of the tradeoff between $||H||_1$ and the violation of (P2). When enforcing (P1) in combination with either (P3) or (P4), the impact of the weighted combinations considered is less conclusive, with Figure 3.8 exhibiting a more consistent tradeoff trend while Figure 3.9 illustrates a considerable amount of volatility in the relationship between $||H||_1$ and (P2) violation.

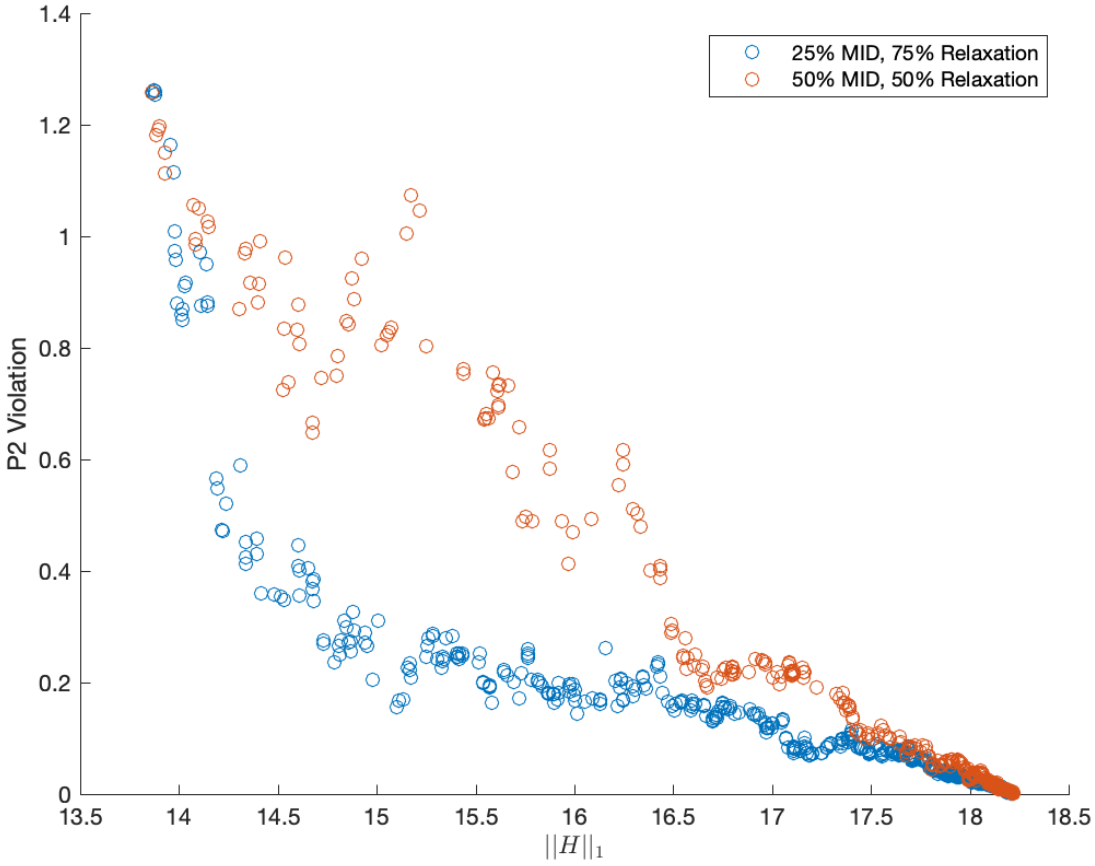


Figure 3.7: Obj. val. vs. (P2) violation of $H1$

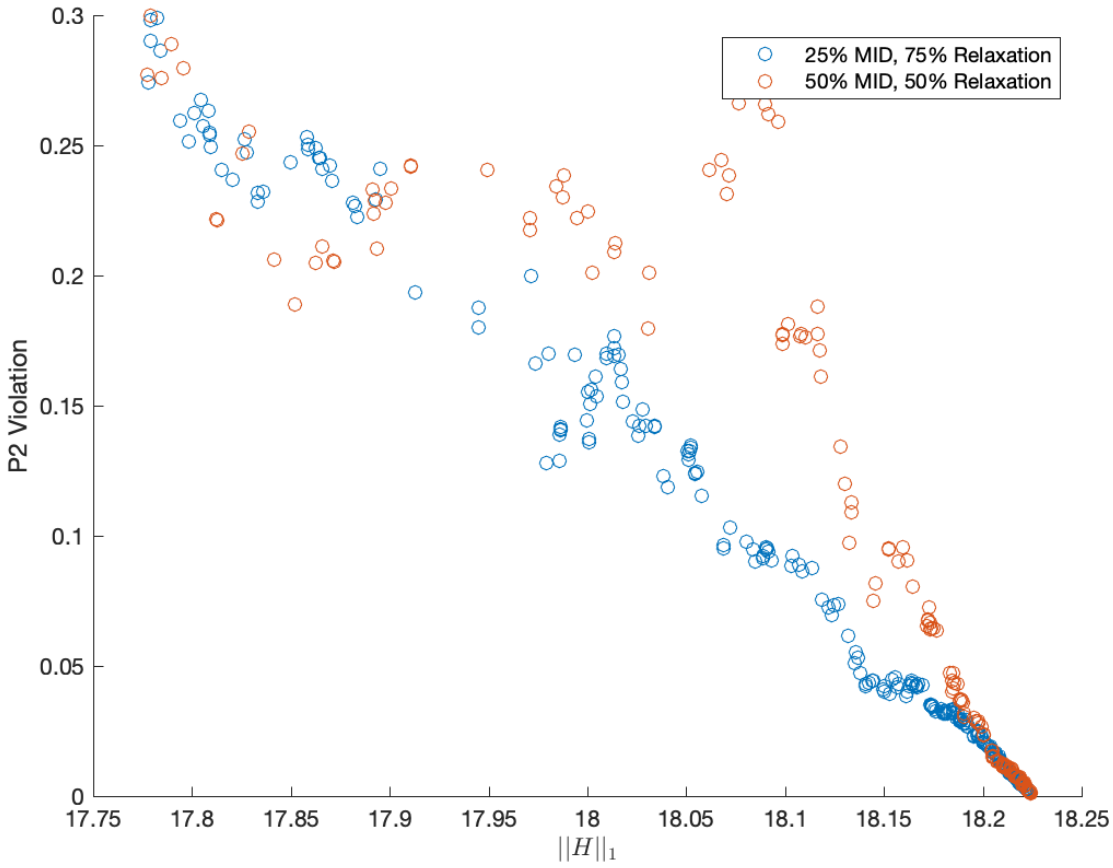


Figure 3.8: Obj. val. vs. (P2) violation of $H13$

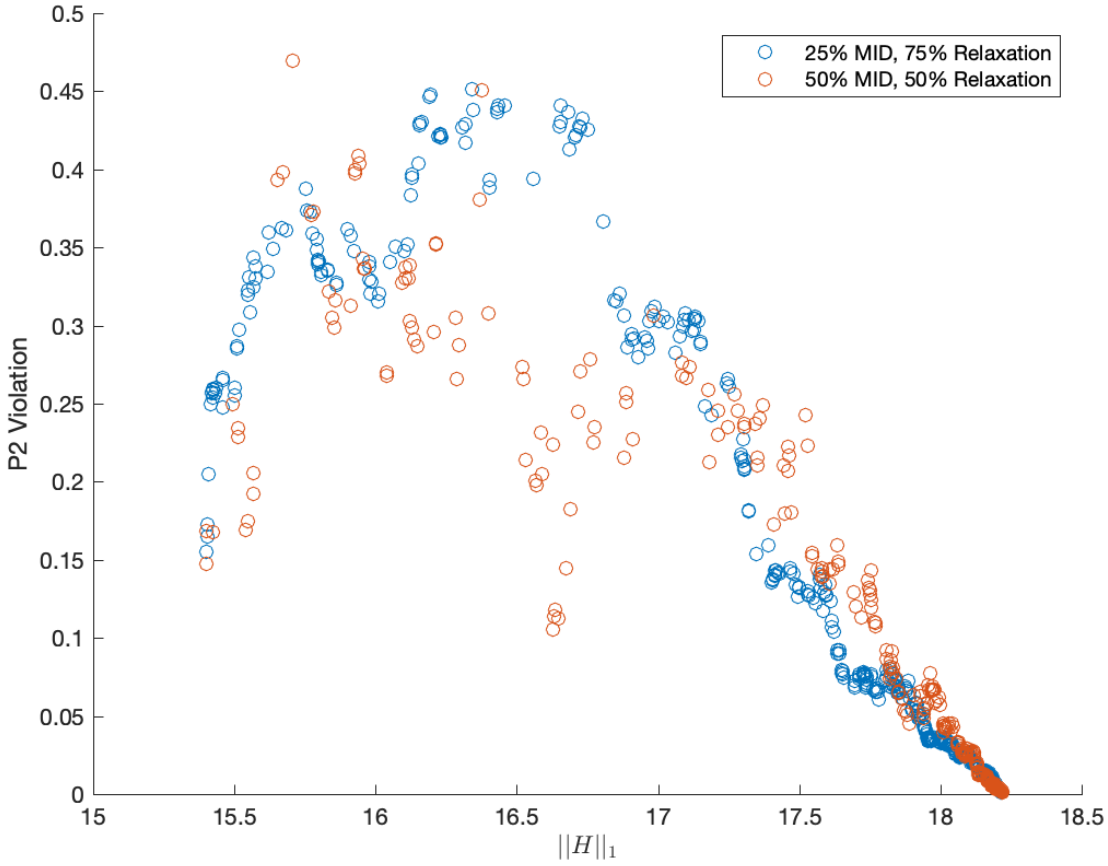


Figure 3.9: Obj. val. vs. (P2) violation of $H14$

3.11.3 Changes in solution norm ratios

As seen in Chapter 2 with Figures 2.2 and 2.3, $H13$ and $H14$ represented solutions that satisfied combinations of the linear M-P properties resulting in a least-squares or 2-norm solution minimizer, with $H134$ representing a solution that generates a minimizer for both types of problems. In particu-

lar, we defined least-squares and 2-norm ratios to measure the quality of a sparse generalized inverse against the true MPP. Solutions H that satisfy M-P properties (P1) and (P3) resulted in a least-squares ratio equal to one and 2-norm ratio greater than one, signifying a solution of lesser quality than the MPP. Similarly, solutions H that satisfy the M-P properties (P1) and (P4) resulted in a 2-norm ratio equal to one and least-squares ratio greater than one (again signifying a solution of lesser quality than the MPP with respect to the least-squares problem). Therefore, a natural extension would be to explore the impact of our relaxation of (P2), along with the subsequent strengthening via our diving heuristic, in relation to the least-squares and 2-norm ratios when considering solutions $H13$ and $H14$.

The experiment focuses on two weightings for the branching point selection using the midpoint, using weighted combinations of 25/75 and 50/50 to provide a preliminary comparison.

For the setup of this experiment, the box constraints used in the relaxation (denoted by δ) are defined as the maximal entry-wise difference between the MPP of A (denoted as A^+) and given solution H

Figure 3.10 illustrates the impact of the 25/75 weight on the 2-norm solution ratio for $H13$, where within the first 50 dive iterates, there is a reduction in the ratio from 15% to 1%. Conversely, the 50/50 weight exhibits a large amount of volatility in the 2-norm solution ratio, reaching values close 25%, before settling close to a ratio of one after approximately 60 dive iterates.

Moving to Figure 3.11, the impact of the branching point weighted combinations appears to be much more pronounced with respect to the least-squares solution norm, with the 25/75 weighted pair exhibiting a much more gradual reduction in the ratio and the 50/50 weighted pair again appearing to generate some volatility in the norm solution ratio (although ultimately

settling close to a ratio of one).

With these preliminary observations, much more exploration and testing will be necessary to get a sense of how our relaxation of (P2), along with the proposed strengthening method, might prove useful in understanding the relationship between maintaining the relative sparsity of initial solutions like $H13$ and $H14$ while also improving upon their 2-norm and least-squares ratios, respectively.

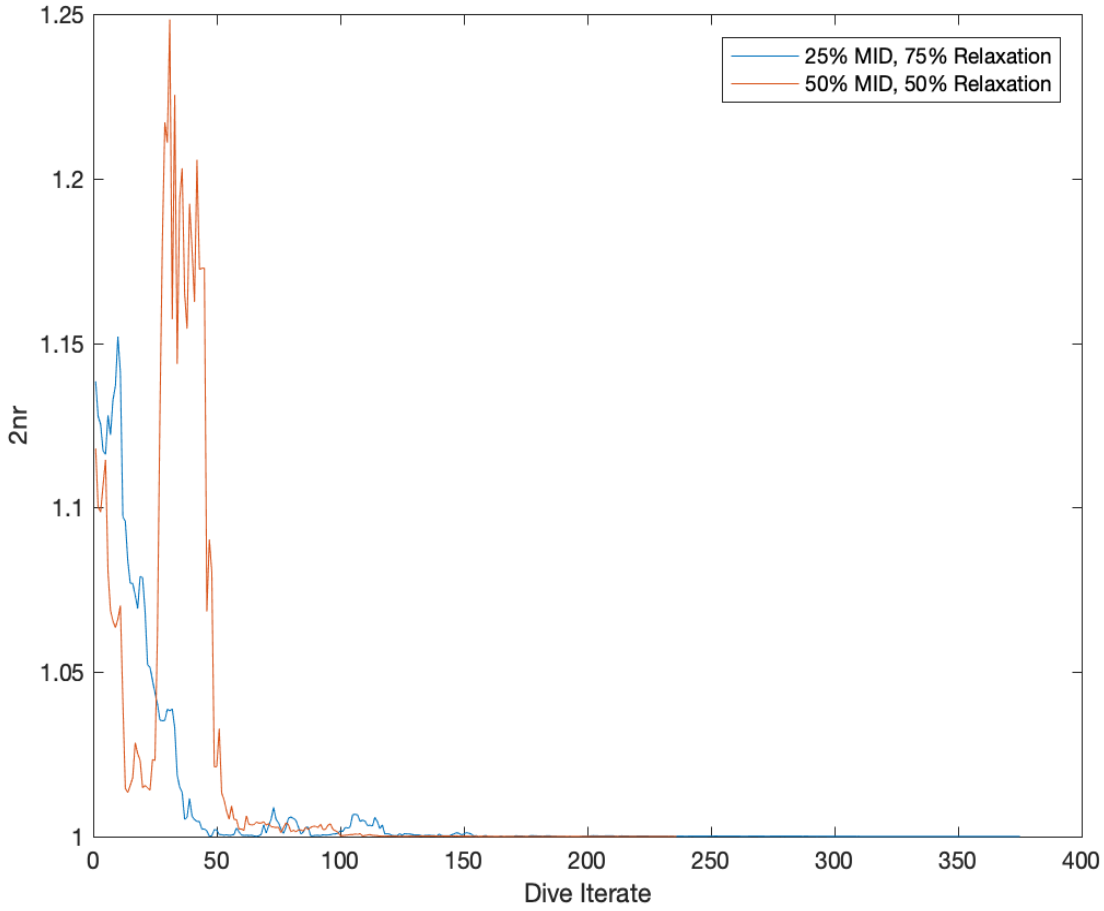


Figure 3.10: 2-norm solution ratios: $H13$ vs. MPP

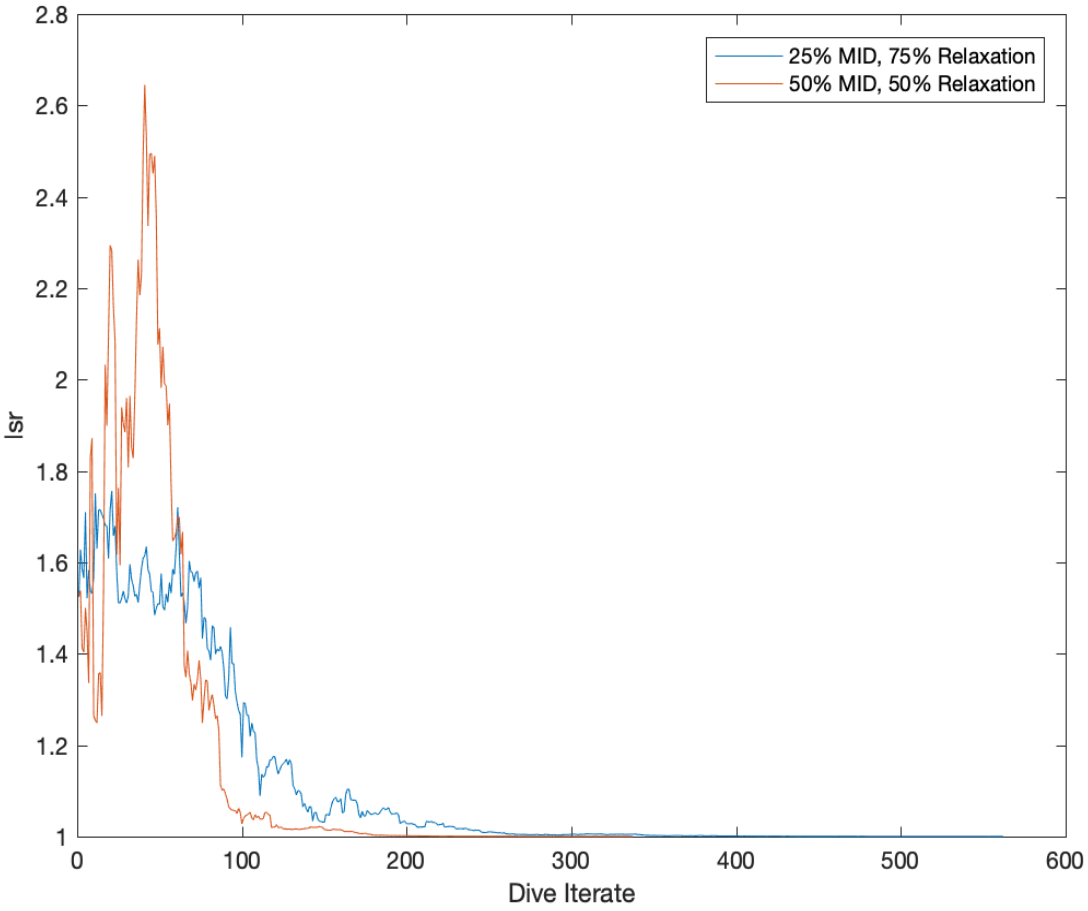


Figure 3.11: Least-squares norm solution ratios: $H14$ **vs.** MPP

3.12 Conclusions and further questions

We have provided convex settings to approximate (P2), such as convex semi-definiteness constraints, reformulation linearization techniques, and convex

quadratic/secant lifting inequalities. Of the various methods considered, such as McCormick/RLT and the quadratic/secant lifting inequalities, we implemented a diving heuristic to iteratively strengthen the relaxations, generating a sequence of solutions which allowed for an illustration of the tradeoff between solution sparsity and violation of (P2) with respect to the Frobenius norm. The choice of branching-point weight selection provides a collection of solutions converging to the MPP, and where one wants to be on this spectrum would be dependent on the application in question.

We note that our experiments were focused on the change in sparsity as we iteratively strengthened the approximations of (P2), but not (directly) on the rank, even though we used singular value information to construct the cuts, determine which secant cuts to refine in the diving heuristic, and as a means to measure violation of (P2). Further exploration into how the rank of a given solution H changes as violation of (P2) decreases, and how to define that in a numerical setting, would be a natural extension.

Although not emphasized in this chapter, some of the greatest limitations are due to scalability and/or numerical stability; so further work would require a more balanced, system-specific approach with regards to the formulation and implementation of these models. In particular, given the heavy nature of the constraint sets considered in all of these models, it would be an interesting next step to focus on methods to reduce the number of constraints in a structured and systematic way, as well as compare against some of the semi-definite programming (SDP) formulations, as they provide a more natural framework for working with rank.

Chapter 4

Sparse-inverse/low-rank Decomposition via Woodbury

This chapter is based on [FFL16a]

4.1 Introduction

We wish to consider general matrix decomposition problems of the form:

$$\begin{aligned} \underset{A_1 \dots A_n}{\text{minimize}} \quad & \sum_{k=1}^{N} \bar{\tau}_k \|\sigma(\phi_k(A_k))\|_0 \\ \text{subject to} \quad & \sum_{k=1}^{N} A_k = \bar{C}, \end{aligned} \tag{4.1}$$

with real input data $\bar{C} \in \mathbb{R}^{n \times n}$, $\bar{\tau}_k > 0$, $\sum_{k=1}^{N} \bar{\tau}_k = 1$, and $\phi_k : \mathbb{R}^{n \times n} \to \mathbb{R}^{n_k \times n_k}$ given, where each n_k is specific to ϕ_k, for $k = 1, 2, \dots, N$, $\sigma(\cdot)$ denotes the vector of singular values, and $\|\cdot\|_0$ counts the number of nonzeros. If ϕ_k is the identity map, then $\|\sigma(\phi_k(A_k))\|_0 = \text{rank}(A_k)$. Let $\text{dvec}(A_k) \in \mathbb{R}^{n^2 \times n^2}$ be

a diagonal matrix with the n^2 components of A_k on the diagonal. If $\phi_k(A_k)$ = dvec(A_k), then $\|\sigma(\phi_k(A_k))\|_0 = \|A_k\|_0$ — the sparsity of A_k. Indeed, in this manner, we can see as a special case the well-known rank-sparsity decomposition problem [Cha+11]:

$$\min \left\{ \bar{\tau}\|A\|_0 + (1-\bar{\tau})\text{rank}(B) \ : \ A + B = \bar{C} \right\}, \tag{$\mathcal{D}_0$}$$

We are particularly interested in situations in which some of the ϕ_k are nonlinear, and we initially focus our attention on $\phi_1(A_1) = \text{dvec}(A^{-1})$, and ϕ_2 the identity map, arriving at

$$\begin{array}{ll} \text{minimize} & \bar{\tau}\|A^{-1}\|_0 + (1-\bar{\tau})\text{rank}(B) \\ \text{subject to} & A + B = \bar{C}, \end{array} \tag{$\mathcal{P}$}$$

where $0 < \bar{\tau} < 1$.

The estimation of a covariance matrix is an important problem in many areas of statistical analysis. For a covariance matrix $\bar{A}$, its inverse $\bar{H}$ is called a *precision matrix* or a *concentration matrix*, but note that the definition can vary (see [CLL11], for example). If $\bar{A}$ is the covariance matrix of Gaussian random variables $X_1, X_2, \ldots, X_n$, then the interpretation of $\bar{h}_{ij} = 0$ is that X_i and X_j are conditionally independent (conditioning on the other $n-2$ random variables); see appendix (A.1) for details. It is common to consider the situation where the precision matrix $\bar{H} = \bar{A}^{-1}$ has an unknown, but sparse structure (see [CLL11], [FHT08], and [Hsi+11]). We can reasonably consider the situation in which we observe data which we summarize via the sample covariance matrix $\bar{C}$ as the sum of a true covariance matrix $\bar{A}$, with the assumption that $\bar{H} = \bar{A}^{-1}$ has unknown but sparse structure, and $\bar{A}$ is obscured by low rank noise $\bar{B}$. Our goal is to recover $\bar{A}$ (and $\bar{B}$). We note that in the literature, it is $\bar{H}$ that is assumed to be subject to the low

rank noise.

A closely related viewpoint is that of recovery, where $\bar{A}$ and $\bar{B}$ are known, with our goal being to recover these exactly, or with some established tolerable range [Cha+11]. To establish the recovery framework, we have input matrices $\bar{H}$ and $\bar{B}$, where $\bar{H}$ is sparse and $\bar{B}$ has low rank. We define $\bar{A} := \bar{H}^{-1}$, and $\bar{C} := \bar{A} + \bar{B}$. In the context of recovery, we apply some optimization method with $\bar{C}$ as the input matrix. The outputs are values A and B, where from A we define $H := A^{-1}$. From this output, we are interested in, for example, whether

1) H is sparse and, by some appropriate measure (e.g. Frobenius norm), is close to $\bar{H}$,

2) B is low rank and, by some appropriate measure (e.g. comparing the singular values and singular vectors), is close to $\bar{B}$.

4.1.1 Convex approximation

The problem $\mathcal{D}_0$ is ordinarily approached by using the (element-wise) 1-norm as an approximation of $\|\cdot\|_0$ and using the nuclear norm (sum of the singular values) as an approximation of rank. So we are led to the approximation

$$\min \left\{ \bar{\tau}\|A\|_1 + (1-\bar{\tau})\|B\|_* \right) \ : \ A + B = \bar{C} \right\}, \qquad (\mathcal{D}_1)$$

where $\|A\|_1 := \sum_{i,j} |a_{ij}|$ and $\|B\|_*$ denotes the sum of the singular values of B. This approach has some very nice features. First of all, because we have genuine norms now in the objective function, this approximation $\mathcal{D}_1$ is a convex optimization problem, and so we can focus our attention on seeking a local optimum of $\mathcal{D}_1$ which will then be a global optimum of $\mathcal{D}_1$. Still,

the objective function of $\mathcal{D}_1$ is not differentiable everywhere, and so we are not really out of the woods. However, the approximation $\mathcal{D}_1$ can be re-cast (see [Cha+11, Appendix A]) as a semidefinite-optimization problem

$$\begin{aligned} &\min\ \bar{\tau}\mathbf{e}'S\mathbf{e} + (1-\bar{\tau})\frac{1}{2}\left(\mathrm{tr}(W_1)+\mathrm{tr}(W_2)\right) \\ &A+B=\bar{C}, \qquad -S\le A\le S, \qquad \begin{pmatrix} W_1 & B \\ B' & W_2 \end{pmatrix} \succeq 0, \end{aligned}$$

which is efficiently solvable in principle (see [WSV00]). We note that semidefinite-optimization algorithms are not at this point very scalable. Nonetheless, there are first-order methods for this problem that do scale well and allow us to quickly get good approximate solutions for large instances (see [Boy+10]).

Also, it is interesting to note that to solve $\mathcal{D}_0$ globally (with additional natural constraints bounding the feasible region), a genuine relaxation of $\mathcal{D}_0$ closely related to $\mathcal{D}_1$ should be employed (see [LZ14]).

4.1.2 Generating test problems via the recovery theory

Another extremely nice feature of the convex approximation $\mathcal{D}_1$ is a "recovery theory". Loosely speaking it says the following: If we start with a sparse matrix $\bar{A}$ that does not have low rank, and a low-rank matrix $\bar{B}$ that is not sparse, then there is a non-empty interval $\mathcal{J} := [\bar{\tau}_\ell, \bar{\tau}_u] \subset [0,1]$ so that for all $\bar{\tau} \in \mathcal{J}$, the solution of the approximation $\mathcal{D}_1$ is uniquely $A = \bar{A}$ and $B = \bar{B}$.

The recovery theory suggests a practical paradigm for generating test problems for algorithms for $\mathcal{D}_1$.

Procedure 1

1. Generate a random sparse matrix $\bar{A}$ that with high probability will not have low rank. For example, for some natural number $\ell \ll \min\{m, n\}$, randomly choose $\ell \cdot \min\{m, n\}$ entries of $\bar{A}$ to be non-zero, and give those entries values independently chosen from some continuous distribution.

2. Generate a random low-rank matrix $\bar{B}$ that with high probability will not be sparse. For example, for some natural number $k \ll \min\{m, n\}$, make an $m \times k$ matrix $\bar{U}$ and a $k \times n$ matrix $\bar{V}$, with entries chosen independently from some continuous distribution, and let $\bar{B} := \bar{U}\bar{V}$.

3. Let $\bar{C} := \bar{A} + \bar{B}$.

4. Perform a search on $[0, 1]$ to find a value $\bar{\tau}^*$ so that the solution of $\mathcal{D}_1$ with $\bar{\tau} = \bar{\tau}^*$ is $A = \bar{A}$ and $B = \bar{B}$.

5. Output: $\bar{C}$, $\bar{\tau}^*$.

The recovery theory tells us that there will be a value of $\bar{\tau}^*$ for which the solution of $\mathcal{D}_1$ with $\bar{\tau} = \bar{\tau}^*$ is $A = \bar{A}$ and $B = \bar{B}$. What is not completely clear is that there is a disciplined manner of searching for such a $\bar{\tau}^*$ (step 4). Let $A_{\bar{\tau}}, B_{\bar{\tau}}$ be a solution of $\mathcal{D}_1$, with the notation emphasizing the dependence on $\bar{\tau}$. We define the univariate function

$$f(\bar{\tau}) := \|\bar{A} - A_{\bar{\tau}}\|_F = \|\bar{B} - B_{\bar{\tau}}\|_F = \frac{1}{2}\left(\|\bar{A} - A_{\bar{\tau}}\|_F + \|\bar{B} - B_{\bar{\tau}}\|_F\right).$$

Clearly, for $\bar{\tau} = 0$, the solution of $\mathcal{D}_1$ will be $B = 0$, $A = \bar{C}$ and $f(0) = \|\bar{B}\|_F$. Likewise, for $\bar{\tau} = 1$, the solution of $\mathcal{D}_1$ will be $A = 0$, $B = \bar{C}$ and $f(1) = \|\bar{A}\|_F$. For $\bar{\tau}^* \in \mathcal{I}$, f is minimized with $f(\bar{\tau}^*) = 0$, And we can hope that f is quasiconvex and we may quickly find a minimizer via a bisection search.

4.2 Sparse-inverse/low-rank decomposition

Now, we turn our attention to a closely related problem — which is our main focus in the present chapter. We assume now that $\bar{G}$ is an order-n *square* input matrix, $0 < \bar{\tau} < 1$, and our goal is to solve the Sparse-*inverse*/low-rank decomposition problem:

$$\min \left\{ \bar{\tau} \|E^{-1}\|_0 + (1-\bar{\tau}) \mathrm{rank}(F) \ : \ E + F = \bar{G} \right\}. \qquad (\mathcal{H}_0)$$

Note that, generally, it may be that $\bar{G}$ is not invertible, but in the approach that we present here, we will assume that $\bar{G}$ is invertible.

The problem $\mathcal{H}_0$ can capture an interesting problem in statistics. In that setting, $\bar{G}$ can be a sample covariance matrix. Then E can be the true covariance matrix that we wish to recover. In some settings, the inverse of E (known as the *precision matrix*) can be of unknown sparse structure — a zero entry in the inverse of E identifies when a pair of variables are conditionally (on the other $n-2$ variables) independent. We do note that for this application, because the sample covariance matrix and the true covariance matrix are positive semidefinite, there are alternative approaches, based on convex approximations, that are very attractive (see [SMG10] and the references therein). So our approach can best be seen as having its main strength for applications in which $\bar{G}$ is not positive semidefinite.

4.2.1 An algorithmic approach via the Woodbury identity

The algorithmic approach that we take is as follows.

Procedure 2

1. Let $\bar{C} := \bar{G}^{-1}$.

2. Apply *any* approximation method for $\mathcal{D}_0$, yielding some A (and B). For example, we can solve $\mathcal{D}_1$.

3. Output $E := A^{-1}$ and $F := \bar{G} - E$.

Our methodology is justified by the *Woodbury matrix identity* (see [Hag89]). In step 2, we find a decomposition $A + B = \bar{G}^{-1}$, with A sparse and B low rank. Now, suppose that $\text{rank}(B) = k$. Then it can be written as $B = UV$, where U is $n \times k$ and V is $k \times n$. By the Woodbury identity, we have

$$\bar{G} = \bar{C}^{-1} = (A+B)^{-1} = (A+UV)^{-1} = A^{-1} - A^{-1}U(I+VA^{-1}U)^{-1}VA^{-1}.$$

Because A is sparse, we have that E^{-1} is sparse. Finally, we have $F = -A^{-1}U(I + VA^{-1}U)^{-1}VA^{-1}$ which has rank no more than k.

4.2.2 Generating test problems without a recovery theory

We could try to work with the approximation

$$\min \left\{ \bar{\tau}\|E^{-1}\|_1 + (1-\bar{\tau})\|F\|_* \right) \ : \ E + F = \bar{G} \right\} \qquad (\mathcal{H}_1)$$

of $\mathcal{H}_0$, but $\mathcal{H}_1$ is not a convex optimization problem, and there is no direct recovery theory for it. But we can exploit the correspondence (via the Woodbury identity) with $\mathcal{D}_1$ to generate test problems for the non-convex problem $\mathcal{H}_1$. In analogy with Procedure 1 of §4.1.2, we employ the following methodology, which incorporates our heuristic Procedure 2.

Procedure 3

1. Generate a random sparse square invertible matrix $\bar{A}$. This may have to be done with a few trials to ensure that $\bar{A}$ is invertible. Let $\bar{E} := \bar{A}^{-1}$.

2. Generate a random low-rank square matrix $\bar{B} := \bar{U}\bar{V}$ that with high probability is not sparse, as described in step 2 of Procedure 1.

3. Let $\bar{F} := -\bar{A}^{-1}\bar{U}(I + \bar{V}\bar{A}^{-1}\bar{U})^{-1}\bar{V}\bar{A}^{-1}$, and let $\bar{G} := \bar{E} + \bar{F}$.

4. Let $\bar{C} := \bar{G}^{-1}$. Search on $[0, 1]$ to find a $\bar{\tau}^*$ seeking to minimize $f(\bar{\tau}) := \|\bar{A} - A_{\bar{\tau}}\|_F = \|\bar{B} - B_{\bar{\tau}}\|_F$.

5. Output: $\bar{G}$, $\bar{\tau}^*$.

Because of the way we have engineered $\bar{F}$ in Procedure 3, we take advantage of the ordinary recovery theory for $\mathcal{D}_1$.

4.3 Computational experiments

We carried out some preliminary computational experiments for Procedure 3, using $n = 75$. We did six experiments, each with the rank of $\bar{B}$ at $k = 3, 6, 9, 12, 15, 18$. For each value of k, we chose $\bar{A}^{-1}$ to have $(k + 1)n$ non-zeros. So, as k increases, the rank of $\bar{B}$ is increasing and the number of non-zeros in $\bar{A}^{-1}$ is increasing. Therefore, we can expect that as k increases, the "window of recovery" (i.e., the set of $\bar{\tau}$ so that $f(\bar{\tau}) = 0$) gets smaller and perhaps vanishes; and once it vanishes, we can expect that the minimum value of $f(\bar{\tau})$ is increasing with k. We can see that this is all borne out in Fig. 4.1. Next, we focus on the $k = 15$ case, where the minimum of $f(\bar{\tau})$ is substantially above 0. Even in such a case, we can see in Fig. 4.2 that there is substantial recovery, attesting to the efficacy of our heuristic Procedure 2.

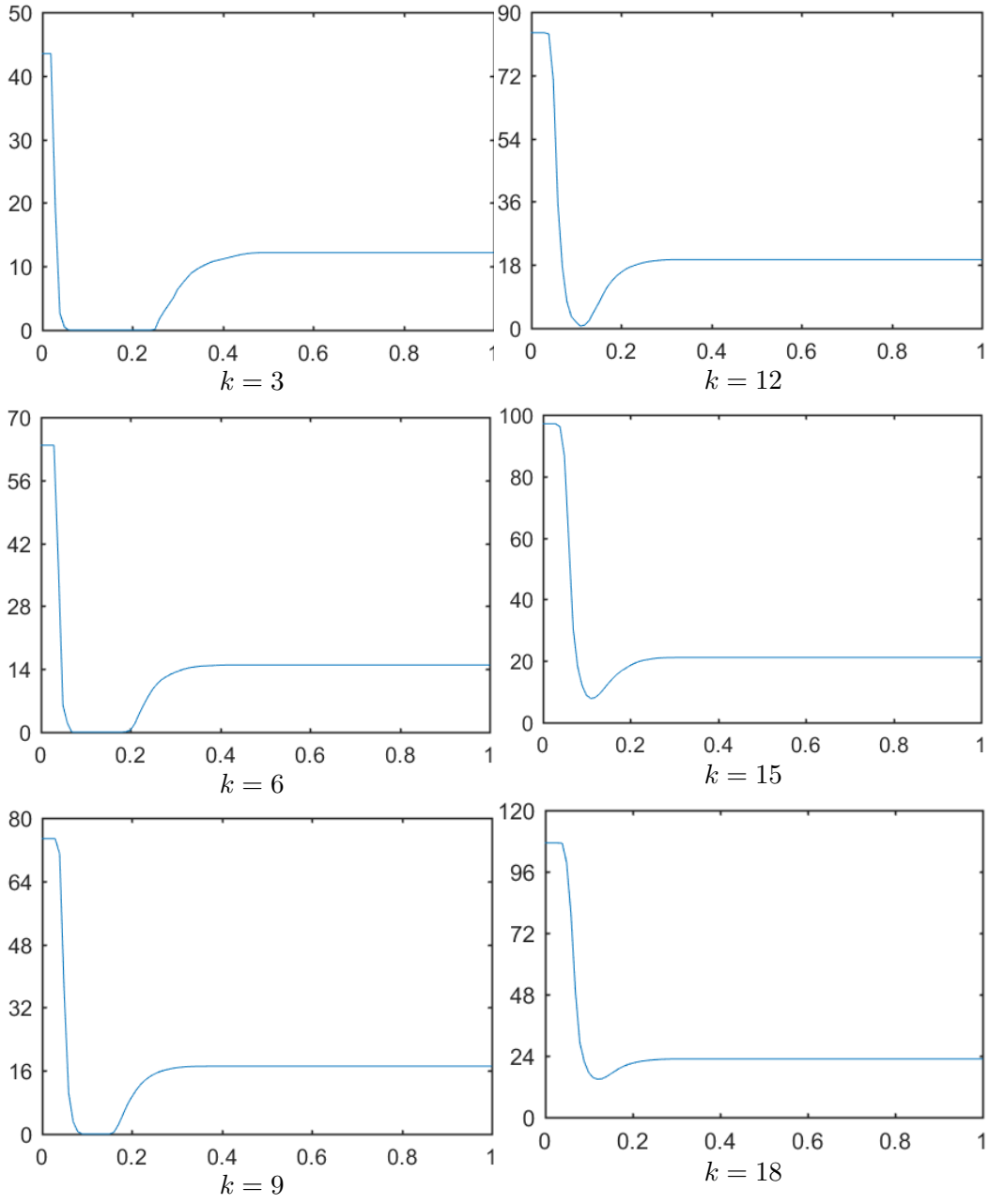
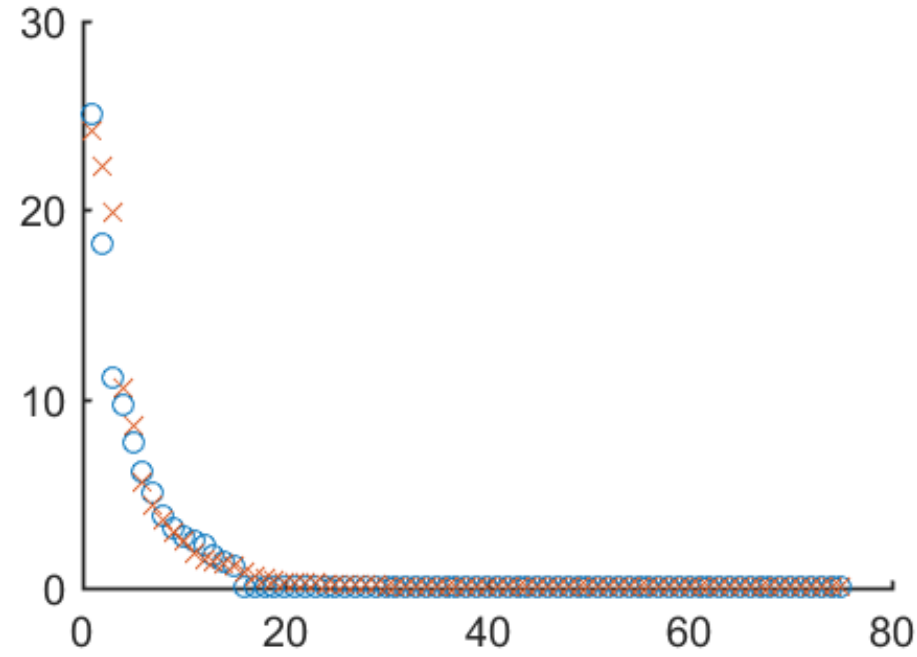
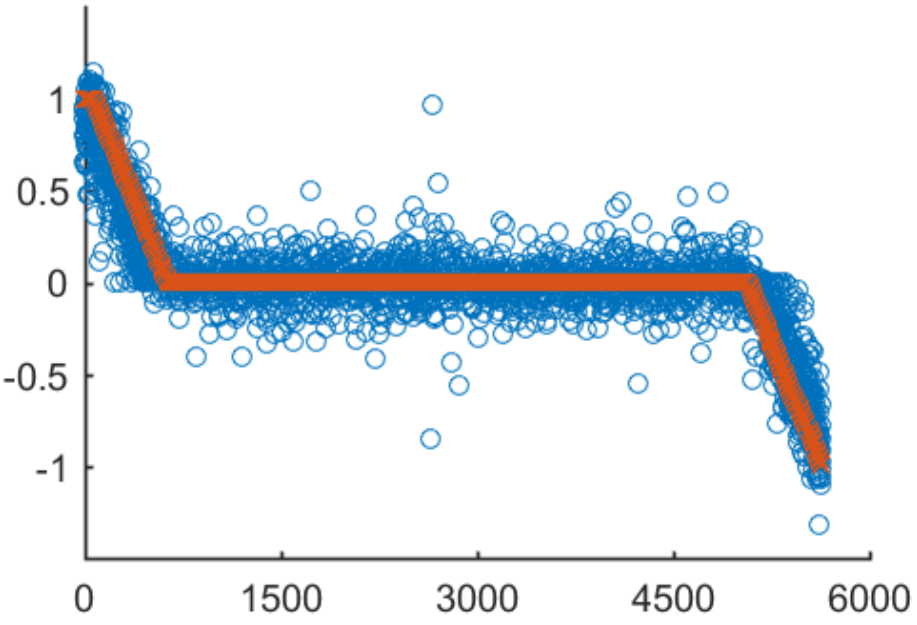


Figure 4.1: $f(\bar{\tau})$ **vs** $\bar{\tau}$ $\quad (n = 75)$



Singular values: input $\bar{F}$; output $\bar{G} - A_{\bar{\tau}}^{-1}$



Entries: sorted input $\bar{E}^{-1}$ vs. output $A_{\bar{\tau}}$

Figure 4.2: $n = 75,\ k = 15$

4.4 Conclusions

We presented a heuristic and a means of generating test problems for the sparse-inverse/low-rank decomposition problem on invertible input. Our method can also be used for generating a starting point for local optimization of $\mathcal{H}_1$.

We are presently working on a new approach to $\mathcal{H}_0$ based on a convex relaxation of $\mathcal{H}_1$. This new approach is much more computationally intensive than the method that we presented here, which we leverage for validating our new approach. Our new approach does not require that the input matrix be invertible. In fact, it can equally apply to even-more-general sparse-*pseudoinverse*/low-rank decomposition problems (see [FFL16b]).

Chapter 5

Computational Techniques for Sparse-Inverse/Low-Rank Decomposition

In this chapter, we develop convex relaxation ideas for the sparse-inverse/low-rank decomposition model ($\mathcal{P}$).

5.1 SDP Approach

Using a now standard idea, we can approximate ($\mathcal{P}$) from §4.1 with

$$\begin{array}{ll} \text{minimize} & \bar{\tau}\|A^{-1}\|_1 + (1-\bar{\tau})\|B\|_* \\ \text{subject to} & A + B = \bar{C}. \end{array} \qquad (\mathcal{R})$$

Employing a known technique (see [Cha+11], for example), we can reformulate ($\mathcal{R}$) as

$$\begin{array}{ll} \text{minimize} & \bar{\tau} e' S e + \dfrac{1}{2}(1-\bar{\tau})(tr(W_1) + tr(W_2)) \\ \text{subject to} & A + B = \bar{C} \\ & -S \leq H \leq S \\ & HA = I_n \\ & \begin{bmatrix} W_1 & B \\ B' & W_2 \end{bmatrix} \succeq \mathbf{0}_{2n}, \end{array} \qquad (\mathcal{R}')$$

where the $n \times n$ matrix variable H models A^{-1}.

The main difficulty that we face is that H and A are both variables, and so the constraint $HA = I_n$ is not convex. It should be noted that we do not know whether the relaxation would be improved by also using $AH = I_m$.

5.2 A convex relaxation

5.2.1 An SDP relaxation

We wish to relax the constraint $HA = I_n$, as it is not convex. We propose to relax ($\mathcal{R}'$) as the convex optimization problem

$$\begin{aligned}
\text{minimize} \quad & \bar{\tau} e' S e + \frac{1}{2}(1-\bar{\tau})(tr(W_1) + tr(W_2)) && (\mathcal{S}) \\
\text{subject to} \quad & A + B = \bar{C} && (5.1) \\
& H + S \geq \mathbf{0}_n && (5.2) \\
& -H + S \geq \mathbf{0}_n && (5.3) \\
& \begin{bmatrix} W_1 & B \\ B' & W_2 \end{bmatrix} \succeq \mathbf{0}_{2n} && (5.4) \\
& \frac{1}{2}\langle \bar{Q}, X_{ij} \rangle = \delta_{ij} && \forall i, j = 1, \ldots, n \quad (5.5) \\
& x_{ij} = \begin{pmatrix} h_i' \\ a_j \end{pmatrix} && \forall i, j = 1, \ldots, n, \quad (5.6) \\
& X_{ij} - x_{ij} x_{ij}' \succeq \mathbf{0}_{2n} && \forall i, j = 1, \ldots, n \quad (5.7)
\end{aligned}$$

where $X_{ij} \in \mathbb{S}^{2n \times 2n}$, $\bar{Q} \in \mathbb{R}^{2n \times 2n}$ has the form

$$\bar{Q} = \begin{bmatrix} \mathbf{0}_n & I_n \\ I_n & \mathbf{0}_n \end{bmatrix},$$

and h_i is the i^{th} row of H and a_j is the j^{th} column of A. The motivation for the constraints (5.5) and (5.7) come from a re-expession of $h_i a_j = \delta_{ij}$. The matrix inner product on the left-hand side of (5.5) can be re-expressed

as

$$\frac{1}{2}\left(h_i, a_j'\right)\begin{bmatrix}\mathbf{0}_n & I_n\\ I_n & \mathbf{0}_n\end{bmatrix}\begin{pmatrix}h_i'\\ a_j\end{pmatrix} = \frac{1}{2}\left(a_j', h_i\right)\begin{pmatrix}h_i'\\ a_j\end{pmatrix} = h_i a_j. \tag{5.8}$$

(5.8) may be written as

$$x_{ij}'\bar{Q}x_{ij} = \langle \bar{Q}, x_{ij}x_{ij}'\rangle. \tag{5.9}$$

The outer product $x_{ij}x_{ij}'$ defines a matrix X_{ij} i.e., $X_{ij} = x_{ij}x_{ij}'$, but such a constraint is not convex, therefore we consider the relaxation $x_{ij}x_{ij}' \preceq X_{ij}$, which can be expressed as the positive-semidefiniteness constraint $X_{ij} - x_{ij}x_{ij}' \succeq \mathbf{0}_{2n}$ or equivalently

$$\begin{bmatrix}1 & x_{ij}'\\ x_{ij} & X_{ij}\end{bmatrix} \succeq \mathbf{0}_{2n+1}. \tag{5.10}$$

Therefore we have a linear constraint with respect to variables X_{ij} in (5.5) and a convex constraint relating X_{ij} to x_{ij} in (5.7).

Lemma 9. *$(\mathcal{S})$ has a strictly feasible solution.*

Proof. A strictly feasible solution to $(\mathcal{S})$ can be constructed as follows:

(1) For (5.1), choose an arbitrary split of $\bar{C}$ with A invertible and $B = \bar{C} - A$.

(2) For (5.2), (5.3), set $H = A^{-1}$, and choose S such that $s_{ij} > |H_{ij}|$ for all $i, j = 1, \ldots, n$.

(3) For (5.4), choose W_1, W_2 such that

$$W_1 = W_2 = \Delta_1 I_n,$$

for sufficiently large $\Delta_1 > 0$. Then W will be strictly diagonally dominant and positive definite.

(4) For (5.7), let $x_{ij} = \begin{pmatrix} h_i' \\ a_j \end{pmatrix}$ for all $i, j = 1, \ldots, n$.

(5) For (5.5), define

$$X_{ij} = \begin{pmatrix} h_i' \\ a_j \end{pmatrix} \begin{pmatrix} h_i & a_j' \end{pmatrix} + \Delta_2 I_{2n} \qquad \forall i, j = 1, \ldots, n,$$

with $\Delta_2 > 0$.

□

We define matrices $\bar{Q}^+$, Z_{ij} as

$$\bar{Q}^+ = \begin{bmatrix} 0 & \vec{0}_{2n}' \\ \vec{0}_{2n} & \bar{Q} \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)},$$

$$Z_{ij} = \begin{bmatrix} x_{ij}^{(0)} & x_{ij}' \\ x_{ij} & X_{ij} \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)}, \qquad \forall i, j = 1, \ldots, n.$$

The row vectors h_i and column vectors a_j are implicit in the definition of Z_{ij}, so we need to include explicit linear constraints of the form

$$\begin{aligned} \langle \bar{R}_\ell, Z_{ij} \rangle &= h_{i\ell} \qquad \forall \ell = 1, \ldots, n, \\ \langle \bar{K}_\ell, Z_{ij} \rangle &= a_{\ell j} \qquad \forall \ell = 1, \ldots, n, \end{aligned} \tag{$\mathcal{L}$}$$

where $\bar{R}_\ell$ and $\bar{K}_\ell$, (both $\in \mathbb{R}^{(2n+1)\times(2n+1)}$), are defined as

$$\bar{R}_\ell = \frac{1}{2}\begin{bmatrix} 0 & e'_\ell & \vec{0}'_n \\ e_\ell & \mathbf{0}_n & \mathbf{0}_n \\ \vec{0}_n & \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \qquad \text{and } \bar{K}_\ell = \frac{1}{2}\begin{bmatrix} 0 & \vec{0}'_n & e'_\ell \\ \vec{0}_n & \mathbf{0}_n & \mathbf{0}_n \\ e_\ell & \mathbf{0}_n & \mathbf{0}_n \end{bmatrix},$$

where for $\bar{R}_\ell$, the only nonzero terms are ones in the $(1, \ell+1)$ and $(\ell+1, 1)$ positions, while for $\bar{K}_\ell$ the only nonzero terms are ones in the $(1, n+\ell+1)$ and $(n+\ell+1, 1)$ positions. This ensures that the matrix variables Z_{ij} exhibit the proper structure with respect to the composite vectors x_{ij}.

Furthermore, we need the additional linear constraint

$$x_{ij}^{(0)} = 1, \qquad \forall i, j = 1, \ldots, n,$$

which we write as

$$\langle \bar{E}^+, Z_{ij} \rangle = 1 \qquad \forall i, j = 1, \ldots, n,$$

where

$$\bar{E}^+ = \begin{bmatrix} 1 & \vec{0}'_{2n} \\ \vec{0}_{2n} & \mathbf{0}_{2n} \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)}.$$

Therefore, since we have

$$\begin{aligned} Z_{ij} \succeq \mathbf{0}_{2n+1} &\iff X_{ij} - x_{ij}x'_{ij} \succeq \mathbf{0}_{2n}, \text{ and} \\ \langle \bar{Q}, X_{ij} \rangle &\iff \langle \bar{Q}^+, Z_{ij} \rangle \end{aligned}$$

we can rewrite $(\mathcal{S})$ in the canonical form

$$
\begin{array}{lll}
\text{minimize} & \bar{\tau}\displaystyle\sum_{i,j=1}^{n} s_{ij} + \frac{1}{2}(1-\bar{\tau})\langle I_{2n}, W\rangle & \underline{\text{dual variables}} \\
\text{subject to} & a_{ij} + (1/2)\langle \bar{E}_{ij}, W\rangle = \bar{c}_{ij} \qquad \forall i,j = 1,\dots,n & (y_{ij}) \\
& h_{ij} + s_{ij} \geq 0 \qquad \forall i,j = 1,\dots,n & (g_{ij}) \\
& -h_{ij} + s_{ij} \geq 0 \qquad \forall i,j = 1,\dots,n & (v_{ij}) \\
& (1/2)\langle \bar{Q}^{+}, Z_{ij}\rangle = \delta_{ij} \qquad \forall i,j = 1,\dots,n & (u_{ij}) \\
& \langle \bar{R}_{\ell}, Z_{ij}\rangle - h_{i\ell} = 0 \qquad \forall i,j,\ell = 1,\dots,n & (t_{ij}^{(\ell)}) \\
& \langle \bar{K}_{\ell}, Z_{ij}\rangle - a_{\ell j} = 0 \qquad \forall i,j,\ell = 1,\dots,n & (p_{ij}^{(\ell)}) \\
& \langle \bar{E}^{+}, Z_{ij}\rangle = 1 \qquad \forall i,j = 1,\dots,n & (r_{ij}) \\
& Z_{ij} \succeq \mathbf{0}_{2n+1} \qquad \forall i,j = 1,\dots,n & \\
& W \succeq \mathbf{0}_{2n} & \\
& & (\mathcal{S}')
\end{array}
$$

with variables:

$$
a_{ij} \in \mathbb{R}, \qquad h_{ij} \in \mathbb{R}, \qquad s_{ij} \in \mathbb{R}, \qquad Z_{ij} \in \mathbb{S}_{+}^{(2n+1)\times(2n+1)}, \qquad W \in \mathbb{S}_{+}^{2n\times 2n},
$$

where

$$
\bar{E}_{ij} = \begin{array}{c} \\ \text{i} \\ \\ \\ \\ \text{j+n} \\ \\ \end{array}
\begin{array}{c}
\begin{array}{cc} \text{i} & \text{j+ n} \end{array} \\
\left[\begin{array}{cc} & 1 \\ & \\ & \\ & \\ 1 & \\ & \end{array}\right]
\end{array}
\in \mathbb{R}^{2n\times 2n}.
$$

Note that we can recover B from W via

$$W = \begin{bmatrix} W_1 & B \\ B' & W_2 \end{bmatrix}.$$

Corollary 10. *($\mathcal{S}'$) has a strictly feasible solution.*

5.2.2 The dual SDP

Applying the well-known notion of SDP duality of Section A.2, the dual of the SDP ($\mathcal{S}'$) can be written as

$$\text{maximize} \sum_{i,j=1}^{n} \bar{c}_{ij} y_{ij} + \sum_{i=1}^{n} u_{ii} + \sum_{i,j=1}^{n} r_{ij} \qquad (\mathcal{D})$$

subject to

$$g_{ij} + v_{ij} = \bar{\tau} \qquad \forall i,j = 1,\dots,n \qquad (5.11)$$

$$g_{ij} - v_{ij} - \sum_{\ell=1}^{n} t_{i\ell}^{(j)} = 0 \qquad \forall i,j = 1,\dots,n \qquad (5.12)$$

$$y_{ij} - \sum_{\ell=1}^{n} p_{\ell j}^{(i)} = 0 \qquad \forall i,j = 1,\dots,n \qquad (5.13)$$

$$\frac{1}{2} \sum_{i,j=1}^{n} y_{ij} \bar{E}_{ij} \preceq \frac{1}{2}(1-\bar{\tau}) I_{2n} \qquad (5.14)$$

$$\frac{1}{2} u_{ij} \bar{Q}^{+} + \sum_{\ell=1}^{n} t_{ij}^{(\ell)} \bar{R}_{\ell} + \sum_{\ell=1}^{n} p_{ij}^{(\ell)} \bar{K}_{\ell} + r_{ij} \bar{E}^{+} \preceq 0 \qquad \forall i,j = 1,\dots,n \qquad (5.15)$$

$$g_{ij}, v_{ij} \geq 0 \qquad \forall i,j = 1,\dots,n, \qquad (5.16)$$

with real scalar variables: y_{ij}, u_{ij}, $t_{ij}^{(\ell)}$, $s_{ij}^{(\ell)}$, r_{ij}, g_{ij}, v_{ij}.

Let us consider some simplifications that may help in solving ($\mathcal{D}$). We

may formulate the simplified dual problem as

$$\text{maximize} \sum_{i,j=1}^{n} \bar{c}_{ij} \sum_{\ell=1}^{n} p_{\ell j}^{(i)} + \sum_{i=1}^{n} u_{ii} + \sum_{i,j=1}^{n} r_{ij} \tag{$\mathcal{D}'$}$$

subject to

$$-\bar{\tau} \leq \sum_{\ell=1}^{n} t_{i\ell}^{(j)} \leq \bar{\tau} \qquad \forall i, j = 1, \dots, n \tag{5.17}$$

$$\sum_{i,j=1}^{n} \left(\sum_{\ell=1}^{n} p_{\ell j}^{(i)} \right) \bar{E}_{ij} \preceq (1 - \bar{\tau}) I_{2n} \tag{5.18}$$

$$\frac{1}{2} u_{ij} \bar{Q}^{+} + \sum_{\ell=1}^{n} t_{ij}^{(\ell)} \bar{R}_{\ell} + \sum_{\ell=1}^{n} p_{ij}^{(\ell)} \bar{K}_{\ell} + r_{ij} \bar{E}^{+} \preceq \mathbf{0}_{2n+1} \tag{5.19}$$

$$\forall i, j = 1, \dots, n$$

Lemma 11. *($\mathcal{D}$) is equivalent to ($\mathcal{D}'$)*

Proof. Taking (5.11), (5.12), notice that

$$v_{ij} = \frac{1}{2}(\bar{\tau} + \sum_{\ell=1}^{n} t_{i\ell}^{(j)}),$$
$$g_{ij} = \frac{1}{2}(\bar{\tau} - \sum_{\ell=1}^{n} t_{i\ell}^{(j)}).$$

Thus we may replace (5.11), (5.12), with (5.17), which reflects the nonnegativity of v_{ij} and g_{ij}. Furthermore, with constraint (5.13) we have

$$y_{ij} = \sum_{\ell=1}^{n} p_{\ell j}^{(i)} \qquad \forall i, j = 1, \dots, n,$$

so we substitute $\sum_{\ell=1}^{n} p_{\ell j}^{(i)}$ for y_{ij} in the objective term $\sum_{i,j=1}^{n} \bar{c}_{ij} y_{ij}$ to get

$$\sum_{i,j=1}^{n} \bar{c}_{ij} \sum_{\ell=1}^{n} p_{\ell j}^{(i)},$$

and substitute $\sum_{\ell=1}^{n} p_{\ell j}^{(i)}$ for y_{ij} in (5.14) to get (5.18).

Constraints (5.15) and (5.19) are the same, as (5.19) is unaffected by the above simplifications and substitutions.

□

Theorem 12. *Strong duality holds for ($\mathcal{S}/\mathcal{S}'$) and ($\mathcal{D}/\mathcal{D}'$).*

Proof. By Lemma 9 and Corrollary 10 it can be shown that a strictly feasible solution can be constructed for ($\mathcal{S}/\mathcal{S}'$). Via Slater's condition (17), it is sufficient that if we find a strictly feasible solution to the primal of a convex optimization problem, then this implies that the duality gap is zero, i.e., that strong duality holds. □

We do not know whether ($\mathcal{D}/\mathcal{D}'$) has a strictly feasible solution in general.

5.2.3 Disjunctive programming

With our definition of the nonconvex constraint $HA = I_n$ in terms of x_{ij}, X_{ij}, and subsequent relaxation into the constraints $X_{ij} - x_{ij}x'_{ij} \succeq 0$ for all $i, j = 1, \ldots, n$, we hope to construct a disjunctive cutting plane that tightens our original relaxations ($\mathcal{S}/\mathcal{S}'$) (in the spirit of [LR08], [SBL10a], [SBL10b]).

Secant inequalities

Applying the ideas of [LR08] and [SBL10a], we can develop a convex relaxation of the *nonconvex* constraints

$$X_{ij} - x_{ij}x'_{ij} \preceq 0 \qquad \forall i, j = 1, \dots, n^2, \tag{5.20}$$

so as to better approximate $X_{ij} = x_{ij}x'_{ij}$ when added to $(\mathcal{S}/\mathcal{S}')$.

The constraint (5.20) could equivalently be modeled by the inequalities $\nu'(X_{ij} - x_{ij}x'_{ij})\nu \leq 0$, which can be written as

$$(\nu' x_{ij})^2 \geq \langle \nu\nu', X_{ij} \rangle, \tag{5.21}$$

for all $\nu \in \mathbb{R}^{2n}$. The nonconvex inequality (5.21) is therefore a valid inequality for the SDP relaxation $(\mathcal{S}/\mathcal{S}')$. As the authors in [SBL10a] point out, it is possible to convexify (5.21) by replacing the concave quadratic function $-(\nu' x_{ij})^2$ with its secant on an interval $[\eta_L(\nu), \eta_U(\nu)]$. The convex relaxation of (5.21) is then given by the secant inequality

$$(\nu' x_{ij})(\eta_L(\nu) + \eta_U(\nu)) - \eta_L(\nu)\eta_U(\nu) \geq \langle \nu\nu', X_{ij} \rangle.$$

The interval $[\eta_L(\nu), \eta_U(\nu)]$ represents the range of the linear function $\nu' x_{ij}$ in the feasible set of $(\mathcal{S}/\mathcal{S}')$. More specifically,

$$\eta_L(\nu) := \min\{\nu' x_{ij} : \Sigma\},$$

and

$$\eta_U(\nu) := \max\{\nu' x_{ij} : \Sigma\},$$

where (Σ) is defined as the set of (S, A, H, W, Z_{ij}) subject to constraints

$$
\begin{aligned}
& a_{ij} + (1/2)\langle \bar{E}_{ij}, W \rangle = \bar{c}_{ij} && \forall i, j = 1, \dots, n \\
& h_{ij} + s_{ij} \geq 0 && \forall i, j = 1, \dots, n \\
& -h_{ij} + s_{ij} \geq 0 && \forall i, j = 1, \dots, n \\
& (1/2)\langle \bar{Q}^{+}, Z_{ij} \rangle = \delta_{ij} && \forall i, j = 1, \dots, n \\
& \langle \bar{R}_{\ell}, Z_{ij} \rangle - h_{i\ell} = 0 && \forall i, j, \ell = 1, \dots, n \\
& \langle \bar{K}_{\ell}, Z_{ij} \rangle - a_{\ell j} = 0 && \forall i, j, \ell = 1, \dots, n \\
& \langle \bar{E}^{+}, Z_{ij} \rangle = 1 && \forall i, j = 1, \dots, n \\
& Z_{ij} \succeq \mathbf{0}_{2n+1} && \forall i, j = 1, \dots, n \\
& W \succeq \mathbf{0}_{2n}.
\end{aligned}
\tag{Σ}
$$

Finally, in [SBL10a], the authors note that if $\hat{X}_{ij}$ and $\hat{x}_{ij}$ are obtained from the solution to $(\mathcal{S}/\mathcal{S}')$, and $\hat{X}_{ij} \neq \hat{x}_{ij}\hat{x}'_{ij}$, then $\hat{X}_{ij} - \hat{x}_{ij}\hat{x}'_{ij}$ has at least one positive eigenvalue. Furthermore, if the vector ν is chosen as the unit-length eigenvector corresponding to any positive eigenvalue of $\hat{X}_{ij} - \hat{x}_{ij}\hat{x}_{ij}$, then the constraint (5.21) would be violated by the solution of the relaxation. This observation guides the choice of the vector ν in the inequalities.

In order to strengthen $(\mathcal{S}/\mathcal{S}')$, we can apply the disjunctive cuts proposed in [LR08]. The idea is to divide the interval $[\eta_L(\nu), \eta_U(\nu)]$ into $k \geq 2$ intervals $[\eta_t, \eta_{t+1}]$, for $t = 1, \dots, k$, such that $\eta_L(\nu) := \eta_1 < \eta_2 < \dots < \eta_k < \eta_{k+1} := \eta_U(\nu)$.

Disjunctive programming over the SDP relaxation

Problem ($\mathcal{S}'$) can be rewritten as

$$
\begin{array}{lll}
\text{minimize} & \bar{\tau}\langle ee', S\rangle + \dfrac{1}{2}(1-\bar{\tau})\langle I_{2n}, W\rangle & \\
\text{subject to} & \langle E_{ij}, A\rangle + \langle \bar{E}_{ij}, W\rangle = \langle E_{ij}, \bar{C}\rangle & \forall i,j = 1,\dots,n \\
& \langle E_{ij}, H\rangle + \langle E_{ij}, S\rangle \geq 0 & \forall i,j = 1,\dots,n \\
& -\langle E_{ij}, H\rangle + \langle E_{ij}, S\rangle \geq 0 & \forall i,j = 1,\dots,n \\
& (1/2)\langle \bar{Q}, Z_{ij}\rangle = \delta_{ij} & \forall i,j = 1,\dots,n \\
& \langle \bar{R}_\ell, Z_{ij}\rangle - h_{i\ell} = 0 & \forall i,j,\ell = 1,\dots,n \\
& \langle \bar{K}_\ell, Z_{ij}\rangle - a_{\ell j} = 0 & \forall i,j,\ell = 1,\dots,n \\
& \langle \bar{E}^+, Z_{ij}\rangle = 1 & \forall i,j = 1,\dots,n \\
& W \succeq 0 & \\
& Z_{ij} \succeq 0. &
\end{array}
\qquad (\mathcal{S}'')
$$

Furthermore, for each $t = 1,\dots,k$, the inequalities

$$
\begin{gathered}
\eta_t \leq v' x_{ij} \leq \eta_{t+1} \\
(\nu' x_{ij})(\eta_t + \eta_{t+1}) - \eta_t \eta_{t+1} \geq \langle \nu\nu', X_{ij}\rangle
\end{gathered}
$$

can be rewritten as

$$
\begin{aligned}
-\langle \bar{N}, Z_{ij}\rangle &\geq -\eta_{t+1}, \\
\langle \bar{N}, Z_{ij}\rangle &\geq \eta_t, \\
-\langle \bar{D}_t, Z_{ij}\rangle &\geq \eta_t \eta_{t+1},
\end{aligned}
\qquad (5.22)
$$

where

$$\bar{N} := \begin{bmatrix} 0 & \frac{1}{2}v' \\ \frac{1}{2}v & \mathbf{0}_{2n} \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)},$$

and

$$\bar{D}_t := \begin{bmatrix} 0 & \vec{0}'_{2n} \\ \vec{0}_{2n} & vv' \end{bmatrix} - (\eta_t + \eta_{t+1})\bar{N} \in \mathbb{R}^{(2n+1)\times(2n+1)}.$$

We define

$$R_t := \{(S, A, W, H, Z_{ij}) : -\langle \bar{N}, Z_{ij} \rangle \geq -\eta_{t+1}, \\ \langle \bar{N}, Z_{ij} \rangle \geq \eta_t, -\langle \bar{D}_t, Z_{ij} \rangle \geq \eta_t \eta_{t+1}\},$$

for all $t = 1, \ldots, k$. Our goal is to construct a linear inequality of the form

$$\langle \Gamma_1, S \rangle + \langle \Gamma_2, A \rangle + \langle \Gamma_3, W \rangle + \langle \Gamma_4, H \rangle + \sum_{i,j} \langle \Gamma_{ij}, Z_{ij} \rangle \geq \beta \tag{5.23}$$

that is valid for

$$R := \text{convcl}\left(\cup_{t=1}^{k} (S \cap R_t)\right),$$

where S is the feasible set of problem ($\mathcal{S}''$). In order to construct the valid inequality, we consider $z_t :=$

$$
\begin{array}{lll}
\text{minimize} & \langle \Gamma_1, S\rangle + \langle \Gamma_2, A\rangle + \langle \Gamma_3, W\rangle + \langle \Gamma_4, H\rangle + \sum_{i,j}\langle \Gamma_{ij}, Z_{ij}\rangle & \underline{\text{dual var.}} \\
\text{subject to} & \langle E_{ij}, A\rangle + \langle \bar{E}_{ij}, W\rangle = \langle E_{ij}, \bar{C}\rangle \qquad \forall i,j = 1,\dots,n & (y_{ij}) \\
& \langle E_{ij}, H\rangle + \langle E_{ij}, S\rangle \geq 0 \qquad \forall i,j = 1,\dots,n & (g_{ij}) \\
& -\langle E_{ij}, H\rangle + \langle E_{ij}, S\rangle \geq 0 \qquad \forall i,j = 1,\dots,n & (v_{ij}) \\
& (1/2)\langle \bar{Q}, Z_{ij}\rangle = \delta_{ij} \qquad \forall i,j = 1,\dots,n & (u_{ij}) \\
& \langle \bar{R}_\ell, Z_{ij}\rangle - h_{i\ell} = 0 \qquad \forall i,j,\ell = 1,\dots,n & (t_{ij}^{(\ell)}) \\
& \langle \bar{K}_\ell, Z_{ij}\rangle - a_{\ell j} = 0 \qquad \forall i,j,\ell = 1,\dots,n & (p_{ij}^{(\ell)}) \\
& \langle \bar{E}^+, Z_{ij}\rangle = 1 \qquad \forall i,j = 1,\dots,n & (r_{ij}) \\
& -\langle \bar{N}, Z_{ij}\rangle \geq -\eta_{t+1} \qquad \forall i,j = 1,\dots,n & (\rho_{ij}^{(t)}) \\
& \langle \bar{N}, Z_{ij}\rangle \geq \eta_t \qquad \forall i,j = 1,\dots,n & (\gamma_{ij}^{(t)}) \\
& -\langle \bar{D}_t, Z_{ij}\rangle \geq \eta_t\eta_{t+1}, \qquad \forall i,j = 1,\dots,n & (\kappa_{ij}^{(t)}) \\
& W \succeq 0 & \\
& Z_{ij} \succeq 0. &
\end{array}
$$

and choose β such that $z_t \geq \beta$ for all $t = 1,\dots,k$.

By strong duality, we have $z_t :=$

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i,j=1}^{n} \bar{c}_{ij} y_{ij} + \sum_{i=1}^{n} u_{ii} + \sum_{i,j=1}^{n} (-\rho_{ij}^{(t)} + \gamma_{ij}^{(t)} \eta_t - \kappa_{ij}^{(t)} \eta_t \eta_{t+1}) + \sum_{i,j=1}^{n} r_{ij} \\
\text{subject to} \quad & g_{ij} + v_{ij} = (\Gamma_1)_{ij} \qquad \forall i, j = 1, \dots, n \\
& y_{ij} - \sum_{\ell=1}^{n} p_{\ell j}^{(i)} = (\Gamma_2)_{ij} \qquad \forall i, j = 1, \dots, n \\
& \frac{1}{2} \sum_{i,j=1}^{n} y_{ij} \bar{E}_{ij} \preceq \Gamma_3 \\
& g_{ij} - v_{ij} - \sum_{\ell=1}^{n} t_{i\ell}^{(j)} = (\Gamma_4)_{ij} \qquad \forall i, j = 1, \dots, n \\
& \frac{1}{2} u_{ij} \bar{Q}^{+} + \sum_{\ell=1}^{n} t_{ij}^{(\ell)} \bar{R}_\ell + \sum_{\ell=1}^{n} p_{ij}^{(\ell)} \bar{K}_\ell + r_{ij} \bar{E}^{+} \\
& \qquad - \rho_{ij}^{(t)} \bar{N} + \gamma_{ij}^{(t)} \bar{N} - \kappa_{ij}^{(t)} \bar{D}_t \preceq \Gamma_{ij} \qquad \forall i, j = 1, \dots, n \\
& g_{ij}, v_{ij}, \rho_{ij}^{(t)}, \gamma_{ij}^{(t)}, \kappa_{ij}^{(t)} \geq 0, \qquad \forall i, j = 1, \dots, n.
\end{aligned}
$$

Finally, searching for a valid inequality that is violated by a given solution $(\hat{S}, \hat{A}, \hat{W}, \hat{H}, \hat{Z}_{ij})$, we set $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_{ij}$ $\forall i, j = 1, \dots, n$, and β by solving the following problem:

$$
\begin{aligned}
\delta := \quad & \min \langle \Gamma_1, \hat{S} \rangle + \langle \Gamma_2, \hat{A} \rangle + \langle \Gamma_3, \hat{W} \rangle + \langle \Gamma_4, \hat{H} \rangle + \sum_{i,j} \langle \Gamma_{ij}, \hat{Z}_{ij} \rangle - \beta \\
\text{s.t.} \quad & \sum_{i,j=1}^{n} \bar{c}_{ij} y_{ij} + \sum_{i=1}^{n} u_{ii} + \sum_{i,j=1}^{n} (-\rho_{ij}^{(t)} + \gamma_{ij}^{(t)} \eta_t - \kappa_{ij}^{(t)} \eta_t \eta_{t+1}) + \sum_{i,j=1}^{n} r_{ij} \geq \beta \\
& \forall t = 1, \ldots, k \\
& g_{ij} + v_{ij} = (\Gamma_1)_{ij} \qquad \forall i, j = 1, \ldots, n \\
& y_{ij} - \sum_{\ell=1}^{n} p_{\ell j}^{(i)} = (\Gamma_2)_{ij} \qquad \forall i, j = 1, \ldots, n \\
& \frac{1}{2} \sum_{i,j=1}^{n} y_{ij} \bar{E}_{ij} \preceq \Gamma_3 \\
& g_{ij} - v_{ij} - \sum_{\ell=1}^{n} t_{i\ell}^{(j)} = (\Gamma_4)_{ij} \qquad \forall i, j = 1, \ldots, n \\
& \frac{1}{2} u_{ij} \bar{Q}^{+} + \sum_{\ell=1}^{n} t_{ij}^{(\ell)} \bar{R}_\ell + \sum_{\ell=1}^{n} p_{ij}^{(\ell)} \bar{K}_\ell + r_{ij} \bar{E}^{+} \\
& \qquad - \rho_{ij}^{(t)} \bar{N} + \gamma_{ij}^{(t)} \bar{N} - \kappa_{ij}^{(t)} \bar{D}_t \preceq \Gamma_{ij} \qquad \forall i, j, t \\
& g_{ij}, v_{ij}, \qquad \forall i, j, t \\
& \rho_{ij}^{(t)}, \gamma_{ij}^{(t)}, \kappa_{ij}^{(t)} \geq 0, \qquad \forall i, j, t
\end{aligned}
$$

If $\delta < 0$, the valid inequality (5.23) obtained from the solution of the above problem is violated by $(\hat{S}, \hat{A}, \hat{W}, \hat{H}, \hat{Z}_{ij})$.

It should be noted that the feasible region is a cone, so we need to add a normalization constraint. As noted in [FLT11], [LR08], and [SBL10a], we

may consider normalization constraints of the form

$$\begin{aligned}
&\sum_{i,j} |\Gamma_1(i,j)| \leq 1, \\
&\sum_{i,j} |\Gamma_2(i,j)| \leq 1, \\
&\sum_{i,j} |\Gamma_3(i,j)| \leq 1, \\
&\sum_{i,j} |\Gamma_4(i,j)| \leq 1, \\
&\sum_{k,\ell} |\Gamma_{ij}(k,\ell)| \leq 1 \qquad \forall i,j = 1,\dots,n,
\end{aligned}$$

although there may be other, more appropriate normalizations to use.

5.3 Remarks

Although we have presented a process to formulate the low-rank/sparse-inverse decomposition problem, along with a procedure to generate test problems (as seen in Chapter 4), we did not conduct any computational experiments using disjunctive cuts. Some preliminary experiments were conducted for the semi-definite programming (SDP) relaxation using MATLAB, CVX and MOSEK, but many questions remained regarding how to better handle the nonconvex constraint, which in our writeup is a left-inverse of A, but could equally be considered as a right-inverse or a pseudoinverse. With the difficulties in coding the formulations, as well as the computational/numerical limitations of existing SDP solvers, we chose to shift our focus to dealing with the nonconvex constraints dealing with the sparse-inverse and addressing the square and non-square scenarios.

Appendix A

Supplementary Material

Appendix

A.1 Conditional Independence Structure of Gaussian Random Variables

We can consider a covariance matrix A for n Gaussian random variables $X_1, \ldots, X_n$. We assume that $X_1, \ldots, X_n$ are partitioned into X and Y, so A has the form

$$A = \begin{bmatrix} A_{XX} & A_{XY} \\ A'_{XY} & A_{YY} \end{bmatrix}.$$

Via the Schur complement, we can see that

$$(A^{-1})_{XX} = \left(A_{XX} - A_{XY}(A_{YY})^{-1}A'_{XY}\right)^{-1}.$$

From this, we can see that the interpretation of $(A^{-1})_{XX}$ is that it is the *inverse* of the covariance matrix for X conditioned on Y. If $X = (X_i, X_j)$

and Y comprises the remaining random variables, then define a, b, c by

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} := (A^{-1})_{XX}.$$

Then

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}^{-1} = \frac{1}{ac - b^2} \begin{bmatrix} c & -b \\ -b & a \end{bmatrix}$$

is the covariance matrix for X conditioned on Y. So we can see that X_i and X_j are conditionally independent when $-b = 0$ (equivalently, when $b = 0$). Therefore the sparsity pattern of A^{-1} directly gives the conditional independence structure of $X_1, \ldots, X_n$.

A.2 Dual of an SDP

Most of the notation in this subsection is local to this and the subsequent subsection, with the dimensions of the matrices being implicit.

SDP duality: basic

Basic SDP duality is usually presented as follows (see [WSV00]). We have a single psd matrix variable X, and the primal form is

$$\min \left\{ \langle C, X \rangle \ : \ \langle E^i, X \rangle = b_i \text{ for } i \in \mathcal{E}, \ X \succeq \mathbf{0} \right\}.$$

The dual has the simple but rather different form

$$\max \left\{ \sum_{i \in \mathcal{E}} y_i b_i \ : \ \sum_{i \in \mathcal{E}} y_i E^i \preceq C \right\}.$$

SDP duality: general

In application, it is often convenient to have many psd matrix variables and additional scalar variables.

Below, we take the finite index sets $\mathcal{P}$, $\mathcal{U}$, $\mathcal{E}$ and $\mathcal{G}$ to be disjoint. We suppose that we have square and symmetric psd matrix variables X_k, for $k \in P$ and unrestricted scalar variables z_j, $j \in \mathcal{U}$. Note that the matrix variables can be of varying sizes. In particular, a 1×1 psd matrix variable is a non-negative scalar variable.

We take a general linear objective function

$$\min \sum_{k \in \mathcal{P}} \langle C_k, X_k \rangle + \sum_{j \in \mathcal{U}} d_j z_j \ . \tag{A.1}$$

We take some general linear equations and inequalities

$$\sum_{k \in \mathcal{P}} \langle E_k^i, X_k \rangle + \sum_{j \in \mathcal{U}} a_j^i z_j \left\{ \begin{array}{c} \geq \\ = \end{array} \right\} b_i, \text{ for } i \in \left\{ \begin{array}{c} \mathcal{G} \\ \mathcal{E}. \end{array} \right\} \tag{A.2}$$

So, our primal SDP is (A.1), subject to (A.2),

$$X_k \succeq \mathbf{0}, \text{ for } k \in \mathcal{P} \tag{A.3}$$

and

$$z_j \text{ unrestricted, for } j \in \mathcal{U}. \tag{A.4}$$

It is convenient to assume, without loss of generality, that the data matrices C_k and E_k^i are symmetric.

For the dual SDP, we will have non-negative scalar variables y_i, $i \in \mathcal{G}$, and unrestricted scalar variables y_i, $i \in \mathcal{E}$.

The dual objective function is

$$\max \sum_{i \in \mathcal{G} \cup \mathcal{E}} y_i b_i \tag{A.5}$$

Next, we have dual constraints corresponding to the primal matrix variables

$$\sum_{i \in \mathcal{G} \cup \mathcal{E}} y_i E_k^i \preceq C_k \ , \ \text{for } k \in \mathcal{P}. \tag{A.6}$$

Note that for k in which X_k is 1×1, the corresponding constraint (A.6) is a scalar '$\leq$' inequality. We have further dual constraints corresponding to the primal scalar variables

$$\sum_{i \in \mathcal{G} \cup \mathcal{E}} y_i a_j^i = d_j \ , \ \text{for } j \in \mathcal{U}. \tag{A.7}$$

So our dual SDP is (A.5), subject to (A.6), (A.7),

$$y_i \geq 0, \text{ for } i \in \mathcal{G}. \tag{A.8}$$

and

$$y_i \text{ unrestricted, for } i \in \mathcal{E}. \tag{A.9}$$

Observation 13. *If there are no matrix variables in the primal that are bigger than* 1×1*, then the primal is a linear-optimization problem ((A.3) reduces to scalar non-negativity constraints). In this case, the dual is also a linear-optimization problem, as (A.6) reduces to scalar inequalities, and we get a familiar dual pair of linear-optimization problems.*

Observation 14. *If* $\mathcal{U} = \mathcal{G} = \emptyset$ *and* $|\mathcal{P}| = 1$*, then we get the familiar basic dual pair of SDPs of §A.2*

Observation 15. *If we define square symmetric matrix variables via the linear equations*

$$S_k = C_k - \sum_{i \in \mathcal{G} \cup \mathcal{E}} y_i E_k^i \ , \ \textit{for } k \in \mathcal{P}, \tag{A.10}$$

then we can see the dual in a form that is essentially in the form of the primal.

Definition 16. *Let p^* denote the optimal value associated with the optimal solution to the primal SDP (A.1), (A.2), and (A.4) and d^* denote the optimal value associated with the optimal to the dual SDP (A.5), (A.6), (A.7), and (A.8). If the equality $p^* = d^*$ holds, then* strong duality *holds.*

Lemma 17. *(**Slater's Condition**) Given an SDP with objective function of the form (A.1) subject to (A.2), (A.3), and (A.4), if there exists a strictly feasible solution that satisfies (A.2), (A.3), and (A.4), then strong duality holds (i.e, there is zero duality gap between the primal and dual optimal values).*

Bibliography

[Ach07] T. Achterberg. "Constraint integer programming". PhD thesis. Berlin Institute of Technology, 2007. ISBN: 978-3-89963-892-9. URL: http://opus.kobv.de/tuberlin/volltexte/2007/1611/.

[AH17] A. A. Ahmadi and G. Hall. "Sum of squares basis pursuit with linear and second order cone programming". *Algebraic and geometric methods in discrete mathematics*. Vol. 685. Contemporary Mathematics. American Mathematical Society, Providence, RI, 2017, pp. 27–53.

[Ber06] T. Berthold. "Primal Heuristics for Mixed Integer Programming". MA thesis. Technische Universität Berlin, 2006.

[Ber08] T. Berthold. "Heuristics of the Branch-Cut-and-Price-Framework SCIP". *Operations Research Proceedings 2007*. Ed. by J. Kalcsics and S. Nickel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 31–36. ISBN: 978-3-540-77903-2.

[Boy+10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". *Foundations and Trends in Machine Learning* 3.1 (2010), pp. 1–122.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CG80] R. Cline and T. Greville. "A Drazin inverse for rectangular matrices". *Linear Algebra and its Applications* 29 (1980), pp. 53–62.

[Cha+11] V. Chandrasekaran, S. Sanghave, P. Parrilo, and W. A.S. "Rank-Sparsity Incoherence for Matrix Decomposition". *SIAM Journal on Optimization* 21.2 (2011), pp. 572–596.

[Che90] Y. Chen. "The generalized Bott-Duffin inverse and its applications". *Linear Algebra and its Applications* 134 (1990), pp. 71–91. ISSN: 0024-3795.

[CLL11] T. Cai, W. Liu, and X. Luo. "A Constrained ℓ_1 Minimization Approach to Sparse Precision Matrix Estimation". *Journal of the American Statistical Association* 106.494 (2011), pp. 594–607.

[DG17] I. Dokmanić and R. Gribonval. "Beyond Moore-Penrose Part II: The Sparse Pseudoinverse". working paper or preprint. June 2017. URL: https://hal.inria.fr/hal-01547283.

[DKV13] I. Dokmanić, M. Kolundžija, and M. Vetterli. "Beyond Moore-Penrose: Sparse pseudoinverse". *ICASSP 2013, pp. 6526–6530*. 2013.

[DRL05] E. Danna, E. Rothberg, and C. Le Pape. "Exploring relaxation induced neighborhoods to improve MIP solutions". *Mathematical Programming, Series A* 102 (2005), pp. 71–90.

[EN07] J. Eckstein and M. Nediak. "Pivot, Cut, and Dive: a heuristic for 0-1 mixed integer programming". *Journal of Heuristics* 13 (2007), pp. 471–503.

[FFL16a] V. Fuentes, M. Fampa, and J. Lee. "Low-rank/Sparse-Inverse Decomposition". *Operations Research Proceedings 2016*. 2016, pp. 111–117.

[FFL16b] V. Fuentes, M. Fampa, and J. Lee. "Sparse pseudoinverses via LP and SDP relaxations of Moore-Penrose". *CLAIO 2016*. 2016, pp. 343–350.

[FFL19] V. Fuentes, M. Fampa, and J. Lee. "Diving for sparse partially reflexive generalized inverses". *WGCO 2019*. 2019.

[FHT08] J. Friedman, T. Hastie, and R. Tibshirani. “Sparse inverse covariance estimation with the graphical lasso”. *Biostatistics* 9.3 (2008), pp. 432–441.

[FL18a] M. Fampa and J. Lee. *Efficient treatment of bilinear forms in global optimization*. arXiv:1803.07625. 2018.

[FL18b] M. Fampa and J. Lee. “On sparse reflexive generalized inverse”. *Operations Research Letters* 46.6 (2018), pp. 605–610.

[FLT11] M. Fischetti, A. Lodi, and A. Tramontani. “On the separation of disjunctive cuts”. *Mathematical Programming* 128.1–2, Ser. A (2011), pp. 205–230.

[GB08] M. Grant and S. Boyd. “Graph implementations for nonsmooth convex programs”. *Recent Advances in Learning and Control. pp. 95–110*. Springer, 2008.

[GB15] M. Grant and S. Boyd. *CVX, version 2.1*. 2015.

[GKL17] D. Gerard, M. Köppe, and Q. Louveaux. “Guided dive for the spatial branch-and-bound”. *Journal of Global Optimization* 68.4 (Aug. 2017), pp. 685–711. ISSN: 1573-2916.

[GV96] G. Golub and C. Van Loan. *Matrix Computations (3rd Ed.)* Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[Hag89] W. Hager. “Updating the Inverse of a Matrix”. *SIAM Review* 31.2 (1989), pp. 221–239. ISSN: 0036-1445.

[Hsi+11] C. Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. “Sparse Inverse Covariance Matrix Estimation Using Quadratic Approximation”. *Advances in Neural Information Processing Systems* 24 (2011), pp. 2330–2338.

[KK03] S. Kim and M. Kojima. “Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations”. *Computational Optimization and Applications* 26.2 (2003), pp. 143–154.

[LR08] J. Lee and F. Rendl. “Improved bounds for Max-Cut via disjunctive programming”. Personal notes. 2008.

[LZ14] J. Lee and B. Zou. "Optimal rank-sparsity decomposition". *Journal of Global Optimization* 60.2 (2014), pp. 307–315.

[Mey73] C. Meyer. "Generalized inversion of modified matrices". *SIAM Journal on Applied Mathematics* 24 (1973), pp. 315–323.

[Pat13] G. Pataki. "Strong Duality in Conic Linear Programming: Facial Reduction and Extended Duals". *Computational and Analytical Mathematics. pp. 613–634*. Springer, 2013.

[Rie92] K. Riedel. "A Sherman-Morrison-Woodbury identity for rank augmenting matrices with application to centering". *SIAM Journal on Matrix Analysis and Applications* 13.2 (1992), pp. 659–662.

[RM71] C. Rao and S. Mitra. *Generalized Inverse of Matrices and Its Applications*. Probability and Statistics Series. Wiley, 1971.

[SA99] H. D. Sherali and W. P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Vol. 31. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, 1999, pp. xxiv+514. ISBN: 0-7923-5487-7.

[SBL10a] A. Saxena, P. Bonami, and J. Lee. "Convex relaxations of non-convex mixed integer quadratically constrained programs: Extended formulations". *Mathematical Programming, Series B* 124.1-2 (2010), pp. 383–411.

[SBL10b] A. Saxena, P. Bonami, and J. Lee. "Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations". *Mathmatical Programming, Series A* 130.2 (2010), pp. 359–413.

[SL18] E. Speakman and J. Lee. "On branching-point selection for trilinear monomials in spatial branch-and-bound: the hull relaxation". *Journal of Global Optimization* 72.2 (2018), pp. 129–153.

[SMG10] K. Scheinberg, S. Ma, and D. Goldfarb. "Sparse inverse covariance selection via alternating linearization methods". *Neural Information Processing Systems 2010*. 2010, pp. 2101–2109.

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming : theory, algorithms, and applications*. International Series in Operations Research & Management Science. Boston, London: Kluwer Academic, 2000. ISBN: 0-7923-7771-0. URL: http://opac.inria.fr/record=b1099098.

[Zha05] F. Zhang, ed. *The Schur complement and its applications*. Vol. 4. Numerical Methods and Algorithms. Springer-Verlag, New York, 2005, pp. xvi+295.