Implicit Loss of Surjectivity and Facial Reduction: Theory and Applications

by

Haesol Im

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Kim-Chuan Toh
Professor, Dept. of Mathematics
National University of Singapore

Supervisor: Henry Wolkowicz
Professor, Dept. of Combinatorics and Optimization
University of Waterloo

Internal Member: Walaa Moursi
Assistant Professor, Dept. of Combinatorics and Optimization
University of Waterloo

Internal Member: Stephen Vavasis
Professor, Dept. of Combinatorics and Optimization
University of Waterloo

Internal-External Member: Yaoliang Yu
Associate Professor, Computer Science
University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Facial reduction, pioneered by Borwein and Wolkowicz, is a preprocessing method that is commonly used to obtain strict feasibility in the reformulated, reduced constraint system. The importance of strict feasibility is often addressed in the context of the convergence results for interior point methods. Beyond the theoretical properties that the facial reduction conveys, we show that facial reduction, not only limited to interior point methods, leads to strong numerical performances in different classes of algorithms. In this thesis we study various consequences and the broad applicability of facial reduction.

The thesis is organized in two parts. In the first part, we show the instabilities accompanied by the absence of strict feasibility through the lens of facially reduced systems. In particular, we exploit the implicit redundancies, revealed by each nontrivial facial reduction step, resulting in the implicit loss of surjectivity. This leads to the two-step facial reduction and two novel related notions of singularity. For the area of semidefinite programming, we use these singularities to strengthen a known bound on the solution rank, the Barvinok-Pataki bound. For the area of linear programming, we reveal degeneracies caused by the implicit redundancies. Furthermore, we propose a preprocessing tool that uses the simplex method.

In the second part of this thesis, we continue with the semidefinite programs that do not have strictly feasible points. We focus on the doubly-nonnegative relaxation of the binary quadratic program and a semidefinite program with a nonlinear objective function. We closely work with two classes of algorithms, the splitting method and the Gauss-Newton interior point method. We elaborate on the advantages in building models from facial reduction. Moreover, we develop algorithms for real-world problems including the quadratic assignment problem, the protein side-chain positioning problem, and the key rate computation for quantum key distribution. Facial reduction continues to play an important role for providing robust reformulated models in both the theoretical and the practical aspects, resulting in successful numerical performances.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Henry Wolkowicz. Having him as my advisor was the greatest blessing that ever happened to me throughout my academic journey. I could not have asked for a more perfect academic father than him. He never spoke a word about what it takes to be a good researcher, yet, led by example, I naturally grew to learn what it means by sitting next to him. From early mornings to late nights that he spared for me were tiny steps for becoming a better researcher. For the past few years, *I really enjoyed doing math with him.* His warm genuine nature and insightful guidance will be deeply cherished in my heart for many years to come.

I would like to express my love to my significant other, William Yeung. The countless meals and coffees that he had made for me were my physical source of energy. His support for the choices I make has been my emotional source of energy. Knowing that he is always going to be there for me is my pillar for moving forward. He definitely deserves a PhD degree in understanding Haesol, if there was such a field of study. I am excited for what the future holds for us and cannot wait to see what adventures await us in the next chapters of our lives.

I am deeply grateful to the committee members Walaa Moursi, Kim-Chaun Toh, Stephen Vavasis and Yaoliang Yu for being part of my academic journey. I am also grateful to many staff members of the department. I would like to acknowledge Melissa Cambridge and Carol Seely-Morrison for their fantastic and dependable support for the entire time I stayed here. Finally I would like to express my appreciation to Chris Calzonetti for his exceptional assistance in the delivery of my examination and many requests that I had made for using computing facilities.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Facial reduction (**FR**), first proposed by Borwein and Wolkowicz [21,22], is a preprocessing method that is often used to obtain strict feasibility in the reformulated, reduced constraint system of an optimization problem. The importance of strict feasibility is often addressed in the context of interior point methods, generally for convergence results. Beyond the theoretical properties that facial reduction conveys, we show that it leads to strong numerical performances in different classes of algorithms, not only limited to interior point methods. In this thesis we study various consequences and the broad applicability that **FR** accompanies.

In the first half of the thesis, we study interesting properties of the system produced by **FR**. We focus on the constraint system represented as the intersection of an affine subspace and a closed convex cone. The surjectivity of the linear map that restricts the feasible region is a standard assumption in optimization problems. Although redundant equalities do not alter the feasible region, the absence of surjectivity results in numerical instability. Moreover, the surjectivity provides the uniqueness of central paths in the convergence result for interior point methods. However, when strict feasibility fails, the linear map that restricts the feasible region necessarily loses the surjectivity in conjunction with the cone constraint. These redundancies are recognized in the literature. Nevertheless, the extensive implications have not yet been realized. In the addition to the known notion of the singularity degree, the unrealized redundancies give rise to the novel definitions on singularities that we name the max-singularity degree and the implicit problem singularity. Furthermore, these singularities give rise to the two-step facial reduction algorithm.

In the absence of strict feasibility, we examine interesting properties that stem from these singularities both in the areas of semidefinite programming and linear programming. In the area of semidefinite programming, we understand the numerical difficulties through the lens of the facially reduced system. We then use the new notions of singularity to improve the known bounds on the rank of feasible points known as the Barvinok-Pataki bound. In the area of linear programming, strict feasibility is seldom a concern. Albeit, the absence of strict feasibility causes difficulties when the simplex and interior point methods are used. The implicit loss of surjectivity immediately results in the degeneracy of each of the basic feasible solutions hence general computational efficiency of the simplex method may be weakened. Furthermore, the lack of strict feasibility results in the positive implicit problem singularities. This results in ill-conditioning and loss of precision, when finding search directions of interior point methods. We propose a preprocessing method that we can apply as an extension of the two-phase simplex method. Owing to the high accuracy that the simplex method provides, we produce an accurate facially reduced system.

In the second half of the thesis, we study the broad applicability of the facial reduction technique that appears in various steps of applications. We show how the two seemingly different classes of algorithms, the splitting method and the interior point method, can benefit from the **FR** approach. We connect the splitting method to the doubly-nonnegative relaxation of the binary quadratic problems and provide a simple derivation for the doubly-nonnegative relaxation. **FR** provides a natural splitting of variables that allows for an effective separation of the polyhedral and nonpolyhedral subproblems. Furthermore, **FR** leads to the straightforward discovery of redundant constraints and the prior knowledge of the dual optimal solutions. This leads to the development of a variant of the Peaceman-Rachford splitting method. We apply our approach to the two real-world problems, the protein side-chain positioning problem and the quadratic assignment problem. We exhibit competitive numerical results accomplished by our reformulation technique as well as the choice of algorithm.

We develop the (projected) Gauss-Newton interior point method for semidefinite programs that are over the Hermitian matrices and have a nonlinear objective function. We derive this framework by forming the nonlinear least squares problem that stems from the first-order optimality conditions. We then apply our method to one of the challenging real-world problems that arises in quantum information theory, the key rate computation of the quantum key distribution. This problem not only fails to have a strictly feasible point, but also the objective function fails to be differentiable. We overcome both difficulties by using the **FR** technique. In particular, we reformulate the objective function to grant the differentiability via the **FR** technique, that is conventionally considered to improve the property of the feasible region.

1.1 Background

We list the common notations used throughout in this thesis and describe the basic model of interest.

1.1.1 Notations

We list some notations used in this thesis. We use these notations without further explanations in later chapters.

We work with finite dimensional Euclidean spaces throughout this thesis. We let $\mathbb{E}^n$ denote the n-dimensional Euclidean space. We list the spaces that we closely work with. Let $\mathbb{R}^n, \mathbb{C}^n$ denote the Euclidean vector space of n-coordinates over the real and complex space, respectively; and we use superscripts to denote the dimensions, e.g., $x \in \mathbb{R}^n$. We let $\mathbb{R}^{m\times n}$ and $\mathbb{C}^{m\times n}$ denote the set of m-by-n real matrices and complex matrices, respectively. For $X, Y \in \mathbb{R}^{m\times n}$, let $\langle X, Y\rangle$ denote the usual trace inner product of X and Y, $\operatorname{trace}(X^TY)$. Analogously, for $X, Y \in \mathbb{C}^{m\times n}$, we let $\langle X, Y\rangle = \operatorname{trace}(X^*Y)$, where X^* denotes the conjugate transpose of the matrix X. We use the notation

$$\operatorname{trace}(XY) = \langle X, Y\rangle = X \bullet Y$$

interchangeably when X, Y are symmetric.

Given $x \in \mathbb{R}^n$, we use subscripts to designate a particular element of x, i.e., we let x_i denote the i-th element of x. Similarly, for $X \in \mathbb{R}^{m\times n}$, we use $X_{i,j}$ to denote the (i, j)-th element of X. Given a vector or a matrix, we often adopt the MATLAB notation to extract partial elements. For example, for $x \in \mathbb{R}^n$ and $\mathcal{I} \subset \{1,\dots,n\}$, we let $x(\mathcal{I})$ denote the subvector of x that correspond to the index set $\mathcal{I}$. For $X \in \mathbb{R}^{m\times n}$, we use $X(:,\mathcal{I})$ to denote the submatrix of X for which the columns correspond to the index set $\mathcal{I}$. We let I_n be the n-by-n identity matrix and we omit the subscript n when the dimension is clear. We let e_i denote the i-th column of the identity matrix, the standard unit basis element in $\mathbb{R}^n$. We use the notation $\bar{e}_m$ to denote the all-ones vector of the length m. When the length of the all-ones vector is clear, we omit the bar symbol ($^-$) and the subscript, and simply use e.

We let $\mathbb{R}^n_+$ ($\mathbb{R}^n_{++}$, resp.) denote the nonnegative (positive, resp.) orthant of n-coordinates. We let $\mathbb{S}^n$ and $\mathbb{H}^n$ denote the space of n-by-n symmetric matrices and n-by-n Hermitian matrices, respectively, i.e.,

$$\mathbb{S}^n := \{X \in \mathbb{R}^{n\times n} : X_{i,j} = X_{j,i},\ \forall i,j\},\ \ \mathbb{H}^n := \{X \in \mathbb{C}^{n\times n} : X^*_{i,j} = X_{j,i},\ \forall i,j\}.$$

A matrix $X \in \mathbb{S}^n$ (or $\mathbb{H}^n$) is called *positive semidefinite* if $\langle x, Xx\rangle \geq 0$ for all $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$). The set of n-by-n positive semidefinite matrices is denoted by $\mathbb{S}^n_+$ (or $\mathbb{H}^n_+$) and we use the notation $X \succeq 0$ to denote the membership $X \in \mathbb{S}^n_+$ (or $\mathbb{H}^n_+$). A matrix $X \in \mathbb{S}^n$ (or $\mathbb{H}^n$) is called *positive definite* if $\langle x, Xx\rangle > 0$ for all nonzero $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$). The set of n-by-n positive definite matrices is denoted by $\mathbb{S}^n_{++}$ (or $\mathbb{H}^n_{++}$) and we use the notation $X \succ 0$ to denote the membership $X \in \mathbb{S}^n_{++}$ (or $\mathbb{H}^n_{++}$).

Given a matrix X, we use $\operatorname{range}(X)$ and $\operatorname{null}(X)$ to denote the range and the nullspace of X, respectively. Given a matrix $X \in \mathbb{S}^n$, we use $\lambda_{\min}(X)$ ($\lambda_{\max}(X)$, resp.) to denote the minimum (maximum, resp.) eigenvalue of X. Given a matrix $X \in \mathbb{R}^{n\times n}$, we use $\operatorname{diag}(X)$ to denote the vector consists of the diagonal entries of X. Given a vector $x \in \mathbb{R}^n$, we let $\operatorname{Diag}(x)$ denote the diagonal matrix with x placed along its diagonal entries. Given a positive integer m, we often use the notation $[m]$ to mean the set of positive integers $\{1,\dots,m\}$. Given a collection of matrices $\{A_i\}_{i=1}^m$, we let $\operatorname{BlkDiag}(A_1,\dots,A_m)$ denote the block diagonal matrix with the i-th diagonal block A_i. Given a set $\mathcal{S} \subseteq \mathbb{E}^n$, we let $\mathcal{S}^\perp$ denote the *orthogonal* complement of $\mathcal{S}$. Given a set $\mathcal{C} \subseteq \mathbb{E}^n$, we use $\operatorname{int}(\mathcal{C})$ (resp. $\operatorname{relint}(\mathcal{C})$) to denote the interior of $\mathcal{C}$ (resp. relative interior of $\mathcal{C}$).

1.1.2 Basic Model

Let $\mathcal{A}$ be a surjective linear map from $\mathbb{S}^n$ to $\mathbb{R}^m$ and let $b \in \mathbb{R}^m$. Given an affine subspace

$$L = \{X \in \mathbb{S}^n : \mathcal{A}(X) = b\},$$

a *spectrahedron* is defined as the intersection of L and the positive semidefinite cone, $\mathbb{S}^n_+$:

$$\mathcal{F} = \{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b\}. \tag{1.1.1}$$

Throughout the thesis, we use m to denote the number of affine constraints in $\mathcal{F}$. In this thesis, we call a minimization problem over the spectrahedron $\mathcal{F}$ a *semidefinite programming, **SDP***:

$$p^* = \min_{X\in\mathbb{S}^n} \{f(X) : \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\}, \tag{1.1.2}$$

where $f : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$ is an extended convex function. We define the spectrahedron in $\mathbb{H}^n_+$ and the **SDP** over $\mathbb{H}^n_+$ analogously.

For the first half of this thesis, we focus on the feasible region (1.1.1) only. A point $X \in \mathcal{F}$ with the property $X \in \mathbb{S}^n_{++}$ is called a *strictly feasible* point. We extensively study the properties that the linear map $\mathcal{A}$ produces in the absence of strict feasibility. In addition, we study interesting properties when $\mathcal{F}$ is reduced to a polyhedron in $\mathbb{R}^n_+$. For the second half of the thesis, we consider the optimization problem (1.1.2). We study how the semidefinite program (1.1.2) is reformulated after **FR** and how they fit into the two classes of the algorithms, the splitting method and the interior point method.

1.2 Contributions and Organization

We summarize the contributions and the organization of this thesis. In Chapter 2 we present some of the background in linear algebra, convex analysis, results in **SDP**, splitting methods and the Gauss-Newton method needed in this thesis.

The main results are organized in two parts throughout Chapters 3, 4, 5 and 6. Part I concerns the implicit loss of surjectivity. In Chapters 3 and 4, we elaborate on the constraint redundancies that stem from the absence of strict feasibility. We focus on semidefinite programs in Chapter 3. The contributions include

- the recognition of the implicit loss of surjectivity;
- the new notions of singularity (max-singularity degree and implicit problem singularity);
- the importance of the two-step facial reduction;
- a strengthened Barvinok-Pataki bound on **SDP** rank.

We focus on linear programs in Chapter 4. The contributions include

- the degeneracy of the individual basic feasible solution of a linear program;
- the development of preprocessing method that uses the simplex;
- a new perspective on the algorithmic difficulties of the simplex and interior methods in the absence of strict feasibility.

In Part II, we study the broad applicability of the **FR** technique that appear in various contexts of applications. In Chapter 5 we show the unified derivation for the doubly-nonnegative (**DNN**) relaxation of the binary quadratic problems with the unit row-sum constraints. We then develop a variant of the Peaceman-Rachford splitting method. The contributions include:

- We understand the structural properties embedded in the **DNN** relaxation of the binary quadratic problem with the unit row-sum constraints;
- We derive the splitting method that uses information on the known elements of the dual optimal solutions;
- We apply the developed framework to the two classes of real-world problems, the protein side-chain positioning problem and the quadratic assignment problem.

In Chapter 6 we extend the Gauss-Newton interior point method that is applicable to **SDP** over the Hermitian matrices with a differentiable nonlinear objective function. We apply the framework to a challenging real-world problem, the key rate computation for quantum key distribution. We recognize that the **FR** technique continues to play an important role in quantum information theory. The contributions include:

- We recognize the **FR** that stems from the partial trace operator in quantum information theory;
- We show that the **FR** is not only limit to improve properties on the feasible region. We can use **FR** to improve properties of the objective function;
- We develop the Gauss-Newton interior point method for **SDP**s over the Hermitian matrices with a differentiable nonlinear objective function.

In Chapter 7 we make conclusions and list interesting open problems and future works.

FR, as a preprocessing tool, serves as a medium for delivering strong numerical performances of many different classes of algorithms. We see the interplay between **FR** and the simplex method, the splitting methods, the interior point methods. Walking through this thesis, we recognize various circumstances that **FR** takes place. The **FR** process can be done using many approaches:

1. Solve an auxiliary system directly for **FR** (Section 2.3.2);
2. Use the simplex method for **FR** in linear programs (Section 4.2.2);
3. Exploit the data structure in the **SDP** relaxations for **FR** (Section 5.1);
4. Use the property that stems from the partial trace for **FR** (Section 6.3.1);
5. Use the **FR** technique to the objective function for differentiability (Section 6.3.2).

Chapter 2

Preliminaries

We now present some of the background in linear algebra, convex analysis, results in **SDP**, splitting methods and the Gauss-Newton method needed in this thesis. In Section 2.1 we provide basic notions in linear algebra and various maps used in this thesis. In Section 2.2, we introduce basic notions of convex analysis and some related results. In Section 2.3 we present interesting results in semidefinite programming with emphasis on facial reduction. In Section 2.4 we present algorithms that are used in this thesis including the Peaceman-Rachford splitting method and the Gauss-Newton method.

2.1 Linear Algebra

Let $\mathcal{D}, \mathcal{R}$ be vector spaces with its underlying scalar field F. A function $T : \mathcal{D} \to \mathcal{R}$ is called *linear* if it satisfies

$$T(\alpha x + \beta y) = \alpha T(x) + \beta T(y),$$

for all $x, y \in \mathcal{D}$ and $\alpha, \beta \in F$. Let $\mathcal{D}, \mathcal{R}$ be inner product spaces. Given a linear map $T : \mathcal{D} \to \mathcal{R}$, the *adjoint* of T, denoted by T^*, is the unique linear map from $\mathcal{R}$ to $\mathcal{D}$ satisfying

$$\langle T(x), y\rangle = \langle x, T^*(y)\rangle, \ \forall x \in \mathcal{D}, \ y \in \mathcal{R}. \tag{2.1.1}$$

Adjoints provide flexibility when working with two spaces through linear maps that include duality concepts. We make an extensive use of the adjoint equation (2.1.1).

In this thesis, we closely work with three inner product spaces, $\mathbb{R}^n$, $\mathbb{S}^n$ and $\mathbb{H}^n$. We note that $\mathbb{H}^n$ is a vector space when the scalar field F is chosen to be $\mathbb{R}$. However $\mathbb{H}^n$ is not a vector space if the underlying scalar field is $\mathbb{C}$ since the scalar multiplication is not closed; for example, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{H}^2$, but $i \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} i & 0 \\ 0 & i \end{bmatrix} \notin \mathbb{H}^2$. Throughout this thesis, we work with vector spaces *defined over the scalar field* $\mathbb{R}$.

We define inner products for the spaces $\mathbb{R}^n, \mathbb{S}^n, \mathbb{C}^n$ and $\mathbb{C}^{n\times n}$. We equip the space $\mathbb{R}^n$ with the standard inner product $\langle\cdot,\cdot\rangle_{\mathbb{R}^n} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$\langle x, y\rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} x_i y_i.$$

We equip the space $\mathbb{S}^n$ with the standard trace inner product $\langle\cdot,\cdot\rangle_{\mathbb{S}^n} : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}$

$$\langle X, Y\rangle_{\mathbb{S}^n} = \operatorname{trace}(XY)$$

and the induced norm is $\|X\|_F = \sqrt{\operatorname{trace}(XX)}$. Given a matrix $X \in \mathbb{C}^{m\times n}$, we use the notation $\Re(X)$ and $\Im(X)$ for the real part of X and the imaginary part of X, respectively. We endow the space $\mathbb{C}^n$ with the inner product $\langle\cdot,\cdot\rangle_{\mathbb{C}^n} : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{R}$ defined by

$$\langle x, y\rangle_{\mathbb{C}^n} := \Re(x^* y) = \Re(x)^T \Re(y) + \Im(x)^T \Im(y).$$

We equip the space $\mathbb{C}^{n\times n}$ with the inner product $\langle\cdot,\cdot\rangle_{\mathbb{C}^{n\times n}} : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} \to \mathbb{R}$ defined by

$$\langle X, Y\rangle_{\mathbb{C}^{n\times n}} := \Re(\langle X, Y\rangle) = \operatorname{trace}\left(\Re(X)^T \Re(Y)\right) + \operatorname{trace}\left(\Im(X)^T \Im(Y)\right). \tag{2.1.2}$$

The induced norm is $\|X\|_F = \sqrt{\operatorname{trace}(X^*X)}$. For two Hermitian matrices X, Y, we always have $\operatorname{trace}(XY) \in \mathbb{R}$. For the set of Hermitian matrices, we have the inner product $\langle\cdot,\cdot\rangle : \mathbb{H}^n \times \mathbb{H}^n \to \mathbb{R}$ that is evaluated as

$$\langle X, Y\rangle_{\mathbb{H}^n} = \operatorname{trace}(XY).$$

The *triangular number,* $t(n)$, is defined as $t(n) = \binom{n+1}{2} = n(n+1)/2$. The dimension of $\mathbb{S}^n$, $\dim(\mathbb{S}^n)$, is $t(n)$ as there are only $t(n)$ basis elements needed to span the upper triangular n-by-n matrices. The dimension of $\mathbb{H}^n$ is n^2 as there are real and imaginary parts of the off-diagonal elements. The dimension of $\mathbb{C}^{n\times n}$ over $\mathbb{R}$ is $2n^2$; the first n^2 results from the real part and the remaining n^2 results from the imaginary part.

We let vec denote the usual vectorization map that stacks the columns of a real matrix into a single vector. We use $\operatorname{T}_{\text{upper}} : \mathbb{R}^{n\times n} \to \mathbb{R}^{t(n-1)}$ to define the vectorization mapping of the strict upper triangular part of a real matrix M. We often work with isometric mappings in this thesis. For example, we define $\operatorname{svec} : \mathbb{S}^n \to \mathbb{R}^{t(n)}$ by

$$\operatorname{svec}(M) = \begin{pmatrix} \operatorname{diag}(M) \\ \sqrt{2}\,\operatorname{T}_{\text{upper}}(M) \end{pmatrix}.$$

We define the mapping Hvec, analogously:

$$\operatorname{Hvec} : \mathbb{H}^n \to \mathbb{R}^{n^2}, \quad \operatorname{Hvec}(M) = \begin{pmatrix} \operatorname{diag}(M) \\ \sqrt{2}\,\operatorname{T}_{\text{upper}}(\Re(M)) \\ \sqrt{2}\,\operatorname{T}_{\text{upper}}(\Im(M)) \end{pmatrix}.$$

We define the mapping $\operatorname{Cvec} : \mathbb{C}^{m\times n} \to \mathbb{R}^{2mn}$ by

$$\operatorname{Cvec}(M) = \begin{pmatrix} \operatorname{vec}(\Re(M)) \\ \operatorname{vec}(\Im(M)) \end{pmatrix}.$$

We often use the following two matrix factorizations. Let $X \in \mathbb{S}^n_+$. Then X has a representation

$$X = QDQ^T, \quad \text{for some } Q \text{ orthogonal}, \ D \in \mathbb{S}^n_+. \tag{2.1.3}$$

The factorization (2.1.3) is the standard *spectral decomposition* of X. A matrix X with $\operatorname{rank}(X) = r$ has a *compact spectral decomposition*

$$X = QDQ^T = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} \hat{D} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix}^T = U\hat{D}U^T,$$

where $Q = \begin{bmatrix} U & V \end{bmatrix}$ and $\hat{D} \in \mathbb{S}^r_{++}$. An analogous factorization follows for $X \in \mathbb{H}^n_+$.

Given a matrix $X \in \mathbb{R}^{m\times n}$, a *QR decomposition* of X is a factorization of the form

$$X\Pi = QR, \text{ where } \Pi \text{ is a permutation matrix}^1, Q \in \mathbb{R}^{m\times m} \text{ orthogonal}, R \in \mathbb{R}^{m\times n} \text{ upper triangular.}$$

The QR decompositions have important properties listed below, see [76, Section 5.4.1]:

1. $\operatorname{range}(X) = \operatorname{range}\left(Q(:, 1 : \operatorname{rank}(X)\right)$;
2. The last $m - \operatorname{rank}(X)$ rows of R are 0.

Due to these two properties, the QR decomposition provides a robust numerical tool for determining the rank of the matrix X.

2.2 Convex Analysis Background and Positive Semidefinite Matrices

In this section we present some background in convex analysis.

Let $\mathbb{E}^n$ denote the n-dimensional Euclidean space. A set $\mathcal{C} \subseteq \mathbb{E}^n$ is *convex* if,

$$\forall x, y \in \mathcal{C}, \ \lambda \in [0, 1] \implies \lambda x + (1 - \lambda)y \in \mathcal{C}. \tag{2.2.1}$$

Let $\mathcal{D} \subseteq \mathbb{E}^n$ be a convex set. A function $f : \mathcal{D} \to \mathbb{R} \cup \{+\infty\}$ is *convex* if it satisfies

$$f(\lambda x + (1 - \lambda)y) \le \lambda f(x) + (1 - \lambda)f(y), \ \forall x, y \in \mathcal{D}, \ \lambda \in [0, 1]. \tag{2.2.2}$$

If $x \notin \mathcal{D}$, we define $f(x) = +\infty$. Given a function $f : \mathcal{D} \to \mathbb{R} \cup \{+\infty\}$, the *subdifferential* of f at $x \in \mathbb{E}^n$, denoted by $\partial f(x)$, is the set

$$\partial f(x) = \{s \in \mathbb{E}^n : f(y) \ge f(x) + \langle s, y - x\rangle, \ \forall y \in \mathcal{D}\}.$$

The inequality in the definition of $\partial f(x)$ is often called the subgradient inequality.

A *cone* $\mathcal{K} \subseteq \mathbb{E}^n$ is a set of points that satisfies

$$\mathcal{K} = \{\alpha x : x \in \mathcal{K}, \ \alpha \in \mathbb{R}_+\}.$$

[1] Π is also for sparse R.

A cone $\mathcal{K} \subseteq \mathbb{E}^n$ is *pointed* if $x \in \mathcal{K}$ and $-x \in \mathcal{K}$ imply $x = 0$. The *dual cone* of $\mathcal{K}$, denoted by $\mathcal{K}^*$, is the set of points

$$\mathcal{K}^* = \{y \in \mathbb{E}^n : \langle x, y \rangle \geq 0, \ \forall x \in \mathcal{K}\}.$$

A cone $\mathcal{K} \subseteq \mathbb{E}^n$ is called *self-dual* if $\mathcal{K} = \mathcal{K}^*$. In this thesis, we closely work with the cones $\mathbb{R}^n_+$, $\mathbb{S}^n_+$ and $\mathbb{H}^n_+$; they are convex, closed, pointed, self-dual and have nonempty interior. Given a convex set $\mathcal{C} \subseteq \mathbb{E}^n$, the *normal cone* to $\mathcal{C}$ at $\bar{x} \in \mathcal{C}$ is

$$\mathcal{N}_{\mathcal{C}}(\bar{x}) = \{s \in \mathbb{E}^n : \langle s, x - \bar{x} \rangle \leq 0, \ \forall x \in \mathcal{C}\}.$$

We list important definitions related to the faces of $\mathbb{S}^n_+$. The definitions related to faces of $\mathbb{S}^n_+$ naturally generalize to the ones of $\mathbb{H}^n_+$.

Definition 2.2.1. *Let $\mathcal{K} \subseteq \mathbb{E}^n$ be a closed convex cone.*

1. *(face) A convex cone F is a face of $\mathcal{K}$ (denoted $F \trianglelefteq \mathcal{K}$) if,*

   $$\text{for } x, y \in \mathcal{K} \text{ with } \{\lambda x + (1-\lambda)y : \lambda \in (0,1)\} \subseteq F, \text{ we have } x, y \in F.$$

2. *(minimal face) The minimal face of $\mathcal{C} \subseteq \mathcal{K}$, $\operatorname{face}(\mathcal{C}, \mathcal{K})$, is the intersection of all faces of $\mathcal{K}$ containing $\mathcal{C}$.*

3. *(exposed face, exposing vector) A face F is exposed if it is the intersection of $\mathcal{K}$ and a hyperplane. In other words, F admits the representation*

   $$F = \mathcal{K} \cap z^{\perp}, \text{ for some } z \in \mathcal{K}^*.$$

   The vector z is called an exposing vector of F.

Given a convex set $\mathcal{C} \subseteq \mathbb{E}^n$, a point $x \in \mathcal{C}$ is called an *extreme point* if, for all $y, z \in \mathcal{C}$, $x = \frac{1}{2}(y+z)$ implies $x = y = z$. Every extreme point is itself a face and it has the dimension 0.

Faces of $\mathbb{R}^n_+$ display simple representations. If $F \trianglelefteq \mathbb{R}^n_+$, then there exists an index set $\mathcal{I} \subseteq \{1, \ldots, n\}$ such that

$$F = \{x \in \mathbb{R}^n_+ : x_i = 0, \ i \in \mathcal{I}\}.$$

The exposing vector for the face F is any vector $z \in \mathbb{R}^n_+$ such that $\operatorname{supp}(z) = \{1, \ldots, n\} \setminus \mathcal{I}$.

Below is a well-known characterization for the face of $\mathbb{S}^n_+$.

Proposition 2.2.2. *(Characterization of faces of $\mathbb{S}^n_+$) Let $F \subseteq \mathbb{S}^n_+$ be a closed convex cone. Let $\hat{X} \in \operatorname{relint}(F)$ and $\operatorname{rank}(\hat{X}) = r$ with the spectral decomposition*

$$\hat{X} = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix}^T.$$

Then the following are equivalent:

1. *$F \trianglelefteq \mathbb{S}^n_+$;*

2. *$F = \{Y \in \mathbb{S}^n_+ : \operatorname{range}(Y) \subset \operatorname{range}(\hat{X})\}$;*

3. $F = \{Y \in \mathbb{S}^n_+ : \operatorname{null}(Y) \supset \operatorname{null}(\hat{X})\}$;

4. $F = P\mathbb{S}^r_+ P^T$;

5. $F = \mathbb{S}^n_+ \cap (QQ^T)^\perp$.

Lemma 2.2.3 provides important properties of faces of $\mathbb{S}^n_+$.

Lemma 2.2.3. *1. $\mathbb{S}^n_+$ is facially exposed, i.e., every face F of $\mathbb{S}^n_+$ has a representation*

$$F = \mathbb{S}^n_+ \cap Z^\perp, \text{ for some } Z \in (\mathbb{S}^n_+)^* = \mathbb{S}^n_+.$$

2. A face $F \trianglelefteq \mathbb{S}^n_+$ can be characterized by a point in its relative interior. For $\hat{X} \in \operatorname{relint}(F)$, we have

$$F = \{X \succeq 0 : \operatorname{range}(X) \subseteq \operatorname{range}(\hat{X})\} \quad \text{and} \quad \operatorname{relint}(F) = \{X \succeq 0 : \operatorname{range}(X) = \operatorname{range}(\hat{X})\}.$$

Item 1 of Lemma 2.2.3 provides the existence of the object that a facial reduction algorithm tries to find; see Section 2.3.2. The implication of Item 2 of Lemma 2.2.3 is that, an element in the relative interior of a face F characterizes the face (see [38, Proposition 2.2.5] or [139, Corollary 18.1.2]). An alternative characterization for Item 2 of Lemma 2.2.3 is presented in [144]; an element of maximum rank in the face of $\mathbb{S}^n_+$ characterizes the face.

We list properties of the positive semidefinite matrices that we use in subsequent chapters.

Fact 2.2.4. *Let $X, Y \in \mathbb{S}^n_+$. Then the following hold.*

1. $\langle X, Y \rangle = 0 \iff XY = 0$.

2. $\operatorname{range}(X) + \operatorname{range}(Y) = \operatorname{range}(X + Y)$.

3. $[X, Y \succeq 0 \text{ and } X + Y = 0] \implies [X = Y = 0]$.

The followng is a well-known test for failure of positive semidefiniteness of a matrix, and follows from having a negative definite 2×2 principal minor.

Fact 2.2.5. *A matrix $X \in \mathbb{S}^n$ with the element '0' on the i-th diagonal is not positive semidefinite if the i-th row or column of X contain a nonzero element.*

2.3 Interesting Results in Semidefinite Programming

In this section we provide some preliminary results in the area of semidefinite programming. We first present the basic framework for semidefinite programming in Section 2.3.1. In Section 2.3.2, we present some known results on facial reduction.

2.3.1 Results in Semidefinite Programming

For simplicity, we discuss semidefinite programming over the space of real symmetric matrices. Analogous arguments follow for the Hermitian matrices. For given matrices $A_i \in \mathbb{S}^n$ for $i \in [m]$, we define the linear transformation $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ by

$$(\mathcal{A}(X))_i = \langle A_i, X \rangle, \quad \text{for } i \in [m].$$

Let $b \in \mathbb{R}^m$ be given and we define an affine set $L := \{X \in \mathbb{S}^n : \mathcal{A}(X) = b\}$. A *spectrahedron* is defined as the intersection of L and the positive semidefinite cone:

$$\mathcal{F} = \{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b \in \mathbb{R}^m\}. \tag{2.3.1}$$

In this thesis, we call minimizing an extended convex function $f : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$ over a spectrahedron a *semidefinite programming* (**SDP**):

$$p^* = \inf_{X \in \mathbb{S}^n} \{ \ f(X) \ : \ \mathcal{A}(X) = b, \ X \succeq 0 \ \}. \tag{2.3.2}$$

If the objective function is linear, i.e., $f(X) = \langle C, X \rangle, C \in \mathbb{S}^n$, then we obtain the standard primal-dual pair of **SDP**:

$$\begin{array}{llll} & p^* = & \inf\limits_{X \in \mathbb{S}^n} & \langle C, X \rangle \\ (\mathcal{P}) & & \text{subject to} & \mathcal{A}(X) = b \\ & & & X \succeq 0 \end{array} \qquad \begin{array}{llll} & d^* = & \sup\limits_{y \in \mathbb{R}^m} & \langle b, y \rangle \\ (\mathcal{D}) & & \text{subject to} & \mathcal{A}^*(y) \preceq C. \end{array} \tag{2.3.3}$$

We have that *weak-duality* $d^* \le p^*$ always holds for the primal-dual pair (2.3.3). And moreover,

$$\langle C, X \rangle \ge \langle b, y \rangle, \quad \text{for all } X \text{ feasible to } (\mathcal{P}), \ y \text{ feasible to } (\mathcal{D}).$$

The well-known duality results for **LP** do not extend completely to **SDP** in general. For the primal-dual pair (2.3.3), even if the value p^* is finite, that does not necessarily mean that $p^* = d^*$ or d^* is attained by a point y feasible for $(\mathcal{D})$. There are instances where: the primal-dual pair has infinite gap between p^* and d^* ([50, Example 2.3.1]), instances that have positive but finite duality gap ([50, Example 2.3.2]), and instances that have zero duality but an optimal value is not attained ([50, Example 2.3.3]). However, under the property of *strict feasibility*, we can establish *strong duality* for the pair (2.3.3).

Definition 2.3.1 (strict feasibility)**.** *We say that the constraint system of $(\mathcal{P})$ is strictly feasible if there exists a positive definite matrix X $(X \succ 0)$ that satisfies $\mathcal{A}(X) = b$. The constraint system of $(\mathcal{D})$ is strictly feasible if there exists $y \in \mathbb{R}^m$ such that $\mathcal{A}^*(y) \prec C$.*

Strict feasibility is generally referred as the *Slater condition.*

Definition 2.3.2 (strong duality for $(\mathcal{P})$)**.** *Suppose that p^* is finite and strict feasibility holds for $(\mathcal{P})$, then*

1. *the primal optimal value and dual optimal value are equal (i.e., $p^* = d^*$); and*
2. *the dual optimal value d^* is attained (i.e., there exists $y \in \mathbb{R}^m$ feasible to $(\mathcal{D})$ such that $d^* = \langle b, y \rangle$).*

Strong duality for $(\mathcal{D})$ holds analogously. Note that the dual of the dual is the primal. In addition, note that the converse implication of Definition 2.3.2 is false. Even when the primal and the dual optimal values are finite and attained, the Slater condition can fail; see [151, Example 2.19]. There are numerous instances where strict feasibility fails and we cannot expect strong duality to hold for these instances. Throughout this thesis we see interesting outcomes carried by the absence of strict feasibility. In Section 2.3.2, we present the preprocessing scheme called *facial reduction* that helps avoid these issues. Given an instance of $(\mathcal{P})$ that lacks strict feasibility, the facial reduction scheme forms an equivalent problem in a smaller dimensional space so that the reformulated problem possesses strictly feasible points.

We now present a constraint qualification that is closely related to the stability of solutions.

Definition 2.3.3 (Mangasarian-Fromovitz)**.** *The Mangarasian-Fromovitz constraint qualification (**MFCQ**) holds for* $(\mathcal{P})$ *if the two conditions below hold:*

1. *There is a strictly feasible point to* $(\mathcal{P})$*; and*
2. *The linear map* $\mathcal{A}$ *is surjective.*

The **MFCQ** is a stronger condition than the Slater condition. The additional requirement, Item 2, is important for establishing the stability of the equality system. If the linear map $\mathcal{A}$ fails to be surjective, then there always exists a perturbation to the right-hand-side $b \in \mathbb{R}^m$ that renders the equality $\mathcal{A}(X) = b$ inconsistent. Such perturbations are easily detected by the use of a QR decomposition of a matrix representation of $\mathcal{A}$. We can rephrase Item 2 of Definition 2.3.3 and we often refer to these alternatives in Chapters 3 and 4:

1. No redundant equalities in the feasible system $\mathcal{A}(X) = b$;
2. The matrix representation of the linear map $\mathcal{A}$ is full-row rank, i.e., the rank is equal to m.

Item 1 of Definition 2.3.3 (Slater condition) cannot be weakened while still maintaining stability. Even when $\mathcal{A}$ is given surjective, the lack of strict feasibility *implicitly* makes $\mathcal{A}$ lose surjectivity. In particular, in Section 2.3.2, we observe that the lack of strict feasibility inevitably causes $\mathcal{A}$ to forfeit surjectivity in conjuction with the cone constraint $X \in \mathbb{S}^n_+$. We discuss the consequences of this observation throughout Chapters 3 and 4.

2.3.2 Facial Reduction and Singularity Degree

In this section we provide some well-known results on facial reduction and its related definitions. We discuss the goal of facial reduction and the number of steps of the facial reduction algorithm for **SDP**. We then discuss facial reduction for **LP**s.

Why We Perform Facial Reduction Facial reduction, **FR**, is a useful preprocessing framework for acquiring a model with strict feasibility. First introduced by Borwein and Wolkowicz [21, 22] in the 80's, facial reduction appears in many places in the literature e.g., [129, 133, 135, 146]. The most attractive by-product of **FR** is the *stability* of the reformulated model. Facial reduction has proven successful in many applications, especially those that arise from **SDP** relaxations of hard combinatorial optimization problems [27, 28, 48, 79, 93, 123, 166]. It is particularly useful when solving a class of problems that bear common and special structures as it avoids the need for performing **FR** repeatedly.

We first present an important lemma.

Lemma 2.3.4. *(Theorem of the alternative) [50, Theorem 3.1.3] For the feasible constraint system (2.3.1), exactly one of the following statements holds:*

1. *There exists $X \succ 0$ such that $\mathcal{A}(X) = b$ (strict feasibility holds),*
2. *There exists $y \in \mathbb{R}^m$ such that*

$$\boxed{\mathcal{A}^*(y) \in \mathbb{S}^n_+ \setminus \{0\}\ ,\ \langle b, y\rangle = 0.} \tag{2.3.4}$$

Lemma 2.3.4 plays a central role in facial reduction algorithms. We emphasize that whenever a spectrahedron fails to have a strictly feasible point, there always exists $y \in \mathbb{R}^m$ that certifies the lack of strict feasibility. We make extensive use of the *auxiliary system* (2.3.4) in this thesis.

Facial reduction for $\mathcal{F}$ is a process of identifying the minimal face of $\mathbb{S}^n_+$ containing the feasible set $\mathcal{F}$. In other words, the **FR** process tries to find face$(\mathcal{F}, \mathbb{S}^n_+)$, where $\mathcal{F}$ is given in (2.3.1) (see Figure 2.3.1 for a graphical representation.). Since $\mathbb{S}^n_+$ is facially exposed (see Lemma 2.2.3), the process can be characterized as revealing an exposing vector for face$(\mathcal{F}, \mathbb{S}^n_+)$. Below, we present a pseudo-code for the **FR** algorithm. More details can be found in [22, 50, 134, 144].

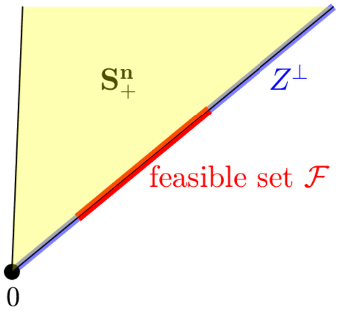

Figure 2.3.1: An exposing vector that exposes the minimal face containing $\mathcal{F}$

Algorithm 2.3.1 Pseudo-Code for Facial Reduction Algorithm for Spectrahedron

```
Require: data $(\mathcal{A}, b)$ for spectrahedon $\mathcal{F} = \{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b\}$, $r = n$
while there exists a solution $y$ satisfying $\mathcal{A}^*(y) \in \mathbb{S}^r_+ \setminus \{0\}$, $\langle b, y\rangle = 0$ do
    Find a nonzero exposing vector $Z$, i.e., face$(\mathcal{F}, \mathbb{S}^r_+) \subseteq \mathbb{S}^r_+ \cap Z^\perp$
    Compute a full column rank matrix $V$ such that range$(V)$ = null$(Z)$
    Set $\mathcal{A} \leftarrow \mathcal{A}_V(\cdot) = \mathcal{A}(V(\cdot)V^T)$, $r \leftarrow$ rank$(V)$
end while
```

Finding an *exposing vector* Z in Algorithm 2.3.1 is generally done by identifying a solution to the auxiliary system (2.3.4). More specifically, $0 \neq Z = \mathcal{A}^*(y) \succeq 0$ in (2.3.4) serves as an exposing vector, i.e.,

$$\text{for all } X \text{ feasible to } \mathcal{F},\ 0 = \langle b, y\rangle = \langle \mathcal{A}(X), y\rangle = \langle X, \mathcal{A}^*(y)\rangle \implies \text{range}(X) \subseteq \text{null}(\mathcal{A}^*(y)). \tag{2.3.5}$$

Namely, the nonzero exposing vector $\mathcal{A}^*(y)$ confines the range of feasible points and it gives rise to line 3 in Algorithm 2.3.1. Every feasible solution X lies in the hyperplane normal to $\mathcal{A}^*(y)$. The implication in (2.3.5) follows from Item 1 of Fact 2.2.4.

Since the exposing vector $Z = \mathcal{A}^*(y)$ confines the range of the feasible points, we can capture all feasible points in the congruence

$$VRV^T \in \mathbb{S}^n_+, \text{ where } R \in \mathbb{S}^r_+ \text{ for some } r \le n \text{ and } \operatorname{range}(V) = \operatorname{null}(Z). \tag{2.3.6}$$

We immediately reduce the dimension of the unknown variable X at each iteration. Instead of working with the cone $\mathbb{S}^n_+$, **FR** allows us to work with a much smaller cone $\mathbb{S}^{n-\operatorname{rank}(Z)}_+$.

We define terminologies related to the exposing vector and V that appear in (2.3.6).

Definition 2.3.5. *Let Z be an exposing vector for a face F, i.e., $F \subseteq \mathbb{S}^n_+ \cap Z^\perp$. Let V be a matrix that satisfies*

$$\textit{for all } X \in F, \textit{ we have } X = VRV^T, \textit{ for some } R \succeq 0.$$

1. *The exposing vector Z is called maximal if it is of the highest rank over all exposing vectors.*
2. *We call the matrix V a facial range vector for F.*
3. *A facial range vector with the minimum number of columns is called a minimal facial range vector.*

Let $V \in \mathbb{R}^{n\times r}$ be a minimal facial range vector for $\mathcal{F}$ and

$$\operatorname{face}(\mathcal{F}, \mathbb{S}^n_+) = V\mathbb{S}^r_+V^T.$$

Then the spectrahedron $\mathcal{F}$ has the following alternative representation

$$\mathcal{F} := \{X \in \mathbb{S}^n : \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\} = \{VRV^T \in \mathbb{S}^n : \mathcal{A}(VRV^T) = b, R \in \mathbb{S}^r_+\}. \tag{2.3.7}$$

Furthermore, we obtain an equivalent reformulation for $(\mathcal{P})$ in (2.3.3):

$$\begin{array}{rcl} p^* & = & \inf\limits_X \{\langle C, X\rangle : \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\} \\ & = & \inf\limits_R \{\langle C, VRV^T\rangle : \mathcal{A}(VRV^T) = b, R \in \mathbb{S}^r_+\} \\ & = & \inf\limits_R \{\langle V^TCV, R\rangle : \mathcal{A}_V(R) = b, R \in \mathbb{S}^r_+\}, \end{array} \tag{2.3.8}$$

where $\mathcal{A}_V(\cdot) = \mathcal{A}(V(\cdot)V^T)$. We also note that there exists a strictly feasible $\hat{R} \in \mathbb{S}^r_{++}$ such that the equality $\mathcal{A}_V(R) = b$ holds. Thus, Slater's condition (Definition 2.3.1) holds for the problem

$$\inf_R \{\langle V^TCV, R\rangle : \mathcal{A}_V(R) = b, R \in \mathbb{S}^r_+\}. \tag{2.3.9}$$

Hence we can establish strong duality (Definition 2.3.2) for (2.3.9).

We often choose facial range vectors V with orthonormal columns. We utilize the simplifications that the orthonormal columns provide; in Section 5.2.4 we use the property $V^TV = I$ to obtain the projection subroutine of the splitting method; in Section 6.3.2 we use the orthonormality to perform **FR** applied to the objective function. It is known that every **FR** step results in at least one equality constraint becoming redundant, see e.g., [144, Section 3.5]. That is, the reduced map $\mathcal{A}_V$ is no longer surjective after **FR**, thus indicating an implicit singularity. We revisit this important property of **FR** and provide a short proof in Chapters 3 and 4.

Length of Facial Reduction Algorithm The **FR** process in Algorithm 2.3.1 does not necessarily end in one iteration. The number of **FR** iterations have a natural upper bound n. The minimum number of **FR** iterations has a special terminology along with important properties.

Definition 2.3.6 (Singularity degree)**.** *[145, 146] Given a spectrahedron $\mathcal{F}$, the* singularity degree *of $\mathcal{F}$, denoted by $\boldsymbol{sd}(\mathcal{F})$, is the smallest number of facial reduction steps needed for identifying the minimal face of $\mathbb{S}^n_+$ containing $\mathcal{F}$.*

The singularity degree,[2] first proposed by Sturm [146], has connections to backward error bounds. A recent development on forward error bounds using singularity degree is given in [144,145]. They argue that a higher singularity degree correlates with a worse error bound and irregular convergence.

It is known that the length, number of iterations, of the **FR** algorithm is the shortest, least, if we choose exposing vectors in

$$\operatorname{relint}\left(\{\mathcal{A}^*(y) : \mathcal{A}^*(y) \succeq 0, \langle b, y\rangle = 0\}\right).$$

In other words, a shortest **FR** algorithm can be achieved by finding a maximum rank solution $\mathcal{A}^*(y)$ of the system (2.3.4) at every iteration.

It is known that the singularity degree admits a tighter upper bound than n.

Fact 2.3.7. *[144, 145] Let $\mathcal{F}$ be a nonempty spectrahedron such that $\mathcal{F} \neq \{0\}$. Then the singularity degree of $\mathcal{F}$ satisfies the following bound:*

$$\boldsymbol{sd}(\mathcal{F}) \leq \min\{n-1, m\}.$$

The original proof for the bound Fact 2.3.7 is nontrvial. We revisit Fact 2.3.7 and provide a considerably simplified proof in Section 3.2 (Corollary 3.2.9) in conjuction with the Barvinok-Pataki bound.

There are different techniques for **FR** known in the literature. Steps for **FR** can be done by simply observing the structure of the data matrices $A_i \in \mathbb{S}^n$. If one of the constraints has the form

$$A_i \succeq 0, \ \operatorname{trace}(A_i X) = 0, \ \text{ for some } i, \tag{2.3.10}$$

then A_i itself is an exposing vector. The *sieve facial reduction method* [167] uses this idea. **FR** that uses the elementary operations and rotations of the data is proposed in [119]. **FR** can also be performed by exploiting special structure of the problem, and there are many successful applications of this type e.g., [27, 28, 48, 79, 123, 166].

[2] [146] uses the terminologies, the level of singularity or the degree of singularity.

2.3.3 Facial Reduction in Linear Programming, LP

In this section we discuss **FR** applied to the class of **LP**s. The ideas above for **FR** for **SDP** naturally apply to **LP** by changing the partial order from the cone $\mathbb{S}^n_+$ to the nonnegative orthant $\mathbb{R}^n_+$. However detailed descriptions of **FR** for **LP** rarely appear in the literature. We provide the details on how the facial range vector and exposing vector are formed for the class of **LP**.

In this section we let

$$\mathcal{F} := \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\},$$

where $A \in \mathbb{R}^{m\times n}$ is a matrix with linearly independent rows. The action of **FR** on the set $\mathcal{F}$ has a simple interpretation:

detect variables that are *fixed at* 0.

We describe how the set $\mathcal{F}$ is represented after **FR**. Suppose that strict feasibility fails for $\mathcal{F}$. Then Lemma 2.3.4 implies that there must exist a nonzero $0 \neq z = A^T y, y \in \mathbb{R}^m$, satisfying

$$\langle x, z\rangle = \langle x, A^T y\rangle = \langle Ax, y\rangle = \langle b, y\rangle = 0,\ \forall x \in \mathcal{F}. \tag{2.3.11}$$

Hence, every $x \in \mathcal{F}$ is perpendicular to the nonnegative vector $A^T y$:

$$A^T y = \sum_{i=1}^{m} y_i a_i = z = \begin{pmatrix} z^+ > 0 \\ 0 \end{pmatrix},\ z^+ \in \mathbb{R}^{s_z},\ s_z < n.$$

We call this vector $z = A^T y$ an *exposing vector* for $\mathcal{F}$, and let the cardinality of its support be $s_z = |\{i : z_i > 0\}|$. Then $z = \sum\limits_{j=1}^{s_z} z_{t_j} e_{t_j}$, where t_j is in increasing order. We now have

$$0 = \langle z, x\rangle \ \text{ and } \ x, z \in \mathbb{R}^n_+ \implies x_i z_i = 0,\ \forall i,$$

i.e., the positive elements in z fix the corresponding elements in x to zero. Then $x = \sum\limits_{j=1}^{n-s_z} x_{s_j} e_{s_j}$, where s_j is in a increasing order. We define the matrix with unit vectors for columns

$$V = \begin{bmatrix} e_{s_1} & e_{s_2} & \dots & e_{s_{n-s_z}} \end{bmatrix} = I_n(:, \operatorname{supp}(z)^c) \in \mathbb{R}^{n\times(n-s_z)}.$$

Then, as we obtained an equivalent representation for the spectrahedron in (2.3.7), we similarly have

$$\mathcal{F} = \{x \in \mathbb{R}^n_+ : Ax = b\} = \{x = Vv \in \mathbb{R}^n : AVv = b, v \in \mathbb{R}^{n-s_z}_+\}. \tag{2.3.12}$$

We call this matrix $V \in \mathbb{R}^{n\times(n-s_z)}$ a *facial range vector*; see Definition 2.3.5. The facial range vector confines the range that every feasible x can have. We use the identification (2.3.12) throughout Chapter 4. The operation AV has a simple interpretation. The role of the facial range vector V is to discard the elements of variable of x (or the columns of the matrix A) that are identically 0 in the set $\mathcal{F}$. Similar to the **SDP** case, the system (2.3.12) also contains redundant equalities and we derive important consequences in Chapter 4.

FR for **LP** exhibits some attractive properties that **FR** for **SDP** does not have. One such property concerns the singularity degree. We recall that the singularity degree for a spectrahedron can exceed one. However, for **LP**s, it is known that **FR** can be done in *one* iteration, i.e., $\mathbf{sd}(\mathcal{F}) \leq 1$; see [50, Theorem 4.4.1]. This is due to the fact that the image $A(\mathbb{R}^n_+)$ is a polyhedron. Thus, $A(\mathbb{R}^n_+)$

is facially exposed. Therefore, face$(b, A(\mathbb{R}^n_+))$, the minimal face of $A(\mathbb{R}^n_+)$ containing b, is exposed.

Another nice property of **FR** for **LP** follows from the sparsity pattern of the data matrix A. Maintaining the sparsity is important for solving **LP**s with large sizes. We recall that the **FR** for **SDP** alters the linear map $\mathcal{A}$ by

$$(\mathcal{A}_V)_i = (\mathcal{A}(V(\cdot)V^T))_i = \langle V^T A_i V \ , \ \cdot \rangle.$$

That is, the data matrix A_i is replaced by the matrix $V^T A_i V$, where V is a facial range vector. This multiplication generally changes the sparsity pattern of the data matrix. Furthermore, if V is chosen to be dense, it renders the data matrix $V^T A_i V$ dense. Unlike **FR** for **SDP**s, the **FR** performed on **LP**s does not alter the sparsity pattern of the data matrix A other than deleting columns and rows.

2.4 Algorithms

We now present some preliminary discussions on the two classes of algorithms that are presented and discussed in Part II of this thesis. In Section 2.4.1 we introduce splitting methods that arise in problems with two linearly related variables. We place a particular interest on the **SDP** relaxations of hard combinatorial optimization problems. These relaxations do not have strictly feasible points and this brings the need for **FR**. The splitting method provides a convenient framework for capturing the natural variable splitting provided by the **FR**. In Section 2.4.2, we introduce the Guass-Newton method and its applications for finding optimal primal-dual solution pair for **SDP**. The Gauss-Newton method provides a useful framework for handling the overdetermined nonlinear system that originates from the complementarity condition of the optimal primal-dual pair. Moreover, the Gauss-Newton method provides a stable computation for the search directions and avoids the need for using the symmetrization that often appear in the interior point methods for **SDP**.

2.4.1 Splitting Methods

Splitting methods allow effective ways to distribute constraints that are difficult to engage simultaneously. Let $\mathcal{X} \subseteq \mathbb{R}^{n_A}, \mathcal{Y} \subseteq \mathbb{R}^{n_B}$ be well-understood closed convex sets and let $f : \mathbb{R}^{n_A} \to \mathbb{R}$, $g : \mathbb{R}^{n_B} \to \mathbb{R}$ be convex functions. Let $A \in \mathbb{R}^{m \times n_A}$, $B \in \mathbb{R}^{m \times n_B}$. Suppose that we are given a problem of the form

$$\min_{x,y} \{ f(x) + g(y) : Ax + By = b, x \in \mathcal{X}, y \in \mathcal{Y} \}. \tag{2.4.1}$$

The two variables in (2.4.1) are linked by the linear equation $Ax + By = b$. The *augmented Lagrangian*, $\mathcal{L}_A$, of (2.4.1) is

$$\mathcal{L}_A(x, y, z) := f(x) + g(y) + \langle z, Ax + By - b \rangle + \frac{1}{2} \| Ax + By - b \|_2^2.$$

Splitting methods solve (2.4.1) by the following sequence of iterations (with some modifications if needed):

$$\begin{array}{rcll} x^{k+1} & = & \min\limits_{x\in\mathcal{X}} \mathcal{L}_A(x, y^k, z^k) & (x\text{-subproblem}) \\ y^{k+1} & = & \min\limits_{y\in\mathcal{Y}} \mathcal{L}_A(x^{k+1}, y, z^k) & (y\text{-subproblem}) \\ z^{k+1} & = & z^k + \alpha \left(Ax^{k+1} + By^{k+1} - b\right) & (\text{dual update with steplength } \alpha). \end{array} \tag{2.4.2}$$

The update rules (2.4.2) are called by the *alternating direction method of multipliers*, **ADMM**, [71]. The solutions for the x- and y-subproblems can often be found analytically; and this promotes the general efficiency of the algorithm.

There are many applications that are posed in the form (2.4.1) in the literature, e.g., finding a common point in the intersection of two sets and ℓ_1-regularization. Many applications of this type can be found in [11,24,72] and the references therein. In this thesis, we place a particular interest on a beautiful application of splitting methods that stems from **FR**. Recall the intermediate problem from (2.3.8):

$$\min_R \{ \langle C, VRV^T \rangle : \mathcal{A}(VRV^T) = b, \ R \in \mathbb{S}^r_+ \}.$$

By assigning $Y = VRV^T$ to the constraint set, we obtain the equivalent problem below:

$$\min_{R,Y} \{ \langle C, Y \rangle : \mathcal{A}(Y) = b, R \in \mathbb{S}^r_+, \ Y = VRV^T \}.$$

We then obtain a problem with two variables where their relation is connected by a linear equation $Y = VRV^T$. Thus, (2.4.2) immediately applies. We may add additional constraints to the variables R and Y that are generally difficult to engage at the same time. Successful applications include [28, 79, 123]. We discuss the application of this type throughout Chapter 5.

2.4.2 The Gauss-Newton Method and Perturbed Optimality Conditions

The *Gauss-Newton method* (e.g., see [43, Chapter 10]) is commonly used for minimizing a sum of squares of nonlinear functions. One of the main advantages of the Gauss-Newton method over the traditional Newton method is that it does not require second-order derivatives of the functions when computing search directions. Let $c : \mathbb{R}^n \to \mathbb{R}^m$ be a continuous vector-valued function and we let $c_i : \mathbb{R}^n \to \mathbb{R}$ be the i-th component of the function c. Suppose that we wish to solve the following nonlinear least squares problem:

$$\min_x f(x) := \frac{1}{2} \sum_{i=1}^m (c_i(x))^2 . \tag{2.4.3}$$

Let J_x be the Jacobian of c evaluated at a point $x \in \mathbb{R}^n$. Using the chain rule and the product rule, we obtain the first and the second order derivatives of f:

$$\nabla f(x) = J_x^T c(x) \ \text{ and } \ \nabla^2 f(x) = J_x^T J_x + \sum_{i=1}^m c_i(x) \cdot \nabla^2 c_i(x).$$

The traditional Newton's method uses the exact Hessian $\nabla^2 f(x)$ for computing the *Newton search direction* $-(\nabla^2 f(x))^{-1} \nabla f(x)$. The Guass-Newton method exploits the fact that we expect the values $|c_i(x)|$ to be small near a optimum, and we therefore discard the second-order derivatives from the Hessian $\nabla f^2(x)$. We get an approximate Hessian $J_x^T J_x$ and the *Gauss-Newton search direction* as follows:

$$\Delta x = -(J_x^T J_x)^{-1} J_x^T c(x).$$

This is equivalent to finding a least squares solution to the overdetermined linear system

$$J_x \Delta x = c(x).$$

The algorithm proceeds with the update $x \leftarrow x + \alpha \Delta x$, where α is a properly chosen steplength.

An important application of the Gauss-Newton method is for finding solutions satisfying the first-order optimality conditions of semidefinite programs. This approach is proposed by [77,99] and motivated by the fact that the optimality conditions for an **SDP** form an overdetermined, bi-linear system. Solving the optimality conditions can be posed in the framework of the Gauss-Newton method. We proceed with the **SDP** with the linear objective function:

$$\min_X \{\langle C, X\rangle : \mathcal{A}(X) = b, \ X \succeq 0\}.$$

We first note that many interior-point based methods try to solve the optimality conditions by solving a sequence of perturbed problems while driving a perturbation (barrier) parameter $\mu \downarrow 0$. The usual approach is to add a barrier term $\mu \log\det(X)$ to the Lagrangian, i.e.,

$$B_\mu(X, y) := \langle C, X\rangle + \langle y, \mathcal{A}(X) - b\rangle - \mu \log\det(X).$$

We obtain perturbed optimality conditions with positive barrier parameter μ as follows. After differentiating the barrier function $B_\mu(X, y)$ with respect to X, we obtain the term μX^{-1}. We set $Z = \mu X^{-1}$, which serves as the dual variable associated with X. Multiplying by X on both sides, we get the equation $XZ - \mu I = 0$, *perturbed complementary slackness*. Hence, the perturbed optimality conditions are

$$\begin{array}{llll} \text{dual feasibility} & (\nabla_x B_\mu = 0) & F_\mu^d(X, y, Z) & := & C + \mathcal{A}^*(y) - Z = 0 \\ \text{primal feasibility} & (\nabla_y B_\mu = 0) & F_\mu^p(X) & := & \mathcal{A}(X) - b = 0 \\ \text{perturbed complementary slackness} & & F_\mu^c(X, Z) & := & XZ - \mu I = 0. \end{array} \tag{2.4.4}$$

Here, the parameter μ gives a measure of the duality gap.

In order to employ the Gauss-Newton method for an optimal triple (X, y, Z) for the primal-dual pair, we construct a real-valued function p_μ that plays the role of f in (2.4.3). Given a fixed $\mu > 0$, the function p_μ is constructed as a sum of squares of nonlinear functions,

$$p_\mu(X, y, Z) = \frac{1}{2}\|F_\mu^d(X, y, Z)\|_F^2 + \frac{1}{2}\|F_\mu^p(X)\|_2^2 + \frac{1}{2}\|F_\mu^c(X, Z)\|_F^2.$$

We emphasize that the system (2.4.4) is overdetermined due to the complementarity $XZ - \mu I$; the product of two symmetric matrices is *not* symmetric in general[3]. Therefore, we cannot apply the Newton's method directly to the system (2.4.4), since the linearization of (2.4.4) yields an overdetermined system, i.e., not a square system.

The function p_μ plays the role of f in (2.4.3). We then apply the Gauss-Newton method to

[3] This does not occur in linear programs since the product of two diagonal matrices remains diagonal.

solve the nonlinear least squares problem $\min\limits_{X,y,Z} \; p_\mu(X,y,Z)$, while $\mu \downarrow 0$. The search direction d_{GN} is obtained by solving the over-determined linear system

$$F'_\mu(X,y,Z)d_{GN} = -F_\mu(X,y,Z), \tag{2.4.5}$$

where F'_μ is the Jacobian of $F_\mu = \left[F^d_\mu; F^p_\mu; F^c_\mu\right]$.

We obtain the *Gauss-Newton direction,* d_{GN}, as the least squares solution

$$d_{GN} = -\left(F'_\mu(X,y,Z)\right)^\dagger F_\mu(X,y,Z),$$

where $\cdot^\dagger$ denotes the Moore-Penrose generalized inverse. In other words, the Gauss-Newton direction is the least squares solution of the linearization $F'_\mu d_{GN} = -F_\mu$. In practice, we do *not* compute the inverse explicitly. Owing to a full column rank assumption, the Gauss-Newton direction is a descent direction (see [43,99]), since

$$\langle \nabla p_\mu, d_{GN}\rangle = \left\langle (F'_\mu)^* F_\mu, -\left((F'_\mu)^* F'_\mu\right)^{-1} (F'_\mu)^* F_\mu \right\rangle < 0. \tag{2.4.6}$$

The inequality follows from the fact that $-\left((F'_\mu)^* F'_\mu\right)^{-1}$ is negative definite.

Instead of solving the system (2.4.5) for the direction $d_{GN} = (\Delta X, \Delta y, \Delta Z)$, we may attempt to solve a smaller system by making variable substitutions or a reduced representation of equalities. In implementations, the steplengths α are made to maintain (sufficient) positive definiteness of the variables X, Z.

The vector-valued function $F_\mu(X,y,Z)$ has the domain and range in different spaces due to the complementarity condition $XZ = \mu I$. There are many techniques available in the literature to overcome this issue [2,87,121,158,165]. Successful search directions are proposed by applying the symmetrized similarity transformation H_P to the complementarity equation:

$$H_P(M) = PMP^{-1} + P^{-T}MP^T.$$

The well-known choices for P in the literature are: $P = I$ (AHO), $P = Z^{\frac{1}{2}}$ (HKM) and $P = \left(X^{\frac{1}{2}}ZX^{\frac{1}{2}}\right)^{\frac{1}{4}} X^{-\frac{1}{2}}$ (NT). We note that the Gauss-Newton approach does not require these symmetrization steps.

Part I

On the Implicit Loss of Surjectivity

Chapter 3

Two-Step Facial Reduction and Implicit Loss of Surjectivity

This chapter is directly motivated by the fact:

*A*t least one linear constraint becomes redundant after each step of **FR**.

Although these redundancies are recognized in the literature, extensive implications have not yet been realized. We examine interesting properties that stem from these redundancies in both **SDP** and **LP**. In particular, each nontrivial step of **FR** reveals the implicit loss of surjectivity in the linear constraints. In this chapter, we elaborate on these redundancies in **SDP**. They give rise to two novel definitions of singularities and we discuss interesting consequences. The redundancies carried by the lack of strict feasibility highlights the importance of the two-step facial reduction algorithm. Furthermore, the redundancies together with the new notions of singularity produce a strengthened Barvinok-Pataki bound.

Pathological behaviours of **SDP** in the absence of strict feasibility are recognized in the literature by many researchers. For example, Sturm [146] introduces the measure singularity degree and relates it to the forward and backward error bounds with respect to $\mathcal{F}$ and shows the bad convergence behaviour under a high singularity degree. Sremac et al., [145] further relate the singularity degree to obtain a lower bound the forward error. Drusvyatskiy et al., [49] show the slow convergence rate of the alternating projection under a high singularity degree. Pataki et al., [119,130] use elementary matrix operations and rotations to understand the pathologies that arise in **SDP**. We understand a source of difficulty under the absence of strict feasibility by using structural properties accompanied by facially reduced system. We provide a comprehensible interpretation and we achieve this by using simple linear algebra.

Contributions and Outline The contribution of this chapter is threefold:

1. We introduce new notions of singularities and highlight the importance of the two-step facial reduction.
2. We use the facially reduced system of the standard spectrahedron to understand the numerical difficulties in the absence of strict feasibility.

3. We use the new notions of singularities to improve the Barvinok-Pataki bound.

This chapter is organized as follows. In Section 3.1 we introduce the two-step facial reduction and present new related notions. In Section 3.2 we use the new notions of singularity to improve the Barvinok-Pataki bound [8, 128].

3.1 The Two-Step Facial Reduction Algorithm

In this section we introduce the two-step facial reduction and address its importance. We recall from (2.3.8) that after **FR**, the feasible set $\mathcal{F}$ is reduced to, with generally $r < n$,

$$\left\{R \in \mathbb{S}^r_+ : \langle V^T A_i V, R\rangle = b_i,\ i \in [m]\right\}. \tag{3.1.1}$$

We observe the equalities that remain in the facially reduced system. That some redundant constraints arise after each step of **FR** algorithm is proved using the property $\operatorname{null}(\mathcal{A}^*) \subsetneq \operatorname{null}(\mathcal{A}^*_V)$, where V is a facial range vector; see e.g., [144]. We now provide a simpler proof using simple arguments from linear algebra.

Lemma 3.1.1. *Suppose that the exposing vector $Z = \mathcal{A}^* y \neq 0$ in an iteration of **FR**. Then at least one linear equality constraint becomes redundant.*

Proof. (Implicit redundancies in spectrahedra) Let $0 \neq Z = \mathcal{A}^*(y)$ be the exposing vector satisfying the auxiliary system (2.3.4). Let $V \in \mathbb{R}^{n\times r}$ be a facial range vector satisfying $\operatorname{null}(\mathcal{A}^*(y)) = \operatorname{range}(V)$. As $\mathcal{A}^*(y)V = 0$, we see that

$$V^T\mathcal{A}^*(y)V = \sum_{i=1}^m y_i V^T A_i V = 0. \tag{3.1.2}$$

After the reduction the constraints have the equivalent form $\operatorname{trace}(V^T A_i V R) = b_i,\ \forall i$. Since $y \in \mathbb{R}^m$ is a nonzero vector, the matrices in $\{V^T A_i V\}_{i=1,\ldots,m} \subseteq \mathbb{S}^r$ are not linearly independent. □

Lemma 3.1.1 immediately implies that the **FR** process reveals the implicit loss of surjuctivity of the linear map $\mathcal{A}$ that defines the feasible set $\mathcal{F}$. Even when the complete **FR** is performed in the sense that the reformulated system has a strictly feasible point, failure to remove the redundant equalities leaves the system ill-posed. Hence, after each iteration of **FR**, redundant equalities should be removed. This gives rise to the *two-step facial reduction*. Let $\bar{m}$ be the cardinality of a maximal linearly independent subset of $\{V^T A_i V\}_{i=1}^m$, where V is a facial range vector. Let

$$P_{\bar{m}} : \mathbb{R}^m \to \mathbb{R}^{\bar{m}}$$

be the simple projection that chooses a maximal linearly independent members in the set $\{V^T A_i V\}_{i=1}^m$. Then we obtain the following:

$$\{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b\} \quad = \quad \{X = VRV^T : P_{\bar{m}}\mathcal{A}_V(R) = P_{\bar{m}}b,\ R \succeq 0\}. \tag{3.1.3}$$

We elaborate on the importance of this projection by relating it to the *distance to infeasibility* in Section 3.1.1 below. Algorithm 2.3.1 and (3.1.3) give rise to Algorithm 3.1.1, the *two-step facial reduction* algorithm. We illustrate Algorithm 3.1.1 in Examples 3.2.12 and 3.2.13.

Algorithm 3.1.1 The Two-Step Facial Reduction

```
Require: data $(\mathcal{A}, b)$ for spectrahedron $\mathcal{F} = \{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b\}$, set $r = n$
while there exists a solution $y$ satisfying $\mathcal{A}^*(y) \in \mathbb{S}^r_+ \setminus \{0\}$, $\langle b, y\rangle = 0$ do
    step 1
        Find a nonzero exposing vector $Z$, i.e., $\text{face}(\mathcal{F}, \mathbb{S}^r_+) \subseteq \mathbb{S}^r_+ \cap Z^\perp$
        Compute a full column rank matrix $V$ such that $\text{range}(V) = \text{null}(Z)$
    step 2
        Obtain a projection $P_{\bar{m}}$ that identifies maximal linearly independent data in $\{V^T A_i V\}_i$
        Set $\mathcal{A} \leftarrow P_{\bar{m}}\mathcal{A}(V(\cdot)V^T)$, $b \leftarrow P_{\bar{m}} b$, $r \leftarrow \text{rank}(V)$
end while
```

We now recall the notion of singularity degree, the smallest number of **FR** steps (see Definition 2.3.6). We can obtain the smallest number of **FR** steps by finding an exposing vector Z of maximum rank at every iteration in Algorithm 3.1.1. We also recall that the implicit redundant constraints are induced by the vector y satisfying the auxiliary system (2.3.4); see the proof of Lemma 3.1.1. Namely, *any* vector y that satisfies (2.3.4) induces redundant equalities. This observation gives rise to two novel and related notions of singularity degree.

Definition 3.1.2. *Let $\{R \succeq 0 : P_{\bar{m}}\mathcal{A}_V(R) = P_{\bar{m}}b \in \mathbb{R}^{\bar{m}}\}$ be the facially reduced system of $\mathcal{F}$ that satisfies the **MFCQ**.*

1. *The max-singularity degree of $\mathcal{F}$, denoted **maxsd**$(\mathcal{F})$, is the largest number of nontrivial facial reduction iterations for finding the minimal face, $\text{face}(\mathcal{F}, \mathbb{S}^n_+)$.*

2. *The implicit problem singularity, denoted **ips**$(\mathcal{F})$, is the number of implicit redundant equalities in $\mathcal{F}$, i.e., **ips**$(\mathcal{F}) = m - \bar{m}$.*

It is clear that **maxsd**$(\mathcal{F}) \leq$ **ips**$(\mathcal{F})$ since every solution y to the auxiliary system (2.3.4) yields at least one redundant constraint. With these new definitions, we conclude the following relations.

Proposition 3.1.3. *Given a spectrahedron $\mathcal{F}$, it holds that*

$$\textbf{\textit{sd}}(\mathcal{F}) \leq \textbf{\textit{maxsd}}(\mathcal{F}) \leq \textbf{\textit{ips}}(\mathcal{F}). \tag{3.1.4}$$

Various relations hold with the inequalities (3.1.4). We show in Example 3.2.12 below that the inequalities (3.1.4) can hold as equality. In Section 5.5, we show that **sd**, **maxsd** and **ips** are three different notions and their values can be very different, i.e., the equalities in (3.1.4) can be strict.

The implicit redundancies in the facially reduced system were first recognized from the **SDP** relaxation for the quadratic assignment problem [166] in the 90's. Surprisingly, after **FR** all the constraints other than a *portion* of the so-called gangster constraints became redundant. Additional constraints are added to the **SDP** relaxation and some structured redundancies are recognized by forming the dual of the dual and a facial range vector. These implicit redundancies are understood as arising from special structural properties embedded in this particular class of instances, rather than properties found in an arbitrary spectrahedron.

The implicit redundancies of the equality system $\mathcal{A}(X) = b$ are recognized using Lemma 3.1.1. The discovery of the redundancies is not only limited to the system represented in the standard trace inner product form. In Section 5.1 we realize the redundancies by using the embedded structure of the problems. In Section 6.3 we recognize the redundancies of the constraint represented in the sum of matrix congruences.

3.1.1 Instability Originating from Implicit Redundancies in Equality Constraints

In this section we examine numerical difficulties that arise when complete **FR** is not performed. Instability issues arising in the absence of strict feasibility are known in the literature, e.g., [50, 145, 146]. We aim to study this instability through the lens of **FR** and linear algebra. We place a particular interest on the instabilities that stem from the implicit loss of surjectivity of the linear map $\mathcal{A}$ in conjunction with the cone $\mathbb{S}^n_+$. We relate to the notion of the *distance to infeasibility* to the facially reduced system.

The *distance to infeasibility*, pioneered for cone optimization by Renegar, is a measure of the smallest perturbations of the data of a problem that renders the problem infeasible. In our setting, we can use the following simplification of the distance to infeasibility from [136] by restricting the perturbations to b, i.e., we can force infeasibility using only perturbations in b;

$$\operatorname{dist}(b, \mathcal{F} = \emptyset) := \inf \left\{ \|b - \tilde{b}\| \ : \ \{X : \mathcal{A}(X) = \tilde{b}, \ X \in \mathbb{S}^n_+\} = \emptyset \right\}.$$

Many interesting bounds, condition numbers, are shown in [136, 137] for **LP** under the assumption that the distance to infeasibility is *positive* and known.

Computing the distance to infeasibility is a challenging task, see e.g., [56, 131]. It is known that a positive distance to infeasibility of $\mathcal{F}$ implies that strict feasibility holds for $\mathcal{F}$; see e.g., [63, 64]. The contrapositive of this statement is that, if strict feasibility fails for $\mathcal{F}$, then the distance to infeasibility is 0. We revisit this statement with the facially reduced system (2.3.12). We provide an elementary proof that there is an arbitrarily small perturbation for the data vector b of $\mathcal{F}$ that yields the set $\mathcal{F}$ infeasible, i.e., $\operatorname{dist}(b, \mathcal{F} = \emptyset) = 0$. Furthermore, we provide an explicit perturbation that renders the set $\mathcal{F}$ empty and show how it is related to the implicit redundancies.

Suppose that $\mathcal{F}$ fails strict feasibility. Recall the representation (2.3.12) for $\mathcal{F}$. Let $\hat{A}$ and $\hat{A}_V$ be the matrices that represent the isometric realizations of $\mathcal{A}$ and $\mathcal{A}_V$, respectively, i.e.,

$$\hat{A} = \begin{bmatrix} \operatorname{svec}(A_1)^T \\ \vdots \\ \operatorname{svec}(A_m)^T \end{bmatrix} \in \mathbb{R}^{m \times t(n)}, \quad \hat{A}_V = \begin{bmatrix} \operatorname{svec}(V^T A_1 V)^T \\ \vdots \\ \operatorname{svec}(V^T A_m V)^T \end{bmatrix} \in \mathbb{R}^{m \times t(r)}.$$

Let $\hat{A}_V = \widehat{Q}\widehat{R}$ be a QR decomposition of $\hat{A}_V$, where $\widehat{Q} \in \mathbb{R}^{m \times m}$ orthogonal, $\widehat{R} \in \mathbb{R}^{m \times t(r)}$ upper triangular. We write $\widehat{Q} = \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 \end{bmatrix}$ so that $\operatorname{range}(\widehat{Q}_1) = \operatorname{range}(\hat{A}_V)$[1]. Then, by the orthogonality of $\widehat{Q}$, we have

$$\mathcal{A}(X) = \hat{A} \operatorname{svec}(X) = \hat{A}_V \operatorname{svec}(R) = b \iff \widehat{Q}^T \hat{A} \operatorname{svec}(X) = \widehat{R} \operatorname{svec}(R) = \widehat{Q}^T b. \tag{3.1.5}$$

Since $\hat{A}_V$ is a rank deficient matrix (see Lemma 3.1.1), the upper triangular matrix $\widehat{R}$ is of the form

$$\widehat{R} = \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} \in \mathbb{R}^{m \times t(r)} \text{ and } \bar{R} \in \mathbb{R}^{\operatorname{rank}(\hat{A}_V) \times t(r)} \text{ with nonzero diagonal.} \tag{3.1.6}$$

[1] We may assume that the columns of $\hat{A}_V$ and the entries of $\operatorname{svec}(R)$ are permuted accordingly to satisfy $\operatorname{range}(\widehat{Q}_1) = \operatorname{range}(\hat{A}_V)$.

Since $b \in \operatorname{range}(\hat{A}_V)$, the last $m - \operatorname{rank}(\hat{A}_V)$ entries of $\widehat{Q}^T b$ are 0, i.e.,

$$\widehat{Q}^T b = \begin{pmatrix} \widehat{Q}_1^T b \\ \widehat{Q}_2^T b \end{pmatrix} = \begin{pmatrix} \widehat{Q}_1^T b \\ 0 \end{pmatrix}.$$

Here, if the facial range vector V completely characterizes $\operatorname{face}(\mathcal{F}, \mathbb{S}^n_+)$, then the implicit problem singularity of $\mathcal{F}$ is equal to $m - \operatorname{rank}(\hat{A}_V)$. Consequently, the unrealized implicit loss of surjectivity produces the system

$$\begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} \operatorname{svec}(R) = \begin{pmatrix} \widehat{Q}_1^T b \\ 0 \end{pmatrix}, \ R \succ 0. \tag{3.1.7}$$

Any perturbation on the last $m - \operatorname{rank}(\hat{A}_V)$ equations in (3.1.7) that cuases the system inconsistency renders the system (3.1.7) infeasible while maintaining the dimension of $\operatorname{relint}(\mathcal{F})$. Namely, large value for $\mathbf{ips}(\mathcal{F}) = m - \operatorname{rank}(\hat{A}_V)$ is a good measure of the ill-posedness of a problem. For instance, replacing the right-hand-side vector in (3.1.7) by $\begin{pmatrix} \widehat{Q}_1^T b \\ \phi \end{pmatrix}$ with any nonzero vector ϕ renders (3.1.7) infeasible. Replacing the data matrix in (3.1.7) by $\begin{bmatrix} \bar{R} \\ \Phi \end{bmatrix}$ for which Φ contains a row $\operatorname{svec}(T)^T$ with positive definite T also renders (3.1.7) infeasible.

We now present a class of perturbations to b that maintains the feasibility of the set $\mathcal{F}$ as well as a special perturbation to b that forces $\mathcal{F}$ to be infeasible. Such perturbations can be found using linear combinations of the columns of $\widehat{Q}_1$ or $\widehat{Q}_2$, respectively. We relate this observation to the solution of the auxiliary system (2.3.4) in the proof of Proposition 3.1.4 below.

Proposition 3.1.4. *Suppose that strict feasibility fails for $\mathcal{F}$ and let $\mathcal{F}$ have the representation*

$$\mathcal{F} = \{X \succeq 0 : \mathcal{A}(X) = b\} = \{VRV^T : \hat{A}_V \operatorname{svec}(R) = b, R \succeq 0\}.$$

Then the following hold.

1. *For all $\Delta b \in \operatorname{range}(\hat{A}_V)$ with sufficiently small norm, the set $\{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b + \Delta b\}$ is feasible.*
2. *Let $\bar{y}$ be a solution to the auxiliary system (2.3.4). Then perturbing the right-hand-side vector b of $\mathcal{F}$ in the direction $\bar{y}$ makes the system $\mathcal{F}$ infeasible. In other words, the distance to infeasibility of $\mathcal{F}$ is 0, i.e., $\operatorname{dist}(b, \mathcal{F} = \emptyset) = 0$.*

Proof. Let Δb be any perturbation in $\operatorname{range}(\hat{A}_V)$. Let $\widehat{Q}\widehat{R} = \hat{A}_V$ be a QR decomposition of $\hat{A}_V$. In particular, let $\widehat{R}$ follow the form (3.1.6) and $\widehat{Q} = \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 \end{bmatrix}$ so that $\operatorname{range}(\widehat{Q}_1) = \operatorname{range}(\hat{A}_V)$. Then

$$\begin{array}{rrcl} & \mathcal{A}(X) & = & \hat{A}_V \operatorname{svec}(R) = b + \epsilon \Delta b \in \mathbb{R}^m \\ \iff & \widehat{R} \operatorname{svec}(R) & = & \widehat{Q}^T b + \epsilon \widehat{Q}^T \Delta b \in \mathbb{R}^m \\ \iff & \bar{R} \operatorname{svec}(R) & = & \widehat{Q}_1^T b + \epsilon \widehat{Q}_1^T \Delta b \in \mathbb{R}^{\operatorname{rank}(\hat{A}_V)}. \end{array} \tag{3.1.8}$$

The last equivalence holds since $\mathcal{A}(X) = b$ and $\Delta b \in \operatorname{range}(\hat{A}_V) = \operatorname{range}(\widehat{Q}_1)$. Since the system $\bar{R} \operatorname{svec}(R) = \widehat{Q}_1^T b$ satisfies the **MFCQ**, the distance to infeasibility of this system is positive. Thus, the perturbed system $\{R \succeq 0 : \bar{R} \operatorname{svec}(R) = \widehat{Q}_1^T b + \epsilon \widehat{Q}_1^T \Delta b\}$ remains feasible. Therefore, by (3.1.8), perturbing b along the direction $\Delta b \in \operatorname{range}(\hat{A}_V)$ maintains the feasibility and this concludes the proof for Item 1.

For Item 2 we present a perturbation Δb to b that renders $\mathcal{F}$ infeasible. By (3.1.2), we have a nonzero vector $\bar{y} \in \mathbb{R}^m$ that satisfies (2.3.4). Then we have

$$\bar{y} \in \operatorname{null}\left((\hat{A}_V)^T\right) = \operatorname{range}(\hat{A}_V)^\perp = \operatorname{range}(\widehat{Q}_2) \implies \bar{y} = \widehat{Q}_2 \bar{u} \text{ for some nonzero } \bar{u}. \tag{3.1.9}$$

We recall Farkas' lemma:

$$\{y \in \mathbb{R}^m : \mathcal{A}^*(y) \succeq 0,\ \langle b, y\rangle < 0\} \neq \emptyset \implies \mathcal{F} = \emptyset.$$

Now, for any $\epsilon > 0$, setting $\Delta b_\epsilon = -\epsilon \bar{y}$ yields

$$\mathcal{A}^*(\bar{y}) \succeq 0,\ \langle b, \bar{y}\rangle = 0 \implies \mathcal{A}^*(\bar{y}) \succeq 0,\ \langle b + \Delta b_\epsilon, \bar{y}\rangle < 0. \tag{3.1.10}$$

Hence, by letting $\epsilon \to 0^+$, we see that the distance to infeasibility, $\operatorname{dist}(b, \mathcal{F} = \emptyset)$, is equal to 0. □

We note that the instability discussed in this section essentially originates from the observation made in Lemma 3.1.1, i.e., redundant equalities arise in the facially reduced system. Facially reduced system allows us to exploit the root of potential instability when the right-hand-side vector b is perturbed. As shown in (3.1.9), the redundancies that originate from the loss of surjectivity precisely give rise to the perturbation to b that renders infeasibility. Moreover, the certificate vector y of the system (2.3.4) is indeed in the range of $\widehat{Q}_2$ that originates from these redundancies. Although the distance to infeasibility is 0 in the absence of strict feasibility, Proposition 3.1.4 suggests that a carefully chosen perturbation of b does not have an impact on the feasibility of $\mathcal{F}$.

The distance to infeasibility directly impacts the measure of well-posedness of the problem, [63, 64, 137]. Given the pair $d = (A, b)$ of the data for an instance $(\mathcal{P})$, the *condition measure* of $(\mathcal{P})$ is defined by

$$\mathcal{C}(d) := \frac{\|d\|}{\inf\{\|\Delta d\| : d + \Delta d \text{ yields } (\mathcal{P}) \text{ infeasible}\}}.$$

The value $\mathcal{C}(d)$ is a measure of well-posedness of the problem $(\mathcal{P})$. Since $\operatorname{dist}(b, \mathcal{F} = \emptyset) = 0$, we have $\mathcal{C}(d) = \infty$. Namely, when strict feasibility fails for $(\mathcal{P})$, the problem is ill-posed.

How to Remove the Redundant Constraints Let $M \in \mathbb{R}^{m \times n}$ be a given matrix. We summarize some available methods for extracting a maximal linearly independent subset of columns of M. (We apply that to M^T and rows below.)

The first method uses a rank-revealing QR decomposition[2]. Let $MI(:, \pi) = QR$ be a QR decomposition such that π is a permutation vector, Q is an orthogonal matrix and R is an upper triangular matrix with a non-increasing diagonal in absolute value. The matrix $I(:, \pi)$ permutes the columns of M. If M has linearly independent columns, then the matrix R contains zeros on its diagonal. Let r be the number of the nonzero diagonal entries of R. Then, $\pi(1 : r)$ returns the subset of columns indices of M that are linearly independent. Another available method makes use of artificial variables [39, Box 8.2]. It constructs $\begin{bmatrix} I & M^T \end{bmatrix}$ and set the initial basis matrix to be the first m columns. Then it performs a variant of the two-phase simplex method to drive the initial

[2] An implementation can be found at https://www.mathworks.com/matlabcentral/fileexchange/77437

basic variables out of the basis one by one. When such an operation is not applicable, a linearly dependent row of M^T is detected. Computational improvements of this method are made in [4]. An alternative numerical method for assessing the rank deficiency is presented in [32, 142].

One of the standard assumptions in **SDP** is the surjectivity of $\mathcal{A}$. Hence, we may use the aforementioned methods above to remove redundant constraints as a part of the preprocessing phase. To find redundant equalities in $\mathcal{A}_V(R) = b$ after the first step of **FR**, we form the isometric realization of the equalities, i.e., $\langle V^T A_i V, R\rangle = \operatorname{svec}(V^T A_i V)^T \operatorname{svec}(R)$, $\forall i \in [m]$. We then form the matrix

$$\hat{A}_V = \begin{bmatrix} \operatorname{svec}(V^T A_1 V)^T \\ \vdots \\ \operatorname{svec}(V^T A_m V)^T \end{bmatrix}$$

and find linearly dependent rows of $\hat{A}_V$ by applying one of the aforementioned methods. We include a graphical illustration of the two-step **FR** process; see Figure 3.1.1. The i-th row of matrix $\hat{A}$ is $\operatorname{svec}(A_i)^T$ below:

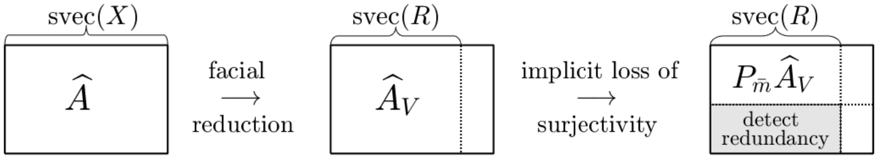

Figure 3.1.1: A graphical illustration of the two-step **FR**

3.2 A Strengthened Barvinok Pataki Bound

In this section we introduce the *Barvinok-Pataki bound* and how the singularity notions from Section 3.1 give rise to a strengthened Barvinok-Pataki bound. We tighten this bound by adding information from the *implicit problem singularity* and *max-singularity degree* of the spectrahedron. We see that this new bound depends not only on the number of affine constraints but also on the *geometry and stability* of the spectrahedron; see Theorem 3.2.7. We first provide some known results on Barvinok-Pataki bound.

Theorem 3.2.1 ([128, Theorem 2.1])**.** *Suppose that $X \in F$, where F is a face of the feasible set of* (2.3.2)*. Let $d = \dim F$, $r = \operatorname{rank} X$. Then*

$$t(r) \le m + d. \tag{3.2.1}$$

Theorem 3.2.2 ([8, Theorem 1.1])**.** *Let $\mathcal{L} \subset \mathbb{S}^n$ be an affine manifold such that the intersection $\mathcal{F} = \mathbb{S}^n_+ \cap \mathcal{L} \neq \emptyset$ and $\operatorname{codim} \mathcal{L} \le t(r+1) - 1$ for some nonnegative integer r. Then there exists $X \in \mathcal{F}$ such that $\operatorname{rank} X \le r$.*

Theorem 3.2.3 ([8, Theorem 1.2])**.** *Let $r > 0, n \ge r + 2$. Let $\mathcal{L} \subset \mathbb{S}^n$ be an affine manifold such that the intersection $\mathcal{F} = \mathbb{S}^n_+ \cap \mathcal{L} \neq \emptyset$ and bounded, and $\operatorname{codim} \mathcal{L} = t(r+1)$, for some nonnegative integer r. Then there exists $X \in \mathcal{F}$ such that $\operatorname{rank} X \le r$.*

Remark 3.2.4. *Theorems 3.2.1 to 3.2.3 all concern bounds on the rank of a feasible point of a spectrahedron. We continue with some remarks for the three theorems above.*

Given the number of constraints, Theorem 3.2.1 gives an upper bound on the rank of a solution. The most well-known application of Theorem 3.2.1 is the case of extreme points. An extreme point X of a convex set C is a point that cannot be expressed as a convex combination of any two distinct points in C. The minimal face containing an extreme point X is 0-dimensional, i.e., $\dim(\text{face}(\{X\})) = 0$. *From (3.2.1), we conclude that*

$$t(\text{rank}(X)) \leq m, \quad \textit{for all extreme points } X \in \mathcal{F}. \tag{3.2.2}$$

Theorem 3.2.2 is a consequence of [10, Theorem 1.3] (see also [9, Section IV.10.3]), and can be interpreted as follows. For the feasible constraint system of (2.3.2), there is a solution X such that its rank is bounded by $\left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor$. We may obtain an equivalent bound by defining the smallest $r \in \mathbb{N}$ satisfying $\binom{r+2}{2} > m$. Therefore if we have $\binom{r+2}{2} - 1 \geq m$, where m is the number of linearly independent constraints, we obtain the statement in Theorem 3.2.2.

Theorem 3.2.3 is stated for a bounded spectrahedron. Suppose that we are given a triple (r, m, n), where r is an upper bound on the target rank; $m = \binom{r+2}{2}$ is the number of linearly independent constraints; and the embedding space $\mathbb{S}^n$ satisfies $n \geq r + 2 \geq 3$. Then there exists a point $X \in \mathcal{F}$ such that $\text{rank}(X) \leq r$.

In this thesis we refer to the *Barvinok-Pataki bound* as given in (3.2.2) and in the following.

Theorem 3.2.5. *(Barvinok-Pataki bound [8, 128]) Every extreme point $X \in \mathcal{F}$ satisfies*

$$t(\text{rank}(X)) \leq m.$$

The *Barvinok-Pataki bound* [8, 128] guarantees the existence of a feasible point X of rank r satisfying $t(r) \leq m$. In other words, the rank of extreme points only depends on the number of affine constraints.

The Barvinok-Pataki bound shows its importance in many areas of applications. In particular, the Barvinok-Pataki bound provides targets on rank when low-rank solutions are desired. Low rank targets are used in splitting methods when using nonconvex, low rank, projections onto the **SDP** cone, e.g., [123]. Furthermore, having a *low rank target* provides efficiency in low rank **SDP** algorithms where the nonlinear formulation $X \leftarrow VV^T$ is used, e.g., [25, 26, 80]. Low rank optimal solutions arising in **SDP** relaxations, e.g., [5, 101, 158], and the references therein. Applications that arise in the **SDP** relaxations of protein folding problems, rank three solutions for the Gram matrices are essential since molecules exist in three-dimensional space, e.g., [110]. Rank-one solutions to semidefinite relaxations, liftings, of many nonconvex combinatorial optimization problems are of particular importance, as they guarantee that the global optimum has been found, e.g., [6, 28, 79]. Further applications include the psd-rank and related notions in [58], trust-region subproblems [107], and optimal power flow problems [115, 120].

3.2.1 The Improved Barvinok-Pataki Bound

In this section we tighten the Barvinok-Pataki bound by employing the max singularity degree and implicit problem singularity; see Definition 3.1.2. This is motivated by the dimension reduction property of **FR** and the redundancies in the equality constraint system. We first show that the rank of feasible points are unchanged after facial reduction.

Lemma 3.2.6. *Let $V \in \mathbb{R}^{n\times r}$ be a minimal facial range vector containing a convex set $\mathcal{C}$, i.e., $V\mathbb{S}^r_+V^T \supseteq \mathcal{C}$. Then, for $R \in \mathbb{S}^r_+$ and $VRV^T \in \mathcal{C}$, we have* $\operatorname{rank}(VRV^T) = \operatorname{rank}(R)$.

Proof. Suppose that $\operatorname{rank}(R) = \hat{r} \le r$. Then R has the spectral decomposition $R = \sum_{i=1}^{r} \lambda_i x_i x_i^T$, where some eigenvalues λ_i are possibly 0. Then we have

$$VRV^T = \sum_{i=1}^{r} \lambda_i Vx_i(Vx_i)^T.$$

Let $X_r = \begin{bmatrix} x_1 & \cdots & x_r \end{bmatrix}$ and consider the equation $\begin{bmatrix} Vx_1 & \cdots & Vx_r \end{bmatrix} a = VX_r a = 0$ for $a \in \mathbb{R}^r$. Then we have

$$V^TVX_r a = 0 \implies X_r a = 0 \implies a = 0.$$

That is, $\{Vx_i\}_{i=1}^r$ is a set of linearly independent vectors. Since VRV^T is the sum of $\hat{r}$ number of rank one matrices that are linearly independent, we conclude $\operatorname{rank}(VRV^T) = \operatorname{rank}(R)$. □

Given a facially reduced **SDP** with the variable in the congruence VRV^T and $R \in \mathbb{S}^r_+$, we have $\operatorname{rank}(VRV^T) = \operatorname{rank}(R)$, when V is a minimal facial range vector with orthonormal columns. Namely, the solution rank of the original **SDP** is completely determined by the solution rank of the facially reduced problem. Using Fact 2.3.7 and Lemma 3.1.1, we show a tighter upper bound on rank.

Theorem 3.2.7. *(A strengthened Barvinok-Pataki bound) Suppose that the singularity degree of the nonempty spectrahedron $\mathcal{F}$ satisfies $\textbf{sd}(\mathcal{F}) > 0$. Then every extreme point $X \in \mathcal{F}$ with $r = \operatorname{rank}(X)$ satisfies*

$$t(r) \le \min\left\{t(n - \textbf{maxsd}(\mathcal{F})),\ m - \textbf{ips}(\mathcal{F})\right\}.$$

Proof. We note that every feasible point of $\mathcal{F}$ is in the cone $\tilde{V}\mathbb{S}^{n-q}_+\tilde{V}^T$, for some facial range vector $\tilde{V} \in \mathbb{R}^{n\times(n-q)}$ and $q \ge \textbf{maxsd}(\mathcal{F})$. That is, every feasible point can be embedded in the cone $\mathbb{S}^{n-\textbf{maxsd}(\mathcal{F})}_+$. Hence every $X \in \mathcal{F}$ satisfies $\operatorname{rank}(X) \le m - \textbf{maxsd}(\mathcal{F})$. Since t is monotonic over the positive real line, $t(\operatorname{rank}(X)) \le t(n - \textbf{maxsd}(\mathcal{F}))$ follows.

We recall Proposition 3.1.3. We immediately have

$$m - \textbf{ips}(\mathcal{F}) \le m - \textbf{maxsd}(\mathcal{F}) \le m - \textbf{sd}(\mathcal{F}).$$

Then the upper bound $m - \textbf{ips}(\mathcal{F})$ of $t(r)$ follows from Theorem 3.2.5 and Lemma 3.2.6. □

3.2.2 Immediate Consequences and Examples

We provide immediate consequences of Theorem 3.2.7. Corollary 3.2.8 below explicitly shows that the singularities improve the bound on rank in Theorem 3.2.5.

Corollary 3.2.8. *Let $\boldsymbol{sd}(\mathcal{F}) > 0$. Then every extreme point X to (2.3.1) satisfies*

$$\operatorname{rank}(X) \le \left\lfloor \frac{\sqrt{1+8\min\left\{t(n-\boldsymbol{maxsd}(\mathcal{F})),\ m-\boldsymbol{ips}(\mathcal{F})\right\}}-1}{2} \right\rfloor.$$

Proof. This follows from the definition of the triangular number and the integrality of the rank function. □

We recall Fact 2.3.7 ([144,145]), an upper bound on the singularity degree, $\mathbf{sd}(\mathcal{F}) \le m$. We extend Fact 2.3.7 by using the strengthened Barvinok-Pataki bound, Theorem 3.2.7.

Corollary 3.2.9. *For the spectrahedron $\mathcal{F}$, the followings holds:*

$$\boldsymbol{sd}(\mathcal{F}) \le \boldsymbol{maxsd}(\mathcal{F}) \le \boldsymbol{ips}(\mathcal{F}) \le m - \max\{\ t(\operatorname{rank}(X)) :\ \textit{extreme point } X \textit{ of } \mathcal{F}\ \} \le m.$$

Proof. Let X be any extreme point of $\mathcal{F}$. Then Theorem 3.2.7 provides

$$t(\operatorname{rank}(X)) \le m - \mathbf{ips}(\mathcal{F}) \implies \mathbf{ips}(\mathcal{F}) \le m - t(\operatorname{rank}(X)). \tag{3.2.3}$$

Since (3.2.3) holds for any extreme point X, we obtain

$$\mathbf{ips}(\mathcal{F}) \le m - \max\{\ t(\operatorname{rank}(X)) :\ \text{extreme point } X \text{ of } \mathcal{F}\ \}.$$

By Proposition 3.1.3, we obtain

$$\mathbf{sd}(\mathcal{F}) \le \mathbf{maxsd}(\mathcal{F}) \le \mathbf{ips}(\mathcal{F}) \le m - \max\{\ t(\operatorname{rank}(X)) :\ \text{extreme point } X \text{ of } \mathcal{F}\ \}.$$

□

An analogous result of Theorem 3.2.7 follows for spectrahedra in $\mathbb{H}^n_+$. The triangular number, $t(r)$ in Theorem 3.2.5, originates from the dimension of $\operatorname{face}(\mathcal{F}, \mathbb{S}^n_+)$. We recall from Section 2.1 that the dimension of $\mathbb{H}^n$ is n^2. Thus, the analogous result of the Barvinok-Pataki bound for the spectrahedron in $\mathbb{H}^n_+$ is

$$(\operatorname{rank}(X))^2 \le m \implies \operatorname{rank}(X) \le \lfloor \sqrt{m} \rfloor,^{3} \tag{3.2.4}$$

where X is any extreme point of $\mathcal{F} \subseteq \mathbb{H}^n_+$; see [107, Theorem 5.1]. The strengthened bound, Theorem 3.2.7, immediately extends to the Hermitian case. Thus, we obtain Corollary 3.2.10 below.

Corollary 3.2.10. *Let $\mathcal{F}$ be a spectrahedron in $\mathbb{H}^n_+$. Then for all extreme point $X \in \mathcal{F}$, it holds that*

$$(\operatorname{rank}(X))^2 \le \min\left\{(n-\boldsymbol{maxsd}(\mathcal{F}))^2,\ m-\boldsymbol{ips}(\mathcal{F})\right\}.$$

□

Remark 3.2.11 below discusses the attainment in the upper bound $\min\{t(n-\mathbf{maxsd}(\mathcal{F})), m-\mathbf{ips}(\mathcal{F})\}$, i.e., how the minimizer of $\min\{t(n-\mathbf{maxsd}(\mathcal{F})), m-\mathbf{ips}(\mathcal{F})\}$ is determined by the relationships among $m, n, \mathbf{maxsd}(\mathcal{F})$ and $\mathbf{ips}(\mathcal{F})$. We include the relation for interest.

[3] Theorem 3.2.5 gives rise to the bound $\operatorname{rank}(X) \le \left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor$. Since $\lfloor \sqrt{m} \rfloor \le \left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor$, a spectrahedron in $\mathbb{H}^n_+$ gives rise to a tighter bound on the rank than a spectrahedron in $\mathbb{S}^n_+$.

Remark 3.2.11. *Let $\mathcal{F}$ be a spectahedron with $\boldsymbol{sd}(\mathcal{F}) > 0$. Then the following relation holds:*

$$\begin{aligned}&\min\{t(n-\boldsymbol{maxsd}(\mathcal{F})), m-\boldsymbol{ips}(\mathcal{F})\}\\ =& \begin{cases} t(n-\boldsymbol{maxsd}(\mathcal{F})), & \text{if } \left|n-\boldsymbol{maxsd}(\mathcal{F})+\frac{1}{2}\right| \le \sqrt{\frac{1}{4}+2m-2\,\boldsymbol{ips}(\mathcal{F})},\\ m-\boldsymbol{ips}(\mathcal{F}), & \text{otherwise.}\end{cases}\end{aligned}$$

Proof. We let $s = \mathbf{maxsd}(\mathcal{F})$ and $i = \mathbf{ips}(\mathcal{F})$ for simplicity. Then, we have

$$t(n-s) \le m-i \iff s^2-(2n+1)s+(n^2+n-2m+2i) \le 0.$$

Let $q(s) := s^2-(2n+1)s+(n^2+n-2m+2i)$. Then the root formula of the quadratic $q(s)$ yields $\left((2n+1) \pm \sqrt{1+8m-8i}\right)/2$. Thus, $q(s) \le 0$ when $|n-s+\frac{1}{2}| \le \sqrt{\frac{1}{4}+2m-2i}$ and the statement follows. □

We give an elementary example to illustrate the advantage of the new improved Barvinok-Pataki bound. That is, when the singularities are known, we have a tighter upper bound on the rank of a solution.

Example 3.2.12. *Consider the spectrahedron $\mathcal{F} = \{X \in \mathbb{S}^4_+ : \operatorname{trace}(A_iX) = b_i,\ i = 1,2,3\}$ with the data*

$$A_1 = \begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix},\ A_2 = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix},\ A_3 = \begin{bmatrix}0&0&0&1\\0&0&0&0\\0&0&1&0\\1&0&0&0\end{bmatrix},\ b = \begin{pmatrix}1\\0\\0\end{pmatrix}.$$

The Barvinok-Pataki bound, Theorem 3.2.5, gives the bound $\operatorname{rank}(X) \le 2$, for all feasible points $X \in \mathcal{F}$. We now see that the knowledge of the singularity degree tightens the rank bound.

We obtain an exposing vector by solving the auxiliary system (2.3.4) for $y \in \mathbb{R}^3$:

$$\mathcal{A}^*(y) = \begin{bmatrix}y_1&0&0&y_3\\0&0&0&0\\0&0&y_3&0\\y_3&0&0&y_2\end{bmatrix} \in \mathbb{S}^4_+ \setminus \{0\},\ b^Ty = y_1 = 0.$$

*By Fact 2.2.5, we see that $y_1 = 0 \implies y_3 = 0$. Hence, $\operatorname{Diag}([0;0;0;1])$ is a maximum rank exposing vector and we complete the first round of **FR** with the facial range vector*

$$V_1 = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\end{bmatrix}. \tag{3.2.5}$$

The second constraint becomes redundant. We now have the new data

$$V_1^TA_1V_1 = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix},\ V_1^TA_3V_1 = \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix},$$

and proceed to the next iteration for ***FR****. By solving the auxiliary system* (2.3.4) *we obtain*

$$V_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \textit{and} \quad V_2^T V_1^T A_1 V_1 V_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{3.2.6}$$

and the third constraint becomes redundant. We note that $\bar{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ *is a Slater point and the* ***FR*** *algorithm terminates. The* ***FR*** *algorithm terminated with two iterations, i.e.,* $\boldsymbol{sd}(\mathcal{F}) = 2$*. It is also clear that* $\boldsymbol{maxsd}(\mathcal{F}) = 2$*. Since* $\mathcal{F}$ *has* 3 *equality constraints and* 1 *equality constraint remains in the facially reduced system,* $\boldsymbol{ips}(\mathcal{F}) = 3 - 1 = 2$*.*

We apply Theorem 3.2.7 *to* $\mathcal{F}$*. Since* $t(n-\boldsymbol{maxsd}(\mathcal{F})) = t(4-2) = 3$ *and* $m-\boldsymbol{ips}(\mathcal{F}) = 3-2 = 1$*, we conclude that every extreme point* X *of* $\mathcal{F}$ *satisfies* $t(\operatorname{rank}(X)) \leq 1$*. The only rank satisfying this bound is* $\operatorname{rank}(X) = 1$*. The point* $\operatorname{Diag}([1;0;0;0])$ *certifies the existence of a rank* 1 *solution.*

We remark that the algorithm in [119] provides a means of evaluating ranks of feasible points in an **SDP**. This is done using elementary row operations and rotations to the data to reveal special structure. More specifically, the data matrices A_2 and A_3 in Example 3.2.12 can be viewed as in special forms introduced in [119] and hence we can restrict every feasible point X in $0 \oplus \mathbb{S}^2_+$. Thus every point $X \in \mathcal{F}$ has rank at most 2.

We now provide an elementary example that the strengthened Barvinok-Pataki bound on the optimal set can be tighter than the one on the feasible set.

Example 3.2.13. *We now apply Theorem* 3.2.7 *to provide useful information on the rank of the optimal solution of an* ***SDP****. With* A_1, A_2, A_3 *defined in Example* 3.2.12*, we define additional data matrices*

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ A_5 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Consider the following ***SDP***

$$\begin{array}{rcll} p^* & = & \min\limits_{X} & \operatorname{trace}(CX) \\ & & s.t. & \operatorname{trace}(A_i X) = b_i, \ i = 1, \ldots, 5 \\ & & & X \in \mathbb{S}^4_+. \end{array} \tag{3.2.7}$$

In order to compute the singularity degree of the feasible region $\mathcal{F}$ *to* (3.2.7) *we consider the auxiliary system* (2.3.4)

$$\mathcal{A}^*(y) = \begin{bmatrix} y_1 & y_5 & 0 & y_3 \\ y_5 & 0 & y_4 & 0 \\ 0 & y_4 & y_3 & 0 \\ y_3 & 0 & 0 & y_2 \end{bmatrix} \in \mathbb{S}^4_+ \setminus \{0\}, \ b^T y = y_1 = 0.$$

By Fact 2.2.5*, we have* $y_1 = 0 \implies y_5 = y_3 = 0 \implies y_4 = 0$*. Thus,* $\operatorname{Diag}([0;0;0;1])$ *is a maximum rank exposing vector and we complete the first round of* ***FR*** *with the facial range vector* V_1 *defined* *in* (3.2.5). *The second constraint becomes redundant and we proceed to the next* ***FR*** *step with*

$$V_1^T A_1 V_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ V_1^T A_3 V_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$V_1^T A_4 V_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},\ V_1^T A_5 V_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then a maximum rank exposing vector to the facially reduced spectrahedron may be chosen with $\mathrm{Diag}([0;0;1])$ *and hence we obtain the facial range vector* V_2 *from* (3.2.6). *The third and the fourth constraints become redundant and we are left with*

$$V_2^T V_1^T A_1 V_1 V_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ V_2^T V_1^T A_5 V_1 V_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

It is clear that $\bar{X} = I_2$ *is feasible and positive definite. Thus we again have* $\boldsymbol{sd}(\mathcal{F}) = \boldsymbol{maxsd}(\mathcal{F}) = 2$ *and* $\boldsymbol{ips}(\mathcal{F}) = 3$.

After ***FR****, we obtain the following facially reduced* ***SDP***

$$\begin{array}{rcl} p^* & = & \min_X \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \bullet X \\ & & s.t. \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \bullet X = 1,\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \bullet X = 0 \\ & & X \in \mathbb{S}^2_+. \end{array} \tag{3.2.8}$$

The optimal value p^* *to* (3.2.8) *(and* (3.2.7)*) is* 0. *We now consider the singularity degree of the optimal set*

$$\mathcal{F}^* := \mathcal{F} \cap \{X \in \mathbb{S}^4 : \mathrm{trace}(CX) = p^* = 0\}.$$

By a similar approach we obtain that $\boldsymbol{sd}(\mathcal{F}^*) = \boldsymbol{maxsd}(\mathcal{F}^*) = 3$ *and* $\boldsymbol{ips}(\mathcal{F}^*) = 5$. *We note that* $t(n - \boldsymbol{sd}(\mathcal{F}^*)) = t(4-3) = 1$ *and* $(m+1) - \boldsymbol{sd}(\mathcal{F}^*) = 1$. *Thus every extreme points* X^* *of the optimal set* $\mathcal{F}^*$ *holds* $\mathrm{rank}(X^*) \le 1$. *Therefore all extreme points of the optimal set are rank* 1. *The point* $X^* = \mathrm{Diag}([1;0;0;0])$ *meets the bound.*

An Application to the SDP Relaxation of the Minimum Bisection Problem The strengthened Barvinok-Pataki bound, Theorem 3.2.7, provides a tighter bound on the rank to the **SDP** relaxation of the minimum bisection problem. The improved bound on the rank can be used when a nonlinear algorithm for solving **SDP** using low-rank factorization $X \leftarrow VV^T$ is used; see [25,26,80].

Let $G = (V, E)$ be a simple undirected graph. Let W represent the weight each edge E of G, i.e., $W_{i,j}$ is the weight on the edge $(i,j) \in E$. Let $|V| = 2k = n$. The minimum partition problem is to find a vertex partition $V = V_1 \cup V_2$ such that $|V_1| = |V_2| = k$. The **SDP** relaxation for the minimum bisection problem (see [25, Section 4.3]) is

$$\min \left\{ \frac{1}{4} \langle \mathrm{Diag}(We) - W, X \rangle : \mathrm{diag}(X) = e,\ e^T X e = 0,\ X \in \mathbb{S}^n_+ \right\}. \tag{3.2.9}$$

The data of the objective function is called the Laplacian matrix. We note that the feasible region of (3.2.9) and the feasible region of the **SDP** relaxation of the max-cut problem [73] only differ by the constraint $e^T X e = 0$.[4] From the constraint of (3.2.9), we obtain a trivial exposing vector, i.e.,

$$0 = e^T X e = \operatorname{trace}(e^T X e) = \operatorname{trace}(ee^T X).$$

This **FR** approach is observed in (2.3.10). A minimal facial range vector is $V = \begin{bmatrix} I_{n-1} \\ -e^T \end{bmatrix}$, and $V\hat{R}V^T$ with $\hat{R} = \frac{n}{n-1} I_{n-1} - \frac{1}{n-1} ee^T$ is a point in the relative interior of the feasible set. Hence **sd**, **maxsd** of the feasible set of (3.2.9) are one. Moreover, **ips** is also one.

We now use this information to obtain a tighter bound on rank for some selected value of n. The number of affine constraints in (3.2.9) is $m = n + 1 = 2k + 1$. Let $X \in \mathbb{S}^{n+1}_+$ be an extreme point. Then the Barvinok-Pataki bound (Theorem 3.2.5) yields

$$r_1 = \operatorname{rank}(X) \le \left\lfloor \frac{\sqrt{1 + 8(n+1)} - 1}{2} \right\rfloor,$$

whereas the strengthened Barvinok-Pataki bound (Corollary 3.2.8) yields

$$r_2 = \operatorname{rank}(X) \le \left\lfloor \frac{\sqrt{1 + 8 \cdot n} - 1}{2} \right\rfloor.$$

Indeed, we obtain tighter bounds on the rank of extreme points for some n. In the table below, we display some n's such that r_1 and r_2 yield different bounds on the rank.

$n = 2k$	2	14	20	44	54	90	104	152	170	230	252	324	350	434	464	560	594
r_1	2	5	6	9	10	13	14	17	18	21	22	25	26	29	30	33	34
r_2	1	4	5	8	9	12	13	16	17	20	21	24	25	28	29	32	33

3.2.3 A Sufficient Condition for Strict Feasibility

We revisit the statement Theorem 3.2.7; the absence of strict feasibility means that every extreme point X of $\mathcal{F}$ satisfies $t(\operatorname{rank}(X)) \le m - 1$. The contrapositive of this statement gives a sufficient condition for the strict feasibility. Here we provide a proof of the contrapositive of Theorem 3.2.7, independent of the results established in this chapter.

Theorem 3.2.14. *If there exists an extreme point X such that $t(\operatorname{rank}(X)) = m$, then strict feasibility holds for $\mathcal{F}$.*

Proof. Let X be an extreme point that satisfies $t(\operatorname{rank}(X)) = m$. By the compact spectral decomposition, we have $X = VRV^T$ with $\operatorname{rank}(R) = r$, and

$$b_i = \langle A_i, X \rangle = \langle A_i, VRV^T \rangle = \langle V^T A_i V, R \rangle, \ \forall i \in [m].$$

We argue that the set of matrices $\{V^T A_i V\}_{i=1}^m$ spans $\mathbb{S}^{\operatorname{rank}(X)}$.

We note that the cardinality of the set $\{V^T A_i V\}_{i=1}^m$ is $m = t(r)$. Hence, we are left with showing the linear independence. Suppose to the contrary that the matrices in the set $\{V^T A_i V\}_{i=1}^m$ are not

[4] The constraint $e^T X e = d$, for some $d \in \mathbb{N}$, is used when a general graph partition problem [95] is considered.

linearly independent. Then the linear system

$$\{W \in \mathbb{S}^r \ : \ \langle V^T A_i V, W \rangle = 0, \ i \in [m]\}$$

contains a nonzero solution E. Since $R \in \mathbb{S}^r_{++}$, for a sufficiently small $\epsilon > 0$, we get

$$R \pm \epsilon E \ \succeq 0, \ \langle V^T A_i V, R \pm \epsilon E \rangle = 0, \ \forall i \in [m].$$

We note that $R = \frac{1}{2}(R + \epsilon E) + \frac{1}{2}(R - \epsilon E)$ and R is an extreme point of $\mathcal{F}$. Thus

$$R = R + \frac{1}{2}\epsilon E \implies E = 0.$$

Hence this proves the linear independence of the set $\{V^T A_i V\}_{i=1}^m$.

We now use contradiction to show that there does not exist a vector y that solves the auxiliary system (2.3.4). Suppose such a y exists. Then,

$$0 = \langle b, y \rangle = \langle \mathcal{A}(X), y \rangle = \langle X, \mathcal{A}^*(y) \rangle = \langle V R V^T, \mathcal{A}^*(y) \rangle = \langle R, V^T \mathcal{A}^*(y) V \rangle = \left\langle R, \sum_{i=1}^m y_i V^T A_i V \right\rangle.$$

It is clear that $V^T \mathcal{A}^*(y) V = \sum_{i=1}^m y_i V^T A_i V \succeq 0$. Since y is nonzero and $\{V^T A_i V\}_{i=1}^m$ has linearly independent matrices, $V^T \mathcal{A}^*(y) V$ is not a zero matrix. Since R is positive definite, $\left\langle R, \sum_{i=1}^m y_i V^T A_i V \right\rangle > 0$. This yields a contradiction. □

An extreme point of a spectrahedron $\mathcal{F}$, with the property $t(\text{rank}(X)) = m$, may not be found depending on the given number of affine constraints, since the triangular numbers evaluated at integers do not generate all integers. For example, when $m = 5$, a point X such that $t(\text{rank}(X)) = 5$ does not exist. However, this sufficient condition is appealing when applied to the class of **LP** since this phenomenon does not occur when $\mathbb{S}^n_+$ is replaced by $\mathbb{R}^n_+$. In Chapter 4 below, we discuss this sufficient condition applied to **LP** in detail. Furthermore, we present an interesting algorithm related to it.

Chapter 4

Degeneracy in Linear Programming

The simplex method [41] and the interior point method are the most popular algorithms for solving linear programs. Unlike general conic programs, linear programs with a finite optimal value do not require strict feasibility in order to establish strong duality. Hence strict feasibility is seldom a concern. In this chapter, we discuss that the degeneracy that arises from lack of strict feasibility necessarily causes difficulties in both simplex and interior point methods.

Degeneracy could result in cycling. There are many anti-cycling rules, see e.g., [19,42,66,84,148] and the references therein. However, techniques for the resolution of degeneracy often result in *stalling* [18,35,117,140], i.e., result in taking a large number of iterations before leaving a degenerate point and can even fail to leave with current techniques [84]. Degeneracies are known to cause numerical issues when interior point methods are used, e.g., [83]. For example, degeneracy can result in singularity of the Jacobian of the optimality conditions, and thus also result in ill-posedness and loss of accuracy [77].

Continuing with the discussion made in Section 3.1.1, the facially reduced system reveals the implicit loss of surjectivity of the linear map of the equality constraint system. In particular, we show that the absence of strict feasibility implies that every basic feasible solution is degenerate. We prove the results using facial reduction and simple linear algebra. Furthermore, we include an efficient preprocessing method that can be performed as a direct extension of phase-I of the two-phase simplex method. We show that this can be used to avoid the loss of precision for many classical problems in the literature.

Contributions and Outline The contribution of this chapter is twofold:

1. We discuss the implicit loss of surjectivity tailored to the class of linear programs in the absence of strict feasibility.
   - We show that every basic feasible solution of a standard linear program is degenerate.
   - We discuss the various consequences of implicit redundancies for the simplex and interior point methods.
2. We propose and illustrate an efficient preprocessing scheme that can be performed as an extension of phase-I of the two-phase simplex method. This technique allows for eliminating variables fixed at 0, and thus regularizing and simplifying the **LP**.

This chapter is organized as follows. In Section 4.1 we discuss notions of degeneracy, the main result and immediate corollaries. In Section 4.2 we present an efficient preprocessing method that can be used as an extension of phase-I of the two-phase simplex method. In Section 4.3 we discuss the results in Sections 4.1 and 4.2 and make connections to the known results in the literature. In Section 4.4 we present numerical results that display the importance of the preprocessing for linear programs.

4.1 Lack of Strict Feasibility and Degeneracy

We first list some analogous results for **SDP** in Section 3.1 tailored to **LP**. We recall, from Section 2.3.3, that the feasible set $\mathcal{F}$ has the reduced representation

$$\mathcal{F} = \{x \in \mathbb{R}^n_+ : Ax = b\} = \{x = Vv : AVv = b, v \geq 0\}, \tag{4.1.1}$$

where V is a facial range vector. We let

$$\mathcal{I}_0 := \{\ i :\ x_i = 0,\ \forall x \in \mathcal{F}\ \} \text{ and } \mathcal{I}_+ := \{1, \ldots, n\} \setminus \mathcal{I}_0.$$

We choose V to be a submatrix of the identity matrix I_n, i.e., $V = I_n(:, \mathcal{I}_+)$. The action of V is to identify variables that are fixed at 0. We call the variable x_i, with $i \in \mathcal{I}_0$, an *exposed variable* or a variable *fixed at* 0.

If $\mathcal{F}$ fails strict feasibility, the theorem of the alternative, Lemma 2.3.4, gives rise to a certificate vector y that yields implicit redundant constraints. For completeness we include an elementary proof tailored to **LP** as well.

Lemma 4.1.1. *Suppose that $\mathcal{F}$ does not have a strictly feasible point. Then, at least one linear constraint becomes redundant after each step of **FR**.*

Proof. (Implicit redundancies in polyhedra) Let $z = A^T y$ be the exposing vector satisfying the auxiliary system (2.3.4) with $\mathbb{S}^n_+$ replaced by $\mathbb{R}^n_+$ and $\mathcal{A}^*(y)$ replaced by $A^T y$. And let V be a facial range vector induced by z. Then

$$0 = V^T z = V^T A^T y = (AV)^T y. \tag{4.1.2}$$

Since $y \in \mathbb{R}^m$ is a nonzero vector, AV contains a linearly dependent row. □

We now list three important definitions related to a feasible region in standard form.

Definition 4.1.2. *Let $\mathcal{B} \subset \{1, \ldots, n\}, |\mathcal{B}| = m$, be a given index set.*

1. *A point $x \in \mathcal{F}$ is called a basic feasible solution, **BFS**, if $A(:, \mathcal{B})$ is nonsingular and $x_i = 0$, for all $i \in \{1, \ldots, n\} \setminus \mathcal{B}$.*

2. *A basic feasible solution $x \in \mathcal{F}$ is nondegenerate if $x_i > 0$, for all $i \in \mathcal{B}$.*

3. *A basic feasible solution $x \in \mathcal{F}$ is degenerate if $x_i = 0$, for some $i \in \mathcal{B}$.*

It is well-known that the simplex method iterates from a **BFS** to a **BFS**. It is clear, from the definition, that every **BFS** has at most m positive entries.

We now present the main result Theorem 4.1.3 of this section.

Theorem 4.1.3. *Suppose that strict feasibility of $\mathcal{F}$ fails. Then every basic feasible solution of $\mathcal{F}$ is degenerate.*

We provide two proofs. First proof utilizes an algebraic approach by using the definition of the **BFS**. The second proof is inspired by a geometric approach by using extreme points. Both proofs rely heavily on Lemma 3.1.1. In Section 4.1.1 we include immediate corollaries of the main result and interesting discussions.

An Algebraic Proof of Theorem 4.1.3 via the Definition of BFS

Proof. Since there is no strictly feasible point in $\mathcal{F}$, there exists a facial range vector V, and as in (2.3.12) we have

$$\mathcal{F} = \{x = Vv \in \mathbb{R}^n \ : \ AVv = b, \ v \in \mathbb{R}^{n-s_z}_+\}.$$

By Lemma 3.1.1, AV has at least one redundant row. By permuting the columns of A, we may assume that the facial range vector V is of the form

$$V = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \text{ and } r = n - s_z.$$

We partition the index set $\{1, \dots, n\}$ as

$$\{1, \dots, n\} = \mathcal{I}_+ \cup \mathcal{I}_0, \text{ where } \mathcal{I}_+ = \{1, \dots, r\} \text{ and } \mathcal{I}_0 = \{r+1, \dots, n\}.$$

Then we have $A = \begin{bmatrix} A(:, \mathcal{I}_+) & A(:, \mathcal{I}_0) \end{bmatrix}$. Let $\bar{x} \in \mathcal{F}$ be a **BFS** with a basis $\mathcal{B}$, i.e.,

$$\mathcal{B} \subset \{1, \dots, n\}, \ |\mathcal{B}| = m, \ \det(A(:, \mathcal{B})) \neq 0, \quad \text{and} \quad A(:, \mathcal{B})\bar{x}(\mathcal{B}) = b.$$

Suppose $\mathcal{B} \subseteq \mathcal{I}_+$. We note, by Lemma 3.1.1 again, that $A(:, \mathcal{I}_+) = AV$ has redundant rows, i.e., $\text{rank}(A(:, \mathcal{I}_+)) < m$. Hence $\bar{x}$ must include a basic variable in $\mathcal{I}_0$ and this concludes that every **BFS** is degenerate. □

A Geometric Proof of Theorem 4.1.3 using Extreme Points We now provide an alternative proof of Theorem 4.1.3. We aim to provide a geometric point of view. We first present Corollary 4.1.4 below as a corollary of Theorem 3.2.1 ([128, Theorem 2.1]). The arguments in [128, Theorem 2.1] can be altered to work with the polyhedral set and we include a proof for completeness.

Corollary 4.1.4. *Suppose that $x \in F$, where F is a face of the set $\mathcal{F}$. Let $d = \dim F$. Then the number of nonzero entries of $x \in F$ is at most $m + d$.*

Proof. Let $x \in F$ and let r be the number positive entries in x. Let $\bar{x} \in \mathbb{R}^r$ be the vector obtained by discarding the 0 entries in x. This is readily given by the following matrix-vector multiplication $\bar{x} = I(\text{supp}(x), :)x$, where $\text{supp}(x)$ is the support of x, the set of indices $\{i : x_i > 0\}$.

Let $\bar{A} \in \mathbb{R}^{m\times r}$ be the matrix after removing the columns of A that are not in the support of x, i.e., $\bar{A} = A(:, \text{supp}(x))$. We note that $\bar{x}$ is a particular solution to the system $\bar{A}z = b$ and $\bar{x} > 0$.

Suppose to the contrary that $r > m + d$. Since $r - m > d$, there exists at least $d + 1$ linearly independent vectors, say $v_1, \ldots, v_{d+1} \in \mathbb{R}^r$, satisfying $\bar{A}v_i = 0$, $\forall i = 1, \ldots, d + 1$. For each $i \in \{1, \ldots, d + 1\}$ and for $\epsilon \in \mathbb{R}$, we define

$$\begin{array}{ll} v_{i,+} := \bar{x} + \epsilon v_i, & v_{i,-} := \bar{x} - \epsilon v_i, \\ x_{i,+} := I(:, \text{supp}(x)) \left(\bar{x} + \epsilon v_i\right), & x_{i,-} := I(:, \text{supp}(x)) \left(\bar{x} - \epsilon v_i\right). \end{array}$$

For a sufficiently small ϵ, we have $x_{i,+}, x_{i,-} \in \mathcal{F}$. We note that $x = \frac{1}{2}(x_{i,+} + x_{i,-})$, $\forall i$. Hence, by the definition of face, $x_{i,+} \in F$, $\forall i$. Therefore, F contains vectors $\{x_{i,+}\}_{i=1,\ldots,d+1} \cup \{x\}$ that are affinely independent and hence $\dim(F) \geq d + 1$. □

An extreme point is itself a face and the dimension of this face is 0. Hence, we obtain Corollary 4.1.5 by writing Corollary 4.1.4 through the lens of extreme points.

Corollary 4.1.5. *Every extreme point $x \in \mathcal{F}$ has at most m positive entries.*

We now restate the main result of this paper Theorem 4.1.3 in the language of extreme points and number of rows of A.

Theorem 4.1.6. *Suppose that strict feasibility of $\mathcal{F}$ fails. Then every extreme point $x \in \mathcal{F}$ has at most $m - 1$ positive entries.*

Proof. Since strict feasibility fails for $\mathcal{F}$, we have $\mathcal{F} = \{x = Vv \in \mathbb{R}^n : AVv = b,\ v \in \mathbb{R}^{n-s_z}_+\}$; see (2.3.12). From Lemma 3.1.1, we note that at least one equality in $AVv = b$ is redundant. Let $P_{\bar{m}}AVv = P_{\bar{m}}b$ be the system obtained after removing redundant rows of AV; see (3.1.3). Then, by Corollary 4.1.5, every extreme point of the set $\{v \in \mathbb{R}^{n-s_z}_+ : P_{\bar{m}}AVv = P_{\bar{m}}b\}$ has at most $m - 1$ nonzero entries. Hence, the statement follows. □

4.1.1 Immediate Consequences and Examples

In this section we elaborate Theorem 4.1.3 and Theorem 4.1.6 and their immediate consequences. Theorem 4.1.3 and Theorem 4.1.6 are equivalent owing to the well-known characterization [82, Theorem 2.21]:

$$x \in \mathcal{F} \text{ is a basic feasible solution} \iff x \in \mathcal{F} \text{ is an extreme point.}$$

We highlight that Theorem 4.1.3 and Theorem 4.1.6 do not merely state the existence of a *single* degenerate **BFS**. They prove that *every* **BFS** is degenerate. Developing a pivot rule that prevents the simplex method from visiting degenerate points is not possible as it can never stay away from the degeneracies when strict feasibility fails.

Examples, Contrapositive and Converse We now present some immediate corollaries of Theorem 4.1.3 and provide related examples. We first provide an example that illustrates Theorem 4.1.3. That is, when strict feasibility fails, all **BFS**s are degenerate.

Example 4.1.7. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix} 1 & 1 & 3 & 5 & 2 \\ 0 & 1 & 2 & -2 & 2 \end{bmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Consider the vector $y = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$. Then

$$A^T y = \begin{pmatrix} 1 & 0 & 1 & 7 & 0 \end{pmatrix}^T \text{ and } b^T y = 0.$$

*Hence, Lemma 2.3.4 certifies that $\mathcal{F}$ does not contain a strictly feasible point. There are exactly six feasible bases in $\mathcal{F}$; the **BFS** associated with $\mathcal{B} = \{\{1,2\},\{2,3\},\{2,4\}\}$ is $x = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix}^T$ and the **BFS** associated with $\mathcal{B} \in \{\{1,5\},\{3,5\},\{4,5\}\}$ is $x = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}^T$. Clearly, all **BFS**s are degenerate.*

Below, we obtain an interesting statement by writing the contrapositive of Theorem 4.1.3. Similarly, we provide Example 4.1.9 below to illustrate Corollary 4.1.8.

Corollary 4.1.8. *If there exists a nondegenerate basic feasible solution, then there exists a strictly feasible point in $\mathcal{F}$.*

Example 4.1.9. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix} 1 & 0 & -2 & 3 & -4 \\ 0 & -1 & -2 & 3 & 1 \end{bmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

*The system $\mathcal{F}$ has exactly four feasible bases; the **BFS** associated with $\mathcal{B} \in \{\{1,4\},\{2,4\},\{4,5\}\}$ is $x = \begin{pmatrix} 0 & 0 & 0 & 1/3 & 0 \end{pmatrix}^T$ and the **BFS** associated with $\mathcal{B} = \{1,5\}$ is $x = \begin{pmatrix} 5 & 0 & 0 & 0 & 1 \end{pmatrix}^T$. We note that the **BFS** associated with $\mathcal{B} = \{1,5\}$ is nondegenerate. As Corollary 4.1.8 states, the system $\mathcal{F}$ has a strictly feasible point, and it is verified by the point $\frac{1}{10}\begin{pmatrix} 4 & 1 & 1 & 4 & 1 \end{pmatrix}^T$.*

Corollary 4.1.8 provides a useful check for strict feasibility when the simplex method is used. If there is any simplex iteration that yields a nondegenerate **BFS**, then it is useful to record the occurrence. We emphasize that recording the occurrence of a nondegenerate iteration is inexpensive and the occurrence gives a *certificate of the stability* of the instance. We revisit Corollary 4.1.8 in Section 4.2.1 below and present an efficient algorithm for obtaining a strictly feasible point from a nondegenerate **BFS**.

We exhibit Example 4.1.10 below to show that the converse of Theorem 4.1.3 and Theorem 4.1.6 is not true. In other words, there is an instance that holds strict feasibility and every **BFS** is degenerate.

Example 4.1.10. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 & -2 \\ 1 & -3 & 2 & 1 & -2 \end{bmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

*$\mathcal{F}$ has exactly four feasible bases and all of them are degenerate; the **BFS** associated with $\mathcal{B} \in \{\{1,2\},\{1,4\}\}$ is $x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T$ and the the **BFS** associated with $\mathcal{B} \in \{\{2,3\},\{3,4\}\}$ is $x = \begin{pmatrix} 0 & 0 & 1/2 & 0 & 0 \end{pmatrix}^T$. However, $\mathcal{F}$ contains a strictly feasible point $\frac{1}{10}\begin{pmatrix} 1 & 1 & 5.5 & 3 & 1 \end{pmatrix}^T$.*

The assignment problem also serves as an example for showing that the converse of Theorem 4.1.3 and Theorem 4.1.6 is not true. For the assignment problem of the order n, the feasible set can be considered to be the doubly stochastic matrices. The extreme points are the permutation matrices by the Birkhoff-Neumann theorem [17, 153]. Therefore, each extreme point has exactly n positive elements while there are $m = 2n - 1$ linearly independent constraints.

Degree of Degeneracy and Two Types of Degeneracy We now further discuss the effects of the implicit redundancies of $\mathcal{F}$ in the absence of strict feasibility.

Definition 4.1.11. *Given a basic feasible solution $\bar{x} \in \mathcal{F}$, we let the degree of degeneracy of $\bar{x}$ be the number of 0's among its basic variables.*

By exploiting the facially reduced system we can estimate how degenerate the **BFS**s of $\mathcal{F}$ are. Items 2 and 3 of Corollary 4.1.12 below are closely related to the implicit problem singularity.

Corollary 4.1.12. *Suppose that strict feasibility fails for $\mathcal{F}$ and let $\mathcal{F}$ have the facial range vector representation (4.1.1). Recall the set of indices $\mathcal{I}_0 = \{i \in \{1, \dots, n\} : x_i = 0, \ \forall x \in \mathcal{F}\}$. Let $\bar{x} \in \mathcal{F}$ be a basic feasible solution with basis $\mathcal{B}$. Then, the followings hold.*

1. *The basis $\mathcal{B}$ has an nonempty intersection with $\mathcal{I}_0$, i.e., $\mathcal{I}_0 \cap \mathcal{B} \neq \emptyset$;*
2. *The degree of degeneracy of $\bar{x}$ is at least $m - \operatorname{rank}(AV)$. In other words, the degree of degeneracy of $\bar{x}$ is lower bounded by **ips**$(\mathcal{F})$;*
3. *At least $m - \operatorname{rank}(AV)$ number of basic indices of $\bar{x}$ are contained in $\mathcal{I}_0$.*

Proof. Let $\bar{x} \in \mathcal{F}$ be a **BFS** and let $\mathcal{B}$ be a basis for $\bar{x}$. Item 1 follows from the proof of Theorem 4.1.3 and the definition of the set $\mathcal{I}_0$. For Item 2, we note that $A(:, \mathcal{B})$ contains m linearly independent columns. The matrix $A(:, \mathcal{B})$ can contain at most $\operatorname{rank}(AV)$ number of columns from AV. Thus, $\bar{x}(\mathcal{B})$ must contain at least $m - \operatorname{rank}(AV)$ number of zeros. Item 3 is a direct consequence of Item 1 and Item 2. □

Item 1 of Corollary 4.1.12 implies that when degeneracy occurs in the absence of strict feasibility, some basic indices must come from $\mathcal{I}_0$. Hence there are *two types* of degeneracy that can occur; a degenerate basic variable has *two distinct origins*, $\mathcal{I}_0$ and $\mathcal{I}_+$. This relates to the work [68] in the sense that it distinguishes degenerate **LP**s in two types. An **LP** instance is called *primal weak degenerate*, if the instance has a degenerate **BFS**, but the dual optimal set is bounded; this relates to Example 4.1.10. An **LP** instance is called *primal strong degenerate*, if the instance has a degenerate **BFS**, and the dual optimal set is unbounded; this relates to Example 4.1.7. The instability under the primal strong degeneracy is shown in [68] by introducing a perturbation to b that yields divergent primal objective values.

Items 2 and 3 of Corollary 4.1.12 provide the minimum number of elements of $\mathcal{I}_0$ each **BFS** must contain. The more implicit redundant equalities the system $\mathcal{F}$ contains, the more degenerate basic variables from $\mathcal{I}_0$ are discovered. We illustrate this graphically in Figure 4.1.1 below. We emphasize that the **ips**$(\mathcal{F})$ *lower bounds* the degree of degeneracy of *every* **BFS** of $\mathcal{F}$.

$\mathcal{I}_+$
$\mathcal{I}_0$
$P_{\bar{m}}AV$
redundancy
$\mathcal{I}_+$
$\mathcal{I}_0$
$P_{\bar{m}}AV$
redundancy

Figure 4.1.1: A graphical illustration of the relationship between the implicit redundancies and the degree of degeneracy: the **BFS**s of the system on the right-hand-side have a greater lower bound on the degree of degeneracy than the system on the left-hand-side.

The arguments used for Corollary 4.1.12 are rather algebraic. The geometric argument used in the proof of Theorem 4.1.6 also provides a similar result for Corollary 4.1.12. For any extreme point $x \in \mathcal{F}$, the number of nonzero elements of x, $|\operatorname{supp}(x)|$, satisfies

$$|\operatorname{supp}(x)| \leq m - \mathbf{ips}(\mathcal{F}) \implies \mathbf{ips}(\mathcal{F}) \leq m - |\operatorname{supp}(x)|.$$

Since this holds for all extreme points of $\mathcal{F}$, an analogous result of Corollary 3.2.9 follows for the polyhedron $\mathcal{F}$:

$$\mathbf{sd}(\mathcal{F}) \leq \mathbf{maxsd}(\mathcal{F}) \leq \mathbf{ips}(\mathcal{F}) \leq \ \hat{d} := \min_{\mathbf{BFS}\, x \,\in\, \mathcal{F}} \{\text{degree of degeneracy of } x\}. \tag{4.1.3}$$

The shortest **FR** steps for the polyhedron $\mathcal{F}$, $\mathbf{sd}(\mathcal{F})$, is at most 1, thus the inequality $\mathbf{sd}(\mathcal{F}) \leq \hat{d}$ does not provide useful information. However, the relation (4.1.3) provides two meaningful consequences related to $\mathbf{maxsd}(\mathcal{F})$ and $\mathbf{ips}(\mathcal{F})$:

1. The inequality $\mathbf{maxsd}(\mathcal{F}) \leq \hat{d}$ implies that the number of nontrivial **FR** steps can never exceed the degree of degeneracy of a least degenerate **BFS** of $\mathcal{F}$;

2. The inequality $\mathbf{ips}(\mathcal{F}) \leq \hat{d}$ shows that it is useful to record the minimum degree of degeneracy observed throughout the simplex iterations. This gives an estimate for the number of implicitly redundant equalities of $\mathcal{F}$.

If $\mathcal{F}$ contains a nondegenerate **BFS**, we get $\hat{d} = 0$ in (4.1.3). Hence, $\mathbf{sd}(\mathcal{F}) = \mathbf{maxsd}(\mathcal{F}) = \mathbf{ips}(\mathcal{F}) = 0$ and it provides an alternative way to view Corollary 4.1.8. We comment that evaluating and recording the degree of degeneracy of a **BFS** are not expensive operations.

The loss of surjectivity provides a method for checking if there is an unrealized exposed variable in a special case. The following special case is of interest.

Proposition 4.1.13. *Let $\mathcal{E}$ be a set of indices containing some exposed variables of $\mathcal{F}$. If* $\operatorname{rank}(A) = \operatorname{rank}(A(:, \mathcal{E}^c))$*, there is an exposed variable that is not contained in $\mathcal{E}$.*

Proof. Let $\mathcal{I}$ be the set of indices of all exposed variables. By Lemma 4.1.1 implies $\operatorname{rank}(A(:, \mathcal{I}^c)) < \operatorname{rank}(A)$. If $\operatorname{rank}(A) = \operatorname{rank}(A(:, \mathcal{E}^c))$, it means that $\mathcal{E} \subsetneq \mathcal{I}$. □

4.2 Efficient Preprocessing for Facial Reduction and Strict Feasibility

In this section we present an efficient preprocessing method for producing a facially reduced system. In Section 4.2.1 we discuss how to compute a strictly feasible point using a nondegenerate **BFS** and its variant. In Section 4.2.2 we present an algorithm for testing strict feasibility and finding an accurate exposing vector.

4.2.1 Towards a Strictly Feasible Point from a Nondegenerate BFS

By Corollary 4.1.8, the existence[1] of a nondegenerate **BFS** guarantees the existence of a strictly feasible point. In this section, we propose an algorithm for acquiring a Slater point from a nondegenerate **BFS** and include a generalization. The argument is this section also provides a constructive proof of Corollary 4.1.8.

Let $\bar{x} \in \mathcal{F}$ be a *nondegenerate* **BFS**. Without loss of generality, we assume that the basic variables $\bar{x}_{\mathcal{B}}$ of $\bar{x}$ are located at the last m entries of $\bar{x}$. We fix a positive scalar $\hat{\gamma} \in (0,1)$ and an index $j \in \{1,\ldots,n-m\}$. For some $\alpha \geq 0$, we consider the simplex method ratio test type inequality

$$\hat{\gamma}\bar{x}_{\mathcal{B}} + \alpha (A_{\mathcal{B}})^{-1}A_j \geq 0. \tag{4.2.1}$$

Since $\bar{x}_{\mathcal{B}} > 0, \hat{\gamma} > 0$, there exists a positive α that maintains the inequality (4.2.1). Let

$$\alpha^* = \min\left\{1,\ \max\{\alpha \in \mathbb{R}_+ : \hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha(A_{\mathcal{B}})^{-1}A_j \geq 0\}\right\}. \tag{4.2.2}$$

Then we decompose

$$\hat{\gamma}\bar{x}_{\mathcal{B}} = \left(\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j\right) + \alpha^*(A_{\mathcal{B}})^{-1}A_j$$

and observe

$$\begin{array}{rcl} b & = & A_{\mathcal{B}}\bar{x}_{\mathcal{B}} \\ & = & (1-\hat{\gamma})A_{\mathcal{B}}\bar{x}_{\mathcal{B}} + \hat{\gamma}A_{\mathcal{B}}\bar{x}_{\mathcal{B}} \\ & = & (1-\hat{\gamma})A_{\mathcal{B}}\bar{x}_{\mathcal{B}} + A_{\mathcal{B}}\left(\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j + \alpha^*(A_{\mathcal{B}})^{-1}A_j\right) \\ & = & A_{\mathcal{B}}(\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j) + \alpha^* A_j. \end{array}$$

If we set $x_j = \alpha^* > 0$ and replace $\bar{x}_{\mathcal{B}}$ by $\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j$, then we have increased the cardinality of the positive entries of a solution. We note that $\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j$ only has strictly positive entries since it is a sum of positive vector and a nonnegative vector;

$$\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j = \underbrace{(1-\hat{\gamma})\bar{x}_{\mathcal{B}}}_{\text{positive}} + \underbrace{\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j}_{\text{nonnegative}}.$$

We can continuously increase the number of positive entries of a solution one by one for each $j \in \{1,\ldots,n-m\}$. Moreover we can achieve this by a compact vectorized operation. The main idea is that we can choose $\hat{\gamma}$ in (4.2.1) independently for each $j \in \{1,\ldots,n-m\}$. Let γ_j be a

[1] Determining the existence of a degenerate **BFS** is an NP-complete problem; see [34].

positive real number such that $0 < \gamma := \sum_{j=1}^{n-m} \gamma_j < 1$. Then, we have

$$\bar{x}_{\mathcal{B}} = (1-\gamma)\bar{x}_{\mathcal{B}} + \gamma\bar{x}_{\mathcal{B}} = (1-\gamma)\bar{x}_{\mathcal{B}} + \sum_{j=1}^{n-m} \gamma_j \bar{x}_{\mathcal{B}}.$$

We set an auxiliary matrix

$$\Theta = \begin{bmatrix} \gamma_1 \bar{x}_{\mathcal{B}} & \cdots & \gamma_{n-m}\bar{x}_{\mathcal{B}} \end{bmatrix} - (A_{\mathcal{B}})^{-1} A(:, 1:n-m) \in \mathbb{R}^{m \times (n-m)}$$

and perform (4.2.2) on each column j of Θ to obtain the vector θ^*:

$$\theta^*_j := \begin{cases} \max(\Theta(:,j)) & \text{if } \max(\Theta(:,j)) \leq 1, \\ 1 & \text{otherwise.} \end{cases}$$

Then the point

$$\begin{bmatrix} \theta^* \\ \bar{x}_{\mathcal{B}} - (A_{\mathcal{B}})^{-1} A(:, 1:n-m)\theta^* \end{bmatrix}$$

is a strictly feasible point to $\mathcal{F}$. Hence, this operation provides a constructive proof of Corollary 4.1.8.

We now extend the aforementioned procedure for obtaining a strictly feasible point using any feasible solution $\bar{x} \in \mathcal{F}$ such that the submatrix $A(:, \operatorname{supp}(\bar{x}))$ is full-row rank. We partition $\bar{x} \in \mathcal{F}$ as follows

$$\bar{x} = \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{B}_2} \\ \bar{x}_{\mathcal{N}} \end{pmatrix}, \text{ where } \operatorname{supp}(\bar{x}) = \mathcal{B}_1 \cup \mathcal{B}_2, \ \operatorname{rank}(A(:,\mathcal{B}_1)) = m, \quad \text{and } \mathcal{N} = \{1, \ldots, n\} \setminus \operatorname{supp}(\bar{x}). \tag{4.2.3}$$

We partition A using the same partition $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{N}$:

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} & A_{\mathcal{N}} \end{bmatrix} \bar{x} = b \iff \begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{N}} \end{pmatrix} = \bar{b} := b - A_{\mathcal{B}_2}\bar{x}_{\mathcal{B}_2}.$$

Then we can apply the aforementioned procedure to the system

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{N}} \end{pmatrix} = \bar{b}$$

and distribute positive weights to $\bar{x}_{\mathcal{N}}$ using $\bar{x}_{\mathcal{B}_1}$. Finally, we find a strictly feasible point to $\mathcal{F}$. This process is summarized in Algorithm 4.2.1. Furthermore, Algorithm 4.2.1 provides a constructive proof for Proposition 4.2.1 below.

Proposition 4.2.1. *Let $x \in \mathcal{F}$ be a solution such that* $\operatorname{rank}(A(:, \operatorname{supp}(x))) = m$. *Then, $\mathcal{F}$ has a strictly feasible point.*

4.2.2 Towards an Exposing Vector: Phase I Part B and Strict Feasibility Testing

We now present an efficient preprocessing procedure for detecting identically 0 variables, constructing exposing vectors and the facially reduced **LP**, i.e., given a **BFS** $\bar{x}$, we solve special subproblems

Algorithm 4.2.1 Compute a Slater Point

```
Require: Given $A \in \mathbb{R}^{m \times n}$, $x \in \mathcal{F}$ partitioned as in (4.2.3).
Choose any $\gamma \in \mathbb{R}^{|\mathcal{N}|}_{++}$ such that $\sum_{j=1}^{|\mathcal{N}|} \gamma_j < 1$.
Compute
    $\Theta = \begin{bmatrix} x_{\mathcal{B}_1} & \cdots & x_{\mathcal{B}_1} \end{bmatrix} \operatorname{Diag}(\gamma) - A_{\mathcal{B}_1}^{-1} A_{\mathcal{N}}.$
Compute $\theta^* \in \mathbb{R}^{|\mathcal{N}|}_{++}$, where for each $j \in \{1, \dots, |\mathcal{N}|\}$,
    $\theta^*_j := \begin{cases} \max(\Theta(:,j)) & \text{if } \max(\Theta(:,j)) \leq 1, \\ 1 & \text{otherwise.} \end{cases}$
Set $x = \begin{pmatrix} x_{\mathcal{B}_1} - (A_{\mathcal{B}_1})^{-1} A_{\mathcal{N}} \theta^* \\ x_{\mathcal{B}_2} \\ \theta^* \end{pmatrix}$.
```

using the simplex method with the initial point $\bar{x}$. By the end of the process, we obtain either

1. a certificate y that produces an exposing vector $A^T y$ (Slater condition fails), or
2. a strictly feasible point (Slater condition holds).

The process has two applications. First, since the only requirement of this process is the **BFS**, the procedure can be used as an extension of the phase-I of the two-phase simplex method to obtain the equivalent facially reduced problem that satisfies strict feasibility. Second, the procedure can be used as a post-optimum diagnosis. By recording a **BFS** with the smallest degree of degeneracy, we can improve tests for stability.

We now describe the proposed preprocessing method. Let $\mathcal{B}$ be a degenerate initial basis of $\mathcal{F}$ and let $\bar{x}$ be the **BFS** associated with $\mathcal{B}$. Without loss of generality, we assume that basic variables are located at the first m entries of $\bar{x}$. Let d be the degree of degeneracy of $\bar{x}$. We further assume that the degenerate basic variables are located at the first d entries of $\bar{x}$. We let $\mathcal{B}_0 := \{1, \dots, d\}$. We consider the following problem:

$$p^*_i = \max_x \{x_i \,:\, Ax = b,\, x \geq 0\},\ i \in \mathcal{B}_0. \tag{4.2.4}$$

For simplicity, we let $i = 1$. We solve (4.2.4) using the simplex method from the initial **BFS** $\bar{x}$. That is, we do not need to repeat the typical phase-I of the two-phase simplex method in order to find an initial feasible basis. The optimal value p^*_1 of (4.2.4) is clearly lower bounded by 0. We consider two cases below:

1. Suppose that $p^*_1 > 0$. Then, the variable x_1 is not an identically 0 variable, i.e., $1 \notin \mathcal{I}_0$.
2. Suppose that $p^*_1 = 0$. Then, the variable x_1 is a variable fixed at 0, i.e., $1 \in \mathcal{I}_0$. Let $\mathcal{B}^*$ be an optimal basis for (4.2.4). Then we have

$$y^* = A(:, \mathcal{B}^*)^{-T} e_1,\ \langle b, y^* \rangle = 0 \ \text{ and } A^T y^* \geq e_1, \tag{4.2.5}$$

> where the e_1 is the first column of the identity matrix in the space of appropriate dimension. We note that the dual optimal solution y^* in (4.2.5) produces a solution to the auxiliary system (2.3.4). Therefore, we obtain a *nontrivial* exposing vector since $A^T y^*$ is not the zero vector. Clearly, the first variable x_1 is exposed by $A^T y^*$ since the first element of $A^T y^*$ is *positive*. Furthermore, if $|\operatorname{supp}(A^T y^*)| > 1$, then we find additional variables other than x_1 that are identically 0 in the feasible set.

Let $\{y^j\}$ be a collection of the certificates that are obtained from solving (4.2.4) with the index 1 replaced by $i \in \mathcal{B}_0 = \{1, \dots, d\}$. Then $y^\circ = \sum_j y^j$ is also a certificate, i.e.,

$$A^T y^\circ = \sum_j A^T y^j \geq 0, \;\; A^T y^\circ \neq 0, \quad \text{and} \quad \langle b, y^\circ \rangle = \sum_j \langle b, y^j \rangle = 0,$$

and we obtain a nontrivial exposing vector $A^T y^\circ$ for the system $\mathcal{F}$. We can now delete the identified identically zero variables along with the corresponding columns of A. We then find and delete redundant rows to obtain a smaller **LP**. By summarizing the two cases above, we obtain an efficient preprocessing method Algorithm 4.2.2.

The following allows for simplifications in Algorithm 4.2.2.

Lemma 4.2.2. *Let $\mathcal{B}$ be an initial basis containing the index i for the problem* (4.2.4)*. Then the index i always remains in the basis throughout the iterations.*

Proof. Without loss of generality, we let $i = 1$. We argue that 1 is not chosen to leave the basis. Let $y^* = (A_{\mathcal{B}}^T)^{-1} c_{\mathcal{B}}$ and $\bar{A} = A_{\mathcal{B}}^{-1} A$. Suppose that the reduced cost at the index j is positive. Then

$$0 < \bar{c}_j = c_j - A_j^T y^* = -A_j^T y^* = -A_j^T (A_{\mathcal{B}}^T)^{-1} e_1 = -\bar{A}_{1j}.$$

Since $\bar{A}_{1j} < 0$, the index 1 is not chosen to leave the basis $\mathcal{B}$. □

The following special case is of interest; no simplex pivoting steps are required to determine strict feasibility when the degree of degeneracy of a **BFS** is one.

Theorem 4.2.3. *(preprocessing for degree of degeneracy 1) Given a basis $\mathcal{B}$, let $\bar{x}$ be the **BFS** with the degree of degeneracy exactly one. Let $\mathcal{N} = \{1, \dots, n\} \setminus \mathcal{B}$ and let $y^* = (A_{\mathcal{B}}^T)^{-1} c_{\mathcal{B}}$. Then strict feasibility fails for $\mathcal{F}$ if, and only if, y^* satisfies $A_{\mathcal{N}}^T y^* \geq 0$.*

Proof. Let $\bar{x}$ be a degenerate **BFS** associated with the basis $\mathcal{B}$. Without loss of generality, we assume $1 \in \mathcal{B}$ and 1 is the degenerate index. We consider the problem

$$p_1^* = \max\{x_1 \,:\, Ax = b,\, x \geq 0\}.$$

We note that $\langle b, y^* \rangle = 0$ since $\langle b, y^* \rangle$ is identical to the current objective value '0'. The backward direction is clear by Lemma 2.3.4. Now suppose that strict feasibility fails. Suppose to the contrary that $A_{\mathcal{N}}^T y^* \geq 0$ fails. Then there exists j such that $A_j^T y^* < 0$, $j \in \mathcal{N}$. Note that, by Lemma 4.2.2, that 1 is not chosen to leave the basis. Thus, there is an index $k \neq 1, k \in \mathcal{B}$ that leaves the basis. Since all other basic variables are positive, we obtain a positive step length and we improve the objective value, which yields a contradiction to $p_1^* = 0$. □

Algorithm 4.2.2 Preprocessing Phase I Part B; Towards Strict Feasibility

```
Require: A BFS $\bar{x}$ with corresponding basis $\mathcal{B}$; set $\mathcal{B}_0 = \{i \in \mathcal{B} : \bar{x}_i = 0\}$
Initialize: $x^\circ = \bar{x}$, $y^\circ = 0 \in \mathbb{R}^m$, $\mathcal{J}_0 = \emptyset$, $\mathcal{B}_* \leftarrow \mathcal{B}_0$
while $\mathcal{B}_0 \neq \emptyset$ and $\mathcal{B}_* \neq \emptyset$ do
    Pick $i \in \mathcal{B}_0$; starting from the initial BFS $\bar{x}$, solve for primal-dual optima $x^*, y^*$
        $$x^* = \text{argmax}_x\{x_i : Ax = b, x \geq 0\}, \quad p^* = x_i^* = b^T y^*$$
    $\mathcal{S} \leftarrow \text{supp}(x^*)$
    $\mathcal{B}_* \leftarrow$ degenerate basic indices for $x^*$
    if $\mathcal{B}_0 \neq \emptyset$ and $\mathcal{B}_* \neq \emptyset$ then
        if $p^* = 0$ (strict feasibility fails) then
            Use dual certificate $y^*$ to satisfy (2.3.4)
            $y^\circ \leftarrow y^\circ + y^*$
            $\mathcal{J}_0 \leftarrow \mathcal{J}_0 \cup (\text{supp}(A^T y^*) \cap \mathcal{B})$
            $\mathcal{B}_0 \leftarrow \mathcal{B}_0 \setminus \{\mathcal{S} \cup \mathcal{J}_0\}$
        else
            $\mathcal{B}_0 \leftarrow \mathcal{B}_0 \setminus \mathcal{S}$
        end if
        Choose $\gamma \in (0,1)$ and set $x^\circ \leftarrow \gamma x^\circ + (1-\gamma)x^*$
    end if
end while
if $\mathcal{J}_0 \neq \emptyset$ then
    $z^\circ = A^T y^\circ$ (exposing vector)
    $\mathcal{R} \leftarrow$ redundant row indices of $A\,(:, \text{supp}(z^\circ)^c)$
    $A \leftarrow A(\mathcal{R}^c, \text{supp}(z^\circ)^c)$,  $b \leftarrow b(\mathcal{R}^c)$
else
    Run Algorithm 4.2.1 with $x^\circ$ and $\det(A_{\mathcal{B}}) \neq 0$ (use $x^*$ and $\mathcal{B}_*$, if $\mathcal{B}_* = \emptyset$)
end if
```

Upon termination of Algorithm 4.2.2, we can always determine whether the system $\mathcal{F}$ has a strictly feasible point or not. Algorithm 4.2.2 terminates in a finite number of iterations since we remove at least one element from the set $\mathcal{B}_0$ at each iteration. We emphasize that we do not need to solve the auxiliary **LP**s for all $i \in \{1, \dots, n\}$. We solve (4.2.4) only for the degenerate basic indices of the predetermined basis $\mathcal{B}$. However, upon termination of Algorithm 4.2.2, it is possible that we have not obtained face$(\mathcal{F}, \mathbb{R}^n_+)$, the minimal face containing $\mathcal{F}$. Although the complete **FR** for **LP** can be completed in one iteration, one step termination is possible only when we find a solution y of (2.3.4) so that $A^T y$ is in the relative interior of the conjugate face of face$(\mathcal{F}, \mathbb{R}^n_+)$. In this case, we can rerun Algorithm 4.2.2 with the facially reduced system. For finding an initial basis for the second trial, we may use the efficient basis recovery scheme, e.g., see [159, Chapter 7].

One of the nice features of Algorithm 4.2.2 is that we do not need to search for a new initial basis for each iteration; the initial basis $\mathcal{B}$ remains the same. Therefore, our approach can be directly employed after the standard phase-I of the two-phase simplex method.

We do not need a lot of pivoting steps to determine if p_i^* is zero or positive. If $p_i^* = 0$, the initial $\mathcal{B}$ is indeed a basis that gives the optimal value. However the dual feasibility may not be obtained immediately[2]. Thus, there may be additional pivoting steps required to obtain dual feasibility. However, since the optimal value is obtained at $\mathcal{B}$, we do not expect that the optimal basis search to be time-consuming. We recall from Lemma 4.2.2 that the index i in (4.2.4) never leaves the basis. For the case $p_i^* \in (0, \infty)$, the optimal value p_i^* does not need to be found. Hence once a basis that gives a positive support on i is found, we can terminate the maximization problem in Algorithm 4.2.2 immediately and concern x° only. In the case of $p_i^* = \infty$, we can perform the following operation. Let $\mathcal{B}_c$ be A basis that indicates $p_i^* = \infty$ and let j be an entering variable that indicates the unboundedness. Then by setting

$$x^\circ(j) \leftarrow 1, \;\; x^\circ(\mathcal{B}_c) \leftarrow x_{\mathcal{B}_c} - A_{\mathcal{B}_c}^{-1} A_j \text{ and } x^\circ((\{j\} \cup \mathcal{B}_c)^c) = 0,$$

we obtain a feasible solution x° that yields a positive objective value.

We often get an exposing vector that reveals more than one element in the set $\mathcal{I}_0$ from solving (4.2.4). Without loss of generality, let $p_1^* = 0$ in (4.2.4) and let y^* be a dual feasible solution. Suppose $A^T y^* = e_1$, i.e., $A^T y^*$ only reveals exactly one exposed variable. Then $y^* \in \text{null}(A(:, 2:n)^T)$. Since the number of columns of the data matrix A is often significantly greater than the number of rows, $y^* \in \text{null}(A(:, 2:n)^T)$ often implies that $y^* = 0$. If $y^* = 0$, we cannot obtain $A^T y^* = e_1$.

When we have an instance of large size and a **BFS** with a very large degree of degeneracy, we may adopt parallel computing for Algorithm 4.2.2 in order to reduce the total computation time. We note again that the initial basis remains the same throughout the iterations. Hence, solving (4.2.4) for individual $i \in \mathcal{B}_0$ can be performed independently. In fact, parallel computing can be used to obtain a strictly feasible solution in Algorithm 4.2.1 as well; the weight vector γ can be chosen independently for each $j \in \mathcal{N}$.

[2] If we have a nondegenerate initial basis, then the dual feasibility is immediately obtained. However, our initial basis is degenerate.

4.3 Discussions

In this section we discuss the main result in Section 4.1 and make connections to new results and known results in the literature.

Distance to Infeasibility The argument on the distance to infeasibility discussed in Proposition 3.1.4 naturally holds for the class of **LP**. Moreover, the vector $\Delta b = y$ that satisfies the auxiliary system (2.3.4) is a perturbation that makes the set $\mathcal{F}$ empty; see (3.1.10). We omit the proof[3]. We illustrate Proposition 4.3.1 in Section 4.4.1.4 below.

Proposition 4.3.1. *Suppose that strict feasibility fails for $\mathcal{F}$ and let $\mathcal{F}$ have the representation (2.3.12). Then the following hold.*

1. *For all $\Delta b \in \operatorname{range}(AV)$ with sufficiently small norm, the set $\{x \in \mathbb{R}^n_+ : Ax = b + \Delta b\}$ is feasible.*
2. *The distance to infeasibility of $\mathcal{F}$ is 0, i.e., $\operatorname{dist}(b, \mathcal{F} = \emptyset) = 0$.*

We recall that the existence of one nondegenerate **BFS** certifies the **MFCQ** to be satisfied; see Corollary 4.1.8. In fact, given any feasible instance, we can perturb the data to generate an instance containing a nondegenerate **BFS**. Let $\mathcal{B}$ be any feasible basis to $\mathcal{F}$ and let $A_{\mathcal{B}}$ be the basis matrix. Then,

$$Ax = b \iff A_{\mathcal{B}}^{-1}Ax = A_{\mathcal{B}}^{-1}b \in \mathbb{R}^m_+.$$

Adding any perturbation ξ to $A_{\mathcal{B}}^{-1}b$ that yields $A_{\mathcal{B}}^{-1}b + \xi \in \mathbb{R}^m_{++}$ generates a feasible set containing a strictly feasible point. We relate this observation to the ϵ-perturbation method proposed by Charnes (see e.g., [35, 118].). The ϵ-perturbation refers to replacing the right-hand-side vector b by $b + \xi$, where $\xi = (\epsilon, \epsilon^2, \dots, \epsilon^m)^T$ for any sufficiently small $\epsilon > 0$. This special perturbation gives rise to a remarkable property; *every* **BFS** of the perturbed set is *nondegenearate*[4]. When every **BFS** is nondegenerate, the simplex algorithm makes a nontrivial progress at every iteration. As a consequence, when strict feasibility fails, the distance to infeasibility *and* the distance to a 'nice' problem are *both* 0.

Applications to Known Characterizations for Strict Feasibility There are some known characterizations for strict feasibility of $\mathcal{F}$. Using these characterizations we can obtain extensions of Theorem 4.1.3, Theorem 4.1.6, and corollary 4.1.8.

The dual $(\mathcal{D})$ of $(\mathcal{P})$ is

$$(\mathcal{D}) \qquad \max_{y,s} \left\{ b^T y \ : \ A^T y + s = c, \ s \geq 0 \right\}. \tag{4.3.1}$$

It is known that strict feasibility fails for $\mathcal{F}$ if, and only if, the set of optimal solutions for the dual $(\mathcal{D})$ is unbounded; see e.g., [159, Theorem 2.3] and [67]. Then Corollary 4.3.2 follows.

Corollary 4.3.2. *1. Suppose that the set of optimal solutions for the dual $(\mathcal{D})$ is unbounded. Then every basic feasible solution to $\mathcal{F}$ is degenerate.*

[3] The proof in Proposition 3.1.4 directly applies by replacing $\hat{A}_V$ with AV, and (3.1.2) with (4.1.2).

[4] This property is also known as the primal-nondegeneracy.

2. *Suppose that there exists a nondegenerate basic feasible solution to $\mathcal{F}$. Then the set of optimal solutions for the dual $(\mathcal{D})$ is bounded.*

The implication of Item 2 of Corollary 4.3.2 is interesting; *any* nondegenerate **BFS** (if it exists) provides the information on the dual *optimal* set.

It is known that strict feasibility holds for $\mathcal{F}$ if, and only if, $b \in \operatorname{relint}(A(\mathbb{R}^n_+))$; see e.g., [50, Proposition 4.4.1]. Then if one finds a set of indices $\mathcal{I} \subset \{1,\dots,n\}$ such that $A(:,\mathcal{I})$ is nonsingular and $A(:,\mathcal{I})z = b$ has a solution z with positive entries, then $b \in \operatorname{relint}(A(\mathbb{R}^n_+))$.

The implicit redundancies relate to some constraint qualifications that arise in optimization problems. It is clear that an instance that has no Slater point fails to satisfy **MFCQ**, i.e., Item 1 of Definition 2.3.3 fails. It is interesting that Item 2 of Definition 2.3.3 must fail implicitly in the absence of strict feasibility. Furthermore, the implicit redundancies immediately imply that the linear independence constraint qualification (**LICQ**) must fail at every **BFS** of $\mathcal{F}$. A comprehensive sensitivity analysis on parametric programming [60] contains the derivatives of optimal solutions and optimal values with respect to the parameters. Most of the results are established under the **LICQ** assumption in order to use the invertibility of the Jacobian matrix containing the gradients of the active constraints.

Applications to Obtain a Strictly Complementary Primal-Dual Solution We present an application of Algorithm 4.2.1 for computing a strictly complementary primal-dual optimal solution.

Let (x^*, y^*, s^*) be an optimal triple for the standard primal-dual **LP** pair. Let $\mathcal{B}^* \cup \mathcal{N}^* = \{1,\dots,n\}$ be the strict complementary partition of the primal-dual optimal pair. The existence of such a partition is guaranteed by the Goldman-Tucker theorem [74] and the partition $\mathcal{B}^* \cup \mathcal{N}^*$ is unique. We can use Algorithm 4.2.1 to obtain the strict complementary solution for the two cases:

1. Let x^* be a nondegenerate (optimal) **BFS**. Then, $\operatorname{supp}(s^*) = \mathcal{N}^*$ and $\operatorname{supp}(x^*)$ can be extended to complete $\mathcal{B}^*$;
2. Let x^* be an optimal solution such that $A(:,\operatorname{supp}(x^*))$ is full-row rank. Then, $\operatorname{supp}(s^*) = \mathcal{N}^*$ and $\operatorname{supp}(x^*)$ can be extended to complete $\mathcal{B}^*$.

Suppose that we are given a primal-dual optimal solution (x^*, y^*, s^*) of the form

$$\begin{bmatrix} A_{\mathcal{B}} & A_{\mathcal{J}} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} x_{\mathcal{B}} \\ x_{\mathcal{J}} \\ x_{\mathcal{N}} \end{pmatrix} = b, \text{ where } \operatorname{rank}(A_{\mathcal{B}}) = m, \ \begin{pmatrix} x_{\mathcal{B}} \\ x_{\mathcal{J}} \\ x_{\mathcal{N}} \end{pmatrix} \begin{matrix} > \\ = \\ = \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} s_{\mathcal{B}} \\ s_{\mathcal{J}} \\ s_{\mathcal{N}} \end{pmatrix} \begin{matrix} = \\ = \\ > \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.3.2}$$

We claim that $\mathcal{N}^* = \operatorname{supp}(s^*)$. That is, the support of the current dual optimal solution s^* is maximal and hence we obtain the strict complementary partition for free. We rewrite the system $Ax = b$ of (4.3.2) as

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} & A_{\mathcal{J}} \end{bmatrix} \begin{pmatrix} x_{\mathcal{B}_1} \\ x_{\mathcal{B}_2} \\ x_{\mathcal{J}} \end{pmatrix} = b, \ \text{ where } A_{\mathcal{B}} = \begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} \end{bmatrix}, \ x_{\mathcal{B}} = \begin{pmatrix} x_{\mathcal{B}_1} \\ x_{\mathcal{B}_2} \end{pmatrix} \text{ and } \operatorname{rank}(A_{\mathcal{B}_1}) = m.$$

Then, by replacing the data in Algorithm 4.2.1 by

$$\mathcal{N} \leftarrow \mathcal{J},\ A \leftarrow A(:, \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{N}),\ x \leftarrow x^*,$$

we can endow positive weights to $x_{\mathcal{J}}$ while maintaining the primal feasibility. Since we maintain the feasibility of the primal-dual solution without violating the complementarity, we maintain the optimality.

Lack of Strict Feasibility and Interior Point Methods The discussion of degeneracy is usually paired with the simplex methods. The degeneracy does not seem to be a serious concern for the interior point methods and this is supported by the limited number of literature that link degeneracy and interior point methods. The available literatures [83,97] discuss the ill-conditioning that can occur depending on the degeneracy status of an optimal point. Below, we elaborate on a new perspective on the ill-conditioning that arise in the interior point methods.

Many interior point algorithms are derived from the optimality conditions (KKT conditions) using the primal $(\mathcal{P})$ and the dual $(\mathcal{D})$. Let (x_c, y_c, s_c) be the current iterate for the primal-dual pair. The search direction is computed by solving the Newton equation

$$\begin{bmatrix} 0_{n\times n} & A^T & I \\ A & 0_{m\times m} & 0_{m\times n} \\ \mathrm{Diag}(s_c) & 0_{n\times m} & \mathrm{Diag}(x_c) \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = - \begin{pmatrix} r_d \\ r_p \\ r_c \end{pmatrix}, \tag{4.3.3}$$

where r_d, r_p, r_c are the residuals of the dual feasibility, primal feasibility and complementarity, respectively. And many practical interior point methods use block variable elimination and find the search direction Δy by solving the so-called normal equation, a square system

$$AD_cA^T\Delta y = \bar{r},\ \text{where } D_c = \mathrm{Diag}(x_c)\,\mathrm{Diag}(s_c)^{-1} \tag{4.3.4}$$

and $\bar{r} \in \mathbb{R}^m$ is some residual; see e.g., [159, Chapter 11]. It is known that (4.3.4) often becomes ill-conditioned near an optimum and it is the main difficulty that arises in implementing interior point methods. The ill-conditioning of the matrix AD_cA^T under the degeneracy is discussed in [83,97] in terms of the lack of nice positive diagonal elements of D_c.[5] This relates to our results in the sense that all vertices that form the optimal face of $(\mathcal{P})$ are also degenerate in the absence of strict feasibility. Moreover, we show that the ill-conditioning of the matrix AD_cA^T not only originates from the columns of A chosen by D_c but also from the rows of A in the absence of strict feasibility. In particular, a large **ips** is a good indicator of the ill-conditioning.

We partition the matrix $A = \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}$, where $[A_{\mathcal{I}_0}; R_{\mathcal{I}_0}]$ corresponds to the submatrix of A associated with the index set $\mathcal{I}_0$. The submatrix R_{AV} refers to the rows of A that are implicitly redundant due the lack of strict feasibility. Let (x^*, y^*, s^*) be an optimal triple and we let $D^* = \mathrm{Diag}(x^*)\,\mathrm{Diag}(s^*)^{-1}$. As $x_c \to x^*$, i.e., as the iterates get closer to the feasible set $\mathcal{F}$, we

[5] We note that the action of D_c is to scale the columns of A.

observe the limiting behaviour $AD_cA^T \to AD^*A^T$ below:

$$\begin{array}{rcl} AD_cA^T & = & \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix} \operatorname{Diag}(x_c) \operatorname{Diag}(s_c)^{-1} \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}^T \\ \to \quad AD^*A^T & = & \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix} \begin{bmatrix} D^*_{AV} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}^T \\ & = & \begin{bmatrix} (P_{\bar{m}}AV)D^*_{AV}(P_{\bar{m}}AV)^T & (P_{\bar{m}}AV)D^*_{AV}R^T_{AV} \\ R_{AV}D^*_{AV}(P_{\bar{m}}AV)^T & R_{AV}D^*_{AV}R^T_{AV} \end{bmatrix}. \end{array}$$

where D^*_{AV} is the submatrix of D^* with the diagonal associated with $\mathcal{I}_+$. We recall from Lemma 3.1.1 that the rows of R_{AV} are linear combinations of the rows of $P_{\bar{m}}AV$. Therefore, the more implicit redundant constraints $\mathcal{F}$ has, the more '0' singular values AD^*A^T has, i.e., ill-conditioned.

The self-dual embedding [162] is a popular formulation of the primal-dual **LP** pair used for an interior point method. An attractive feature of the self-dual embedding is that a *feasible* initial iterate in the interior of the cone is analytically given. The success of the self-dual embedding technique is supported by the strong performances of solvers such as MOSEK and SeDuMi. Hence, the lack of strict feasibility does not appear to be a concern at first glance. However, under the lack of strict feasibility, we show that we still encounter the ill-conditioning when we look for search directions. For instance, [162, equation (17)] displays the equation as a part of computing the search direction $(d_x; d_y)$:

$$\begin{bmatrix} X^kS^k & -X^kA^T \\ AX^k & 0 \end{bmatrix} \begin{pmatrix} (X^k)^{-1}d_x \\ d_y \end{pmatrix} = \begin{pmatrix} \gamma\mu^k e - X^k s^k \\ 0 \end{pmatrix} - \begin{bmatrix} X^k c & -X^k\bar{c} \\ -b & \bar{b} \end{bmatrix} \begin{pmatrix} d_\tau \\ d_\theta \end{pmatrix}.$$

Here, $X^k = \operatorname{Diag}(x^k)$ and $S^k = \operatorname{Diag}(s^k)$, where x^k, s^k are the current primal-dual iterate. It then uses the back-solve steps to complete the remaining components of the search direction. For simplicity, we set the right-hand-side of the system to be $\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$. By expanding the first block equation, we obtain

$$(X^kS^k)(X^k)^{-1}d_x - X^kA^Td_y = r_1 \iff (X^k)^{-1}d_x = (X^kS^k)^{-1}r_1 + (X^kS^k)^{-1}X^kA^Td_y.$$

We then substitute the equality above into the second block equation, i.e.,

$$\begin{array}{rcl} AX^k(X^k)^{-1}d_x = r_2 & \iff & AX^k\left[(X^kS^k)^{-1}r_1 + (X^kS^k)^{-1}X^kA^Td_y\right] = r_2 \\ & \iff & AX^k(X^kS^k)^{-1}X^kA^Td_y = r_2 - AX^k(X^kS^k)^{-1}r_1 \\ & \iff & AX^k(S^k)^{-1}A^Td_y = r_2 - AX^k(X^kS^k)^{-1}r_1. \end{array}$$

Hence, computing the search direction involves the normal matrix $AX^k(S^k)^{-1}A^T$ that appear in (4.3.4), and this matrix becomes very ill-conditioned as we get close to the feasible region with no strictly feasible point.

Lack of Strict Feasibility in the Dual We consider the facial reduction for the dual $(\mathcal{D})$; see (4.3.1). We denote the feasible set of the dual $(\mathcal{D})$ by

$$\mathcal{F}_{\mathcal{D}} := \{(y,s) \in \mathbb{R}^m \oplus \mathbb{R}^n_+ \ : \ A^Ty + s = c\} = \left\{(y,s) \in \mathbb{R}^m \oplus \mathbb{R}^n_+ \ : \ \begin{bmatrix} A^T & I \end{bmatrix} \begin{pmatrix} y \\ s \end{pmatrix} = c \right\}. \quad (4.3.5)$$

The facial reduction arguments applied to the dual are parallel to the ones given in Section 2.3.2. Hence, we provide a short derivation for the facially reduced system for $\mathcal{F}_{\mathcal{D}}$. We also conclude that the absence of strict feasibility for $\mathcal{F}_{\mathcal{D}}$ implies the dual degeneracy at all **BFS**s.

The following lemma is the theorem of the alternative applied to the set $\mathcal{F}_{\mathcal{D}}$.

Lemma 4.3.3. *[38, Theorem 3.3.10](theorem of the alternative in dual form) Let $\mathcal{F}_{\mathcal{D}}$ in (4.3.5) be feasible. Then, exactly one of the following statements holds:*

1. *There exists $(y,s) \in \mathbb{R}^m \oplus \mathbb{R}^n_{++}$ with $A^T y + s = c$, i.e., strict feasibility holds for $\mathcal{F}_{\mathcal{D}}$;*
2. *There exists $w \in \mathbb{R}^n$ such that*

$$0 \neq w \in \mathbb{R}^n_+, \ Aw = 0 \ \textit{ and } \ \langle c, w\rangle = 0. \tag{4.3.6}$$

We recall that the vector $A^T y$ in (2.3.11) is an exposing vector to the set $\mathcal{F}$. Similarly, the solution vector w to the auxiliary system (4.3.6) plays the role of an exposing vector for $\mathcal{F}_{\mathcal{D}}$:

$$\forall (y,s) \in \mathcal{F}_{\mathcal{D}}, \ \text{ it holds } \ \langle w, s\rangle = \langle w, c - A^T y\rangle = \langle c, w\rangle - \langle Aw, y\rangle = 0 - \langle 0, y\rangle = 0. \tag{4.3.7}$$

We let

$$\mathcal{I}_w = \{1,\ldots,n\} \setminus \operatorname{supp}(w), \ U = I_n(:,\mathcal{I}_w) \ \text{ and } \ s_w = |\operatorname{supp}(w)|. \tag{4.3.8}$$

Then, the facially reduced system of $\mathcal{F}_{\mathcal{D}}$ from (4.3.5) appears

$$\left\{ (y,u) \in \mathbb{R}^m \oplus \mathbb{R}^{n-s_w}_+ \ : \ \begin{bmatrix} A^T & U \end{bmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = c \right\}. \tag{4.3.9}$$

The notion of degeneracy in Definition 4.1.2 naturally extends to an arbitrary polyhedron $P \subseteq \mathbb{R}^n$, e.g., see [15, Section 2]. A point p in P is called a *basic solution* if there are n linearly independent active constraints at p. In addition, if there are more than n active constraints at the point $p \in P$, then the point p is called *degenerate*. Using this definition of the degeneracy, we now show that the absence of strict feasibility of $\mathcal{F}_{\mathcal{D}}$ implies that every basic solution of $\mathcal{F}_{\mathcal{D}}$ is degenerate.

We show that the facially reduced system in (4.3.9) contains a redundant constraint. Let w be a solution to the system (4.3.6), i.e., w is an exposing vector for $\mathcal{F}_{\mathcal{D}}$. Then we have

$$\begin{bmatrix} A \\ U^T \end{bmatrix} w = \begin{bmatrix} Aw \\ U^T w \end{bmatrix} = \begin{bmatrix} 0_m \\ 0_{n-s_w} \end{bmatrix}. \tag{4.3.10}$$

In other words, there is a nontrivial row combination of $\begin{bmatrix} A^T & U \end{bmatrix}$ that yields the 0 vector, i.e., there exists a redundant row in $\begin{bmatrix} A^T & U \end{bmatrix}$. Hence, the facially reduced system contains a redundant constraint. The redundancy immediately implies the dual degeneracy; for any basic solution of $\mathcal{F}_{\mathcal{D}}$, there always exists an implicit redundant equality in $\begin{bmatrix} A^T & I \end{bmatrix} \begin{pmatrix} y \\ s \end{pmatrix} = c$.

A popular method for rewriting an instance with a free variable $x_i \in \mathbb{R}$ into the primal standard form is to write x_i into the difference of two nonnegative variables, i.e.,

$$x_i = x_i^+ - x_i^- \text{ with } x_i^+, x_i^- \geq 0.$$

This equivalent transformation does not seem to cause any difficulties at first glance; at least the primal simplex method does not consider both x_i^+ and x_i^- as basic variables simultaneously in order to form a (nonsingular) basis matrix. Moreover, a method for computing an element-wise positive starting point for an interior point method that uses this type of decomposition is introduced in [106]. However, this equivalent transformation has a significant consequence to the dual program. For any $K \geq \max\{x_i^+, x_i^-\}$, we can maintain the equality

$$x_i = x_i^+ - x_i^- = (x_i^+ + K) - (x_i^- + K).$$

Thus, the primal optimal set is unbounded. This implies that the dual feasible region of the reformulated primal does not have a strictly feasible point. Consequently, the results established for the primal apply to the dual; (i) this decomposition forces all **BFS**s of the dual $(\mathcal{D})$ to be degenerate; (ii) the equality system for the dual feasibility contains implicit redundancies and thus the Newton equation (4.3.3) becomes very ill-conditioned near an optimum.

4.4 Numerical Experiments

We now provide empirical evidences that **FR** is indeed a useful preprocessing tool for reducing the size of problems as well as for improving the *conditioning*. We do this first for interior point methods and then for simplex methods. In particular, this provides empirical evidence that lack of strict feasibility brings out implicit singularity. We use three different solvers in our tests[6]: (i) *linprog* from MATLAB[7]; (ii) *SDPT3*[8]; and (iii) *MOSEK*[9]. MATLAB version 2021a is used to access all the solvers for the tests, and we use their default settings for stopping criteria.

4.4.1 Empirics with Interior Point Methods

In this section we compare the behaviour for finding near-optimal points with instances that do and do not satisfy strict feasibility. More specifically, given a near optimal primal-dual point $(x^*, s^*) \in \mathbb{R}^n_{++} \oplus \mathbb{R}^n_{++}$ obtained from an interior point solver, we observe the condition number, i.e., the ratio of largest to smallest eigenvalues of the normal matrix at (x^*, s^*):

$$\kappa\left(AD^*A^T\right), \quad \text{where } D^* = \operatorname{Diag}(x^*)\operatorname{Diag}(s^*)^{-1}. \tag{4.4.1}$$

There is a comprehensive survey [83] that concerns problems caused by degeneracies when an interior point method is chosen for **LP**s. The survey [83] addresses the effect of degeneracy on the convergence of interior point methods and numerical performance, etc. We show that instances that do not have strictly feasible points tend to have significantly larger condition numbers of the normal equation near the optimum. We also present a numerical experiment on perturbations of the right-hand-side vector b.

[6] All the numerical tests are performed using MATLAB version 2021a on Dell XPS 8940 with 11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz 2.60 GHz with 32 Gigabyte memory.

[7] https://www.mathworks.com/. Version 9.10.0.1669831 (R2021a) Update 2.

[8] https://www.math.cmu.edu/~reha/sdpt3.html, version SDPT3 4.0, [149]

[9] https://www.mosek.com/. Version 8.0.0.60.

4.4.1.1 Generating LPs without Strict Feasibility

Given $m, n, r \in \mathbb{N}$, we construct the data $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ to satisfy (2.3.4) with r as the dimension of the relative interior of $\mathcal{F}$, relint$(\mathcal{F})$.

1. Pick any $0 \neq y \in \mathbb{R}^m$. Let
$$\{y\}^\perp = \operatorname{span}\{a_i\}_{i=1}^{m-1} \quad (= \operatorname{null}(y^T)).$$
We let $R \in \mathbb{R}^{(m-1)\times r}$ be a random matrix, and get
$$A_1 := \begin{bmatrix} a_1 & \dots & a_{m-1}\end{bmatrix} R \in \mathbb{R}^{m\times r}, \quad A_1^T y = 0 \in \mathbb{R}^r.$$
2. Pick any $\hat{v} \in \mathbb{R}^r_{++}$ and set $b = A_1\hat{v}$. We note that $y^T A_1 = 0$ and $\langle b, y\rangle = 0$.
3. Pick any matrix $A_2 \in \mathbb{R}^{m\times(n-r)}$ satisfying $(y^T A_2)_i \neq 0, \ \forall i$. If there exists i such that $(y^T A_2)_i < 0$, then change the sign of the i-th column of A_2 so that we conclude
$$(A_2^T y) \in \mathbb{R}^{n-r}_{++}.$$
4. We define the matrix $A = \begin{bmatrix} A_1 & A_2\end{bmatrix} \in \mathbb{R}^{m\times n}$. Then $\{x \in \mathbb{R}^n_+ : Ax = b\}$ is a polyhedron with a feasible point $\hat{x} = [\hat{v}; 0]$ having r number of positives. The vector y is a solution for the system (2.3.4)
$$0 \lneq z = A^T y = \begin{pmatrix} A_1^T y = 0 \\ A_2^T y > 0\end{pmatrix}, \ b^T y = 0.$$
We then randomly permute the columns of A to avoid the zeros always being at the bottom of the feasible variables x.

For the empirics, we construct the objective function $c^T x$ of $(\mathcal{P})$ as follows. We choose any $\bar{s} \in \mathbb{R}^n_{++}, \bar{y} \in \mathbb{R}^m$ and set $c = A^T\bar{y} + \bar{s}$. Then we have the data for the primal-dual pair of **LP**s and the primal *fails* strict feasibility:

$$(\mathcal{P}_{(A,b,c)}) \quad \min\{\ c^T x : Ax = b,\ x \geq 0\ \} \quad \text{and} \quad (\mathcal{D}_{(A,b,c)}) \quad \max\{\ b^T y : A^T y + s = c,\ s \geq 0\ \}.$$

We note that by choosing $\bar{s} \in \mathbb{R}^n_{++}$, the dual problem $(\mathcal{D}_{(A,b,c)})$ has a strictly feasible point. In order to generate instances with strictly feasible points, we maintain the same data A, c used for the pair $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{D}_{(A,b,c)})$. We only redefine the right-hand-side vector by $\bar{b} = Ax^\circ$, where $x^\circ \in \mathbb{R}^n_{++}$:

$$(\bar{\mathcal{P}}_{(A,\bar{b},c)}) \quad \min\{\ c^T x : Ax = \bar{b},\ x \geq 0\ \} \quad \text{and} \quad (\bar{\mathcal{D}}_{(A,\bar{b},c)}) \quad \max\{\ \bar{b}^T y : A^T y + s = c,\ s \geq 0\ \}.$$

The facially reduced instances of $(\mathcal{P}_{(A,b,c)})$ are denoted by $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$. They are obtained by discarding the variables that are identically 0 in the feasible set $\mathcal{F}$ and the redundant constraints. In other words, the affine constraints of $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are of the form (3.1.3).

4.4.1.2 Condition Numbers

In order to illustrate the differences in condition numbers of the normal matrices, we solve the three families of instances: (i) $(\mathcal{P}_{(A,b,c)})$, strictly feasible fails; (ii) $(\bar{\mathcal{P}}_{(A,\bar{b},c)})$, strictly feasible holds; (iii)

$(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$, facially reduced instances of $(\mathcal{P}_{(A,b,c)})$.

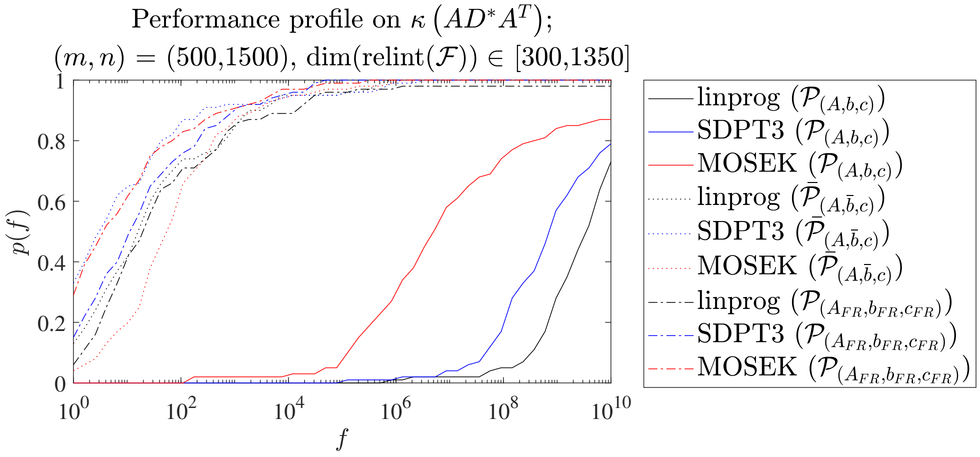

Figure 4.4.1: Performance profile on $\kappa\left(AD^*A^T\right)$ with(out) strict feasibility near optimum; various solvers

In Figure 4.4.1 we use a *performance profile* [47,78] to observe the overall behaviour on different families of instances using the three solvers. The performance profile provides a useful graphical comparison for solver performances. Figure 4.4.1 displays the performance profile on the condition numbers of the normal matrix AD^*A^T near optimal points from different solvers. We generate 100 instances for each family that have $\dim(\text{relint}(\mathcal{F})) \in [300, 1350]$. The instance sizes are fixed with $(m,n) = (500, 1500)$. The vertical axis in Figure 4.4.1 represents the statistics of the performance ratio on $\kappa\left(AD^*A^T\right)$, the condition number of normal matrix near optimum (x^*, s^*); see (4.4.1). Roughly speaking, the vertical axis represents the probability of achieving a performance ratio within a factor of f among all methods used. We used the lower the better statistics. The details of the performance ratio are discussed in [47, 78]. The solid lines in Figure 4.4.1 represent the performance of the instances $(\mathcal{P}_{(A,b,c)})$ that fail strict feasibility. They show that the condition numbers of the normal matrices near optima are significantly higher when strict feasibility fails. That is, when strict feasibility fails for $\mathcal{F}$, the matrix AD^*A^T is more ill-conditioned and it is difficult to obtain search directions of high accuracy. We also observe that facially reduced instances yield smaller condition numbers near optima. We note that the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are equivalent.

4.4.1.3 Stopping Criteria

We now use the three solvers to observe the accuracy of the first-order optimality conditions (KKT conditions) and the running times, for the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$, see Table 4.4.1. We test the average performance of 10 instances of the size $(n,m,r) = (3000, 500, 2000)$. The headers used in Table 4.4.1 provide the following. Given solver outputs (x^*, y^*, s^*), the header 'KKT' exhibits the average of the triple consisting of the primal feasibility, dual feasibility and complementarity;

$$\text{KKT} = \left(\frac{\|Ax^* - b\|}{1 + \|b\|}, \ \frac{\|A^Ty^* + s^* - c\|}{1 + \|c\|}, \ \frac{\langle x^*, s^* \rangle}{n} \right).$$

The headers 'iter' and 'time' in Table 4.4.1 refer to the average of the number of iterations and the running time in seconds, respectively.

		Non-Facially Reduced System	Facially Reduced System
linprog	KKT	(2.44e-15, 2.05e-12, 3.18e-09)	(5.85e-16, 4.74e-16, 9.22e-09)
	iter	22.30	17.90
	time	2.34	0.81
SDPT3	KKT	(8.11e-10, 7.55e-12, 5.65e-02)	(1.43e-11, 3.67e-16, 4.38e-06)
	iter	25.50	19.30
	time	1.73	0.70
mosek	KKT	(7.52e-09, 1.80e-15, 3.27e-06)	(3.85e-09, 3.69e-16, 1.19e-06)
	iter	40.30	10.20
	time	1.40	0.35

Table 4.4.1: Average of KKT conditions, iterations and time of (non)-facially reduced problems

From Table 4.4.1 we observe that facially reduced instances provide significant improvement in first-order optimality conditions, the number of iterations and the running times for all solvers in general. We note that the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are equivalent. Hence, our empirics show that performing facial reduction as a preprocessing step not only improves the solver running time but also the *quality* of solutions.

4.4.1.4 Distance to Infeasibility

In this section we present empirics that illustrate the impact of perturbations of the right-hand-side b when strict feasibility fails. We recall, from Proposition 4.3.1, that there exists an arbitrarily small perturbation of the right-hand-side vector b that renders the problem infeasible. Meanwhile, a carefully chosen perturbation to b does not force the infeasibility.

We follow the steps in Section 4.4.1.1 to generate instances of the order $(n,m) = (1000,200)$ and $r = \text{relint}(\mathcal{F}) = 900$. The objective function $c^T x$ is chosen as presented in Section 4.4.1.1. For the fixed (n,m,r), we generate 10 instances and observe the average performance of these instances as we gradually increase the magnitude of the perturbation. We recall the matrix AV from (2.3.12). We use two types of perturbations for b;

$$\Delta b, \text{ where } \Delta b \in \text{range}(AV)^{\perp}, \quad \Delta \bar{b}, \text{ where } \Delta \bar{b} \in \text{range}(AV).$$

We choose Δb to be the vector y that satisfies (2.3.4). For $\Delta \bar{b}$, we choose AVd, where $d \in \mathbb{R}^r$ is a randomly chosen vector. As we increase $\epsilon > 0$, we observe the performance of the two families of the systems

$$Ax = b_\epsilon := b - \epsilon \Delta b \quad \text{and} \quad Ax = \bar{b}_\epsilon := b - \epsilon \Delta \bar{b}.$$

We use the interior point method from MATLAB's linprog for the test. Figure 4.4.2 contains the average of the first-order optimality conditions evaluated at the solver outputs (x^*, y^*, s^*) of these instances; primal feasibility, dual feasibility and the complementarity.

The horizontal axis of Figure 4.4.2 indicates the degree of the perturbation imposed on the right-hand-side vector b, $\epsilon\|\Delta b\|$ and $\epsilon\|\Delta \bar{b}\|$. The vertical axis indicates the individual component of the first-order optimality. From Figure 4.4.2, we observe that the KKT conditions with the perturbation $\Delta \bar{b}$ display a steady performance regardless of the perturbation degree; see the markers ○,□,△ with the dotted lines. In contrast, the markers ●,■,▲ in Figure 4.4.2 exhibit the performance of the instances that are perturbed with Δb and they display a different performance. In particular, we

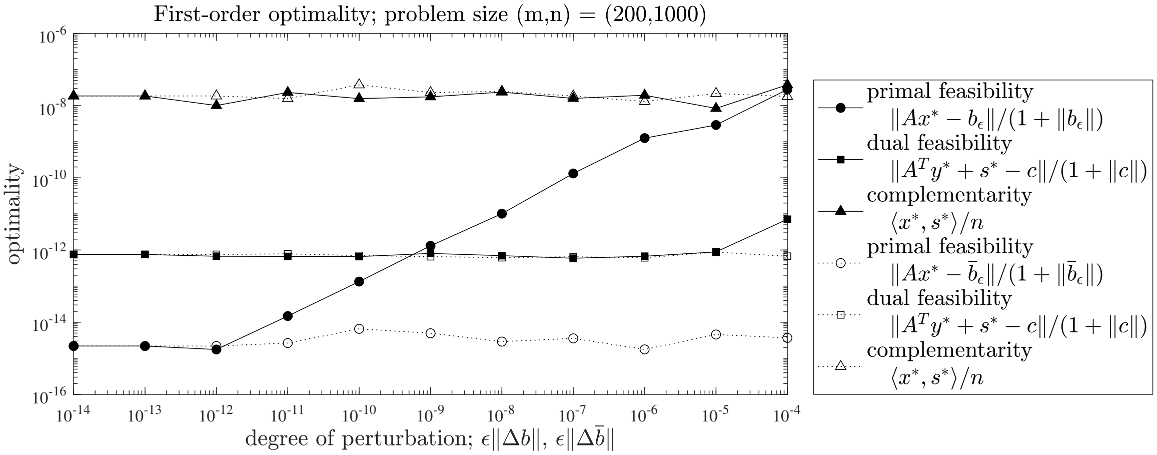

Figure 4.4.2: Changes in the first-order optimality conditions as the perturbation of b increases

see that the relative primal feasibility $\|Ax^* - b_\epsilon\|/(1 + \|b_\epsilon\|)$, marked with $\bullet$, consistently increases as the perturbation magnitude $\epsilon\|\Delta b\|$ increases when strict feasibility fails for $\mathcal{F}$.

4.4.1.5 Empirics on Singular Values maxsd and ips

We show that, in the absence of strict feasibility, the ill-conditioning of the matrix AD_cA^T not only originates form D_c but also originates from the rows of A. We exhibit that the number of small singular values (in the relative sense) of the normal matrix is closely related to the **maxsd** and **ips**. Hence, a large **ips** is a good indicator for ill-conditioning.

We generated instances with different settings for **maxsd** $= 1, 5$ and 10. We recall the generation for the vector y and A_2 in Section 4.4.1.1. For generating and instance with **maxsd** > 1, we generated $Y_c = \text{BlkDiag}(y^1, \dots, y^{\textbf{maxsd}}) \in \mathbb{R}^{m\times \textbf{maxsd}}$ and $A_2 = \text{BlkDiag}(A_2^1, \dots, A_2^{\textbf{maxsd}})$ of appropriate dimension in order to produce the exposing vector $A_2^T \sum_{j=1}^{\textbf{maxsd}} Y_c(:,j) \geq 0$. Each column of Y_c serves as a vector satisfying (2.3.4).

Let $\sigma_{\max}(AD^*A^T)$ be the maximum singular value of AD^*A^T. We count the number of singular values of AD^*A^T that are smaller than $10^{-8} \cdot \sigma_{\max}(AD^*A^T)$. In Table 4.4.2 below, we report the cardinality of

$$\Sigma_0 := \{i : \sigma_i(AD^*A^T) < \sigma_{\max}(AD^*A^T)\}.$$

We test the average performance on the 20 instances of the fixed size $(n, m, r) = (3000, 500, 2000)$. We display the average number of $|\Sigma_0|$. We see from Table 4.4.2 a larger **maxsd** and **ips** value

		maxsd = 1	maxsd = 5	maxsd = 10
linprog	$\lvert\Sigma_0\rvert$	4.10	8.65	13.10
SDPT3	$\lvert\Sigma_0\rvert$	4.75	8.00	34.65
MOSEK	$\lvert\Sigma_0\rvert$	6.45	12.35	14.50

Table 4.4.2: # (rel.) small singular values of AD^*A^T near optimum; average over 20 instances

produce a greater number of small singular values. When there is a significant number of redundant constraints, it is more difficult to obtain a good search direction due to a large number of relatively small singular values.

4.4.2 Empirics with Simplex Method and Problems where Strict Feasibility Fails

In this section we compare the behaviour of the dual simplex method with instances that have strictly feasible points and instances that do not. We also observe the degeneracy issues that arise in the instances from NETLIB[10].

4.4.2.1 Generating Dual LPs without Strict Feasibility

We first show how to generate an instance for the dual feasible set $\mathcal{F}_\mathcal{D}$ that fails strict feasibility. The construction is similar to the one in Section 4.4.1.1. We generate a degenerate problem by constructing a consistent auxiliary system (4.3.6). Given $m, n, r \in \mathbb{N}$, we construct $A \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^n$ that satisfy (4.3.6) with $\dim(\operatorname{relint}(\mathcal{F}_\mathcal{D})) = m + r$.

1. Pick any $0 \neq w \in \mathbb{R}^n_+$ with $|\operatorname{supp}(w)| = n - r$. Let
$$\{w\}^\perp = \operatorname{span}\{d_i\}_{i=1}^{n-1} \subset \mathbb{R}^n \quad \left(= \operatorname{null}(w^T)\right).$$
We let $D \in \mathbb{R}^{(n-1)\times n}$ be the matrix where its rows consist of $\{d_i^T\}_{i=1}^{n-1}$. We let $R \in \mathbb{R}^{m\times(n-1)}$ be a random matrix and we set $A = RD$. We note that $Aw = 0$.
2. Pick $s \in \mathbb{R}^n_+$ so that
$$s_i = \begin{cases} 0 & \text{if } i \in \operatorname{supp}(w) \\ \text{positive} & \text{if } i \notin \operatorname{supp}(w). \end{cases}$$
We note that $\langle w, s\rangle = 0$ holds.
3. Pick $y \in \mathbb{R}^m$ and set $c = A^T y + s$. We note that $\langle c, w\rangle = 0$ holds.

For the empirics, we construct the objective function $b^T y$ of $(\mathcal{D})$ by choosing a vector $\hat{x} \in \mathbb{R}^n_{++}$ and setting $b = A\hat{x}$.

4.4.2.2 Empirics on the Number of Degenerate Iterations

In this section we test how the lack of strict feasibility affects the performance of the dual simplex method. We choose MOSEK for our tests since MOSEK reports the percentage of degenerate iterations as a part of the solver report. MOSEK reports the quantity 'DEGITER(%)', the ratio of degenerate iterations.

Given a set $\mathcal{F}_\mathcal{D}$ and a point $(y, s) \in \operatorname{relint}(\mathcal{F}_\mathcal{D}) \subseteq \mathbb{R}^m \oplus \mathbb{R}^n_+$, let r be the number of positive entries of s, i.e., $r = |\operatorname{supp}(s)|$. In our tests, we gradually increase r for fixed n, m and generate instances for $\mathcal{F}_\mathcal{D}$ as described in Section 4.4.2.1. We then observe the behaviour of the dual simplex method. Table 4.4.3 contains the results. In Table 4.4.3, a smaller value for the header $(r/n)\%$ means that there are more entries of s that are identically 0 in the set $\mathcal{F}_\mathcal{D}$; and the value 0% means that strict feasibility holds. For each triple (n, m, r), we generated 10 instances and we report the average of 'DEGITER(%)' of these instances.

[10] https://www.netlib.org/lp/

		$100\% - (r/n)\%$				
		40	30	20	10	0
(n,m)	(1000, 250)	36.62	10.18	0.01	0.02	0.00
	(2000, 500)	39.72	18.28	0.07	0.15	0.01
	(3000, 750)	25.99	10.66	0.32	0.75	0.02
	(4000, 1000)	29.78	18.25	0.25	0.53	0.02

Table 4.4.3: Average of the ratio of degenerate iterations

We recall Theorem 4.1.3: lack of strict feasibility implies that all **BFS**s are degenerate. However, we observe more, i.e., from Table 4.4.3, the frequency of degenerate iterations increases as r decreases. In other words, higher degeneracy of the set $\mathcal{F}_\mathcal{D}$ yields more degenerate iterations when the dual simplex method is used.

4.4.2.3 NETLIB Problems; Perturbations; Stability

We now illustrate the lack of strict feasibility on instances in the NETLIB[11] data set. We use the following 67 instances that are in the standard form at this link:

25fv47 adlittle* afiro agg* agg2* agg3* bandm* beaconfd* blend bnl1*
bnl2* brandy* cre_a* cre_b* cre_c* cre_d* d2q06c* degen2* degen3* e226*
fffff800* israel lotfi maros_r7 nug05 nug06 nug07 nug08 nug12 nug15
nug20 osa_07* osa_14* qap12 qap15 qap8 sc105* sc205* sc50a* sc50b*
scagr25 scagr7 scfxm1* scfxm2* scfxm3* scorpion* scrs8* scsd1 scsd6 scsd8
sctap1 sctap2 sctap3 share1b share2b ship04l* ship04s* ship08l* ship08s* ship12l*
ship12s* stocfor1 stocfor2 stocfor3 truss wood1p* woodw*

The only preprocessing performed to these instances is removing the redundant rows of the data matrix A. For each instance, a **BFS** is obtained by solving the problem $\min_x\{\langle e, x\rangle : x \in \mathcal{F}\}$ using MOSEK. We use this **BFS** to set the initial **BFS** $\bar{x}$ described at the beginning of Algorithm 4.2.2. We then use Algorithm 4.2.2 to determine if strict feasibility holds or not.

That the feasible linear conic programs has strictly feasible points is generic is shown in [52]. However, there are many real-life instances that do not seem to possess this property.[12] Surprisingly, the Slater condition fails for 37 out of these 67 instances; the instances that fail strict feasibility are marked with the asterisk $*$ in the list above. This has interesting implications for both the interior point and simplex methods. The standard interior point method stopping criteria become complicated by the unbounded dual optimal set. For the primal simplex method, every iteration will always visit degenerate **BFS**s. Therefore preprocessing to eliminate the variables fixed at 0 is important. In addition, in order to motivate robust optimization, it is shown in e.g., [13, 14] that optimal solutions of many of the NETLIB instances are extremely sensitive to perturbations in the data. We now see this to be the case, and we show that **FR** regularizes the problem and avoids this instability.

We first use the instance `degen3` in order to illustrate the consequence of lack of strict feasibility. The data matrix A after removing two redundant rows is 1501-by-2604. After **FR**, we obtain the constraint matrix $P_{\bar{m}}AV$ of size 1226-by-1648. This implies that $2604 - 1648 = 956$ number of variables are identically 0 on the feasible set. Furthermore, $\mathbf{ips}(\mathcal{F}) = 275$ equality constraints are implicitly redundant. By Item 2 of Corollary 4.1.12, without **FR**, the minimum degree of

[11] https://www.netlib.org/lp/

[12] This also continues to hold in the applications that we see in Chapters 5 and 6.

degeneracy of all **BFS**s is at least 275. Namely, the length of the basis is 1501 and every basis contains at least 275 degenerate indices.

We now illustrate that **FR** gives a more robust model with respect to data perturbations using the instance `brandy`. Let (A, b) be the data after removing the redundant equalities constraints. Let $(P_{\bar{m}}AV, P_{\bar{m}}b)$ be the data for the facially reduced system. The data matrices A and $P_{\bar{m}}AV$ have the sizes 193-by-303 and 155-by-260, respectively[13]. We set the perturbation scalars $\epsilon_A = \epsilon_b = 10^{-9}$. We construct a random perturbation matrix Φ, $\|\Phi\|_F = \|A\|_F + 1$, and random perturbation vector ϕ, $\|\phi\|_2 = \|b\|_2 + 1$. We then solve the problem

$$\tilde{p}^* = \max\{\langle c, x\rangle : (A + \epsilon_A \Phi)x = b + \epsilon_b \phi,\ x \geq 0\}.$$

For the facially reduced system, we use the identical perturbation data Φ, ϕ and discard the rows and columns of (A, b) found from **FR**. That is, we use the perturbations $P_{\bar{m}}\Phi V$ and $P_{\bar{m}}\phi$ to the facially reduced system after the scaling $\|P_{\bar{m}}\Phi V\|_F = \|P_{\bar{m}}AV\|_F + 1$ and $\|P_{\bar{m}}\phi\|_2 = \|P_{\bar{m}}b\|_2 + 1$. We then solve

$$\max\{\langle V^T c, v\rangle : (P_{\bar{m}}AV + \epsilon_A P_{\bar{m}}\Phi V)v = P_{\bar{m}}b + \epsilon_b P_{\bar{m}}\phi,\ v \geq 0\}.$$

In this way, we maintain the identical perturbation structure for the original system and the facially reduced system. We also generate a transportation problem and use the aforementioned perturbations. We note that the transportation problems have Slater points but are known to be highly degenerate. The size of the data generated is 49-by-600.

In the experiment, we tested the instances using 100 different perturbation settings. We randomly generated perturbations Φ, ϕ with the density set at 0.1. We used MOSEK simplex with the setting 'MSK_OPTIMIZER_FREE_SIMPLEX'. In Table 4.4.4, the headers ϵ_A and ϵ_b refer to the scalars used for perturbations as described above. The headers (A, b), $(P_{\bar{m}}AV, P_{\bar{m}}b)$ and $(A_{\text{trans}}, b_{\text{trans}})$ refer to the non-facially reduced system, the facially reduced system and the transportation problems, with the perturbations. The integral values in the table indicate the number of times that the solver outputs PRIMAL_AND_DUAL_FEASIBLE. Let p^* be the optimal value for the unperturbed instance `brandy`, and let $\tilde{p}^*$ be the optimal value of a perturbed instance of `brandy`. The non-integral values in the table indicate the average relative difference in the optimal values between p^* and $\tilde{p}^*$. The relative difference is computed using the formula $\frac{|p^* - \tilde{p}^*|}{2|p^* + \tilde{p}^*|}$. For example, the first entry 11 in Table 4.4.4 means that $100 - 11$ out of 100 perturbed instances yield infeasibility or unknown status, i.e., only 11 solved successfully. The entry 4.938e-02 next to 11 indicates the average of $\frac{|p^* - \tilde{p}^*|}{2|p^* + \tilde{p}^*|}$ on those 11 instances. We see in columns (A, b) and $(P_{\bar{m}}AV, P_{\bar{m}}b)$ in Table 4.4.4 demonstrate that the facially reduced problems are more immune to data perturba-

ϵ_A	ϵ_b	(A, b)	$(P_{\bar{m}}AV, P_{\bar{m}}b)$	$(A_{\text{trans}}, b_{\text{trans}})$
1.0e-09	0	(11 , 4.938e-02)	(97 , 6.705e-03)	100
0	1.0e-09	(27 , 2.470e-10)	(100 , 2.208e-10)	100
1.0e-09	1.0e-09	(11 , 1.339e-01)	(96 , 8.719e-03)	100

Table 4.4.4: Number of successful results out of 100 perturbed instances using simplex method on the instance `brandy` and transportation problem

[13] This also means that, without **FR**, every **BFS** has at least 38 degenerate basic variables. At least 19.69 percent of basic variables are always degenerate.

tions; the number of successfully solved perturbed instances are significantly larger and the optimal values under the perturbations are less influenced. The last column indicates that although the instance may have many degenerate **BFS**s, having a strictly feasible point is important in terms of perturbations in data, i.e., this emphasizes the difference between the two types of degeneracy.

4.4.2.4 Preprocessing for LP and Beyond

Our preprocessing method, Algorithm 4.2.2, is also applicable to different classes of problems that have polyhedral feasible regions, e.g., the standard quadratic program [61, 157]

$$\min_x \left\{ x^T Q x + q^T x + \gamma \ : \ Ax = b, \ x \geq 0 \right\},$$

where $Q \in \mathbb{S}^n, q \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$. Moreover, our proposed approach is also applicable to the **LP** relaxation of a mixed integer programming (**MIP**). The popularity of the **MIP** has grown due to its ability to integrate situations that arise in real-life problems. For instance, an **MIP** with binary variables (0 or 1) can model decisions that arise in real life situations; a variable equal to 1 may indicate 'open the gate', and 0 for 'close the gate'[14]. The integer restrictions on the variables lead to the nonconvex constraint set and hence solving an **MIP** is considered a challenging task. The branch-and-bound is a popular method for handling an **MIP** and it requires solving many subproblems that contain the originally given constraints. As a presolving step, we can also employ Algorithm 4.2.2 in order to detect fixed variables via **LP** relaxation. As a case study, we use the instance named `acc-tight5` from MIPLIB[15], the mixed integer programming library. The instance is given in the form

$$\min_x \{c^T x : A_{\text{eq}} x = b_{\text{eq}}, A_{\text{ineq}} x \leq b_{\text{ineq}}, \ell \leq x \leq u, \ x \text{ integer}\}$$

and we transform the instance into the standard form (4.1.1) by adding slack variables. The instance `acc-tight5` contains binary variables, i.e., $\ell = 0, u = e$. With Algorithm 4.2.2, we detect 670 variables are that fixed at 0 from the **LP** relaxation in the standard form. Interpreting this into the original form, 168 out of 670 correspond to the lower bounds that are fixed at 0; 602 out of 670 correspond to the slack variables associated with $A_{\text{ineq}} x \leq b_{\text{ineq}}$ that are fixed at 0, i.e., 602 inequality constraints are implicitly equality constraints.

4.4.3 Concluding Remarks

Throughout Section 4.4 we have shown that fail to eliminate the variables fixed at 0, i.e., lack of strict feasibility, gives rise to implicit problem singularity and this helps explain the numerical difficulties that arise. There are related works [62, 147] that aim to identify inequality constraints that are implicitly equality constraints. Our work relates in the sense that identifying the variables fixed 0 is the process for finding always active constraints within the nonnegativity $x \geq 0$. However, the theoretical consequences of having identically 0 variables do not seem to appear broadly in the literature.

[14] We also see this binary decision problem through the lens of **DNN** relaxation in Chapter 5. More specifically, we consider the case where 1 indicates the membership inclusion to a set whereas 0 indicates the exclusion.

[15] `https://miplib.zib.de`

An essential step for almost all algorithms for **LP** is preprocessing. There are many preprocessing methods for achieving problem reduction by observing the structure of the data; see e.g., [94]. One part of preprocessing is identifying fixed variables. However, identifying variables fixed at 0, facial reduction, has not been actively done due to expense and accuracy problems. Among many advantages of our approach, we point out the accuracy of the exposing vector that our approach produces. The exposing vector that we obtain from the simplex method is accurate within machine accuracy. Interior point methods can be used to identify some exposed variables; [133] produces exposing vectors by using the self-dual embedding; [32] identifies variables fixed at 0 by constructing a merit function that consists of the first-order optimality conditions. As pointed out in [38, 133], the exposing vector that we obtain from the interior point type methods are *approximate*. The difficulties that arise in the approaches using an interior point method is addressed through the numerical experiment in [133] for the classes of **SDP** and this continues to hold true for the case of **LP**.

We investigated the main numerical difficulties that arise with the interior point and simplex methods. For interior point methods, we displayed the importance of strict feasibility using condition numbers and relationships with distance to infeasibility. We also shed light on the main difficulties that arose with the implicit redundant constraints and used the QR decomposition to show how these difficulties come into play. This also relates to the implicit problem singularity, **ips**. A larger **ips** means that there is a higher chance of inducing an infeasible problem under perturbations. A large number of degenerate **BFS**s is believed to cause difficulties for the simplex method. We have shown that the settings for having many identically 0 variables in the dual program yield many degenerate iterations. That the **ips** provides a lower bound on the degree of degeneracy of all **BFS**s adds importance of exploiting the implicit redundancies. We also have shown that many NETLIB instances fail strict feasibility and used selected instances to show the effect of this degeneracy. Moreover, the facially reduced problems are seen to be more robust with respect to data perturbations.

Although degeneracy is a well-known subject, to the best of our knowledge, the relationships between degeneracy and stability are rarely discussed. We showed that the degeneracy at a **BFS** provides useful information on the robustness of the **LP**; the least degenerate **BFS** provides an upper bound on the number of implicitly redundant equalities of the set $\mathcal{F}$. We note that an $\mathcal{F}$ that contains a large number of implicit redundancies is more susceptible to be ill-posed. We have also provided an important modelling perspective on the usual treatment of free variables in the literature, i.e., a free variable x_i is generally replaced by the difference of two nonnegative variables, $x_i \leftarrow x_i^+ - x_i^-$. We have shown that this decomposition results in the absence of strict feasibility for the dual. Consequently, this results in the ill-posed dual problem. Furthermore, all **BFS**s observed from the dual simplex method are degenerate and the Newton equation of the interior point methods is ill-conditioned near an optimal point.

Part II

Broad Application of Facial Reduction

Chapter 5

A Restricted Dual PRSM and its Applications to DNN Relaxation of Binary Quadratic Program

A *binary quadratic problem,* ***BQP***, is a class of optimization problems with a quadratic objective function and variables restricted to be either 0 or 1. Many real-life applications of hard combinatorial optimization problems are posed as **BQP**s with additions of some affine constraints. Special instances of **BQP**s include the protein side-chain positioning problem [27], the quadratic assignment problem [166] and the minimum-cut problem [108]. Solving a **BQP** is NP-hard in general and hence many methods based on heuristics and branch-and-bound are proposed. We approach the problem via **SDP** and **DNN** relaxations.

A recent work by Oliveira et al., [123] uses the *alternating direction method of multipliers* (**ADMM**) to solve the **SDP** (and **DNN**) relaxation of the quadratic assignment problem, **QAP**. The **SDP** relaxation of the **QAP** fails strict feasibility; see [166]. Thus **FR** is invited and this allows for a variable substitution of the form $Y = VRV^T$, where $R \succeq 0$; see Section 2.3.2. Meanwhile, the **SDP** relaxation contains a set of constraints that fixes some elements of VRV^T to be 0 or 1. The **DNN** relaxation is a stronger relaxation than the **SDP** relaxation. Simply put, the **DNN** relaxation of the **BQP** contains polyhedral constraints of the type $0 \leq (VRV^T)_{i,j} \leq 1$, for all i, j. These polyhedral constraints in conjunction with the cone constraint, $R \succeq 0$, are difficult to satisfy simultaneously. The **FR** grants a natural splitting of variables and the **ADMM** framework provides an effective technique for handling these constraints individually; numerically hard problems are divided into simpler subproblems. The approach [123] provides the basic principle of the splitting method for solving facially reduced **SDP** relaxations and the numerical experiments show great success.

In this chapter we focus on the **DNN** relaxations of binary quadratic problems that have a special affine constraint that we name the *unit row-sum* constraint. We provide a simple derivation of the **DNN** relaxation via a direct lifting. Moreoever, we exploit the embedded structures for this class of problems. The **FR** leads to the discovery of the implicit redundant constraints and a prior knowledge on the dual optimal solutions. Continuing with the variable splitting provided by the **ADMM** framework by Oliveira et al., [123], we rather use the (customized) Peaceman-Rachford splitting method to work with the **DNN** relaxations.

Contributions and Outline The contribution of this chapter is threefold.

1. We introduce the structural properties embedded in the **DNN** relaxation of the binary quadratic problems with the unit row-sum constraint.

2. We introduce a variant of the Peaceman-Rachford splitting method that uses known dual optimal elements, and we study the derivation.

3. We discuss the **DNN** relaxations of two classes of binary quadratic problems: the protein side-chain positioning problem, and the quadratic assignment problem.

This chapter is organized as follows. In Section 5.1 we introduce a simple derivation for the **DNN** relaxation of the binary quadratic problem with the *unit row-sum* constraint. We provide common properties of the **DNN** relaxation of the **BQP**. In particular, we show that there are known elements in the set of the dual optimal solutions of the **DNN** relaxation. In Section 5.2 we provide the derivation of the Peaceman-Rachford splitting method via the Lagrangian dual. We then derive a variant **PRSM**, a restricted dual Peaceman-Rachford splitting method (**rPRSM**), by adding information of the known optimal dual elements. Finally we apply our framework to two classes of **DNN** relaxations of real-world **BQP**s: the side-chain positioning problem (Section 5.3), and the quadratic assignment problem (Section 5.4). We derive the **sd**, **maxsd** and **ips** of these two applications in Section 5.5.

5.1 DNN Relaxation of Binary Quadratic Program

In this section we provide a unified derivation for the **DNN** relaxation for binary quadratic problems with the *unit row-sum* constraint. The derivation of the **SDP** relaxation of **BQP** is presented using Lagrangian duality in e.g., see [38,166]. We provide a much simplified derivation that uses a direct lifting with essential constraints.

We define the unit row-sum constraint as follows. We let

$$n_0 := \sum_{i=1}^{p} m_i, \quad \text{where } m_1, \ldots, m_p \text{ are given positive integers.}$$

We define the 0-1 matrix

$$A_u = \operatorname{BlkDiag}(\bar{e}_{m_1}^T, \bar{e}_{m_2}^T, \cdots, \bar{e}_{m_p}^T) = \begin{bmatrix} \bar{e}_{m_1}^T & 0 & 0 & \cdots & 0 \\ 0 & \bar{e}_{m_2}^T & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \bar{e}_{m_p}^T \end{bmatrix} \in \mathbb{R}^{p \times n_0}. \tag{5.1.1}$$

Here, $\bar{e}_{m_i}$ is the vector of all ones of the length m_i. Let x be a vector in $\{0,1\}^{n_0}$. We call the equality

$$A_u x = \bar{e}_p$$

the *unit row-sum constraint*. Any constraint matrix that has exactly one '1' in each column can be represented using the matrix A_u, up to permutation. Many applications of combinatorial optimization problems contain the unit row-sum constraint. For instance, given a collection of p sets with

m_i members in each set, choosing *exactly one* representative from each set can be formulated using the unit row-sum constraint. The protein side-chain positioning problem [27] directly fits into this setting. The quadratic assignment problem [166] and the minimum-cut problem [108] contain the unit row-sum constraints with the addition of special linear constraints.

Let $Q \in \mathbb{S}^{n_0}$ be given. We define the binary quadratic problem (**BQP**) with the unit row-sum constraint as

$$(\mathbf{BQP}) \qquad \begin{array}{rccl} p^*_{\mathbf{BQP}} & = & \min\limits_{x} & x^T Q x \\ & & \text{subject to} & A_u x = e \\ & & & x \in \{0,1\}^{n_0}. \end{array} \tag{5.1.2}$$

The problem (5.1.2) is NP-hard in general and hence many non-polynomial time algorithms are proposed, e.g., heuristic based method and branch-and-bound. We approach the problem via **SDP** and **DNN** relaxations. The **SDP** relaxation is one of the most well-known approaches that avoids the difficulties arising from the NP-hardness of the model (5.1.2).

We obtain the **SDP** relaxation in two steps. The first step employs the *lifting* of the variable $x \in \mathbb{R}^{n_0}$ to a higher dimensional space, e.g., [150]. We lift a vector in $\mathbb{R}^{n_0}$ to a matrix in $\mathbb{S}^{n_0+1}_+$ as follows:

$$Y_x = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} \in \mathbb{S}^{n_0+1}_+. \tag{5.1.3}$$

For the **SDP** relaxation, we index the rows and columns *starting from* 0, i.e., the row and column indices are $\{0,1,2,\ldots,n_0\}$. For a fixed x, the matrix Y_x has the rank one since every column of Y_x is a scalar multiple of the column vector $[1;x]$. Let $\hat{Q} = \text{BlkDiag}(0,Q)$. The objective function has a simple formulation

$$\langle x, Qx \rangle = \left\langle \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}, \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \right\rangle = \left\langle \hat{Q}, Y_x \right\rangle.$$

We now see how the lifting acts on the *unit row-sum* constraint. We begin by following the series of the implications below:

$$\begin{array}{lll} & A_u x = e \in \mathbb{R}^p & \\ \implies & A_u^T A_u x = A_u^T e = e \in \mathbb{R}^{n_0} & \\ \implies & (A_u^T A_u - I)x = e - x & \\ \implies & (A_u^T A_u - I)xx^T = ex^T - xx^T & (\text{multiplication by } x^T) \\ \implies & \text{trace}(A_u^T A_u - I)xx^T = \text{trace}(ex^T - xx^T) = 0 & (\text{since } x \in \{0,1\}^{n_0}). \end{array} \tag{5.1.4}$$

We note that $A_u^T A_u - I$ and xx^T are nonnegative matrices. From (5.1.4), we see that the elements of xx^T that correspond to $\text{supp}(A_u^T A_u - I)$ must be to 0.

We now employ a mapping the so-called *gangster operator* to complete the constraint transformation. Given a set of matrix indices $\mathcal{J} \subseteq \{0,\ldots,n_0\} \times \{0,\ldots,n_0\}$, we define the gangster operator

$$G_{\mathcal{J}} : \mathbb{S}^{n_0+1} \to \mathbb{R}^{|\mathcal{J}|} \text{ by } G_{\mathcal{J}}(Y) = Y_{\mathcal{J}}, \tag{5.1.5}$$

where $Y_{\mathcal{J}}$ is the vectorization of the submatrix of Y that chooses the elements in the index set $\mathcal{J}$.

By abuse of notation,[1] we also consider the gangster operator from $\mathbb{S}^{n_0+1}$ to $\mathbb{S}^{n_0+1}$ to mean

$$G_{\mathcal{J}} : \mathbb{S}^{n_0+1} \to \mathbb{S}^{n_0+1}, \quad (G_{\mathcal{J}})_{i,j} = \begin{cases} Y_{i,j} & \text{if } (i,j) \text{ or } (j,i) \in \mathcal{J}, \\ 0 & \text{otherwise.} \end{cases}$$

Taking (5.1.4) into account, we set $\mathcal{J} \subseteq \{0,\ldots,n_0\} \times \{0,\ldots,n_0\}$ to be

$$\mathcal{J} = \{(0,0)\} \cup \operatorname{supp}(\operatorname{BlkDiag}(0, A_u^T A_u - I)). \tag{5.1.6}$$

Then we obtain the *gangster constraint*

$$\left[(A_u^T A_u - I) \circ xx^T = 0 \text{ and } Y_x(0,0) = 1\right] \implies G_{\mathcal{J}}(Y_x) = e_0 e_0^T =: E_{00}. \tag{5.1.7}$$

The term *gangster* refers to the action of the constraint; $G_{\mathcal{J}}(Y)$ sets many elements of Y associated with $\mathcal{J}$ to be zero (shoots holes in the matrix).

The second step of deriving the **SDP** relaxation is simple. We recall that the rank-one restriction on the lifted variable Y_x. We simply discard that rank constraint on Y_x and work with general positive semidefinite matrices. Finally, we obtain the **SDP** relaxation of the model (5.1.2):

$$\begin{array}{rccl} p^*_{\mathbf{SDP}} & = & \min\limits_{Y} & \langle \hat{Q}, Y \rangle \\ & & \text{subject to} & G_{\mathcal{J}}(Y) = E_{00} \\ & & & Y \succeq 0. \end{array} \tag{5.1.8}$$

We emphasize that the model (5.1.8) accompanies interesting advantages over the model (5.1.2). By relaxing the rank-one constraint on the variable, we now obtain a convex feasible region. Another important property of the transformation is that the objective function $\langle \hat{Q}, Y \rangle$ is now linear in Y. The data matrix Q in (5.1.2) is often indefinite. Hence, even though the feasible region of (5.1.2) was convex, solving (5.1.2) is NP-hard, see e.g., [126]. However, since (5.1.8) is a relaxation of (5.1.2), it is possible to encounter the discrepancy $p^*_{\mathbf{BQP}} > p^*_{\mathbf{SDP}}$. We call this quantity $p^*_{\mathbf{BQP}} - p^*_{\mathbf{SDP}}$, a *relaxation gap*.

The **SDP** relaxation (5.1.8) does not possess a strictly feasible point. One way to see this is as follows:

$$A_u x = e \iff \begin{bmatrix} -e & A_u \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = 0 \implies \begin{bmatrix} -e & A_u \end{bmatrix}^T \begin{bmatrix} -e & A_u \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = 0. \tag{5.1.9}$$

Since

$$K := \begin{bmatrix} -e & A_u \end{bmatrix}^T \begin{bmatrix} -e & A_u \end{bmatrix} \in \mathbb{S}^{n_0+1}_+ \setminus \{0\}, \tag{5.1.10}$$

the matrix K serves as an exposing vector for the feasible set of (5.1.8)[2]. Hence, (5.1.9) proves that strict feasibility always fails for (5.1.8).

In order to obtain a model with a strictly feasible point, we find the minimal face that contains the feasible set. Because the **SDP** relaxations originate from specific applications in general, we usually obtain exposing vectors analytically by exploiting the problem structures rather than solving

[1] This may result in the loss of surjectivity of the mapping $G_{\mathcal{J}}$. However, we later derive robust iterate update rules and this instability does not come into play in our algorithm.

[2] This property holds for arbitrary data matrix A and right-hand-side vector b. Alternatively, we cannot find n_0 linearly independent feasible points in the ground set, see [150].

the auxiliary system (2.3.4) numerically. The embedded structure removes the need for computing the facial range vector repeatedly.

Once we identify the minimal face of $\mathbb{S}_+^{n_0+1}$ that contains the feasible set, we may replace the variable Y with VRV^T, where $V \in \mathbb{R}^{(n_0+1)\times r}$ is a minimal facial range vector with *orthonormal columns*:

$$p^*_{\textbf{SDP}} = \min_R \{\langle \hat{Q}, VRV^T\rangle : G_{\mathcal{J}}(VRV^T) = E_{00},\ R \in \mathbb{S}^r_+\}. \tag{5.1.11}$$

As pointed out earlier, simultaneously engaging the two constraints, $G_{\mathcal{J}}(VRV^T) = E_{00}$ and $R \succeq 0$, is complicated. Assigning $Y = VRV^T$ to (5.1.11) helps avoid this complication:

$$p^*_{\textbf{SDP}} = \min_{R,Y}\{\langle \hat{Q}, Y\rangle : G_{\mathcal{J}}(Y) = E_{00},\ Y = VRV^T,\ R \in \mathbb{S}^r_+\}. \tag{5.1.12}$$

Many applications of this type can be found in the literature [27, 28, 48, 79, 108, 123, 166].

5.1.1 Embedded Properties of SDP Relaxations of Binary Quadratic Programs

In this section we present common structures that are embedded within the **SDP** relaxation of (**BQP**). We take advantage of these structures and apply those properties to the two classes of **SDP** relaxations of binary quadratic programs, **SCP** and **QAP** in Sections 5.3 and 5.4.

We first partition Y below in order to capture the special structures that arise in the **SDP** relaxations of (**BQP**):

$$Y = \begin{bmatrix} 1 & Y_{10}^T & Y_{20}^T & \cdots & Y_{p0}^T \\ Y_{10} & Y_{11} & Y_{12} & \cdots & Y_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{p0} & Y_{p1} & Y_{p2} & \cdots & Y_{pp} \end{bmatrix} \in \mathbb{S}^{n_0+1}, \tag{5.1.13}$$

where

$$Y_{ii} \in \mathbb{S}^{m_i},\ Y_{ij} \in \mathbb{R}^{m_i \times m_j},\ Y_{i0} \in \mathbb{R}^{m_i},\ \forall i, j \in \{1, \ldots, p\}.$$

With the partition (5.1.13) of Y, we obtain Theorem 5.1.1 below.

Theorem 5.1.1. *Let V be a facial range vector that satisfies* $\operatorname{range}(V) \subseteq \operatorname{null}(K)$ *with K given in* (5.1.10)*. Suppose that Y and R satisfy*

$$Y \in \mathbb{S}^{n_0+1},\ R \in \mathbb{S}_+^{n_0+1-p} \text{ with } Y = VRV^T,\ G_{\mathcal{J}}(Y) = E_{00}.$$

Then the following hold.

1. *The first column of Y is equal to the diagonal of Y;*
2. $\operatorname{trace}(Y_{i,i}) = 1$*, for all $i \in \{1, \ldots, p\}$.*

Proof. We let $\bar{e}_t$ denote the all ones vector of length t. We recall that $\operatorname{range}(V) \subseteq \operatorname{null}(K) = \operatorname{null}\left(\begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix}\right)$. Hence we have

$$\begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix} Y = \begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix} VRV^T = 0RV^T = 0. \tag{5.1.14}$$

We then exploit the structure of $\begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix} Y$.

We use $Y_{ij}^{\mathbf{col}\,\ell}$ to denote the ℓ-th column of the (i,j)-th block of Y and $Y_{i0,\ell}$ to denote the ℓ-th coordinate of the vector $Y_{i0} \in \mathbb{R}^{m_i}$. Then expanding $\begin{bmatrix}-\bar{e}_p & A_u\end{bmatrix} Y$ with the block representation (5.1.13) yields

$$\begin{bmatrix}-\bar{e}_p & A_u\end{bmatrix} Y = \begin{bmatrix}a_0 & A_1 & \cdots & A_p\end{bmatrix} \in \mathbb{R}^{p\times(n_0+1)},$$

where

$$a_0 = \begin{bmatrix}-1+\bar{e}_{m_1}^T Y_{10}\\ -1+\bar{e}_{m_2}^T Y_{20}\\ \vdots \\ -1+\bar{e}_{m_p}^T Y_{p0}\end{bmatrix} \in \mathbb{R}^p, \tag{5.1.15}$$

and, for each $i \in \{1,\dots,p\}$,

$$A_i = \begin{bmatrix} -Y_{i0,1}+\bar{e}_{m_1}^T Y_{1i}^{\mathbf{col}\,1} & -Y_{i0,2}+\bar{e}_{m_1}^T Y_{1i}^{\mathbf{col}\,2} & \cdots & -Y_{i0,m_i}+\bar{e}_{m_1}^T Y_{1i}^{\mathbf{col}\,m_i}\\ \vdots & \vdots & \ddots & \vdots\\ -Y_{i0,1}+\bar{e}_{m_j}^T Y_{ji}^{\mathbf{col}\,1} & -Y_{i0,2}+\bar{e}_{m_j}^T Y_{ji}^{\mathbf{col}\,2} & \cdots & -Y_{i0,m_i}+\bar{e}_{m_j}^T Y_{ji}^{\mathbf{col}\,m_i}\\ \vdots & \vdots & \ddots & \vdots\\ -Y_{i0,1}+\bar{e}_{m_p}^T Y_{pi}^{\mathbf{col}\,1} & -Y_{i0,2}+\bar{e}_{m_p}^T Y_{pi}^{\mathbf{col}\,2} & \cdots & -Y_{i0,m_i}+\bar{e}_{m_p}^T Y_{pi}^{\mathbf{col}\,m_i} \end{bmatrix} \in \mathbb{R}^{p\times m_i}.$$

By (5.1.14), we have $A_i = 0, \ \forall i \in \{1,\dots,p\}$. Thus, for each $i \in \{1,\dots,p\}$, the i-th row A_i yields

$$Y_{i0,\ell} = \bar{e}_{m_i}^T Y_{ii}^{\mathbf{col}\,\ell}, \quad \ell \in \{1,\dots,m_i\}.$$

Since $G_{\mathcal{J}}(Y) = E_{00}$ holds, we obtain

$$\operatorname{diag}(Y_{ii}) = Y_{i0}, \ \forall i \in \{1,\dots,p\}.$$

Therefore, we conclude that the first column and the diagonal of Y are identical and it completes the proof for Item 1.

By (5.1.14), we have the vector a_0 from (5.1.15) is 0. Thus, $1 = \bar{e}_{m_i}^T Y_{i0}$, for all $i = 1,\dots,p$. By Item 1, we obtain

$$1 = \bar{e}_{m_i}^T Y_{i0} = \operatorname{trace}(Y_{i,i}), \ \forall i \in \{1,\dots,p\}.$$

Hence, it completes the proof for Item 2. □

We often call the indices for the first column, the first row and the diagonal elements of Y the *arrow indices* since the positioning of these elements resembles an arrow shape; see Figure 5.1.1. The same properties are known to hold from the **SDP** relaxations of the **QAP** [166] and the **SCP** problem [27] via the use of the Lagrangian dual. We rather obtain these properties through the

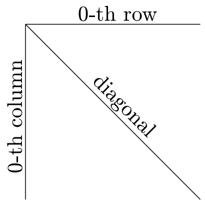

Figure 5.1.1: Arrow indices; the first (0-th) row and column and the diagonal elements

direct lifting. We note that the properties in Theorem 5.1.1 correspond to the implicit redundancies revealed by **FR**. Item 2 of Theorem 5.1.1 gives rise to an additional property on the variable R of (5.1.12).

Corollary 5.1.2. *Let (R, Y) be a solution to (5.1.12). Then we have $\operatorname{tr}(R) = 1 + p$.*

Proof. By Item 2 of Theorem 5.1.1, each diagonal block of Y satisfies $\operatorname{trace}(Y_{i,i}) = 1$. Hence, $Y = VRV^T$ yields

$$1 + p = \operatorname{tr}(Y) = \operatorname{tr}(VRV^T) = \operatorname{tr}(RV^TV) = \operatorname{tr}(R),$$

where the last equality holds since $V^TV = I$. □

5.1.2 Doubly-Nonnegative Relaxation

In this section we introduce the doubly-nonnegative (**DNN**) relaxation of the model (5.1.2) and its related properties.

A cone $\mathcal{K} \subseteq \mathbb{S}^n$ is called a *doubly-nonnegative cone* if

$$\mathcal{K} = \mathbb{S}^n_+ \cap \{X \in \mathbb{S}^n : X_{i,j} \geq 0, \ \forall i, j\}.$$

We note that variables in the ground set of (**BQP**) are in $\{0,1\}^{n_0}$. Hence, with Y_x from the lifting process (5.1.3), we note that

$$Y_x \in \{Y \in \mathbb{S}^{n_0+1} : 0 \leq Y_{i,j} \leq 1, \ \forall i, j\}.$$

That is, we may restrict the lifted variable to the doubly-nonnegative cone, $\mathbb{S}^{n_0+1}_+ \cap \mathbb{R}^{(n_0+1)\times(n_0+1)}_+$. This gives rise to the *doubly-nonnegative relaxation (**DNN**)*:

$$\textbf{(DNN)} \qquad \begin{array}{rcll} p^*_{\textbf{DNN}} & = & \min_{R,Y} & \langle \hat{Q}, Y \rangle \\ & & \text{subject to} & G_{\mathcal{J}}(Y) = E_{00} \\ & & & Y = VRV^T \\ & & & R \succeq 0 \\ & & & 0 \leq Y \leq 1. \end{array} \qquad (5.1.16)$$

Clearly the doubly-nonnegative cone, $\mathbb{S}^n_+ \cap \mathbb{R}^{n\times n}_+$, is a proper subset of the positive semidefinite cone, $\mathbb{S}^n_+$. Hence, the **DNN** relaxation is a tighter relaxation of (**BQP**) than the **SDP** relaxation. As a consequence, we obtain

$$p^*_{\textbf{BQP}} \geq p^*_{\textbf{DNN}} \geq p^*_{\textbf{SDP}} \cdot$$

We show in our numerical experiment that the inequality $p^*_{\textbf{DNN}} \geq p^*_{\textbf{SDP}}$ can be strict. In fact, the **DNN** relaxation often provides a significantly strengthened lower bound.

We note that the model (5.1.16) is not entirely stable. We can immediately observe that the gangster mapping $G_{\mathcal{J}}$ is not surjective as noted above; the equality $Y = VRV^T$ can be eliminated; and the lower and upper bounds on the elements of Y associated with the gangster index set $\mathcal{J}$ are redundant. However, we develop a splitting method below with robust updates that avoid these instabilities. In fact, there is an additional set of redundant constraints to the model (5.1.16) as we see in Proposition 5.1.3 below.

Proposition 5.1.3. *The polyhedral constraints on the arrow indices are redundant. In other words, the inequalities*

$$Y_{i,j} \in [0,1], \ \forall i,j \text{ such that } i \cdot j = 0 \text{ or } i = j,$$

are redundant constraints to (5.1.16).

Proof. Since Y is positive semidefinite, all the diagonal elements of Y must be at least 0. In addition, we recall from Item 2 of Theorem 5.1.1 that $\operatorname{trace}(Y_{ii}) = 1$. Thus, each diagonal element of Y is at most 1. By Item 1 of Theorem 5.1.1, the diagonal of Y is equal to the first row and first column of Y. Hence, it follows that the polyhedral constraint on the first row and first column are also redundant. □

Proposition 5.1.3 suggests that we may discard some of the polyhedral constraints, e.g., the ones dealing with the arrow indices. This observation leads to Theorem 5.1.4 that shows that some elements of the optimal dual solution are known.

Theorem 5.1.4. *Let* (R^*, Y^*) *be an optimal solution pair for* (5.1.16) *and let*

$$\mathcal{Z}_A := \left\{ Z \in \mathbb{S}^{n_0+1} : Z_{i,i} = -\hat{Q}_{i,i}, \ Z_{0,i} = Z_{0,i} = -\hat{Q}_{0,i}, \ i = 1, \ldots, n_0 \right\}.$$

Then there exists a dual multiplier Z^* *associated with the constraint* $Y = VRV^T$ *such that* $Z^* \in \mathcal{Z}_A$ *and* (R^*, Y^*, Z^*) *solves* (5.1.16).

Proof. We define

$$\mathcal{Y}_A = \{ Y : G_{\mathcal{J}}(Y) = E_{00}, \ Y_{i,j} \in [0,1], \ \forall i,j \text{ such that } i \cdot j \neq 0, i \neq j \}.$$

That is, $\mathcal{Y}_A \supseteq \mathcal{Y}$ is obtained after removing the redundant polyhedral constraints associated with the arrow indices of Y; see Proposition 5.1.3. Then (5.1.16) is equivalent to

$$\begin{array}{rl} \min_{R,Y} & \langle \hat{Q}, Y \rangle \\ \text{subject to} & Y = VRV^T \\ & R \in \mathbb{S}^r_+ \\ & Y \in \mathcal{Y}_A. \end{array} \tag{5.1.17}$$

Let (R^*, Y^*, Z^*) be an optimal primal-dual solution for (5.1.17), where Z^* is the dual multiplier associated with the equality $Y = VRV^T$. Then, by the first-order optimality conditions, we have

$$0 \in -V^T Z^* V + \mathcal{N}_{\mathbb{S}^r_+}(R^*), \tag{5.1.18a}$$

$$0 \in \hat{Q} + Z^* + \mathcal{N}_{\mathcal{Y}_A}(Y^*), \tag{5.1.18b}$$

$$Y^* = VR^*V^T, \quad R^* \in \mathbb{S}^r_+, \quad Y^* \in \mathcal{Y}_A. \tag{5.1.18c}$$

By the definition of the normal cone, we have

$$0 \in \hat{Q} + Z^* + \mathcal{N}_{\mathcal{Y}_A}(Y^*) \iff \langle Y - Y^*, \hat{Q} + Z^* \rangle \geq 0, \ \forall Y \in \mathcal{Y}_A.$$

Since the diagonal and the first column and row of $Y \in \mathcal{Y}_A$ except for the first element are unconstrained, as are all the redundant gangster positions, we see that

$$(\hat{Q} + Z^*)_{i,j} = 0, \forall i, j \text{ such that } i \cdot j = 0 \text{ or } i = j.$$

This implies that $Z^* \in \mathcal{Z}_A$ and proves the statement. □

Theorem 5.1.4 raises the following question; how do we utilize the prior knowledge of the elements in the dual optimal solution? Theorem 5.1.4 motivates the development of a variant of the splitting method. We introduce a splitting method that takes advantage of this information in Section 5.2.3 below. We comment that there is an effective method for handling the **DNN** relaxation by majorizing the augmented Lagrangian of the dual problem [161]. However, engaging the known dual elements for this method is not yet available.

We now complete the model by including the two redundant constraints from Corollary 5.1.2 and Proposition 5.1.3. We define the two sets

$$\begin{array}{rcl} \mathcal{R} & = & \{R \in \mathbb{S}^r \ : \ R \in \mathbb{S}^r_+, \ \operatorname{trace}(R) = 1 + p\}, \\ \mathcal{Y} & = & \{Y \in \mathbb{S}^{n_0+1} \ : \ G_{\mathcal{J}}(Y) = E_{00}, \ 0 \le Y \le 1\}. \end{array} \tag{5.1.19}$$

We then obtain the **DNN** relaxation

$$\begin{array}{rccl} p^*_{\mathbf{DNN}} & = & \min\limits_{R,Y} & \langle \hat{Q}, Y \rangle \\ & & \text{subject to} & R \in \mathcal{R} \\ & & & Y \in \mathcal{Y} \\ & & & Y = VRV^T. \end{array} \tag{5.1.20}$$

Solving the model (5.1.20) using the splitting method is convenient in three ways. First, the variables R and Y are linearly related by $Y = VRV^T$. This linear relation connects the two different sets $\mathcal{R}$ and $\mathcal{Y}$ and allows us to focus on the two sets individually. In addition, we will see later that this linear relation allows us to obtain convenient projection formulae for solving the subproblems of the splitting methods.

Secondly, splitting the variables allows us to handle the constraints that are difficult to handle together. The model (5.1.20) can be solved by using a standard solver by incorporating the element-wise nonnegativity on Y through cutting planes [27]. However, this approach becomes more computationally challenging as the number of cutting planes increases. By considering the sets $\mathcal{R}$ and $\mathcal{Y}$ individually, we can avoid the complications that arise from considering the intersection of the two sets; the polyhedral constraint on Y and the cone constraint on R are very expensive to engage at the same time.

Lastly, we can include redundant constraints safely. A model that possesses redundant linear constraints brings out instability issues as we observed throughout Part I. For instance, when an interior point method is used, having redundant constraints in the model damages the model quality with regard to stability. The observed redundancies to the model (5.1.20) are a result of considering the variables R and Y together. However, these constraints are not redundant with respect to the set $\mathcal{R}$ and $\mathcal{Y}$ individually in our subproblems. From Proposition 5.1.3, we observed that the polyhedral constraint on the arrow positions of Y are redundant. However, when considering Y as a member of the set $\mathcal{Y}$, these are not redundant constraints. From Corollary 5.1.2, we observed that $\operatorname{trace}(R) = 1 + p$ is a redundant constraint. However, R being positive semidefinite does not by itself imply that the trace of R is equal to $1 + p$.

We conclude this section with the first-order optimality conditions of the **DNN** relaxation (5.1.20):

$$\begin{aligned} 0 &\in -V^T Z^* V + \mathcal{N}_{\mathcal{R}}(R^*), && (5.1.21a)\\ 0 &\in \hat{Q} + Z^* + \mathcal{N}_{\mathcal{Y}}(Y^*), && (5.1.21b)\\ Y^* &= V R^* V^T, \quad R^* \in \mathbb{S}^r_+,\ Y^* \in \mathcal{Y}. && (5.1.21c)\end{aligned}$$

By the definition of normal cone, we obtain the following equivalent optimality condition that we can evaluate with ease. The proof of Proposition 5.1.5 follows from [90, Proposition 5.3.3].

Proposition 5.1.5 (characterization of optimality for (5.1.20))**.** *The primal-dual triple* (R, Y, Z) *is optimal for* (5.1.20) *if, and only if,* (5.1.21) *holds if, and only if,*

$$R = \mathcal{P}_{\mathcal{R}}(R + V^T Z V), \quad Y = \mathcal{P}_{\mathcal{Y}}(Y - \hat{Q} - Z), \quad Y = V R V^T. \tag{5.1.22}$$

5.2 A Derivation of the Restricted Dual PRSM

In this section we display a detailed derivation of the Peaceman-Rachford splitting method (**PRSM**) via Lagrangian dual. In particular, we start the derivation from the Peaceman-Rachford scheme from the monotone operator theory. By examining the derivation closely, we provide a variant of the **PRSM** that utilizes the known elements of the dual multipliers. [72, Chapter 6] points out that the Peaceman-Rachford splitting method is often faster than the Douglas-Rachford splitting method when it converges.

Let $f : \mathbb{R}^{n_A} \to \mathbb{R}$ and $g : \mathbb{R}^{n_B} \to \mathbb{R}$ be convex functions, and let $A \in \mathbb{R}^{m \times n_A}$, $B \in \mathbb{R}^{m \times n_B}$ and $b \in \mathbb{R}^m$ be the given data. Let $\mathcal{X} \subseteq \mathbb{R}^{n_A}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_B}$ be closed convex sets. We focus on the problem

$$\min_{x,y} f(x) + g(y) \ \text{ subject to } \ Ax + By = b,\ x \in \mathcal{X},\ y \in \mathcal{Y}. \tag{5.2.1}$$

The two variables x and y are constrained in $\mathcal{X}$ and $\mathcal{Y}$, respectively, and they are tied by the linear equality. The splitting method provides an effective means of solving the model (5.2.1). For example, as we observed in Section 5.1, the cone constraint and the polyhedral constraints in the **DNN** relaxation (5.1.20) are difficult to engage simultaneously. Algorithm 5.2.1 is known as the Peaceman-Rachford splitting method (**PRSM**) applied to (5.2.1). Algorithm 5.2.1 can be summarized as follows: alternate minimization of variables x and y interlaced by the dual variable z update.

Algorithm 5.2.1 does not necessarily guarantee the convergence to the optimal solution. The convergence guarantee is available under a more restrictive setting such as the uniform monotonicity of one of the underlying monotone operators or uniform convexity of one of functions f or g; see [11]. A recent work [85] shows that if we use an under-relaxation parameter $\gamma \in (0, 1)$ in the dual update, the iterates converge to an optimal solution strictly. Algorithm 5.2.2 below is the strictly contractive Peaceman-Rachford splitting method for (5.2.1).

Algorithm 5.2.1 Peaceman-Rachford Splitting Method for (5.2.1)

```
  Require: Initial Iterates $y^0, z^0,\ \beta \in \mathbb{R}_{++}$
  while stopping criteria are not satisfied do
    $x^{k+1} = \min_{x\in\mathcal{X}} \mathcal{L}_A(x, y^k, z^k)$    ($x$-subproblem)
    $z^{k+\frac{1}{2}} = z^k + \beta\left(Ax^{k+1} + By^k - b\right)$    (intermediate dual update).
    $y^{k+1} = \min_{y\in\mathcal{Y}} \mathcal{L}_A(x^{k+1}, y, z^{k+\frac{1}{2}})$    ($y$-subproblem)
    $z^{k+1} = z^{k+\frac{1}{2}} + \beta\left(Ax^{k+1} + By^{k+1} - b\right)$    (dual update ).
    $k \leftarrow k+1$
  end while
```

Algorithm 5.2.2 Strictly Contractive Peaceman-Rachford Splitting Method for (5.2.1) [85]

```
  Require: Initial Iterates $y^0, z^0,\ \beta \in \mathbb{R}_{++},\ \gamma \in (0,1)$
  while stopping criteria are not satisfied do
    $x^{k+1} = \min_{x\in\mathcal{X}} \mathcal{L}_A(x, y^k, z^k)$    ($x$-subproblem)
    $z^{k+\frac{1}{2}} = z^k + \gamma\beta\left(Ax^{k+1} + By^k - b\right)$    (intermediate dual update).
    $y^{k+1} = \min_{y\in\mathcal{Y}} \mathcal{L}_A(x^{k+1}, y, z^{k+\frac{1}{2}})$    ($y$-subproblem)
    $z^{k+1} = z^{k+\frac{1}{2}} + \gamma\beta\left(Ax^{k+1} + By^{k+1} - b\right)$    (dual update).
    $k \leftarrow k+1$
  end while
```

5.2.1 Preliminaries on Monotone Operator Theory

In this section we present basic definitions related to the monotone operator theory that we use for our derivations. We use Id to denote the identity operator. We work in finite dimensional Hilbert spaces.

Definition 5.2.1. *Let X be a Hilbert space. Let $\mathscr{A}$ be a set-valued mapping from X to 2^X, where 2^X is a power set of X, i.e., the collection of subsets of X.*

1. *A graph of $\mathscr{A}$ is a set of points*

$$\operatorname{gra}\mathscr{A} := \{(x, x^*) \in X \times X : x^* \in \mathscr{A}x\}.$$

2. *A set-valued operator $\mathscr{A}$ is called monotone, if*

$$\langle x - y, x^* - y^*\rangle \geq 0,\ \forall (x, x^*), (y, y^*) \in \operatorname{gra}\mathscr{A}.$$

3. *Given a monotone operator $\mathscr{A} : X \rightrightarrows X$, the resolvent operator is defined by*

$$J_{\mathscr{A}} := (\operatorname{Id} + \mathscr{A})^{-1}.$$

4. *The* reflection *operator of the monotone operator $\mathscr{A}$ is defined by*

$$\operatorname{refl}_{\mathscr{A}} := 2J_{\mathscr{A}} - \operatorname{Id}.$$

5. *The set of fixed points of the monotone operator $\mathscr{A}$ is defined by*

$$\operatorname{Fix}(\mathscr{A}) = \{x \in X : \mathscr{A}x = x\}.$$

We consider the following inclusion problem; finding a zero of the sum of two monotone operators.

Problem 5.2.2. *(Monotone inclusion problem) Let X be a Hilbert space and let $\mathscr{A}, \mathscr{B} : X \rightrightarrows X$ be monotone operators. The monotone inclusion problem is to*

$$\textit{find } x \in X \textit{ such that } 0 \in \mathscr{A}x + \mathscr{B}x.$$

The Peaceman-Rachford (**PR**) method [132] tries to find a fixed point of $\operatorname{refl}_{\mathscr{B}} \operatorname{refl}_{\mathscr{A}}$, the composition of two reflection operators, via the iterate update regulated by

$$\boxed{w^{k+1} = \operatorname{refl}_{\mathscr{B}} \circ \operatorname{refl}_{\mathscr{A}}(w^k) \quad \text{for } k \in \mathbb{N}.} \tag{5.2.2}$$

It is important to note that the solution set of Problem 5.2.2 has the following characterization (see [11, Proposition 26.1].):

$$\{x \in X : 0 \in \mathscr{A}x + \mathscr{B}x\} = J_{\mathscr{A}} \left(\operatorname{Fix}(\operatorname{refl}_{\mathscr{B}} \operatorname{refl}_{\mathscr{A}})\right). \tag{5.2.3}$$

We emphasize that finding a fixed point of $\operatorname{refl}_{\mathscr{B}} \circ \operatorname{refl}_{\mathscr{A}}$ does *not* yield a solution to Problem 5.2.2; the solution to Problem 5.2.2 is found *by mapping* a point in $\operatorname{Fix}(\operatorname{refl}_{\mathscr{B}} \operatorname{refl}_{\mathscr{A}})$ by the resolvent $J_{\mathscr{A}}$.

Problem 5.2.2 has a beautiful application to the problem of finding a minimizer of $f(x)+g(x)$, the sum of two convex functions. We replace the operators in Problem 5.2.2 by

$$\mathscr{A} \leftarrow \partial f \quad \text{and} \quad \mathscr{B} \leftarrow \partial g,$$

Then, we can use Fermat's rule (see [11, Theorem 16.2].) to find a minimizer of $f(x)+g(x)$:

$$\text{find } x \in X \text{ such that } 0 \in \partial f(x) + \partial g(x). \tag{5.2.4}$$

We assume that a solution that solves (5.2.4) exists in the subsequent sections. Minimizing the sum of two convex functions appears in numerous applications, e.g., see [11, 24, 72] and the references therein.

Finally, we arrive at an alternative interpretation of the **PR** method for (5.2.4):

$$w^{k+1} = \operatorname{refl}_{\partial g} \circ \operatorname{refl}_{\partial f}(w^k) = (2J_{\partial g} - \operatorname{Id}) \circ (2J_{\partial f} - \operatorname{Id})(w^k). \tag{5.2.5}$$

Once the iteration sequence (5.2.5) finds a *fixed point* of the operator $\operatorname{refl}_{\partial g} \operatorname{refl}_{\partial f}$, the solution of the problem (5.2.4) is found by evaluating $J_{\partial f}(w^{k+1})$; see (5.2.3). Throughout Sections 5.2.2 and 5.2.3 below, we show how the iteration sequence (5.2.5) gives rise to the use of prior knowledge on some elements of the dual optimal solutions.

5.2.2 PR Algorithm Applied to Dual Problem

In this section we see how the **PR** algorithm gives rise to the splitting method in Algorithm 5.2.1. A reader who is familiar with the derivation may skip this section and go to Section 5.2.3. That the **PRSM** can be derived using the dual formulation is known in the literature; the method can be viewed as the **PR** algorithm by using the Fenchel conjugate, e.g., see [20, 53, 65, 113]. We aim to include a detailed self-contained derivation tailored to our setting via Lagrangian duality. We closely follow some steps displayed in [54, 65]. This derivation directly leads to the clear interpretation of the restricted dual **PRSM**.

Suppose that we want to solve the problem of the form (5.2.1). By relating the linear constraint with Lagrange multiplier z, we get the Lagrangian of (5.2.1) for $x \in \mathcal{X}, y \in \mathcal{Y}$ as follows:

$$\mathcal{L}(x, y; z) = f(x) + g(y) + \langle z, Ax + By - b\rangle.$$

We split the Lagragian above into

$$\mathcal{L}(x, y; z) = \mathcal{L}_f(x; z) + \mathcal{L}_g(y; z), \text{ where } \mathcal{L}_f(x; z) := f(x) + \langle z, Ax\rangle, \; \mathcal{L}_g(y; z) := g(y) + \langle z, By - b\rangle.$$

Then, the Lagrangian dual problem of (5.2.1) is

$$\max_z \left\{ \min_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathcal{L}(x, y; z) \right\} = \max_z \left\{ \min_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathcal{L}_f(x; z) + \mathcal{L}_g(y; z) \right\}. \tag{5.2.6}$$

We emphasize that the dual functional is separable into the two terms associated with x and y, respectively. This splitting initiates the two minimization problems that appear in Algorithm 5.2.1. We define

$$\hat{d}_f(z) := \min_{x\in\mathcal{X}} \mathcal{L}_f(x; z) \quad \text{and} \quad \hat{d}_g(z) := \min_{y\in\mathcal{Y}} \mathcal{L}_g(y; z).$$

We assume that the minimizers of $\hat{d}_f$ and $\hat{d}_g$ are attained; this is true if $\mathcal{X}$ and $\mathcal{Y}$ are compact. The functions $\hat{d}_f$ and $\hat{d}_g$ are concave functions since they are point-wise minimum of concave functions with respect to z. We let

$$d_f(z) = -\hat{d}_f(z) \quad \text{and} \quad d_g(z) = -\hat{d}_g(z).$$

The dual problem (5.2.6) is equivalent to minimizing the sum of two convex functions:

$$\min_z d_f(z) + d_g(z). \tag{5.2.7}$$

Let $\beta > 0$. The dual problem (5.2.7) is also equivalent to $\min_z \beta d_f(z) + \beta d_g(z)$, i.e., the minimizer is invariant under the positive scalar multiple β. And, the multiplier gives rise to the parameter β that appear in the augmented Lagarangian $\mathcal{L}_A$ in Algorithm 5.2.1. For simplicity, we let $\beta = 1$ in the derivation.

The subdifferentials are monotone operators [90, Proposition 6.1.1]. The subdifferential of a proper, convex, lower-semi-continuous function is maximal [138]. Hence we apply the **PR** algorithm to the problem

$$\text{find } z \in \mathbb{R}^m \text{ such that } 0 \in \partial d_f(z) + \partial d_g(z)$$

and employ the Peaceman-Rachford update rule (5.2.2)

$$w^{k+1} = \text{refl}_{\partial d_g} \circ \text{refl}_{\partial d_f}(w^k). \tag{5.2.8}$$

We break the update (5.2.8) into two steps:

$$w^{k+\frac{1}{2}} = \text{refl}_{\partial d_f}(w^k), \quad w^{k+1} = \text{refl}_{\partial d_g}(w^{k+\frac{1}{2}}).$$

The sequence $\{w^k\}$ is the governing sequence of the problem (5.2.6). The *uncoupling* of the two steps gives rise to two auxiliary problems and these appear as the two subproblems in Algorithm 5.2.1.

We proceed with the derivation by expanding the terms that appear in (5.2.8). We let $z^k = J_{\partial d_f}(w^k)$. Then, by the definition of the reflection operator, we have

$$w^{k+\frac{1}{2}} = \mathrm{refl}_{\partial d_f}(w^k) = (2J_{\partial d_f} - \mathrm{Id})w^k = 2z^k - w^k = z^k + (z^k - w^k). \tag{5.2.9}$$

Now, by the definition of the resolvent operator, we obtain the following relation:

$$\begin{aligned} z^k = J_{\partial d_f}(w^k) \quad &\Longleftrightarrow \quad w^k \in (\partial d_f + \mathrm{Id})z^k \\ &\Longleftrightarrow \quad w^k \in z^k + \partial d_f z^k \\ &\Longleftrightarrow \quad w^k = z^k + p^k, \text{ for some } p^k \in \partial d_f(z^k). \end{aligned} \tag{5.2.10}$$

We emphasize that the equality $z^k = J_{\partial d_f}(w^k)$ indeed allows us to find a point *in the solution set* of the problem (5.2.4); see (5.2.3). Let $p^k = -Ax^k \in \partial d_f(z^k)$, where $x^k \in \operatorname*{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x; z^k)$. Thus, we may replace the first half of the iteration $w^{k+\frac{1}{2}} = \mathrm{refl}_{\partial d_f}(w^k)$ by

$$\boxed{x^k = \operatorname*{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x; z^k), \;\; w^k = z^k - Ax^k, \quad \text{and } w^{k+\frac{1}{2}} = z^k + Ax^k.} \tag{5.2.11}$$

We now obtain an alternative representation of the remaining iteration $w^{k+1} = \mathrm{refl}_{\partial d_g}(w^{k+\frac{1}{2}})$. By setting $z^{k+\frac{1}{2}} = J_{\partial d_g}(w^{k+\frac{1}{2}})$, we follow the expansion made in (5.2.9):

$$w^{k+1} = \mathrm{refl}_{\partial d_g}(w^{k+\frac{1}{2}}) = z^{k+\frac{1}{2}} + (z^{k+\frac{1}{2}} - w^{k+\frac{1}{2}}).$$

We let $y^{k+\frac{1}{2}} = \operatorname*{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y; z^{k+\frac{1}{2}})$ and obtain $-(By^{k+\frac{1}{2}} - b) \in \partial d_g(z^{k+\frac{1}{2}})$. Then, $z^{k+\frac{1}{2}} = J_{\partial d_g}(w^{k+\frac{1}{2}})$ gives rise to the relations below:

$$\boxed{y^{k+\frac{1}{2}} = \operatorname*{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y; z^{k+\frac{1}{2}}), \;\; w^{k+\frac{1}{2}} = z^{k+\frac{1}{2}} - (By^{k+\frac{1}{2}} - b) \;\; \text{and } w^{k+1} = z^{k+\frac{1}{2}} + \Big(By^{k+\frac{1}{2}} - b\Big).} \tag{5.2.12}$$

We combine (5.2.11) and (5.2.12), and produce an alternative representation of (5.2.8):

$$\begin{aligned} x^k &= \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x; z^k) && \text{(5.2.13a)} \\ z^k &= w^k + Ax^k && \text{(5.2.13b)} \\ w^{k+\frac{1}{2}} &= z^k + Ax^k && \text{(5.2.13c)} \\ y^{k+\frac{1}{2}} &= \operatorname{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y; z^{k+\frac{1}{2}}) && \text{(5.2.13d)} \\ z^{k+\frac{1}{2}} &= w^{k+\frac{1}{2}} + (By^{k+\frac{1}{2}} - b) && \text{(5.2.13e)} \\ w^{k+1} &= z^{k+\frac{1}{2}} + (By^{k+\frac{1}{2}} - b). && \text{(5.2.13f)} \end{aligned}$$

We now eliminate the governing sequence, the iterates $w^{k+\frac{1}{2}}$ and w^{k+1} from (5.2.13). We then rewrite (5.2.13c) and (5.2.13e) as below:

$$\begin{aligned} z^k &= w^k + Ax^k && \text{by} \quad (5.2.13b) \\ &= w^k - (By^{k-\frac{1}{2}} - b) + (By^{k-\frac{1}{2}} - b) + Ax^k && \\ &= z^{k-\frac{1}{2}} + Ax^k + By^{k-\frac{1}{2}} - b && \text{by} \quad (5.2.13f) \end{aligned}$$

and

$$\begin{aligned} z^{k+\frac{1}{2}} &= w^{k+\frac{1}{2}} + (By^{k+\frac{1}{2}} - b) && \text{by} \quad (5.2.13e) \\ &= w^{k+\frac{1}{2}} - Ax^k + Ax^k + (By^{k+\frac{1}{2}} - b) && \\ &= z^k + Ax^k + By^{k+\frac{1}{2}} - b && \text{by} \quad (5.2.12). \end{aligned}$$

Using the relation $z^k = z^{k-\frac{1}{2}} + Ax^k + By^{k-\frac{1}{2}} - b$, we can rewrite (5.2.13a):

$$\begin{aligned} & x^k = \operatorname*{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x; z^k) \\ \iff\ & 0 \in \partial f(x^k) + A^T z^k + \partial i_{\mathcal{X}}(x^k) \\ & \quad = \partial f(x^k) + A^T \left(z^{k-\frac{1}{2}} + (Ax^k + By^{k-\frac{1}{2}} - b) \right) + \partial i_{\mathcal{X}}(x^k) \\ \iff\ & 0 \in \partial f(x^k) + A^T z^{k-\frac{1}{2}} + A^T(Ax^k + By^{k-\frac{1}{2}} - b) + \partial i_{\mathcal{X}}(x^k) \\ \iff\ & x^k = \operatorname{argmin}_{x\in\mathcal{X}} f(x) + \langle z^{k-\frac{1}{2}}, Ax\rangle + \tfrac{1}{2}\|Ax + By^{k-\frac{1}{2}} - b\|_2^2 \\ \iff\ & x^k = \operatorname{argmin}_{x\in\mathcal{X}} f(x) + \langle z^{k-\frac{1}{2}}, Ax + Bz^{k-\frac{1}{2}} - b\rangle + \tfrac{1}{2}\|Ax + By^{k-\frac{1}{2}} - b\|_2^2 \\ \iff\ & x^k = \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_A(x, y^{k-\frac{1}{2}}; z^{k-\frac{1}{2}}). \end{aligned}$$

Here, $\partial i_{\mathcal{X}}(x^k)$ is the subdifferential of the indicator function, i, with respect to $\mathcal{X}$ at x^k. Similarly, using the relation $z^{k+\frac{1}{2}} = z^k + Ax^k + By^{k+\frac{1}{2}} - b$, we rewrite (5.2.13d):

$$\begin{aligned} & y^{k+\frac{1}{2}} = \operatorname*{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y; z^{k+\frac{1}{2}}) \\ \iff\ & 0 \in \partial g(y^{k+\frac{1}{2}}) + B^T z^{k+\frac{1}{2}} + \partial i_{\mathcal{Y}}(y^{k+\frac{1}{2}}) \\ & \quad = \partial g(y^{k+\frac{1}{2}}) + B^T \left(z^k + (Ax^k + By^{k+\frac{1}{2}} - b) \right) + \partial i_{\mathcal{Y}}(y^{k+\frac{1}{2}}) \\ \iff\ & 0 \in \partial g(y^{k+\frac{1}{2}}) + B^T z^k + B^T(Ax^k + By^{k+\frac{1}{2}} - b) + \partial i_{\mathcal{Y}}(y^{k+\frac{1}{2}}) \\ \iff\ & y^{k+\frac{1}{2}} = \operatorname{argmin}_{y\in\mathcal{Y}} g(y) + \langle z^k, By\rangle + \tfrac{1}{2}\|Ax^k + By - b\|_2^2 \\ \iff\ & y^{k+\frac{1}{2}} = \operatorname{argmin}_{y\in\mathcal{Y}} g(y) + \langle z^k, Ax^k + By - b\rangle + \tfrac{1}{2}\|Ax^k + By - b\|_2^2 \\ \iff\ & y^{k+\frac{1}{2}} = \operatorname{argmin}_{y\in\mathcal{Y}} \mathcal{L}_A(x^k, y; z^k). \end{aligned}$$

Hence, we obtain the update rules as seen in Algorithm 5.2.1.

5.2.3 PRSM with Known Dual Elements in the Solution

In this section we present the derivation for a variant of the **PRSM** that utilizes the information of the known elements of the dual optimal solutions. We follow the arguments in Section 5.2.2 to derive **rPRSM**, *the restricted dual **PRSM***, Algorithm 5.2.3. We aim to realize **rPRSM** by the composition of the dual problem (5.2.6) with a special affine map.

Let $\bar{z}^*$ be a dual optimal solution of the problem (5.2.1). Suppose that we have an element-wise partial knowledge on the dual optimal solution of the problem (5.2.1). In other words, there is a subset $\mathcal{I}^*$ of $\{1, \ldots, m\}$, and the elements $\bar{z}^*(\mathcal{I}^*)$ are known in advance. We can take advantage of this information by projecting the dual iterates $z^k, z^{k+\frac{1}{2}}$ in Algorithm 5.2.2 onto the set

$$\mathcal{Z}^* := \{z \in \mathbb{R}^m : z_i = \bar{z}_i^*, \ i \in \mathcal{I}^*\}.$$

We define the projection

$$\mathcal{P}_0 : \mathbb{R}^m \to \mathbb{R}^m \text{ by } \left(\mathcal{P}_0(z)\right)_i = \begin{cases} 0 & i \in \mathcal{I}^*, \\ z_i & \text{otherwise.} \end{cases}$$

With the projection $\mathcal{P}_0$ and by setting the initial dual iterate $z^0 \in \mathcal{Z}^*$, we maintain the dual iterates $z^k, z^{k+\frac{1}{2}}$ in the set $\mathcal{Z}^*$ as shown in Algorithm 5.2.3. The convergence of Algorithm 5.2.3 is shown in the class of the **DNN** relaxation of the quadratic assignment problem (**QAP**) [79, Theorem 3.2] via the general convergence theory of semi-proximal strictly contractive **PRSM** [81, 109].

Algorithm 5.2.3 A Restricted Dual Strictly Contractive **PRSM** (**rPRSM**) for (5.2.1)

```
Require: Initial iterates $y^0, z^0 \in \mathcal{Z}^*$, $\beta \in \mathbb{R}_{++}$, $\gamma \in (0,1)$
while stopping criteria are not satisfied do
    $x^{k+1} = \min_{x \in \mathcal{X}} \mathcal{L}_A(x, y^k, z^k)$    ($x$-subproblem)
    $z^{k+\frac{1}{2}} = z^k + \gamma\beta\mathcal{P}_0\left(Ax^{k+1} + By^k - b\right)$    (intermediate dual update).
    $y^{k+1} = \min_{y \in \mathcal{Y}} \mathcal{L}_A(x^{k+1}, y, z^{k+\frac{1}{2}})$    ($y$-subproblem)
    $z^{k+1} = z^{k+\frac{1}{2}} + \gamma\beta\mathcal{P}_0\left(Ax^{k+1} + By^{k+1} - b\right)$    (dual update).
    $k \leftarrow k+1$
end while
```

We note that the dual problem (5.2.6) is equivalent to

$$\max_z \left\{ \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathcal{L}(x, y; z) \right\} = \max_{z \, : \, z_i = \bar{z}_i^*, \ i \in \mathcal{I}^*} \left\{ \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathcal{L}(x, y; z) \right\}.$$

We define a diagonal matrix $D \in \mathbb{S}^m$ and a vector $d \in \mathbb{R}^m$

$$D_{i,i} = \begin{cases} 0 & \text{if } i \in \mathcal{I}^*, \\ 1 & \text{otherwise,} \end{cases} \quad \text{and } d_i = \begin{cases} \bar{z}_i^* & \text{if } i \in \mathcal{I}^*, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2.14}$$

Then, fixing some elements of the dual variable can be realized by

$$\{z : z_i = \bar{z}_i^*, \ i \in \mathcal{I}^*\} = \{Dz + d : z \in \mathbb{R}^m\} \tag{5.2.15}$$

Hence, the dual problem (5.2.6) is equivalent to

$$\max_z \left\{ \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathcal{L}(x, y; Dz + d) \right\}. \tag{5.2.16}$$

We now follow the derivation displayed throughout Section 5.2.2 with the problem (5.2.16). By chain rule [139, Theorem 23.9], we consider Problem 5.2.2 (monotone inclusion problem) with the replacement $\mathscr{A} \leftarrow D\partial d_f(Dz + d)$ and $\mathscr{B} \leftarrow D\partial d_g(Dz + d)$:

$$\text{find } z \in \mathbb{R}^m \text{ such that } 0 \in D\partial d_f(Dz + d) + D\partial d_g(Dz + d).$$

As in (5.2.10), we pick $p^k \in D\partial d_f(z^k)$ and set $p^k = D(-Ax^k)$, where $x^k \in \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x, z^k)$. They yield

$$x^k = \operatorname*{argmin}_{x\in\mathcal{X}} \mathcal{L}_f(x; z^k),\ \ w^k = z^k - DAx^k, \quad \text{and } w^{k+\frac{1}{2}} = z^k + DAx^k.$$

Similarly, we have $-D(By^{k+\frac{1}{2}} - b) \in D\partial d_g(z^{k+\frac{1}{2}})$, where $y^{k+\frac{1}{2}} \in \operatorname{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y, z^{k+\frac{1}{2}})$. Hence, we obtain

$$y^{k+\frac{1}{2}} = \operatorname*{argmin}_{y\in\mathcal{Y}} \mathcal{L}_g(y; z^{k+\frac{1}{2}}),\ \ w^{k+\frac{1}{2}} = z^{k+\frac{1}{2}} - D(By^{k+\frac{1}{2}} - b) \ \text{ and } w^{k+1} = z^{k+\frac{1}{2}} + D\left(By^{k+\frac{1}{2}} - b\right).$$

We again eliminate the governing sequence w^k. The elimination produces the update of the form

$$z^{k+\frac{1}{2}} = z^k + D\left(Ax^k + By^{k-\frac{1}{2}} - b\right) \quad \text{and} \quad z^{k+1} = z^{k+\frac{1}{2}} + D\left(Ax^{k+1} + By^{k+\frac{1}{2}} - b\right). \tag{5.2.17}$$

We recall that D fixes the coordinates in $\mathcal{I}^*$ to be 0. Hence, the primal residuals in (5.2.17) have no contribution on the elements associated with $\mathcal{I}^*$.

We set the initial dual variable $z^0 \in \mathbb{R}^m$ with the property $z_i^0 = \bar{z}_i^*$, for $i \in \mathcal{I}^*$. We define the projection

$$\mathcal{P}_0 : \mathbb{R}^m \to \mathbb{R}^m \text{ by } \left(\mathcal{P}_0(z)\right)_i = \begin{cases} 0 & i \in \mathcal{I}^*, \\ z_i & \text{otherwise.} \end{cases}$$

Then the dual updates are governed by the projection as shown in Algorithm 5.2.3.

5.2.4 A Restricted Dual PRSM for the DNN Relaxation

In this section we present implementation details of Algorithm 5.2.3 for the **DNN** relaxations of (**BQP**) as well as a strategy for obtaining a valid lower bound to the problem. The augmented Lagrangian for (5.1.20) with the Lagrange multiplier Z is

$$\mathcal{L}_A(R, Y, Z) = \langle \hat{Q}, Y\rangle + \langle Z, Y - VRV^T\rangle + \frac{\beta}{2}\|Y - VRV^T\|_F^2,$$

where $\beta > 0$ is a given parameter. The variables R and Y play the roles of x and y in Algorithm 5.2.3, respectively. We obtain explicit update rules tailored to our problem.

For the R-subproblem, we follow the equalities

$$\begin{aligned} R^{k+1} &= \operatorname*{argmin}_{R\in\mathcal{R}} \mathcal{L}_A(R, Y^k, Z^k) \\ &= \operatorname*{argmin}_{R\in\mathcal{R}} -\langle Z^k, VRV^T\rangle + \tfrac{\beta}{2}\|Y^k - VRV^T\|_F^2 \\ &= \operatorname*{argmin}_{R\in\mathcal{R}} \tfrac{\beta}{2}\|Y^k - VRV^T + \tfrac{1}{\beta}Z^k\|_F^2 \\ &= \operatorname*{argmin}_{R\in\mathcal{R}} \tfrac{\beta}{2}\|R - V^T(Y^k + \tfrac{1}{\beta}Z^k)V\|_F^2 \\ &= \mathcal{P}_{\mathcal{R}}(V^T(Y^k + \tfrac{1}{\beta}Z^k)V), \end{aligned}$$

where the fourth equality follows from the choice of the facial range vector V, namely, $V^TV = I$. Thus, the R-subproblem reduces to a projection problem. We now show how to perform the projection $\mathcal{P}_{\mathcal{R}}(W_R)$, where W_R is a given matrix. Let $W_R = U\Lambda U^T$ be a *spectral decomposition* of W_R. Then

$$\mathcal{P}_{\mathcal{R}}(W_R) = U\,\mathrm{Diag}(\mathcal{P}_{\Delta}(\mathrm{diag}(\Lambda)))U^T,$$

where $\mathcal{P}_{\Delta}(\mathrm{diag}(\Lambda))$ denotes the projection of $\mathrm{diag}(\Lambda)$ onto the simplex $\Lambda = \{\lambda \in \mathbb{R}_+^{n_0+1-p} : \lambda^T e = p+1\}$. The projection $\mathcal{P}_{\Delta}(\mathrm{diag}(\Lambda))$ can be performed efficiently, e.g., see [37]. The R-update reduces to the projection of the vector consists of the positive eigenvalues of $V^T(Y^k + \frac{1}{\beta}Z^k)V$ onto the simplex Δ.

We now obtain the explicit formula for the Y-subproblem. We again complete the square as seen in the R-subproblem:

$$\begin{aligned} Y^{k+1} &= \underset{Y\in\mathcal{Y}}{\operatorname{argmin}}\, \mathcal{L}_A(R^{k+1}, Y^k, Z^{k+\frac{1}{2}}) \\ &= \underset{Y\in\mathcal{Y}}{\operatorname{argmin}} \langle \hat{Q}, Y\rangle + \langle Z^{k+\frac{1}{2}}, Y - VR^{k+1}V^T\rangle + \tfrac{\beta}{2}\|Y - VR^{k+1}V^T\|_F^2 \\ &= \underset{Y\in\mathcal{Y}}{\operatorname{argmin}}\, \tfrac{\beta}{2}\left\|Y - \left(VR^{k+1}V^T - \tfrac{1}{\beta}(\hat{Q} + Z^{k+\frac{1}{2}})\right)\right\|_F^2 \\ &= \mathcal{P}_{\mathcal{Y}}\left(VR^{k+1}V^T - \tfrac{1}{\beta}(\hat{Q} + Z^{k+\frac{1}{2}})\right). \end{aligned} \tag{5.2.18}$$

We again obtain a projection. Let $W_Y = VR^{k+1}V^T - \frac{1}{\beta}(\hat{Q} + Z^{k+\frac{1}{2}})$. Then

$$\mathcal{P}_{\mathcal{Y}}(W_Y)_{i,j} = \begin{cases} 1 & \text{if } i = j = 0, \\ 0 & \text{if } (i,j),(j,i) \in \mathcal{J} \setminus (0,0), \\ \min\{1, \max\{(W_Y)_{i,j}, 0\}\} & \text{otherwise.} \end{cases} \tag{5.2.19}$$

The formula (5.2.19) displays the computational efficiency of the projection (5.2.18). We highlight again that this efficiency is the consequence of the variable splitting offered by the **FR**.

We now use Theorem 5.1.4 for the Z-update in Algorithm 5.2.3. The known elements of the dual optimal solutions are the first row, the first column and the diagonal elements excluding the $(0,0)$-th element. Then, the projection for the Z-updates in Algorithm 5.2.3 follows:

$$\begin{aligned} &\mathcal{P}_0 : \mathbb{S}^{n_0+1} \to \mathbb{S}^{n_0+1}, \text{ where} \\ &\quad (\mathcal{P}_0(Z))_{i,j} = \begin{cases} 0, & (i,j) \in \{(i,j) : i = j,\ (0,j),\ (i,0) \text{ for } i,j = 1,\ldots,n_0\}, \\ Z_{i,j}, & \text{otherwise.} \end{cases} \end{aligned}$$

The most time-consuming operation among R, Y, Z updates is the spectral decomposition that appear in the R-update. The spectral decomposition of n-by-n matrix has the complexity $O(n^3)$. We can impose additional constraints to the subproblems that appear in Algorithm 5.2.3 as long as the computational resources allow.

Valid Lower Bound Computation We now discuss a strategy for obtaining a valid lower bound to $p^*_{\mathbf{BQP}}$. Exact solutions of the **DNN** relaxation (5.1.16) provide lower bounds to (**BQP**). However, we often terminate algorithms when the stopping criteria are met for a pre-defined tolerance and we never set the tolerance to be exactly 0 in practice. A near optimal point $\tilde{Y}$ can result in

$$p^*_{\mathbf{DNN}} \leq \langle \hat{Q}, \tilde{Y}\rangle \text{ and } p^*_{\mathbf{BQP}} < \langle \hat{Q}, \tilde{Y}\rangle$$

and may not provide a valid lower bound to $p^*_{\mathbf{BQP}}$. Hence, we provide a method for obtaining a valid lower bound to (**BQP**) by forming the dual of the **DNN** relaxation:

$$\begin{aligned}
p^*_{\mathbf{DNN}} &= \min_{R\in\mathcal{R},Y\in\mathcal{Y}} \max_{Z} \left\{ \langle \hat{Q}, Y\rangle + \langle Z, Y - VRV^T\rangle \right\} \\
&= \max_{Z} \min_{R\in\mathcal{R},Y\in\mathcal{Y}} \left\{ \langle \hat{Q}, Y\rangle + \langle Z, Y - VRV^T\rangle \right\} \\
&= \max_{Z} \left\{ \min_{Y\in\mathcal{Y}} \langle \hat{Q} + Z, Y\rangle + \min_{R\in\mathcal{R}} \langle Z, -VRV^T\rangle \right\} \\
&= \max_{Z} \left\{ \min_{Y\in\mathcal{Y}} \langle \hat{Q} + Z, Y\rangle + \min_{R\in\mathcal{R}} \langle -V^TZV, R\rangle \right\} \\
&= \max_{Z} \left\{ \min_{Y\in\mathcal{Y}} \langle \hat{Q} + Z, Y\rangle - (1+p)\lambda_{\max}(V^TZV) \right\}.
\end{aligned}$$

The second equality holds by [139, Corollary 28.2.2] and [139, Theorem 28.4] and the last equality holds due to the Rayleigh's principle. We define the dual functional

$$d(Z) := \min_{Y\in\mathcal{Y}} \langle \hat{Q} + Z, Y\rangle - (1+p)\lambda_{\max}(V^TZV). \tag{5.2.20}$$

Then, for *any* $\bar{Z} \in \mathbb{S}^{n_0+1}$, $d(\bar{Z})$ provides a valid lower bound to (**BQP**):

$$d(\bar{Z}) \le p^*_{\mathbf{DNN}} \le p^*_{\mathbf{BQP}}. \tag{5.2.21}$$

We note that evaluating (5.2.20) is inexpensive. It requires one eigen decomposition and the remaining computational costs are negligible. We compute the lower bound $d(Z^k)$ at any dual iterate Z^k from Algorithm 5.2.3. We may use $\lceil d(Z^k)\rceil$ for a lower bound when $p^*_{\mathbf{BQP}}$ is known to be an integer.

5.3 Application to Protein Side-Chain Positioning Problem

The *protein side-chain positioning,* ***SCP*** problem is one of the most important subproblems of the protein structure prediction problem. The applications of **SCP** problem extend to ligand binding [103, 114] and protein-protein docking with backbone flexibility [116, 154]. A protein is a macromolecule consisting of a long main chain backbone that provides a set of anchors for a sequence of amino acid side-chains. The backbone is comprised of a repeating triplet of atoms (nitrogen, carbon, carbon) with the central carbon atom being designated as the alpha carbon. An amino acid side-chain is a smaller (1 to 18 atoms) side branch that is anchored to an alpha carbon. The positions of the atoms in a side-chain can be established by knowing the 3D position of its alpha carbon and the dihedral angles defined by atoms in the side-chain. The number of dihedral angles varies from 1 to 4 depending on the length of the side-chain. This is true for 18 of the 20 amino acids with glycine and alanine being exceptions because their low atom counts preclude dihedral angles.

It has been observed that the values of dihedral angles are not uniformly distributed. They tend to form clusters with cluster centers that are equally separated (+60, 180, -60). Consequently, if the dihedral angles are unknown, we at least have a reasonable estimate of their values by appealing to these discretized values. With this strategy being applied, a side-chain with one dihedral angle would have three possible sets of positions for its atoms. We refer to each set of atomic positions as a rotamer. A side-chain with two dihedral angles will have 3 times 3 or 9 different arrangements of the atoms (i.e. 9 rotamers). Three dihedral angles will result in 27 rotamers and four dihedral angles will give 81 rotamers.

In the **SCP** problem we are given a fixed backbone and a designation of the amino acid type for each alpha carbon. To solve the problem it is required that each amino acid is assigned a particular rotameric setting with the objective of avoiding any collisions with neighbouring amino acids that are given their rotameric settings. Avoiding collisions will lower the overall energy of the protein and, in fact, even with all possible collisions circumvented we want to have an energy evaluation that is minimal.

The **SCP** problem has been proven to be NP-hard [1]. The nature of the **SCP** problem has motivated the development of many heuristic based algorithms [7, 23, 30, 45, 141, 160] and many of these approaches rely on the graph structure of the problem. Other approaches for solving **SCP** problems have been proposed. These range from probabilistic approaches [91, 105, 143], integer programming [3, 57, 96], to semidefinite programming [27, 36]. Our approach is based on the **SDP** relaxation. Given a rotamer library, the **SCP** problem can be formulated as an *binary quadratic problem*, **BQP** with the unit row-sum constraint. We then form the **DNN** relaxation discussed throughout Section 5.1.

5.3.1 Problem Formulation as BQP and DNN Relaxation

We now present a mathematical formulation of the **SCP** problem as a **BQP**. We are given a collection of disjoint sets $\mathcal{V}_i, i = 1, \ldots, p$. Each set $\mathcal{V}_i$ has m_i members, $|\mathcal{V}_i| = m_i$, with total $n_0 = \sum_{i=1}^p m_i$ and $\mathcal{V} = \cup_{i=1}^p \mathcal{V}_i$. We call each set $\mathcal{V}_i$ a rotamer set and its members are rotamers. The *protein side-chain positioning problem* seeks to select *exactly one* rotamer v_i from each set $\mathcal{V}_i$, in order to minimize the sum of the weights (energy) on the edges between chosen rotamers, and the energy between each chosen rotamer and the backbone, see Figure 5.3.1[3]. We denote the edge weights between two distinct rotamers (nodes) $u \neq v$ by the matrix entries E_{uv}; while the diagonal entries E_{uu} denote the weight between the rotamer u and the backbone. This yields a symmetric matrix E, where $E_{uv} = \infty$ if both rotamers u, v are in the same set. We note that the multiplication $0\infty = 0$ when adding up the weights (energies). Alternatively, we can set these weights to 0 and add a constraint to choose exactly one rotamer from each set, which is what we

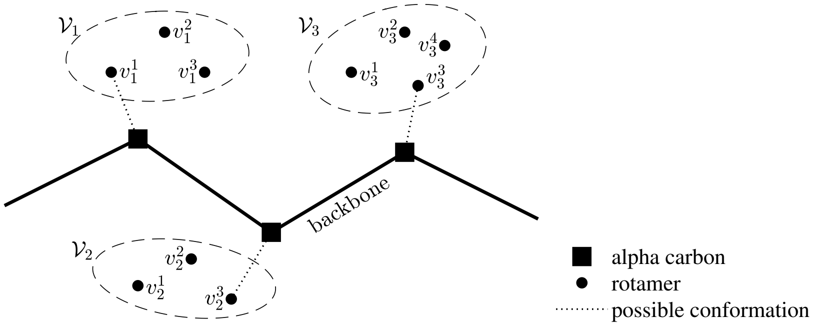

Figure 5.3.1: A diagram of the protein side-chain positioning problem

[3] $\mathcal{V}_i$ indicates the i-th rotamer set and v_i^j indicates the j-th candidate in the i-th rotamer set $\mathcal{V}_i$.

do. Thus each diagonal block of E, of size m_i, can be assumed to be a diagonal matrix. We can make this simplification without loss of generality since we are looking to only choose one rotamer per set $\mathcal{V}_i$.

We cast the settings for the **SCP** problem as the **BQP** over the indicator vector x:

$$\begin{array}{ll} \min\limits_{x} & \sum\limits_{u,v} E_{uv} x_u x_v \\ \text{subject to} & \sum\limits_{u \in \mathcal{V}_k} x_u = 1, \quad k = 1, \dots, p \\ & x = (x_u) \in \{0,1\}^{n_0}. \end{array} \tag{5.3.1}$$

Then we can adopt the notation A_u defined in (5.1.1) to model the *unit row-sum* constraint so that the solutions choose exactly one rotamer for each set. We can rewrite the program (5.3.1) as follows:

$$(\mathbf{BQP_{SCP}}) \qquad \begin{array}{rll} p^*_{\mathbf{BQP_{SCP}}} := & \min\limits_{x} & x^T E x \\ & \text{subject to} & A_u x = \bar{e}_p \\ & & x = \begin{bmatrix} v_1^T & v_2^T & \dots & v_p^T \end{bmatrix}^T \in \{0,1\}^{n_0} \\ & & v_i \in \{0,1\}^{m_i}, \ i = 1, \dots, p. \end{array} \tag{5.3.2}$$

Following the derivation in Section 5.1, we form the **DNN** relaxation for ($\mathbf{BQP_{SCP}}$) in (5.3.2):

$$(\mathbf{DNN}) \qquad p^*_{\mathbf{DNN_{SCP}}} = \min_{R,Y} \ \{\langle \hat{E}, Y \rangle : Y = VRV^T, \ R \in \mathcal{R}, \ Y \in \mathcal{Y}\}, \tag{5.3.3}$$

where $\hat{E} = \operatorname{BlkDiag}(0, E)$ and $\mathcal{R}, \mathcal{Y}$ defined in (5.1.19).

5.3.2 The Strengths of DNN Relaxation

In this section we provide numerical experiments with real-world data and discuss the strengths of the **DNN** relaxation. We observe the useful aspects of the **DNN** relaxation through the numerical experiments. The **DNN** relaxation provides a means of treating an ill-posed data matrix with large positive values, hence we can avoid numerical instabilities. Moreover, we observe that the **DNN** relaxation provides superior performance over the **SDP** relaxation.

5.3.2.1 Implementation Details

Energy Matrix Computation We give an outline for acquiring the energy matrix E in (5.3.2), the data matrix in the objective. Our implementation relies on the usage of a Python script executing as an extension of the UCSF Chimera[4] application. A detailed implementation can be found in [29, Chapter 7]. We used protein data files from the Protein Data Bank (PDB)[5] to obtain the coordinates of all atoms in the protein. To get the energy values required by the algorithm, the native side chain conformations were replaced by rotamers extracted from a rotamer library provided by the Dunbrack Laboratory [51].

Some approaches use an energy evaluation based on a piece-wise linear approximation of the Lennard-Jones potential formula (e.g., [30,160]). Here, we use the Lennard-Jones potential formula,

[4] The UCSF Chimera software can be found in https://www.cgl.ucsf.edu/chimera/download.html.

[5] https://www.rcsb.org/

which provides a more accurate energy value computation. In brief, the Lennard-Jones potential formula engages the Euclidean distance between a pair of atoms with some parameters dependant on the type of amino acids. A more detailed explanation of these energy computations can be found in [29, Chapter 6-7]. We finally use a strategy (known as 'dead end elimination') to reduce the size of the rotamer sets associated with each amino acid. The basic idea behind this strategy is that a rotamer can be removed from its rotamer set if there is another rotamer in that set that gives a better energy value regardless of the rotamer selections for the neighbouring amino acids. Among various approaches for the dead end elimination, we followed the Goldstein's criteria [75].

Let $\mathcal{U}$ be a side-chain conformation of a protein. The energy of the conformation $\mathcal{U}$ is

$$E(\mathcal{U}) = \sum_{i=1}^{n_0} E_{\text{self}}(u_i) + \sum_{i=1}^{n_0-1} \sum_{j=i+1}^{n_0} E_{\text{pair}}(u_i, u_j),$$

where u_i is a side-chain conformation of an amino acid, $E_{\text{self}}(u_i)$ is the energy corresponding to u_i and the backbone, and $E_{\text{pair}}(u_i, u_j)$ is the energy formed by u_i and u_j, a rotamer associated with a neighbouring amino acid. In our formulation, we placed $E_{\text{self}}(u_i)$ along the diagonal of E and $E_{\text{pair}}(u_i, u_j)$ on the appropriate off-diagonal positions of E as shown in Section 5.3.1.

Effective Removal of Collisions We typically observe some very large elements in E. This is due to the collisions between rotamers and they are indicated by *huge* energy values $E_{ij} >> 0$ that are often greater than 10^{10}. These huge values occur due to a part of the Lennard-Jones potential formula that involves the Euclidean distance between two distinct rotamers that goes to the denominator of a fraction.

In general, having very large values in data is prone to numerical instabilities. If every nonzero elements of E are large, the usual approach is to scale E to avoid large values. However, the matrix E often has elements $E_{i,j}$ that are more than 10 digits as well as elements that are 1 digit. When there is a large discrepancy among the elements of E, scaling E would make the relatively small values close to 0 and lead to loss of precision in the solution. However, this ill-posed data does not take place as a problem in our implementation. Recall that we solve the Y-subproblem (5.2.18) as follows:

$$Y^{k+1} = \mathcal{P}_{\mathcal{Y}}\left(G_{\mathcal{J}^c}\left(VR^{k+1}V^T - \frac{1}{\beta}(\hat{E} + Z^{k+\frac{1}{2}})\right)\right) = \mathcal{P}_{\mathcal{Y}}\left(G_{\mathcal{J}^c}\left(-\frac{1}{\beta}\hat{E} + \left[VR^{k+1}V^T - \frac{1}{\beta}Z^{k+\frac{1}{2}}\right]\right)\right).$$

If there is a very large element $(\hat{\imath}, \hat{\jmath})$ in $\hat{E} = \text{blkdiag}(0, E)$, the projection $\mathcal{P}_{\mathcal{Y}}$ sets the $(\hat{\imath}, \hat{\jmath})$-element of Y^{k+1} to 0; see (5.2.19). Hence, for those positions $(\hat{\imath}, \hat{\jmath})$ with very large energy values, the constraint $Y_{\hat{\imath},\hat{\jmath}} = 0$ is implicitly imposed. We can interpret this as having implicit gangster constraints on these elements. Consequently, the large elements do not contribute to the objective value since $\hat{E}_{\hat{\imath},\hat{\jmath}} Y_{\hat{\imath},\hat{\jmath}} = 0$.

We can also take advantage of the large values in the data matrix to eliminate edges in the graph and increase the size of the gangster indices.

Lemma 5.3.1. *Suppose that x is feasible for the ($\boldsymbol{BQP_{SCP}}$) in (5.3.2), and let $u = x^T E x$ be its objective value. Let $N_E = \sum_{\{ij:E_{ij}<0\}} E_{ij}$ and suppose that*

$$E_{i_0 j_0} > u - N_E, \text{ for some } i_0, j_0$$

holds. Then, for any optimal solution x^* to ($\boldsymbol{BQP_{SCP}}$), we have $x^*_{i_0}x^*_{j_0} = 0$.

Proof. Let x^* be an optimal solution to ($\mathbf{BQP_{SCP}}$). Let $\mathcal{U}$ be the set of selected rotamers found in the optimal solution x^*. We note that, for any set S, we have

$$\sum_{(i,j)\in S} E_{i,j} = \sum_{(i,j)\in S\cap\{(i,j):E_{i,j}\geq 0\}} E_{i,j} + \sum_{(i,j)\in S\cap\{(i,j):E_{i,j}<0\}} E_{i,j} \geq 0 + N_E = N_E.$$

Suppose to the contrary that x^* satisfies $x^*_{i_0}x^*_{j_0} = 1$, i.e., $x^*_{i_0} = x^*_{j_0} = 1$. Then we reach the following contradiction:

$$p^*_{\mathbf{BQP_{SCP}}} = \langle x^*, Ex^*\rangle = E_{i_0j_0} + \left(E_{i_0j_0} + \sum_{(i,j)\in U\backslash\{(i_0,j_0)\}} E_{i,j} \right) \ \geq \ E_{i_0j_0} + N_E > u.$$

□

Corollary 5.3.2. *Let i_0 be an index such that $E_{i_0i_0} > u - N_E$, where u and N_E defined in Lemma 5.3.1. Then, for any optimal solution x^* to $\boldsymbol{BQP_{SCP}}$, we have*

$$Y_{x^*} := \begin{pmatrix} 1 \\ x^* \end{pmatrix} \begin{pmatrix} 1 \\ x^* \end{pmatrix}^T \in \left\{ Y \in \mathbb{S}^{n_0+1} : Y(:,i_0) = 0,\ Y(i_0,:) = 0 \right\}.$$

Proof. Let i_0 be an index such that $E_{i_0i_0} > u - N_E$. Then $x^*_{i_0} = 0$ by Lemma 5.3.1. We note that Y_{x^*} is a positive semidefinite matrix. If a diagonal entry of a positive semidefinite is zero, then its corresponding column and row must be 0; see Fact 2.2.5. □

By Lemma 5.3.1 and Corollary 5.3.2, if we detect entries i_0, j_0 that has the property $E_{i_0i_0} > u - N_E$, then we may strengthen the gangster constraint as follows:

$$\mathcal{J} \leftarrow \mathcal{J} \cup \left\{ Y \in \mathbb{S}^{n_0+1} : \begin{array}{ll} Y(i_0,j_0) = Y(j_0,i_0) = 0, & \text{for } i_0 \neq j_0 \text{ such that } \ E_{i_0j_0} > u - N_E \\ Y(:,i_0) = 0,\ Y(i_0,:) = 0, & \text{for } i_0 \text{ such that } \ E_{i_0i_0} > u - N_E \end{array} \right\}.$$

Upper Bound Computation We discuss two strategies for obtaining upper bounds to the **SCP** problem. These strategies are derived from those presented in [27, 36] and we include them here for completeness. We obtain upper bounds by finding feasible solutions to the original integer model in (5.3.2). Let $(R^{\text{out}}, Y^{\text{out}}, Z^{\text{out}})$ be the output of the algorithm.

1. Let $x^{\text{approx}} \in \mathbb{R}^{n_0}$ be the second through to the last elements of the first column of Y^{out}. Note that $0 \leq x^{\text{approx}} \leq 1$. Let $\mathcal{S} = \{x \in \{0,1\}^{n_0} : Ax = \bar{e}_p\}$ be the feasible region of **BQP**. Then, owing to the feasibility, the nearest feasible solution to (**BQP**) from x^{approx} can be found by solving the following projection (see [27, Proposition 5.1]):
$$\operatorname{argmin}_x \left\{ \|x - x^{\text{approx}}\|^2 \ : x \in \mathcal{S} \right\} = \operatorname{argmin}_x \left\{ \langle x, x^{\text{approx}} \rangle \ : x \in \mathcal{S} \right\}. \tag{5.3.4}$$

2. The second approach is based on the Eckart-Young theorem, the best rank-one approximation argument. Let $Y^{\text{out}} = \sum_{i=1}^r \lambda_i v_i v_i^T$ be the compact spectral decomposition, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. And by abuse of notation we set v_i to be the vectors in $\mathbb{R}^{n_0}$ formed by

discarding the first element from v_i. We now let use the most dominant eigen pair of Y^{out} to form x^{approx}, i.e., $x^{\text{approx}} = \lambda_1 v_1$. We again obtain the nearest feasible solution to x^{approx} by solving (5.3.4).

In fact, solving (5.3.4) does not require using any **LP** software; we can obtain the optimal solution for (5.3.4) as follows. We partition x^{approx} into p subvectors of sizes $m_i = |\mathcal{V}_i|$, for $i = 1, \ldots, p$. Let $x^i \in \mathbb{R}^{m_i}$ be the subvector of x^{approx} associated with i-th rotamer set $\mathcal{V}_i$, i.e., $x^{\text{approx}} = [x^1; x^2; \ldots; x^p]$. We define $\hat{x}^i \in \mathbb{R}^{m_i}$ as follows:

$$\hat{x}^i_j = \begin{cases} 1, & \text{if } x^i_j = \max\limits_{\ell \in [m_i]} \{x^i_\ell\} \\ 0, & \text{otherwise.} \end{cases}$$

If there is subvector $\hat{x}^i$ with more than one 1 in its components, we pick only one 1 and set the remaining to be 0. We then form $\hat{x} = [\hat{x}^1; \hat{x}^2; \ldots; \hat{x}^p] \in \mathbb{R}^{n_0}$. It is clear that $\hat{x}$ is feasible for (5.3.1). We use $\hat{x}^T E \hat{x}$ as an upper bound to the **SCP** problem.

Stopping Criteria We terminate **rPRSM** when either of the following conditions is satisfied.

1. Maximum number of iterations, denoted by "maxiter" is achieved.
2. For given tolerance ϵ, the following bound on the primal and dual residuals holds for m_t sequential times:
$$\max \left\{ \|Y^k - VR^kV^T\|_F, \beta \|Y^k - Y^{k-1}\|_F \right\} < \epsilon.$$
3. Let $\{l_1, \ldots, l_k\}$ and $\{u_1, \ldots, u_k\}$ be sequences of lower and upper bounds discussed in (5.2.20) and (5.3.4), respectively. Any of the lower bounds achieve the best upper bound, i.e.,
$$\min\{l_1, \ldots, l_k\} \geq \max\{u_1, \ldots, u_k\}.$$

Parameter Settings We use the following parameters related to the implementation. For β and γ, we choose

$$\beta = \max\{\lfloor 0.5 \cdot n_0/p \rfloor, 1\}, \quad \gamma = 0.99.$$

The parameters related to stopping criteria are:

$$\text{maxiter} = \min\{10^5, p(n_0 + 1) + 10^4\}, \quad \epsilon = 10^{-10}, \quad m_t = 100.$$

For the initial iterates for **rPRSM**, with $\mathcal{Z}_A$ defined in Theorem 5.1.4, we use

$$Y^0 = 0, \; Z^0 = \mathcal{P}_{\mathcal{Z}_A}(Y^0).$$

5.3.2.2 Experiments with Real-World Data

We now provide numerical experiments using the selected instance from the Protein Data Bank[6]. We select instances listed in [30] with proteins that have up to 300 amino acids. The following list defines the column headers used in Table 5.3.1. We use the same headers to the additional numerical experiments that are displayed in Appendix A.1.

[6] https://www.rcsb.org/

1. **name**: instance name;
2. $\boldsymbol{p}$: the number of amino acids;
3. $\boldsymbol{n_0}$: the total number of rotamers;
4. **lbd**: the lower bound obtained by running **rPRSM**;
5. **ubd**: the upper bound obtained by running **rPRSM**;
6. **rel-gap**: relative gap of each instance using **rPRSM**, where

$$\text{relative gap} := 2\,\frac{|\text{best feasible upper bound} - \text{best lower bound}|}{|\text{best feasible upper bound} + \text{best lower bound} + 1|}; \tag{5.3.5}$$

7. **iter**: the number of iterations used by **rPRSM** with tolerance $\epsilon = 10^{-10}$;
8. **time(sec)**: the running time (in seconds) used by **rPRSM**.

Problem Data				**Numerical Results**			**Timing**	
#	**name**	$\boldsymbol{p}$	$\boldsymbol{n_0}$	**lbd**	**ubd**	**rel-gap**	**iter**	**time(sec)**
10	2IGD	50	126	-78.50608	-78.50608	5.39611e-15	500	19.43
20	1VQB	75	406	-96.94940	-96.94940	4.34568e-14	900	179.35
30	2ACY	84	580	-146.32254	-146.32254	1.06468e-14	7800	2610.24
40	2TGI	100	355	-14.03554	-14.03554	2.46249e-13	1300	136.30
50	2SAK	111	214	-239.86975	-239.86975	1.08995e-12	500	25.50
60	2CPL	132	819	-284.97180	-284.97180	9.75693e-15	5900	3292.98
70	1CV8	146	730	-213.13554	-213.13554	3.28738e-13	5600	2572.99
80	2ENG	162	867	82.01797	82.01797	1.33295e-13	14200	8274.48
90	1A7S	179	524	-239.78218	-239.78218	1.00542e-14	1200	314.57
100	1MRJ	208	1178	-295.13711	-295.13711	1.70740e-13	2300	2421.15
110	1EZM	239	1497	-217.36581	-217.36581	3.49620e-13	2300	3876.18
120	1SBP	256	1704	-271.08838	-271.08838	3.59996e-14	40000	609487.29
130	3PTE	284	2006	161.17216	161.17216	5.09815e-15	13500	250604.17

Table 5.3.1: Computational results on selected PDB instances

We observe from the last two columns of Table 5.3.1 that many instances are solved within good relative gaps. In fact, most of the instances display relative gaps that are essentially 0. We recall from (5.3.4) that we obtain the upper bounds via finding feasible solutions to (**BQP**). That we have the relative gap essentially 0 grants us the attainment of the *globally optimal* solutions to the **SCP** problem. Approaches involving heuristic algorithms do not provide a natural means of certifying optimality, relying solely on a comparison of the rotameric solution with the so-called χ_1 and χ_2 angles from the PDB while ignoring optimality of the discretized solution. We highlight that we provide not only the globally optimal solutions but also a way to certify their optimality.

A Tight Relaxation We illustrate the strengths of the **DNN** relaxation by comparing the optimal values of the **DNN** relaxation and the **SDP** relaxation. In our test, we selected five small instances. As discussed above, some elements of the data E are typically very large due to the collisions in rotamers, typically at least 10 digits. These cause numerical difficulties when a standard interior point solver is used. Hence, in our test, we set the entries $E_{i,j} = \min\{10^4, E_{i,j}\}$, $\forall i, j$, in order to avoid the difficulties from having these large elements. We used the **rPRSM** algorithm for **DNN** relaxation and used the SDPT3 for solving the **SDP** relaxation. The displayed values in Table 5.3.2 are the best lower bounds found from **rPRSM** and the optimal reported by

problem #	instance	**DNN** relaxation	**SDP** relaxation
1	1AIE	-46.96	-2460.53
2	2ERL	55.33	-18241.26
3	1CBN	-40.43	-22380.58
4	1RB9	-76.97	-23936.35
5	1BX7	16.96	-23965.88

Table 5.3.2: The solver optimal values of the **DNN** and **SDP** relaxations on selected instances

SDPT3. We observe in Table 5.3.2 that the **DNN** relaxation shows superior performances over the **SDP** relaxations in the relaxation values; the **DNN** relaxation for the **SCP** problem provides a much tighter relaxation than the **SDP** relaxation.

5.4 Application to Quadratic Assignment Problem

The *quadratic assignment problem, **QAP***, is one of the fundamental combinatorial optimization problems in the field of operations research, and includes many important applications. It is arguably one of the hardest of the NP-hard problems. The **QAP** models real-life problems such as facility location. Suppose that we are given a set of n facilities and a set of n locations. For each pair of locations (s,t) a distance B_{st} is specified, and for each pair of facilities (i,j) a weight or flow $A_{i,j}$ is specified, e.g., the amount of supplies transported between the two facilities. In addition, there is a location (building) cost C_{is} for assigning a facility i to a specific location s. The problem is to assign each facility to a distinct location with the goal of minimizing the sum over all facility-location pairs of the distances between locations multiplied by the corresponding flows between facilities, along with the sum of the location costs. This is formulated

$$\min_{\pi\in\Lambda} \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}B_{\pi(i),\pi(j)} + \sum_{i=1}^{n} C_{i,\pi(i)}, \tag{5.4.1}$$

where Λ is the set of all permutations of $\{1,\dots,n\}$.

Applications of the **QAP** include: scheduling, production, computer manufacture, and other fields, see e.g., [55,69,86,98,152]. Moreover, many classical combinatorial optimization problems, including the travelling salesman problem, maximum clique problem, and graph partitioning problem, can all be expressed as a **QAP**. For more information see e.g., [16,33,40,124,125]. There are three main classes for method for solving the **QAP**; heuristic methods, branch-and-bound methods and methods based on the **SDP** relaxations. Among many available methods for solving the **QAP**, we focus on the last type of class.

5.4.1 Problem Formulation as BQP

We adopt the trace inner product reformulation of the **QAP** (5.4.1):

$$\min_{X\in\Pi} \langle AXB - 2C, X\rangle, \tag{5.4.2}$$

where $A,B\in\mathbb{S}^n$, $C\in\mathbb{R}^{n\times n}$, and Π is the set of n-by-n permutation matrices. By the Birkhoff-Neumann theorem [17,153], it is known that Π is equal to the set of extreme points of the doubly stochastic matrices. This leads to an alternative representation of the set Π

$$\Pi = \mathcal{D}_e \cap \mathcal{B}_z,$$

where

$$\begin{array}{rcl} \mathcal{D}_e & := & \{X \in \mathbb{R}^{n\times n} : Xe = e, X^T e = e\}, \\ \mathcal{B}_z & := & \{X \in \mathbb{R}^{n\times n} : X_{ij} \in \{0,1\},\ \forall i,j \in [n]\}. \end{array}$$

The characterization $X \in \Pi = \mathcal{D}_e \cap \mathcal{B}_z$ allows us to formulate the problem (5.4.2) as a **BQP** with linear constraints. We note the characterizations

$$\begin{array}{rcccl} X^T e = e & \iff & e^T X I = e & \iff & (I \otimes e^T)\operatorname{vec}(X) = e; \\ Xe = e & \iff & I X e = e & \iff & (e^T \otimes I)\operatorname{vec}(X) = e. \end{array}$$

Hence, (5.4.2) is equivalent to

$$(\mathbf{BQP_{QAP}}) \qquad \begin{array}{rccl} p^*_{\mathbf{BQP_{QAP}}} & = & \min\limits_x & x^T(B^T \otimes A)x - 2\operatorname{vec}(C)^T x \\ & & \text{subject to} & (I \otimes e^T)x = e \\ & & & (e^T \otimes I)x = e \\ & & & x \in \{0,1\}^{n^2}. \end{array} \qquad (5.4.3)$$

The **QAP** is a particular instance of **BQP** with the unit row-sum constraint and an additional affine constraint. The data matrix $I \otimes e^T$ is in the form of A_u defined in (5.1.1) with $p = n$ and $m_i = n$, for all $i \in [p]$. The additional set of constraints is $(e^T \otimes I)x = e$ and it gives rise to an additional set of gangster indices as we observe in the next section.

5.4.2 DNN Relaxation of QAP

We now derive the **DNN** relaxation of ($\mathbf{BQP_{QAP}}$). We follow the implications made in (5.1.4). By regarding $A_u \leftarrow (I \otimes e^T)$, we obtain the a set of gangster indices from the support of the matrix

$$A_u^T A_u - I = (I \otimes e^T)^T (I \otimes e^T) - I = \left(I \otimes ee^T\right) - I.$$

Similarly, $Xe = e$ gives rise to a variant of the gangster constraint (5.1.7). By regarding $A_u \leftarrow (e^T \otimes I)$, we obtain an additional set of gangster indices comprised of the support of the matrix

$$A_u^T A_u - I = (e^T \otimes I)^T (e^T \otimes I) - I = \left(ee^T \otimes I\right) - I.$$

Consequently, we have the following set of gangster indices for ($\mathbf{BQP_{QAP}}$):

$$\mathcal{J}_Q := \operatorname{supp}\left(\operatorname{BlkDiag}(1, I \otimes ee^T - I)\right) \cup \operatorname{supp}\left(\operatorname{BlkDiag}(1, ee^T \otimes I - I)\right).$$

Given Y following the partition defined in (5.1.13), the action of gangster operator $G_{\mathcal{J}_Q}(Y) = E_{00}$ can be described as follows in plain language; the first gangster set refers to setting the off-diagonal elements of the diagonal blocks to be 0; and the second gangster set refers to setting the diagonal elements of the off-diagonal blocks to be 0.

We follow the derivation discussed in Section 5.1. We now obtain an exposing vector for the **SDP** relaxation of ($\mathbf{BQP_{QAP}}$). We recall from (5.1.9) that the linear map gives rise to an exposing vector. We let

$$H := \begin{bmatrix} (I \otimes e^T) \\ (e^T \otimes I) \end{bmatrix} \in \mathbb{R}^{2n \times n^2}.$$

Then we have the linear equality $Hx = e$ that allows us to form the exposing vector as introduced in (5.1.9):

$$\begin{bmatrix} -e & H \end{bmatrix}^T \begin{bmatrix} -e & H \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = 0.$$

and we obtain the exposing vector

$$K = \begin{bmatrix} -e & H \end{bmatrix}^T \begin{bmatrix} -e & H \end{bmatrix} = \begin{bmatrix} 2n & -2\bar{e}_{n^2}^T \\ -2\bar{e}_{n^2} & H^T H \end{bmatrix} = \begin{bmatrix} 2n & -2\bar{e}_{n^2}^T \\ -2\bar{e}_{n^2} & I \otimes ee^T + ee^T \otimes I \end{bmatrix}. \tag{5.4.4}$$

The exposing vector K is in fact maximal, and it gives the minimal facial range vector V presented in [166]:

$$V = \begin{bmatrix} 1 & 0 \\ \frac{1}{n}e & V_e \otimes V_e \end{bmatrix} \in \mathbb{R}^{(n^2+1)\times((n-1)^2-1)}, \text{ where } V_e = \begin{bmatrix} I_{n-1} \\ -e_{n-1}^T \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}.$$

We replace V with a facial range vector with orthonormal columns. After the lifting, we obtain the linear objective in Y:

$$\langle AXB - 2C, X \rangle = \langle L_Q, Y \rangle, \quad \text{where } L_Q = \begin{bmatrix} 0 & -\operatorname{vec}(C)^T \\ -\operatorname{vec}(C) & B \otimes A \end{bmatrix}.$$

Finally, we obtain the **DNN** relaxation for (**BQP$_{\mathbf{QAP}}$**):

$$p^*_{\mathbf{DNN_{QAP}}} = \min_{R,Y} \left\{ \langle L_Q, Y \rangle : G_{\mathcal{J}_Q}(Y) = E_{00}, \ Y = VRV^T, \ 0 \le Y \le 1, \ R \succeq 0 \right\}. \tag{5.4.5}$$

A strictly feasible point of (5.4.5) can be found by using the barycenter of the rank-one lifted matrices of the ground set Π, e.g., see [166].

5.4.3 Numerical Experiment

We now present numerical results for Algorithm 5.2.3 using the real-world data from QAPLIB[7]. In Section 5.4.3.1 we discuss the upper bounding strategies, list the stopping criteria and the parameter settings. In Section 5.3.2.2 we exhibit the comparative performance between **rPRSM** and [123, **ADMM**]. We aim to show that our proposed approach shows significant improvements on the relative gaps.

5.4.3.1 Implementation Details

Upper Bound Computation Following the same approach from (5.3.4), given $\bar{X} \in \mathbb{R}^{n\times n}$, the nearest permutation matrix X^* from $\bar{X}$ is found by solving

$$X^* = \operatorname*{argmin}_{X \in \Pi} \|X - \bar{X}\|_F^2 = \operatorname*{argmin}_{X \in \Pi} -\langle \bar{X}, X \rangle. \tag{5.4.6}$$

[7] http://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/

Any solution to the problem (5.4.6) yields a feasible solution to the original **QAP**, which gives a valid upper bound $p^*_{\mathbf{BQP_{QAP}}}$. Since the permutation matrices are the extreme points of the set of doubly stochastic matrices $\mathcal{D}_e$, we reformulate the problem (5.4.6) as the **LP**

$$\max_{x\in\mathbb{R}^{n^2}} \left\{\langle \operatorname{vec}(\bar{X}), x\rangle \ : \ (I_n \otimes e^T)x = e, \ (e^T \otimes I_n)x = e, \ x \geq 0\right\}, \tag{5.4.7}$$

and we solve (5.4.7) using a simplex method.

For the choice of $\bar{X}$, [123] present two methods for obtaining upper bounds using a nearest permutation matrix, identical to the ones discussed in the upper bound computation discussed in Section 5.3.2.1. Inspired by the approximation algorithm in [73], we use an additional strategy using a nearest permutation matrix. We let ξ be a random vector in $\mathbb{R}^r$ with elements in $(0,1)$, and in decreasing order. We use ξ to perturb the eigenvalues $\lambda_1,\ldots,\lambda_r$ and form $\bar{X}$ for the upper bound problem (5.4.7) so that:

$$\operatorname{vec}(\bar{X}) = \sum_{i=1}^{r} \xi_i \lambda_i v_i.$$

We repeat this $\max\{1, \min(3 * \lceil \log(n)\rceil, \text{ubest} - \text{lbest}\}$ number of times, where 'ubest' and 'lbest' refer to the best upper and lower bounds achieved during the **rPRSM** routine, respectively. We update the current upper bound 'ubest' if a smaller upper bound is obtained by any of approaches listed above.

Stopping Criteria We terminate the algorithm when at least one of the following list of stopping condition is met. We adopt all stopping criteria used for the **SCP** problem listed in Section 5.3.2.1. That is, we terminate the algorithm when maximum iteration is reached or the best lower bound meets the best upper bound. We terminate the algorithm if the residual error is less than ϵ for the m_t consecutive times. Let $\{\ell_1,\ldots,\ell_k\}$ and $\{u_1,\ldots,u_k\}$ be sequences of lower and upper bounds from (5.2.21), respectively. The lower (resp. upper) bounds do not change for m_ℓ (resp. m_u) sequential times. Finally, we terminate the algorithm when the **KKT** conditions given in Proposition 5.1.5 are satisfied within a pre-defined tolerance $\epsilon > 0$; for a predefined tolerance $\delta > 0$, it holds that

$$\max\left\{\|R^k - \mathcal{P}_{\mathcal{R}}(R^k + V^T Z^k V)\|_F, \ \|Y^k - \mathcal{P}_{\mathcal{Y}}(Y^k - L_Q - Z^k)\|_F, \ \|Y^k - VR^kV^T\|_F\right\} < \delta.$$

Parameter Settings We set the parameter $\beta = \frac{n}{3}$ and the under-relaxation parameter $\gamma = 0.9$ for the dual variable update. We choose the initial iterates[8]

$$Y^0 = \frac{1}{n!}\sum_{X\in\Pi}(1;\operatorname{vec}(X))(1;\operatorname{vec}(X))^T \quad \text{and} \quad Z^0 = \mathcal{P}_{\mathcal{Z}_A}(0).$$

For the parameters related to the stopping condition, we set maxiter $= 40000$, $\epsilon = 10^{-5}$, $m_t = 100$, and $m_\ell = m_u = 100$. We use the **KKT** condition stopping criterion for instances with n larger than 20 and we set the tolerance $\delta = 10^{-5}$ when it is used. We compute the lower and upper bounds every 100 iterations.

[8] The formula for Y^0 is introduced in [166, Theorem 3.1].

5.4.3.2 Experiments with Real-World Data

We now provide numerical experiments using selected instances from the QAPLIB[9]. We use the instances with symmetric, integral data matrices A and B in the model (5.4.2). When A, B are symmetric, $p^*_{\mathbf{BQP_{QAP}}}$ is an even number. Hence, we add 1 when a lower bound obtained is an odd number.

The below is the list of headers used in Table 5.4.1[10].

1. **true-opt**: global optimal value; marked with * when unknown.
2. **lbd**: lower bound from **rPRSM**;
3. **ubd**: upper bound from **rPRSM**;
4. **rel.gap**: relative gap from **rPRSM**, with the formula (5.3.5)
5. **rel.opt.gap**: relative optimality gap from **rPRSM** using the known true optimal value and the lower bound;
6. **rel.gap$^{\mathbf{A}}$**: relative gap from [123, **ADMM**] with tolerance $\epsilon = 10^{-5}$;
7. **iter**: number of iterations by **rPRSM** with tolerance $\epsilon = 10^{-5}$;
8. **iter$^{\mathbf{A}}$**: number of iterations from [123, **ADMM**] with tolerance $\epsilon = 10^{-5}$;
9. **time**: solver **rPRSM** time;
10. **time$^{\mathbf{A}}$**: solver [123, **ADMM**] time.

Additional numerical experiments are displayed in Table A.2.1 and Table A.2.2 in Appendix A.2 and we use the same headers listed above.

We discuss the results exhibited in Table 5.4.1. 45 out of 46 instances are solved with relative gaps just as good as the ones obtained by **ADMM** and these instances are marked with boldface in Table 5.4.1.

[9] http://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/

[10] All the numerical tests are performed using MATLAB version 2021a on Dell XPS 8940 with 11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz 2.60 GHz with 32 Gigabyte memory.

Problem Data			**Numerical Results**					**Timing**			
#	**name**	**true-opt**	**lbd**	**ubd**	**rel.gap**	**rel.opt.gap**	**rel.gap$^{\mathbf{A}}$**	**iter**	**iter$^{\mathbf{A}}$**	**time**	**time$^{\mathbf{A}}$**
1	chr12a	9552	9548	9552	0.04	0.04	0.02	10900	24800	20.13	44.24
2	chr12b	9742	9742	9742	**0**	0	0.08	6500	26700	14.18	46.98
3	chr12c	11156	11156	11156	**0**	0	0	1100	19400	2.36	34.38
4	chr15a	9896	9896	9896	**0**	0	0.28	4400	30900	17.25	127.24
5	chr15b	7990	7990	7990	**0**	0	0.03	2000	20300	7.79	83.69
6	chr15c	9504	9504	9504	**0**	0	0.08	1100	20000	4.64	84.61
7	chr18a	11098	11098	11098	**0**	0	0.00	1100	20600	9.57	183.46
8	chr18b	1534	1534	1642	**6.80**	0	59.83	2372	12600	24.12	126.03
9	chr20a	2192	2192	2192	**0**	0	0.18	2300	33700	35.95	527.66
10	chr20b	2298	2298	2298	**0**	0	0	700	26200	11.80	436.49
11	chr20c	14142	14140	14142	**0.01**	0.01	0.15	35000	33700	578.68	522.53
12	els19	17212548	17208746	17212548	**0.02**	0.02	0.35	16200	40000	219.94	517.13
13	esc16a	68	64	68	**6.02**	6.02	37.74	1406	597	8.34	3.37
14	esc16b	292	290	294	**1.37**	0.69	6.66	557	399	3.14	2.21
15	esc16c	160	154	160	**3.81**	3.81	23.93	748	896	5.11	5.18
16	esc16d	16	14	18	**24.24**	12.90	87.50	891	659	5.93	3.86
17	esc16e	28	28	28	**0**	0	58.97	1500	556	8.68	3.23
18	esc16f	0	0	0	**0**	0	0	0	1	0.00	0.00
19	esc16g	26	26	26	**0**	0	17.86	300	695	1.76	4.01
20	esc16h	996	978	996	**1.82**	1.82	31.76	623	609	3.80	3.51
21	esc16i	14	12	14	**14.81**	14.81	88.89	10200	2044	58.18	11.46
22	esc16j	8	8	8	**0**	0	82.76	100	787	0.60	4.47
23	had12	1652	1652	1652	**0**	0	0	300	11600	0.63	21.90
24	had14	2724	2724	2724	**0**	0	0	400	20300	1.44	68.81
25	had16	3720	3720	3720	**0**	0	0	500	18100	3.13	99.30
26	had18	5358	5358	5358	**0**	0	0.02	1100	34700	11.01	339.89
27	had20	6922	6922	6922	**0**	0	0.13	1800	40000	30.69	655.85
28	nug12	578	568	616	**8.10**	1.74	24.13	4842	2884	11.91	5.79
29	nug14	1014	1012	1022	**0.98**	0.20	1.08	8087	19600	32.76	71.64
30	nug15	1150	1142	1268	**10.45**	0.70	16.33	6965	5812	34.93	27.47
31	nug16a	1610	1600	1610	**0.62**	0.62	0.62	10500	19300	69.99	117.93
32	nug16b	1240	1220	1258	**3.07**	1.63	25.41	7069	2347	43.61	13.46
33	nug17	1732	1708	1756	**2.77**	1.39	2.77	10500	6401	93.09	50.14
34	nug18	1930	1894	1990	**4.94**	1.88	12.84	10500	3988	116.48	40.33
35	nug20	2570	2508	2680	**6.63**	2.44	16.90	11000	2386	205.48	40.16
36	rou12	235528	235528	235528	**0**	0	0	4100	34200	10.27	70.42
37	rou15	354210	350218	360702	**2.95**	1.13	4.89	2596	3946	13.98	19.57
38	rou20	725522	695182	764912	**9.55**	4.27	14.93	5450	1538	96.64	26.80
39	scr12	31410	31410	31410	**0**	0	19.38	300	4268	0.74	8.16
40	scr15	51140	51140	51140	**0**	0	2.67	400	5489	1.97	24.69
41	scr20	110030	106804	132330	**21.35**	2.98	33.40	29200	9705	503.69	158.47
42	tai10a	135028	135028	135028	**0**	0	0.01	1300	21400	2.05	20.95
43	tai12a	224416	224416	224416	**0**	0	0	300	4300	0.62	7.74
44	tai15a	388214	377102	409004	**8.12**	2.90	9.03	2290	2245	12.45	11.17
45	tai17a	491812	476526	534328	**11.44**	3.16	16.25	3897	1399	33.83	11.62
46	tai20a	703482	671676	753334	**11.46**	4.63	19.03	3872	999	70.05	17.62

Table 5.4.1: QAPLIB instances of small size

We have found provably optimal solutions for the instances

chr12b chr12c chr15a chr15b chr15c chr18a chr20a chr20b esc16e esc16f esc16g
esc16j had12 had14 had16 had18 had20 rou12 scr12 scr15 tai10a tai12a.

We also observe from columns **iter** and **iter$^{\mathbf{A}}$** in Table 5.4.1 that **rPRSM** gives reduction in number of iterations in many instances; 38 out of 46 instances use fewer or equal number of iterations using **rPRSM** compared to **ADMM**. For **rPRSM** alone we observe that most of the instances show good bounds with reasonable amount of time. Most of the instances are solved within a minute using the machine described above.

5.5 Implicit Problem Singularities of SCP and QAP

In this section we compute the three different singularity values and the strengthened Barvinok-Pataki bound discussed in Chapter 3 for the **SDP** relaxations of the **SCP** problem and **QAP**. We aim to show that the singularity degree, max-singularity degree and the implicit problem singularity can take different values. The implicit redundant constraints in the **SDP** relaxation of the **QAP** are first observed in the 90's. The redundant constraints are found after finding an appropriate facial range vector and they appear to be realized as a special property embedded in this particular class of problem, rather than a property holds for an arbitrary spectrahedron.

We first realize the linear constraints of (5.3.3) as a set of constraints with the standard inner product. We recall the partitions used for the block representation for the lifted matrix Y in (5.1.13). There are four types of constraints involved. We group the constraint data matrices with the notation $A_i^{\text{type}\,1}$, $A_i^{\text{type}\,2}$, $A_i^{\text{type}\,3}$ and $A_i^{\text{type}\,4}$.

1. Type 1: The $(0,0)$-th entry is equal to 1. This can be imposed by the equality $Y_{00} = E_{00} \bullet Y = 1$, i.e., $A_1^{\text{type}\,1} = E_{00}$.
2. Type 2: The off-diagonal elements of the diagonal blocks are 0. These constraints can be imposed by the equalities $A_i^{\text{type}\,2} \bullet Y = 0$, for $i = 1, \ldots, \sum_{j=1}^{p} t(m_j - 1)$, where $\{A_i^{\text{type}\,2}\}_i$ is a set of matrices
$$e_k e_\ell^T + e_\ell e_k^T, \text{ for } (k,\ell) \in \operatorname{supp}(\operatorname{BlkDiag}(0, A_u^T A_u - I)) \text{ and } k < \ell.$$
3. Type 3: The trace of the diagonal blocks are equal to 1. These constraints can be imposed by the equalities $A_i^{\text{type}\,3} \bullet Y = 1$ for $i = 1, \ldots, p$, where
$$A_i^{\text{type}\,3} = \begin{bmatrix} 0 & 0_{n_0}^T \\ 0_{n_0} & \operatorname{BlkDiag}(0, \ldots, I_{m_i}, \ldots, 0) \end{bmatrix}.$$
4. Type 4: The 0-th row and column are equal to the diagonal. These constraints can be realized by the equalities $A_i^{\text{type}\,4} \bullet Y = 0$, for $i = 1, \ldots, n_0$, where
$$A_i^{\text{type}\,4} = 2e_i e_i^T - e_i e_0^T - e_0 e_i^T.$$

Thus we have the set of equalities that are represented in the standard trace inner product form $(\mathcal{A}(Y))_i = \langle A_i, Y \rangle = b_i, \ \forall i$.

We show that the singularity degree of the **SCP** is exactly one. We use the matrices $A^{\text{type}\,\ell}$, $\ell = 1, 2, 3, 4$, above to form the exposing vector defined in (5.1.10):
$$K_{\mathbf{SCP}} := \begin{bmatrix} -e & A_u \end{bmatrix}^T \begin{bmatrix} -e & A_u \end{bmatrix} = \operatorname{BlkDiag}(p, I_{m_1}, \ldots, I_{m_p}) - e_0 \bar{e}_{n_0+1}^T - \bar{e}_{n_0+1} e_0^T.$$

We verify that $\bar{y} = \begin{pmatrix} p & \bar{e}_{\sum_{j=1}^{p} t(m_j-1)} & -\bar{e}_p & \bar{e}_{n_0} \end{pmatrix}^T \in \mathbb{R}^{1+\sum_{j=1}^{p} t(m_j-1)+p+n_0}$ can be used to form $K_{\mathbf{SCP}}$ and satisfy the auxiliary system (2.3.4):
$$K_{\mathbf{SCP}} = \mathcal{A}^*(\bar{y}) = A_1^{\text{type}\,1} \cdot (p) + \sum_{i=1}^{\sum_{j=1}^{p} t(m_j-1)} A_i^{\text{type}\,2} \cdot (1) + \sum_{i=1}^{p} A_i^{\text{type}\,3} \cdot (-1) + \sum_{i=1}^{n_0} A_i^{\text{type}\,4} \cdot (1)$$

and

$$b^T\bar{y} = 1\cdot(p) + \sum_{i=1}^{\sum_{j=1}^{p} t(m_j-1)} 0\cdot(1) + \sum_{i=1}^{p} 1\cdot(-1) + \sum_{i=1}^{n_0} 0\cdot(1) = p + 0 - p + 0 = 0.$$

Hence, the singularity degree of the **SDP** relaxation of the **SCP** is 1. We recall, from Theorem 5.1.1, that the type 3 and the type 4 constraints are redundant. Therefore, the implicit problem singularity of the **SCP** is $p+n_0$.

We let $\mathcal{F} \subseteq \mathbb{S}_+^{n_0+1}$ be the feasible region of **SCP** problem. We recall that $\text{face}(\mathcal{F}, \mathbb{S}_+^{n_0+1}) \subseteq V\mathbb{S}_+^{n_0+1-p}V^T$, where V is a minimal facial range vector for $\mathcal{F}$. Hence, the maximum **FR** steps, **maxsd**, is at most p. We show that $\textbf{maxsd}(\mathcal{F})$ is lower bounded by p, and consequently implies that $\textbf{maxsd}(\mathcal{F}) = p$. We can decompose the exposing vector $K_{\mathbf{SCP}}$ as follows. We decompose A_u into

$$A_u = \sum_{i=1}^{p} A_u^i, \quad \text{where} \quad A_u^i = \text{BlkDiag}(0, \cdots, \underbrace{\bar{e}_{m_i}}_{i-\text{th block}}, \cdots, 0), \ \forall i \in [p].$$

Then, and we obtain

$$K_{\mathbf{SCP}} = \begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix}^T \begin{bmatrix} -\bar{e}_p & A_u \end{bmatrix} = \sum_{i=1}^{p} \begin{bmatrix} -e_i & A_u^i \end{bmatrix}^T \begin{bmatrix} -e_i & A_u^i \end{bmatrix}.$$

For each $i = 1, \ldots, p$, we note that, $K_i := \begin{bmatrix} -e_i & A_u^i \end{bmatrix}^T \begin{bmatrix} -e_i & A_u^i \end{bmatrix}$ serves as an exposing vector. We can choose appropriate coefficient vector $\bar{y}$ to form each K_i[11]. Therefore, $\textbf{maxsd}(\mathcal{F}) = p$.

Without considering the **FR**, the usual Barvinok-Pataki bound, Theorem 3.2.5, gives the bound on the rank r for which

$$t(r) \le \sum_{j=1}^{p} t(m_j - 1) + p + n_0 + 1.$$

The implicit problem singularity for **SCP** gives a much tighter bound on the rank r

$$t(r) \le \sum_{j=1}^{p} t(m_j - 1) + 1.$$

A similar argument follows for the **QAP**. The **QAP** also has the four types of constraints used for the **SCP** problem above. The **SDP** relaxation for the **QAP** contains additional type of constraints that sets the diagonal elements of the off-diagonal blocks to be 0. This gives rise to $n \cdot t(n-1)$ number of equality constraints. We call group these as the type 5 constraints and they can be represented using the set of matrices $\{A_i^{\text{type 5}}\}_i$ of the form

$$\text{BlkDiag}\left(0, (e_k e_\ell^T + e_k e_\ell^T) \otimes I_n\right), \text{ for } k, \ell \in [n], \ k < \ell.$$

We again find a vector $\bar{y}$ that forms the exposing vector $K_{\mathbf{QAP}}$ given in (5.4.4). With

$$\bar{y} = \begin{pmatrix} 2n & \bar{e}_{n\cdot t(n-1)} & -2\bar{e}_n & 2\bar{e}_{n^2} & \bar{e}_{n\cdot t(n-1)} \end{pmatrix}^T,$$

[11] For instance, in order to form K_1, we can choose the coefficients $\bar{y} = [\bar{y}^1; \bar{y}^2; \bar{y}^3; \bar{y}^4]$ with $\bar{y}^1 = 1$, $\bar{y}^2 = [\bar{e}_{t(m_1)}; 0_{\sum_{j=2}^{p} t(m_j-1)}]$, $\bar{y}^3 = e_1 \in \mathbb{R}^p$ and $\bar{y}^4 = [\bar{e}_{m_1}; 0_{\sum_{j=2}^{p} m_j}]$.

we obtain

$$K_{\mathbf{QAP}} = A_1^{\text{type 1}} \cdot (2n) + \sum_{i=1}^{n \cdot t(n-1)} A_i^{\text{type 2}} \cdot (1) + \sum_{i=1}^{n} A_i^{\text{type 3}} \cdot (-2) + \sum_{i=1}^{n^2} A_i^{\text{type 4}} \cdot (2) + \sum_{i=1}^{n \cdot t(n-1)} A_i^{\text{type 5}} \cdot (1)$$

and

$$b^T \bar{y} = 1 \cdot (2n) + \sum_{i=1}^{n \cdot t(n-1)} 0 \cdot (1) + \sum_{i=1}^{n} 1 \cdot (-2) + \sum_{i=1}^{n^2} 0 \cdot (-2) + \sum_{i=1}^{n \cdot t(n-1)} 0 \cdot (1) = 2n + 0 - 2n + 0 + 0 = 0.$$

Thus, the singularity degree of the **SDP** relaxation of the **QAP** is equal to 1; see also [50]. Since the type 3 and 4 constraints become redundant after **FR**, the implicit problem singularity for the **QAP** is $n + n^2$. The size of the minimal facial range vector V is of the size $(n^2 + 1)$-by-$((n-1)^2 + 1)$. Hence, the **maxsd** $\leq (n^2 + 1) - [(n-1)^2 + 1] = 2n - 1$. We can adopt the same approach given in the **SCP** to obtain the partial exposing vectors, and we obtain **maxsd** $\geq n$. We also note that the original Barvinok-Pataki bound, Theorem 3.2.5, gives the rank bound $t(r) \leq 1 + 2n \cdot t(n-1) + n + n^2 = n^3 + n + 1$ and the strengthened bounds gives the improved bound $t(r) \leq n^3 - n^2 + 1$.

Chapter 6

Gauss-Newton Interior Point Method for Solving SDP with a Nonlinear Objective

Optimization over $\mathbb{H}^n_+$ is essential in the area of quantum information theory. Many problems can be posed as an optimization problem with the objective functions being the fidelity function, the nuclear norm or the (relative) entropy functions; see e.g., [70, 104, 112, 155]. The arguments in the quantum information theory are governed by the object called *density matrix*, a Hermitian positive semidefinite matrix of the unit trace. Therefore, the **SDP** naturally takes place when these problems are posed as an optimization problem. Furthermore, the **FR** plays a critical role with the growing need for developing stable numerical methods in the quantum information theory. The variables in these applications are defined over complex vector spaces and a direct application of optimization algorithms developed over $\mathbb{S}^n_+$ is not available.

In this chapter, we propose the Gauss-Newton interior-point method framework for **SDP**s over the Hermitian matrices with nonlinear objective functions. We then apply our proposed framework to the real-world application that arises in the key rate computation for quantum key distribution, **QKD**. The main goal of this application is to obtain a reliable lower bound to the optimal value. However, the feasible region may fail to contain strictly feasible points resulting in numerical instability. Moreover, the non-smoothness of the objective function adds difficulties. We show that the **FR** technique serves perfectly for avoiding these difficulties and hence shows a great promise for the growing need for stability in the area.

Contributions and Outline The contribution of this chapter is twofold.

1. We extend the existing Gauss-Newton framework [99, 100] to solve a nonlinear **SDP** over the Hermitian matrices.
2. We introduce a successful application of **FR** to the quantum key rate computation for quantum key distribution.

This chapter is organized as follows. In Section 6.1 we introduce the general Gauss-Newton framework for solving a **SDP** with a smooth nonlinear objective function over $\mathbb{H}^n_+$. In Section 6.2

we introduce the properties of the particular application of interest, computing the key rate for quantum key distribution. We explain the problem data and its related properties. In Section 6.3 we discuss our reformulation process via **FR** by using the properties of the model. The outcome of the reformulations are: (i) the constraint set satisfies the **MFCQ**; (ii) the objective function is differentiable over the interior of the positive semidefinite cone.

6.1 Gauss-Newton Method for Minimizing Nonlinear Objective over Spectrahedron in $\mathbb{H}^n_+$

We recall the Gauss-Newton framework for solving the linear **SDP** over a spectrahedron in $\mathbb{S}^n$ introduced in Section 2.4.2. We extend this framework to problems with nonlinear objective function over a spectrahedron in $\mathbb{H}^n$. Let $f : \mathbb{H}^n \to \mathbb{R} \cup \{+\infty\}$ be a continuously differentiable convex function. For a given set of matrices $\{\Gamma_i\}_{i=1}^m$, we let $\Gamma : \mathbb{H}^n \to \mathbb{R}^m$ be the mapping

$$(\Gamma(\rho))_i = \langle \Gamma_i, \rho \rangle = \text{trace}(\Gamma_i \rho)^{1}.$$

Let $\gamma \in \mathbb{R}^m$. We focus on the feasible model

$$\begin{array}{rl} \min\limits_{\rho} & f(\rho) \\ \text{subject to} & \Gamma(\rho) = \gamma \\ & \rho \in \mathbb{H}^n_+. \end{array} \tag{6.1.1}$$

In this section, we assume that the constraint system satisfies the **MFCQ** (Definition 2.3.3).

6.1.1 Optimality Conditions and Gauss-Newton Direction

The perturbed optimality conditions in (2.4.4) naturally extend to the model (6.1.1):

$$\begin{array}{lll} \text{dual feasibility} & F^d_\mu(\rho, y, Z) & := \nabla f(\rho) + \Gamma^*(y) - Z = 0 \\ \text{primal feasibility} & F^p_\mu(\rho) & := \Gamma(\rho) - \gamma = 0 \\ \text{perturbed complementary slackness} & F^c_\mu(\rho, Z) & := Z\rho - \mu I = 0, \ \rho, Z \succ 0. \end{array} \tag{6.1.2}$$

The dual feasibility condition of the *linear* **SDP** would render $\nabla f(\rho)$ a constant matrix. We note that $\nabla f(\rho)$ does not necessarily remain as a constant term when f is not linear. For each $\mu > 0$, we put the individual optimality conditions together and rewrite them as

$$F_\mu(\rho, y, Z) = \begin{bmatrix} \nabla f(\rho) + \Gamma^*(y) - Z \\ \Gamma(\rho) - \gamma \\ Z\rho - \mu I \end{bmatrix} = 0, \ \rho, Z \succ 0. \tag{6.1.3}$$

It is important to notice that the domain and the range of the equality system (6.1.3) are different. This is a subtle difference that distinguishes **SDP** from **LP**. We note that the system (6.1.3) is overdetermined since

$$F_\mu : \mathbb{H}^n \times \mathbb{R}^m \times \mathbb{H}^n \to \mathbb{H}^n \times \mathbb{R}^m \times \mathbb{C}^{n\times n}. \tag{6.1.4}$$

[1] We use the symbols Γ and ρ rather than $\mathcal{A}$ and X in order to emphasize that we work with problems over $\mathbb{H}^n$.

That the system (6.1.3) is overdetermined stems from the fact that the product of two Hermitian matrices is not Hermitian in general. That is, $Z \in \mathbb{H}^n$ and $\rho \in \mathbb{H}^n$ do not necessarily imply $Z\rho \in \mathbb{H}^n$, resulting in $Z\rho - \mu I \in \mathbb{C}^{n\times n}$ in (6.1.4). Many approaches overcome this issue by using the symmetrized similarity transformation, e.g., see [2,121,165].

We define the linear maps

$$\begin{array}{lll} \mathcal{M}_Z : \mathbb{H}^n \to \mathbb{C}^{n\times n} & \text{by} & \mathcal{M}_Z(\Delta X) = Z(\Delta X), \\ \mathcal{M}_\rho \ : \mathbb{H}^n \to \mathbb{C}^{n\times n} & \text{by} & \mathcal{M}_\rho(\Delta X) = (\Delta X)\rho. \end{array}$$

Then the linearization of (6.1.3) gives

$$F'_\mu d_{GN} = \begin{bmatrix} \nabla^2 f(\rho)\Delta\rho + \Gamma^*(\Delta y) - \Delta Z \\ \Gamma(\Delta\rho) \\ Z\Delta\rho + \Delta Z\rho \end{bmatrix} = \begin{bmatrix} \nabla^2 f(\rho) & \Gamma^* & -I \\ \Gamma & 0 & 0 \\ \mathcal{M}_Z & 0 & \mathcal{M}_\rho \end{bmatrix} \begin{pmatrix} \Delta\rho \\ \Delta y \\ \Delta Z \end{pmatrix} \approx -F_\mu. \tag{6.1.5}$$

We obtain the *Gauss-Newton direction,* ***GN*** *direction*, $d_{GN} \in \mathbb{H}^n \times \mathbb{R}^m \times \mathbb{H}^n$ by solving the system

$$(F'_\mu)^* F'_\mu d_{GN} = -(F'_\mu)^* F_\mu, \tag{6.1.6}$$

where the adjoint $(F'_\mu)^*$ is

$$(F'_\mu)^* = \begin{bmatrix} \nabla^2 f(\rho) & \Gamma^* & \mathcal{M}_Z^* \\ \Gamma & 0 & 0 \\ -I & 0 & \mathcal{M}_\rho^* \end{bmatrix}.$$

The **GN** direction, d_{GN}, requires solving the linear system (6.1.6) of the number of unknowns $2n^2 + m$. The adjoint $(F'_\mu)^*$ contains the adjoints $\mathcal{M}_Z^*$ and $\mathcal{M}_\rho^*$. We provide the expressions for $\mathcal{M}_Z^*$ and $\mathcal{M}_\rho^*$ for those interested in solving (6.1.6) directly; see Lemmas B.1.1 and B.1.2 in Appendix B.1. In Section 6.1.2 below, we strive to reduce the size of linear system by making block variable eliminations.

6.1.2 Projected Gauss-Newton Method

The main computational step of interior point methods boils down to the search direction computation. The search directions are obtained by solving the linear system that stem from the optimality conditions. Many practical interior point methods use block variable eliminations to reduce the size of the linear system for computational efficiency, e.g., see [12, Chapter 3], [165]. We also adopt a similar approach for computing the **GN** direction more efficiently.

We note that the system (6.1.5) has zero blocks and ΔZ has a closed form representation

$$\Delta Z = F^d_\mu + \nabla^2 f(\rho)\Delta\rho + \Gamma^*(\Delta y). \tag{6.1.7}$$

We use (6.1.7) to substitute the variable ΔZ that appear in (6.1.5). Then we have

$$\begin{aligned}
(F_\mu^{p,c})' \begin{pmatrix} \Delta\rho \\ \Delta y \end{pmatrix} &= \begin{bmatrix} \Gamma(\Delta\rho) \\ Z\Delta\rho + \left(\nabla^2 f(\rho)\Delta\rho + \Gamma^*(\Delta y)\right)\rho \end{bmatrix} \\
&= \begin{bmatrix} \Gamma & \\ \mathcal{M}_Z + \mathcal{M}_\rho \nabla^2 f(\rho) & \mathcal{M}_\rho \Gamma^* \end{bmatrix} \begin{pmatrix} \Delta\rho \\ \Delta y \end{pmatrix} \\
&\approx -\begin{bmatrix} F_\mu^p \\ F_\mu^c + F_\mu^d \rho \end{bmatrix}.
\end{aligned} \tag{6.1.8}$$

Here, the superscripts p, c in $(F_\mu^{p,c})'$ indicate the primal feasibility and complementarity equations. The reduced system (6.1.8) now has $n^2 + m$ number of unknowns rather than $2n^2 + m$.

We continue to reduce the size of the system by representing the equality $\Gamma(\rho) = \gamma$ using a null-space representation. Let $\{N_i\}_{i=1}^{n^2-m} \subseteq \mathbb{H}^n$ be the set of basis elements for $(\{\Gamma_i\}_{i=1}^m)^\perp$. Define the linear map $\mathcal{N}^* : \mathbb{R}^{n^2-m} \to \mathbb{H}^n$ by

$$\mathcal{N}^*(v) = \sum_{i=1}^{n^2-m} v_i N_i. \tag{6.1.9}$$

Let $\hat{\rho}$ be a particular solution to $\Gamma(\rho) = \gamma$. Then we have

$$\{\rho \in \mathbb{H}^n : \Gamma(\rho) = \gamma\} = \left\{\rho = \mathcal{N}^* v + \hat{\rho} \in \mathbb{H}^n : v \in \mathbb{R}^{n^2-m}\right\}. \tag{6.1.10}$$

With the new representation for the primal feasibility, we can effectively isolate the primal variable ρ and it leads us to a smaller system to solve.

Theorem 6.1.1. *The second projected* ***GN*** *direction* $d_{GN} = \begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix} \in \mathbb{R}^{n^2}$ *can be fuond from the least squares solution of*

$$\left[Z\mathcal{N}^*(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^*(\Delta v)\rho\right] + \left[\Gamma^*(\Delta y)\rho\right] = -F_\mu^c - ZF_\mu^p - (F_\mu^d + \nabla^2 f(\rho)F_\mu^p)\rho. \tag{6.1.11}$$

Proof. By abuse of notation, we use F_μ^p to denote the equivalent primal feasibility equation from (6.1.10):

$$F_\mu^p(\rho, v) = \mathcal{N}^* v + \hat{\rho} - \rho.$$

Then the perturbed optimality conditions in (6.1.3) can be written

$$F_\mu(\rho, v, y, Z) = \begin{bmatrix} F_\mu^d \\ F_\mu^p \\ F_\mu^c \end{bmatrix} = \begin{bmatrix} \nabla f(\rho) + \Gamma^*(y) - Z \\ \mathcal{N}^*(v) + \hat{\rho} - \rho \\ Z\rho - \mu I \end{bmatrix} = 0, \quad \rho, Z \succ 0. \tag{6.1.12}$$

Linearizaing the system (6.1.12) at (ρ, v, y, Z) gives

$$F_\mu' d_{GN} = \begin{bmatrix} \nabla^2 f(\rho)\Delta\rho + \Gamma^*(\Delta y) - \Delta Z \\ \mathcal{N}^*(\Delta v) - \Delta\rho \\ Z(\Delta\rho) + (\Delta Z)\rho \end{bmatrix} \approx -F_\mu = -\begin{pmatrix} F_\mu^d \\ F_\mu^p \\ F_\mu^c \end{pmatrix}. \tag{6.1.13}$$

The second block equation of (6.1.13) gives

$$\Delta\rho = \mathcal{N}^*(\Delta v) + F_\mu^p. \tag{6.1.14}$$

The first block equation of (6.1.13) gives

$$\Delta Z = F_\mu^d + \nabla^2 f(\rho)\Delta\rho + \Gamma^*(\Delta y) = F_\mu^d + \nabla^2 f(\rho)\left(\mathcal{N}^*(\Delta v) + F_\mu^p\right) + \Gamma^*(\Delta y). \tag{6.1.15}$$

We then substitute $\Delta\rho$ and ΔZ into the last block equation of (6.1.13):

$$\begin{aligned} -F_\mu^c &= Z(\Delta\rho) + (\Delta Z)\rho \\ &= Z\left(\mathcal{N}^*(\Delta v) + F_\mu^p\right) + \left(F_\mu^d + \nabla^2 f(\rho)\left(\mathcal{N}^*(\Delta v) + F_\mu^p\right) + \Gamma^*(\Delta y)\right)\rho \\ &= \left[Z\mathcal{N}^*(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^*(\Delta v)\rho\right] + \left[\Gamma^*(\Delta y)\rho\right] + \left[ZF_\mu^p + \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho\right]. \end{aligned}$$

Rearranging the constant terms, we obtain (6.1.11). □

Once we compute d_{GN} from (6.1.11), we obtain the original search direction with the backsolve steps to complete the remaining components; for recovering $\Delta\rho$ we use (6.1.14); for recovering ΔZ we use (6.1.15) and $(\Delta v, \Delta y)$. The steps for obtaining $(\Delta\rho, \Delta y, \Delta Z)$ are summarized in lines 4, 5 and 6 of Algorithm 6.1.1.

Algorithm 6.1.1 Projected Gauss-Newton Interior Point Algorithm

```
1: Initialize: $\hat{\rho} \succ 0$, $\mu \in \mathbb{R}_{++}$, $\eta \in (0,1)$, problem data for (6.1.1)
2: while stopping criteria is not met do
3:    Obtain search direction
4:          solve system (6.1.11) for $(\Delta v, \Delta y)$
5:          backsolve (6.1.14) : $\Delta\rho = \mathcal{N}^*(\Delta v) + F_\mu^p$
6:          backsolve (6.1.15) : $\Delta Z = F_\mu^d + \nabla^2 f(\rho)(F_\mu^p + \mathcal{N}^*(\Delta v)) + \Gamma^*(\Delta y)$
7:    Update iterate
8:          choose steplength $\alpha$
9:          $(\rho, y, Z) \leftarrow (\rho, y, Z) + \alpha(\Delta\rho, \Delta y, \Delta Z)$
10:         $\mu \leftarrow \langle \rho, Z \rangle / n$; $\mu \leftarrow \eta\mu$
11: end while
```

The block variable eliminations result in the reduction in the number of unknowns for solving linear systems and we summarize the reduction below:

system	(6.1.5)	(6.1.8)	(6.1.11)
# of unknowns	$2n^2 + m$	$n^2 + m$	n^2

We emphasize that the backsolve steps, (6.1.14) and (6.1.15), are stable. The stopping criteria, solving (6.1.11), choosing the steplength in Algorithm 6.1.1 are elaborated in Section 6.1.3 below.

6.1.3 Implementation Heuristics and Details

In this section we discuss some implementation details for Algorithm 6.1.1.

Step Lengths The step length α, in Line 8 of Algorithm 6.1.1, is chosen to maintain the variables ρ and Z sufficiently positive definite. We achieve this by line search with backtracking. We can build a quadratic model of f at the current iterate ρ_c economically since we have the gradient and Hessian evaluated at ρ_c in the search direction $\Delta\rho_c$:

$$q(\alpha) := f(\rho_c) + \alpha\langle\nabla f(\rho_c), \Delta\rho_c\rangle + \frac{1}{2}\alpha^2\langle\Delta\rho_c, \nabla^2 f(\rho_c)\Delta\rho_c\rangle.$$

Then $\alpha^* = \operatorname{argmin}_\alpha q(\alpha)$ gives

$$\alpha^* = -\langle\nabla f(\rho_c), \Delta\rho_c\rangle \; / \langle\Delta\rho_c, \nabla^2 f(\rho_c)\Delta\rho_c\rangle.$$

We note that α^* is positive since **GN** direction is a descent direction (see (2.4.6)) and the objective function is convex. In our implementation, we start backtracking from α^* with the backtracking parameter 0.97.

For the iteration where a full step, i.e., $\alpha = 1$, is taken, subsequent iterations maintain the exact primal feasibility (see also [159, page 12].).

Theorem 6.1.2. *Let α be a steplength and consider the update*

$$\rho_+ \leftarrow \rho + \alpha\Delta\rho = \rho + F_\mu^p + \alpha\mathcal{N}^*(\Delta v).$$

1. *If a steplength one is taken ($\alpha = 1$), then the new primal residual is exact, i.e.,*

$$F_\mu^p = \mathcal{N}^*(v_+) + \hat{\rho} - \rho_+ = 0, \text{ where } v_+ = v + \alpha\Delta v.$$

2. *Suppose that the exact primal feasibility is achieved. Then the primal residual is exact throughout the iterations regardless of the steplength.*

Proof. Suppose that the steplength $\alpha = 1$ is taken. Then the new primal residual $(F_\mu^d)_+$ is

$$\begin{array}{rcl} (F_\mu^d)_+ & = & \mathcal{N}^*(v_+) + \hat{\rho} - \rho_+ \\ & = & \mathcal{N}^*(v + \Delta v) + \hat{\rho} - \rho - F_\mu^p - \mathcal{N}^*(\Delta v) \\ & = & \mathcal{N}^*(v) + \hat{\rho} - \rho - F_\mu^p \\ & = & \mathcal{N}^*(v) + \hat{\rho} - \rho - \mathcal{N}^*(v) - \hat{\rho} + \rho \\ & = & 0. \end{array}$$

This shows Item 1. For Item 2, suppose that we reached the exact primal feasibility, i.e., $F_\mu^p = 0$. Then

$$\rho_+ \leftarrow \rho + \alpha\Delta\rho = \rho + \alpha\left(F_\mu^p + \mathcal{N}^*(\Delta v)\right) = \rho + \alpha\mathcal{N}^*(\Delta v) = 0,$$

where the last equality holds since

$$\Gamma(\rho_+) = \Gamma_V(\rho + \alpha\mathcal{N}^*(\Delta v)) = \Gamma(\rho) = \gamma.$$

□

Stopping Criteria Let $\epsilon > 0$ be a pre-defined tolerance. We terminate the algorithm when the optimality conditions are approximately satisfied within the given tolerance. If the algorithm computes lower and upper bounds of the optimal value throughout its execution, we may terminate the algorithm when the gap between the best known lower and upper bounds is within ϵ. We denote the residual of the right-hand-side in (6.1.11) by

$$\phi = -F_\mu^c - ZF_\mu^p - (F_\mu^d + \nabla^2 f(\rho)F_\mu^p)\rho.$$

We define the denominator term by

$$\text{denom} = 1 + \frac{1}{2}\min\left\{\|\rho\|_F + \|Z\|_F, |\text{bestub}| + |\text{bestlb}|\right\},$$

where bestub and bestlb denote the best upper and the lower bounds to the optimal value. Then, for the tolerance ϵ, we terminate the algorithm when

$$\frac{1}{\text{denom}}\max\left\{\text{bestub} - \text{bestlb}, \|\phi\|\right\} < \epsilon.$$

Finally a common way to terminate an algorithm is to impose restrictions on the running time, i.e., setting an upper bound on the number of iterations.

Sparse Nullspace Representation for $\mathcal{N}^*$ in (6.1.9) With Hvec defined in Section 2.1, we let $Mr = \gamma$ be the matrix-vector representation of the system $\Gamma(\rho) = \gamma$, i.e.,

$$r = \text{Hvec}(\rho),\ M(i,:) = \text{Hvec}(\Gamma_i)^T,\ \forall i \in [m].$$

We permute the columns of M to obtain a matrix $\hat{M}$ so that $\hat{M}$ has a well-conditioned nonsingular matrix B at the first m columns. We place the remaining columns of M from the $(m+1)$-th column of $\hat{M}$, i.e., $\hat{M} = \begin{bmatrix} B & E \end{bmatrix}$. Let P be a permutation matrix that permutes the columns of M to form $\hat{M}$, i.e., $\hat{M} = MP$. Following the approach in [77], we construct the matrix $\hat{N} = \begin{bmatrix} B^{-1}E \\ -I \end{bmatrix}$. Then each column of $\hat{N}$ is a basis element of null$(\hat{M})$ since

$$\hat{M}\hat{N} = \begin{bmatrix} B & E \end{bmatrix}\begin{bmatrix} B^{-1}E \\ -I \end{bmatrix} = B \cdot B^{-1}E - E \cdot I = 0.$$

We then choose

$$N_i = \text{HMat}(\hat{M}(i,:)P^T),\ i = 1, \ldots, n^2 - m,$$

to be the data matrices for the mapping $\mathcal{N}^*$ in (6.1.9).

Matrix Representation of the System (6.1.11) Solving the linear system (6.1.11) using a standard software requires the conventional matrix-vector representation, i.e., $Ax = b$. We now provide the matrix representation for the system (6.1.11). We begin by rearranging the left-hand-side of the system (6.1.11):

$$\begin{array}{rl} & \left[Z\mathcal{N}^*(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^*(\Delta v)\rho\right] + \left[\Gamma^*(\Delta y)\rho\right] \\ = & \left[Z\left(\sum\limits_i \Delta v_i \cdot N_i\right) + \nabla^2 f(\rho)\left(\sum\limits_i \Delta v_i \cdot N_i\right)\rho\right] + \left[\left(\sum\limits_i \Delta y_i \cdot \Gamma_i\right)\rho\right] \\ = & \sum\limits_i \Delta v_i \cdot \left(ZN_i + \nabla^2 f(\rho)N_i\rho\right) + \sum\limits_i \Delta y_i \cdot \Gamma_i\rho. \end{array}$$

Using Cvec (see Section 2.1.) to the terms related to Δv, we have the following matrix representation:

$$\begin{bmatrix} \mathrm{Cvec}(ZN_1 + \nabla^2 f(\rho)N_1\rho) & \cdots & \mathrm{Cvec}(ZN_{n^2-m} + \nabla^2 f(\rho)N_{n^2-m}\rho) \end{bmatrix} \begin{pmatrix} \Delta v_1 \\ \vdots \\ \Delta v_{n^2-m} \end{pmatrix}.$$

Similarly, using Cvec to the terms related to Δy, we get

$$\begin{bmatrix} \mathrm{Cvec}(\Gamma_1\rho) & \cdots & \mathrm{Cvec}(\Gamma_m\rho) \end{bmatrix} \begin{pmatrix} \Delta y_1 \\ \vdots \\ \Delta y_m \end{pmatrix}.$$

Then the projected **GN** direction is obtained by

$$\begin{aligned} & \begin{bmatrix} \left[\mathrm{Cvec}(ZN_i + \nabla^2 f(\rho)N_i\rho)\right]_{i=1,\dots,n^2-m} & \left[\mathrm{Cvec}(\Gamma_i\rho)\right]_{i=1,\dots,m} \end{bmatrix} \begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix} \\ = & \ -\mathrm{Cvec}\left(F_\mu^c + ZF_\mu^p + (F_\mu^d + \nabla^2 f(\rho)F_\mu^p)\rho\right). \end{aligned} \tag{6.1.16}$$

That the linearized system is over-determined yields the system (6.1.16) with the n^2 number of unknowns and the $2n^2$ number of equations.

Preconditioning In the course of solving the system (6.1.16), we use a diagonal preconditioning. We use $Jx = b$ to denote the linear system (6.1.16). For each i, we let $d_i = \|Je_i\|_2$, where e_i is the i-th column of the identity matrix. We then scale the columns of J as $J\,\mathrm{Diag}(d)^{-1}$.[2] We note that this preconditioning does not require an excessive computational resource. It is known that this column scaling provides the optimal Ω-condition number; see [44, Proposition 2.1].

6.2 Application to Key Rate Computation for QKD

Quantum key distribution, ***QKD*** is the art of distributing a shared secret between two parties (traditionally known as Alice and Bob) over a public channel, e.g., see [122, Section 12.6.3]. Upon the termination of a **QKD** protocol, the established secret can be used for a secure communication between two parties. **QKD** is a quantum-resistant key establishment protocol, i.e., it is secure even after the quantum computers become available. In the course of a particular **QKD**, a third party (traditionally known as Eve) makes an appearance as an eavesdropper. The core of a security proof of **QKD** protocol is to calculate the secret key rate, the secret key bits obtained per exchange of a quantum signal. An analytic computation for the key rate is a challenging task. Fortunately, it has been shown that the key rate calculation can be posed as a convex optimization problem; see [156]. A tight provable lower bound to this problem provides a tight reliable key rate. Hence, we resort to find a provable lower bound to this convex optimization problem numerically.

The variables in the quantum information theory are governed by positive semidefinite matrices. Hence, the problem naturally lies in the class of **SDP** over the Hermitian matrices. The convex

[2] In MATLAB command, this can be done $(J/\,\mathrm{Diag}(d))\backslash b)\,./d$.

optimization formulation for computing the key rate for **QKD** [156] is

$$\min_{\rho}\{f(\rho) : \Gamma(\rho) = \gamma,\ \rho \in \mathbb{H}^n_+\}, \tag{6.2.1}$$

where $f(\rho)$ is the composition of the quantum relative entropy function and the linear maps; we explain the objective function in detail in 6.2.1 below. The feasible region of (6.2.1) is the standard spectrahedon in $\mathbb{H}^n_+$. The variable ρ is called a *density matrix*, a positive semidefinite matrix with the *unit trace* property $\operatorname{trace}(\rho) = 1$.

The model (6.2.1) may not have a strictly feasible point. Moreover, the objective function $f(\rho)$ is not assumed to be differentiable. An approach [156] is proposed for avoiding these difficulties by perturbing the variables using a small positive multiple of the identity matrix. Instead, we preprocess the model so that the reformulated model satisfies the **MFCQ**, and the objective function is differentiable at any positive definite matrix. This reformulation is greatly inspired by **FR**. We aim to solve the robust reformulated model using the stable interior point method developed in Section 6.1. We also provide an approach for computing provable lower bounds to the optimal value of the problem.

6.2.1 The Problem

In this section we introduce the problem and its related properties. Every Hermitian matrix X has the *spectral decomposition*

$$X = U\Lambda U^*, \quad \text{for some unitary matrix } U, \text{ and real diagonal matrix } \Lambda.$$

Let $\mathbb{H}^n_+$ be the set of n-by-n Hermitian positive semidefinite matrices. For $X = U\Lambda U^* \in \mathbb{H}^n_+$, the *matrix extension of the* log function is defined by

$$\log(X) = \log(U\Lambda U^*) = U \operatorname{Diag}(\log(\Lambda_{1,1}), \ldots, \log(\Lambda_{n,n}))U^*. \tag{6.2.2}$$

See [88] for additional properties of the matrix logarithms.

Definition 6.2.1. *([155, Definition 5.18]) The quantum relative entropy function $D : \mathbb{H}^n_+ \times \mathbb{H}^n_+ \to \mathbb{R}_+ \cup \{+\infty\}$ is defined by*

$$D(\sigma||\delta) = \begin{cases} \operatorname{trace}(\delta \log \delta) - \operatorname{trace}(\delta \log \sigma) & \textit{if } \operatorname{range}(\delta) \cap \operatorname{null}(\sigma) = \emptyset, \\ \infty & \textit{otherwise.} \end{cases} \tag{6.2.3}$$

Here, log is the matrix logarithm. When $\operatorname{range}(\delta) \cap \operatorname{null}(\sigma) = \emptyset$, i.e., $\operatorname{range}(\delta) \subseteq \operatorname{range}(\sigma)$ holds, the finite function value of D can be shown by using the eigenspace associated with 0 eigenvalues of δ, σ. The function value D is nonnegative, and is equal to 0 if, and only if, $\delta = \sigma$. An important property of the quantum relative entropy function follows.

Proposition 6.2.2. *([122, Section 11.3]) The quantum relative entropy function is convex. Furthermore, it is jointly convex.*

A linear map $\mathcal{T} : \mathbb{H}^n \to \mathbb{H}^k$ is called *positive*, if $\rho \in \mathbb{H}^n_+$ implies $\mathcal{T}(\rho) \in \mathbb{H}^k_+$. We now define the two positive maps $\mathcal{G}, \mathcal{Z}$ that are parts of the objective function.

Definition 6.2.3. *The linear map $\mathcal{G} : \mathbb{H}^n \to \mathbb{H}^k$ is defined as a sum of matrix products*

$$\mathcal{G}(\cdot) := \sum_{j=1}^{\ell} K_j(\cdot)K_j^*, \tag{6.2.4}$$

where $K_j \in \mathbb{C}^{k\times n}, \forall j \in [\ell]$ and $\sum\limits_{j=1}^{\ell} K_jK_j^* \preceq I_k$.

The matrix K_j often has more rows than columns, i.e., $k > n$. The linear map $\mathcal{G}$ of the from in (6.2.4) often appears in quantum physics. Each element in $\{K_j\}_{j=1}^{\ell}$ is called a *Kraus operator* and the map (6.2.4) is said to be in a *Kraus representation.* Any linear map in the form (6.2.4) has the adjoint $\mathcal{G}^*(\cdot) := \sum_j K_j^*(\cdot)K_j$.

Definition 6.2.4. *The linear map $\mathcal{Z} : \mathbb{H}^k \to \mathbb{H}^k$ is defined as a sum of matrix products*

$$\mathcal{Z}(\cdot) := \sum_{j=1}^{N} Z_j(\cdot)Z_j^*, \tag{6.2.5}$$

where $Z_j = Z_j^2 = Z_j^* \in \mathbb{H}^n, \forall j \in [N]$ and $\sum_{j=1}^{N} Z_j = I_k$.

The set $\{Z_j\}_{j=1}^{N}$ that follows the properties in Definition 6.2.4 is said to be a *spectral resolution of I*. Given the linear maps $\mathcal{G}, \mathcal{Z}$, we define the objective f in (6.2.1) as the composition of $\mathcal{G}, \mathcal{Z}$ and the quantum relative entropy function D

$$f(\rho) := D(\mathcal{Z}(\mathcal{G}(\rho))||\mathcal{G}(\rho)) = \operatorname{trace}\left(\mathcal{G}(\rho)\log\mathcal{G}(\rho) - \mathcal{G}(\rho)\log\mathcal{Z}(\mathcal{G}(\rho))\right). \tag{6.2.6}$$

The convexity of the quantum relative entropy function is preserved after the composition with the linear maps $\mathcal{G}$ and $\mathcal{Z}$. Finally, we obtain our model

$$\begin{array}{rcll} p^* = & \min\limits_{\rho} & D(\mathcal{Z}(\mathcal{G}(\rho))||\mathcal{G}(\rho)) & (= f(\rho)) \\ & \text{subject to} & \Gamma(\rho) = \gamma & \\ & & \rho \in \mathbb{H}_+^n. & \end{array} \tag{6.2.7}$$

The function f is differentiable only when $\mathcal{G}(\rho)$ is positive definite. However, even when ρ is positive definite, $\mathcal{G}(\rho)$ may fail to be positive *definite.* The gradient of f is not well-defined when the point of differentiation is on the boundary of its domain. Hence we strategize to guarantee positive definiteness of the arguments $\mathcal{Z}(\mathcal{G}(\rho)), \mathcal{G}(\rho)$ of (6.2.6) in Section 6.3 below.

6.2.2 Properties of $\mathcal{G}$ and $\mathcal{Z}$

We now discuss some interesting properties of the maps $\mathcal{G}$, $\mathcal{Z}$ and the objective function f. We note that the term $\operatorname{trace}(\mathcal{G}(\rho)\log\mathcal{Z}(\mathcal{G}(\rho)))$ in (6.2.6) has two distinct components $\mathcal{G}(\rho)$ and $\mathcal{Z}(\mathcal{G}(\rho))$. We exploit the properties of the map $\mathcal{Z}$ to rewrite this term as $\operatorname{trace}\left(\mathcal{Z}(\mathcal{G}(\rho))\log\mathcal{Z}(\mathcal{G}(\rho))\right)$. This allows for symmetric components in each term in (6.2.6). Consequently, it leads us to adopt the idea of **FR** to the objective function effectively.

Using $\sum_{j=1}^{N} Z_j = I$, we can show that the map $\mathcal{Z}$ is a *trace-preserving* map, i.e., $\operatorname{trace}(\delta) = \operatorname{trace}(\mathcal{Z}(\delta))$. However, the map $\mathcal{G}$ is not trace-preserving since $\sum_{j=1}^{\ell} K_jK_j^* \preceq I$ does not necessarily hold with equality. Not only the map $\mathcal{Z}$ is trace-preserving and but is also completely positive, which makes $\mathcal{Z}$ a *quantum channel.*

Proposition 6.2.5. *The map $\mathcal{Z}$ is a projection. Moreover, for $\delta \in \mathbb{H}^k_+$,*

$$\operatorname{trace}\left(\delta \log \mathcal{Z}(\delta)\right) = \operatorname{trace}\left(\mathcal{Z}(\delta) \log \mathcal{Z}(\delta)\right). \tag{6.2.8}$$

Proof. For any i, we have

$$IZ_i = \left(\sum_{j=1}^N Z_j\right) Z_i = Z_i Z_i + \left(\sum_{j\neq i}^N Z_j Z_i\right) \implies \left(\sum_{j\neq i}^N Z_j Z_i\right) = 0 \implies Z_i Z_j = 0, \forall j \neq i.$$

Then $\mathcal{Z} = \mathcal{Z}^2 = \mathcal{Z}^*$ and hence $\mathcal{Z}$ is a projection. Since each Z^j commutes with $\mathcal{Z}(\delta)$, it holds that Z^j commutes with $\log(\mathcal{Z}(\delta))$ by [88, Theorem 1.13]. Then by the linearity and cyclicity of the trace, (6.2.8) holds; see also [111]. □

Owing to Proposition 6.2.5, we can represent the objective (6.2.6) with the symmetric components:

$$f(\rho) := \operatorname{trace}\left(\mathcal{G}(\rho) \log \mathcal{G}(\rho) - \mathcal{Z}(\mathcal{G}(\rho)) \log \mathcal{Z}(\mathcal{G}(\rho))\right). \tag{6.2.9}$$

This new alternative representation allows for simplified **FR** steps.

Lemma 6.2.6 below shows that the objective value of the problem is finite on the feasible set.

Lemma 6.2.6. *Let $X \succeq 0$. Then* $\operatorname{range}(X) \subseteq \operatorname{range}(\mathcal{Z}(X))$.

Proof. Let X be a positive semidefinite matrix with rank r and compact spectral decomposition

$$X = \sum_{i=1}^r \lambda_i u_i u_i^*. \tag{6.2.10}$$

We only focus on the first term $\lambda_1 u_1 u_1^*$:

$$\mathcal{Z}(\lambda_1 u_1 u_1^*) = \sum_{j=1}^n Z_j (\lambda_1 u_1 u_1^*) Z_j^* = \sum_{j=1}^n \lambda_1 (Z_j u_1)(Z_j u_1)^*.$$

We note, from Item 2 of Fact 2.2.4, that

$$\begin{aligned} \operatorname{range}\left(\mathcal{Z}(\lambda_1 u_1 u_1^*)\right) &= \operatorname{range}\left(\lambda_1 (Z_1 u_1)(Z_1 u_1)^* + \lambda_1 (Z_2 u_1)(Z_2 u_1)^* + \cdots + \lambda_1 (Z_n u_1)(Z_n u_1)^*\right) \\ &= \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1). \end{aligned}$$

We also note that

$$u_1 = I u_1 = \left(\sum_{j=1}^n Z_j\right) u_1 = \sum_{j=1}^n Z_j u_1 \in \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1).$$

Hence,

$$\operatorname{range}(\lambda_1 u_1 u_1^*) = \operatorname{range}(u_1) \subseteq \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1) = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^*)).$$

We now consider the first two terms, $\lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^*$, in X in (6.2.10). Similarly,

$$\operatorname{range}(\lambda_1 u_1 u_1^*) \subseteq \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^*)) \quad \text{and} \quad \operatorname{range}(\lambda_2 u_2 u_2^*) \subseteq \operatorname{range}(\mathcal{Z}(\lambda_2 u_2 u_2^*)). \tag{6.2.11}$$

Then

$$\begin{array}{lll} \operatorname{range}(\lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^*) & = \operatorname{range}(\lambda_1 u_1 u_1^*) + \operatorname{range}(\lambda_2 u_2 u_2^*) & \text{by from Item 2 of Fact 2.2.4} \\ & \subseteq \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^*)) + \operatorname{range}(\mathcal{Z}(\lambda_2 u_2 u_2^*)) & \text{by (6.2.11)} \\ & = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^*) + \mathcal{Z}(\lambda_2 u_2 u_2^*)) & \text{by from Item 2 of Fact 2.2.4} \\ & = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^*)) & \text{by linearity of } \mathcal{Z}. \end{array}$$

This completes the proof (The induction steps are clear.). □

A consequence of Lemma 6.2.6 is that

$$\operatorname{range}(\mathcal{G}(\rho)) \subseteq \operatorname{range}(\mathcal{Z}(\mathcal{G}(\rho))) \tag{6.2.12}$$

and the positive semidefiniteness of $\mathcal{G}(\rho), \mathcal{Z}(\mathcal{G}(\rho))$ implies that they are simultaneously diagonalizable. Furthermore, this property grants the finite objective values, see Definition 6.2.1.

We now obtain the formulae for the first and the second order derivatives of f. We assume that $\mathcal{G}(\rho)$ is positive definite for now. Our reformulation process for guaranteeing $\mathcal{G}(\rho) \succ 0$ follows in Section 6.3 below. We do not need to further assume that $\mathcal{Z}(\mathcal{G}(\rho))$ is positive definite since $\mathcal{G}(\rho) \succ 0$ implies $\mathcal{Z}(\mathcal{G}(\rho)) \succ 0$ by Lemma 6.2.6.

Lemma 6.2.7. *Let $\mathcal{H} : \mathbb{H}^n \to \mathbb{H}^k$ be a positive linear map. Let $\rho \in \mathbb{H}^n_+$ be a point satisfying $\mathcal{H}(\rho) \in \mathbb{H}^k_{++}$. Define the composite function $g : \mathbb{H}^k_+ \to \mathbb{R}$ by*

$$g(\rho) = \operatorname{trace}\left(\mathcal{H}(\rho) \log \mathcal{H}(\rho)\right).$$

Then the gradient of g at ρ is

$$\nabla g(\rho) = \mathcal{H}^*\left(\log \mathcal{H}(\rho)\right) + \mathcal{H}^*(I_k),$$

and the Hessian of g at ρ acting on $\Delta\rho$ is

$$\nabla^2 g(\rho)(\Delta\rho) = \mathcal{H}^*\left(\log' \mathcal{H}(\rho)(\mathcal{H}(\Delta\rho))\right).$$

Proof. For any differentiable function h, the first order approximation of $\operatorname{trace}(h(x))$ is $\operatorname{trace}(h(x+\Delta x)) \approx \operatorname{trace}(h(x) + h'(x)\Delta x)$. Hence we obtain

$$\begin{array}{lll} & \langle \nabla g(\rho), \Delta\rho \rangle & \\ = & \operatorname{trace}\left(\frac{d}{d\rho}\left(\mathcal{H}(\rho)\log\mathcal{H}(\rho)\right)(\Delta\rho)\right) & \\ = & \operatorname{trace}\left(\frac{d}{d\rho}\left(\mathcal{H}(\rho)\right)(\Delta\rho)\log\mathcal{H}(\rho) + \mathcal{H}(\rho)\frac{d}{d\rho}\left(\log\mathcal{H}(\rho)\right)(\Delta\rho)\right) & \text{by product rule} \\ = & \left\langle \frac{d}{d\rho}\left(\mathcal{H}(\rho)\right)(\Delta\rho), \log\mathcal{H}(\rho)\right\rangle + \left\langle \mathcal{H}(\rho), \frac{d}{d\rho}\left(\log\mathcal{H}(\rho)\right)(\Delta\rho)\right\rangle & \text{by definition of trace inner product} \\ = & \langle \mathcal{H}\Delta\rho, \log\mathcal{H}(\rho)\rangle + \left\langle \left(\frac{d}{d\rho}\left(\log\mathcal{H}(\rho)\right)\right)^* \mathcal{H}(\rho), \Delta\rho \right\rangle & \text{by definition of adjoint} \\ = & \langle \Delta\rho, \mathcal{H}^*\left(\log\mathcal{H}(\rho)\right)\rangle + \left\langle \left(\frac{d}{d\rho}\left(\log\mathcal{H}(\rho)\right)\right)^* \mathcal{H}(\rho), \Delta\rho \right\rangle & \text{by definition of adjoint} \\ = & \langle \Delta\rho, \mathcal{H}^*\left(\log\mathcal{H}(\rho)\right)\rangle + \langle \mathcal{H}^*(I), \Delta\rho\rangle & \\ = & \langle \mathcal{H}^*\left(\log\mathcal{H}(\rho)\right) + \mathcal{H}^*(I), \Delta\rho\rangle & \text{by linearity.} \end{array}$$

The second last equality holds by the chain rule and the fact that the directional derivative of matrix log at δ in the direction δ is

$$\log'(\delta)(\delta) = \log'(\delta;\delta) = I.$$

Similarly, the Hessian of g at ρ acting on $\Delta\rho$ is

$$\nabla^2 g(\rho)(\Delta\rho) = \frac{d}{d\rho}\mathcal{H}^*\left(\log\mathcal{H}(\rho)\right)(\Delta\rho) = \mathcal{H}^*\frac{d}{d\rho}\left(\log\mathcal{H}(\rho)\right)(\Delta\rho) = \mathcal{H}^*\left(\log'\mathcal{H}(\rho)\right)\mathcal{H}(\Delta\rho).$$

□

With Lemma 6.2.7, we now obtain the derivatives of the objective function f in (6.2.9).

Corollary 6.2.8. *Suppose that $\rho \in \mathbb{H}^n_+$ and $\mathcal{G}(\rho) \in \mathbb{H}^k_{++}$. Then the gradient of f at ρ is*

$$\nabla f(\rho) = \mathcal{G}^*\left(\log\mathcal{G}(\rho)\right) - (\mathcal{Z}\circ\mathcal{G})^*\left(\log\mathcal{Z}\circ\mathcal{G}(\rho)\right) + \mathcal{G}^*(I) - (\mathcal{Z}\circ\mathcal{G})^*(I) \tag{6.2.13}$$

The Hessian at $\rho \in \mathbb{H}^n_+$ acting on the direction $\Delta\rho \in \mathbb{H}^n$ is

$$\nabla^2 f(\rho)(\Delta\rho) = \mathcal{G}^*\left([\log'\mathcal{G}(\rho)(\mathcal{G}\Delta\rho)]\right) - (\mathcal{Z}\circ\mathcal{G})^*\left([\log'(\mathcal{Z}\circ\mathcal{G})(\rho)((\mathcal{Z}\circ\mathcal{G})(\Delta\rho))]\right). \tag{6.2.14}$$

6.3 Reformulation of (QKD) via FR

In this section we reformulate (6.2.7) to obtain a model that satisfies the **MFCQ** and that grants us the differentiability of the objective function. We begin by assigning $\delta = \mathcal{G}(\rho)$ and $\sigma = \mathcal{Z}(\mathcal{G}(\rho))$ to (6.2.9) and rewrite the model (6.2.7) as below

$$\begin{array}{rl} \min\limits_{\rho,\delta,\sigma} & \operatorname{trace}(\delta\log\delta - \sigma\log\sigma) \\ \text{subject to} & \Gamma(\rho) = \gamma \\ & \delta = \mathcal{G}(\rho) \\ & \sigma = \mathcal{Z}(\delta) \\ & \rho \in \mathbb{H}^n_+,\ \ \delta,\sigma \in \mathbb{H}^k_+. \end{array} \tag{6.3.1}$$

We provide a brief summary of the reformulation process. The facial structure of $\mathbb{S}^n_+$ naturally extends to $\mathbb{H}^n_+$ with the transpose sign (T) in Proposition 2.2.2 replaced by the conjugate transpose $(*)$; see [89]. We perform **FR** on the triple (ρ,δ,σ) and we let $V_\rho, V_\delta, V_\sigma$ be the facial range vectors for these variables. Then we have the variables $(R_\rho, R_\delta, R_\sigma)$ of smaller orders as follows:

$$(V_\rho R_\rho V_\rho^*, V_\delta R_\delta V_\delta^*, V_\sigma R_\sigma V_\sigma^*) \in V_\rho\mathbb{H}^{n_\rho}_+V_\rho^* \times V_\delta\mathbb{H}^{n_\delta}_+V_\delta^* \times V_\sigma\mathbb{H}^{n_\sigma}_+V_\sigma^*.$$

As a consequence, we obtain the property

$$R_\rho \succ 0 \implies R_\delta, R_\sigma \succ 0,$$

granting us the differentiability of the modified objective function. After making some algebraic manipulations we get the desired model. The reformulations presented in this section are twofold:

1. We use the properties of the Kronecker product to derive exposing vectors analytically;

2. We perform **FR** to the objective function in order to guarantee differentiability.

6.3.1 Facial Reduction on the Constraint Set

For **FR** on the constraint set, we first observe its structure. The equality constraints $\{\rho : \Gamma(\rho) = \gamma\}$ of the problem (6.3.1) are divided into two groups; the observational and reduced density operator constraint sets, i.e., $S_O \cap S_R$. The set of the *observational constraints* is given by

$$S_O := \{\rho \succeq 0 : \langle P_s^A \otimes P_t^B, \rho\rangle = p_{st}, \forall st\},$$

where $P_s^A \in \mathbb{H}^{n_A}$, $P_t^B \in \mathbb{H}^{n_B}$, $n = n_A n_B$. Let $\{\Theta_j\}_j$ form an orthonormal basis for $\mathbb{H}^{n_A}$ and let $\rho_A \in \mathbb{S}_+^{n_A}$ be given. The *reduced density operator constraints* is given by

$$S_R := \{\rho \succeq 0 \ : \text{trace}_B(\rho) = \rho_A\} = \left\{\rho \succeq 0 \ : \ \langle \Theta_j \otimes I, \rho\rangle = \langle \Theta_j, \rho_A\rangle, \ \forall j = 1, \ldots, n_A^2\right\}.$$

These constraints originate from the partial trace. The *partial trace* is an operation that is often used in the area of quantum information theory. Let $\mathbb{H}^{n_A}$ and $\mathbb{H}^{n_B}$ be two Hilbert spaces. Given a composite system of A and B, the partial trace is used to evaluate the trace of only one component of the composite system. Let $\{b_j\}_{j=1}^{n_B^2}$ be a set of orthonormal basis for $\mathbb{H}^{n_B}$. Then the partial trace over the system B is the map $\text{trace}_B : \mathbb{H}^{n_A n_B} \to \mathbb{H}^{n_A}$ defined by

$$\text{trace}_B(\rho_{AB}) = \sum_j (I_{n_A} \otimes b_j^*)\rho_{AB}(I_{n_A} \otimes b_j).$$

The action of the map trace_B is often called ‘tracing out the system B’. It is known that the adjoint of trace_B is

$$\text{trace}_B^*(W) = W \otimes I_{n_B}.$$

We perform **FR** on the reduced density operator constraints that takes advantage of the Kronecker structure of the given data.

Theorem 6.3.1. *Let* $\text{range}(P) = \text{range}(\rho_A)$*,* $P^*P = I$*, and let* $V = P \otimes I$*. Then*

$$\rho \in S_R \implies \rho = VRV^*, \quad \textit{for some } R \in \mathbb{H}_+^{\text{rank}(\rho_A)\cdot n_B}.$$

Proof. If ρ_A is nonsingular, we choose $V = I$. We assume that $\text{rank}(\rho_A) < n_A$. We may write ρ_A using the spectral decomposition

$$\rho_A = \begin{bmatrix} P & Q \end{bmatrix} \text{BlkDiag}(D, 0) \begin{bmatrix} P & Q \end{bmatrix}^*, \ \text{rank}(D) = \text{rank}(\rho_A).$$

1. We recall that $\text{trace}_B^*(W) = W \otimes I_B$. Then $\rho \in S_R$ implies that

   $$\langle QQ^* \otimes I_{n_B}, \rho\rangle = \langle \text{trace}_B^*(QQ^*), \rho\rangle = \langle QQ^*, \text{trace}_B(\rho)\rangle = \langle W, \rho_A\rangle = 0,$$

   where the first two equalities hold by the property of the partial trace and the third equality holds due to the definition of S_R. Clearly, $QQ^* \otimes I_{n_B}$ serves as an exposing vector for S_R and it follows that $P \otimes I_{n_B}$ is a facial range vector.

2. We now provide an alternative proof that directly uses the auxiliary system (2.3.4). Consider $Z_\Theta = QQ^* \succeq 0$. Since $\{\Theta_j\}_j$ forms a basis, there always exists a vector y such that $Z_\Theta = \sum_j y_j \Theta_j$. Since the reduced density operator constraint holds $\langle \Theta_j, \rho_A \rangle = \theta_j$, we obtain

$$\langle \theta, y \rangle = \sum_j y_j \theta_j = \sum_j y_j \langle \Theta_j, \rho_A \rangle = \left\langle \sum_j y_j \Theta_j, \rho_A \right\rangle = Z_\Theta \rho_A = 0.$$

We recall the auxiliary system (2.3.4) in Lemma 2.3.4.

$$0 \preceq Z_\Theta \otimes I_{n_B} = \left(\sum_j y_j \Theta_j \right) \otimes I_{n_B} = \sum_j y_j \left(\Theta_j \otimes I_{n_B} \right) \neq 0, \quad \text{and } \langle \theta, y \rangle = 0.$$

Hence, by Lemma 2.3.4, $Z_\Theta \otimes I_{n_B}$ serves as an exposing vector for S_R and $P \otimes I$ is a facial range vector.

□

The facial range vector $P \otimes I$ computed in Theorem 6.3.1 is accurate within machine accuracy since it requires one eigen decomposition.

6.3.2 Facial Reduction on the Objective

We now turn our attention to the objective function. The domain of the objective function is possibly restricted to the boundary of the semidefinite cone. Even when the variable ρ is positive definite, the matrices $\mathcal{G}(\rho)$ and $Z(\mathcal{G}(\rho))$ can be singular. For instance, the matrix $\mathcal{G}(\rho)$ is always singular when the cardinality of the set $\{K_j\}_{j=1}^{\ell}$ is 1 and $k > n$. In this case, the objective function is not differentiable and the gradient formula in (6.1.2) is not applicable. Hence, the need arises for guaranteeing the differentiability of the objective function. **FR** has been typically invited for improving the quality of the feasible set as we have seen throughout this thesis. We show that **FR** also provides an effective preprocessing tool for improving the characteristic of the objective function.

We present a lemma that allows for a successful application of **FR** to the objective function.

Lemma 6.3.2. *Let $Y = VRV^*$, $R \succ 0$ be the compact spectral decomposition of a rank deficient matrix Y with $V^*V = I$. Then*

$$\operatorname{trace}(Y \log Y) = \operatorname{trace}(R \log R).$$

Proof. Let $U = \begin{bmatrix} V & P \end{bmatrix}$ be a unitary matrix, where columns of P form an orthonormal basis for the orthogonal complement of range(V). Then $Y = UDU^*$, where $D = \operatorname{BlkDiag}(R, 0)$.

$$\operatorname{trace}(Y \log Y) = \operatorname{trace}(UDU^*U(\log D)U^*) = \operatorname{trace}(D \log D) = \operatorname{trace}(R \log R),$$

where the first equality holds by the definition of the matrix extension of log function (see (6.2.2).) and the last equality holds by $0 \cdot \log 0 = 0$. □

The following result is used to obtain the exposing vectors for the images under the maps $\mathcal{G}, \mathcal{Z}$.

Lemma 6.3.3. *Let $\mathcal{C} \subseteq \mathbb{H}^n_+$ be a given convex set containing a positive definite matrix D. Let $\{Q_i\}_{i=1}^t \subseteq \mathbb{C}^{k\times n}$ be a given set of matrices. Define the linear map $\mathcal{T} : \mathbb{H}^n \to \mathbb{H}^k$ and the matrix $V \in \mathbb{C}^{k\times r}$ with orthonormal columns by*

$$\mathcal{T}(X) = \sum_{i=1}^t Q_i X Q_i^*, \quad \textit{and} \ \operatorname{range}(V) = \operatorname{range}\left(\sum_{i=1}^t Q_i Q_i^*\right).$$

Then the minimal face of $\mathbb{H}^k_+$ containing the image $\mathcal{T}(\mathcal{C})$ is characterized by

$$\operatorname{face}(\mathcal{T}(\mathcal{C}), \mathbb{H}^k_+) = V\mathbb{H}^r_+ V^*.$$

Proof. We first note that $\mathcal{T}(\mathcal{C}) \subseteq \mathbb{H}^k_+$. Let $W \in \mathbb{H}^k_+$ be a maximal exposing vector for $\operatorname{face}(\mathcal{T}(\mathcal{C}), \mathbb{H}^k_+)$. Then

$$\begin{array}{rcl} \langle W, \mathcal{T}(\mathcal{C})\rangle = 0 & \iff & \langle W, Y\rangle = 0, \ \forall Y \in \mathcal{T}(\mathcal{C}) \\ & \iff & \langle \mathcal{T}^*(W), X\rangle = 0, \ \forall X \in \mathcal{C} \\ & \iff & \mathcal{T}^*(W) = 0, \quad \text{since } D \in \mathbb{H}^n_{++}. \end{array}$$

Since $Q_i^* W Q_i \succeq 0, \ \forall i \in [t]$, we have $Q_i^* W Q_i = 0, \ \forall i \in [t]$, due to Item 3 of Fact 2.2.4. Thus, $\operatorname{range}(W) \subseteq \operatorname{null}(Q_i^*), \forall i$. Therefore we obtain the minimal facial range vector V analytically as follows:

$$\operatorname{range}(V) = \operatorname{null}(W) = \operatorname{range}\left(\sum_{i=1}^t Q_i Q_i^*\right).$$

□

Lemmas 6.3.2 and 6.3.3 imply that the **FR** for the equalities $\delta = \mathcal{G}(\rho)$ and $\sigma = \mathcal{Z}(\delta)$ can be done in *one* step; we obtain the greatest reduction on the dimension of the image after one eigen decomposition. This refers to the property of the positive map $\mathcal{T}$ defined in Lemma 6.3.3; the image $\mathcal{T}(\mathcal{C})$ is facially exposed. We emphasize that the **FR** applied to the sum of congruences can be performed within machine accuracy as it only requires a spectral decomposition.

We now elaborate on the step-by-step reformulation process that allows for a Slater point and a differentiable objective function over the positive definite matrices. Let $V_\rho, V_\delta, V_\sigma$ be facial range vectors for the triple (ρ, δ, σ) satisfying the constraints in (6.3.1). Hence variables have the form

$$\begin{array}{rcll} \rho & = & V_\rho R_\rho V_\rho^* \in \mathbb{H}^n_+, & R_\rho \in \mathbb{H}^{n_\rho}_+, \quad n_\rho \leq n; \\ \delta & = & V_\delta R_\delta V_\delta^* \in \mathbb{H}^k_+, & R_\delta \in \mathbb{H}^{k_\delta}_+, \quad k_\delta \leq k; \\ \sigma & = & V_\sigma R_\sigma V_\sigma^* \in \mathbb{H}^k_+, & R_\sigma \in \mathbb{H}^{k_\sigma}_+, \quad k_\sigma \leq k. \end{array}$$

We define the linear maps

$$\begin{array}{rclcrcl} \Gamma_V & : & \mathbb{H}^{n_\rho}_+ \to \mathbb{R}^m & \text{by} & \Gamma_V(R_\rho) & = & \Gamma(V_\rho R_\rho V_\rho^*); \\ \mathcal{G}_V & : & \mathbb{H}^{n_\rho}_+ \to \mathbb{H}^k_+ & \text{by} & \mathcal{G}_V(R_\rho) & = & \mathcal{G}(V_\rho R_\rho V_\rho^*); \\ \mathcal{Z}_V & : & \mathbb{H}^{k_\delta}_+ \to \mathbb{H}^k_+ & \text{by} & \mathcal{Z}_V(R_\delta) & = & \mathcal{Z}(V_\delta R_\delta V_\delta^*). \end{array}$$

Below is the summary of the computations for the facial range vectors $V_\rho, V_\delta, V_\sigma$.

1. We perform **FR** to $\{\rho \in \mathbb{H}^n_+ : \Gamma(\rho) = \gamma\}$ to find $V_\rho \in \mathbb{C}^{n\times n_\rho}$ for $\operatorname{face}(\mathcal{F}_\rho, \mathbb{H}^n_+)$, minimal face containing the feasible set. After **FR**, many of the linear equality constraints end up

being redundant; see Lemma 3.1.1. Let $P_{\bar{m}}$ be the projection that chooses the non-redundant equalities. We discard the redundant equalities using $P_{\bar{m}}$ and carry the well-conditioned equality system

$$\mathcal{F}_\rho := \{R_\rho \in \mathbb{H}_+^{n_\rho} : P_{\bar{m}}\Gamma_{V_\rho}(R_\rho) = P_{\bar{m}}\gamma\}.$$

2. We note that the facially reduced set $\mathcal{F}_\rho$ has a strictly feasible point. We use Lemma 6.3.3 to $\mathcal{F}_\delta := \{\mathcal{G}_V(R_\rho) \in \mathbb{H}_+^k : R_\rho \in \mathcal{F}_\rho\}$ and obtain the facial range vector $V_\delta \in \mathbb{C}^{k\times n_\rho}$

   $$\operatorname{range}(V_\delta) = \operatorname{range}(\mathcal{G}_V(I)).$$

   We choose V_δ with orthornormal columns. We define

   $$\mathcal{F}_\delta := \{R_\delta \in \mathbb{H}_+^{k_\delta} : V_\delta R_\delta V_\delta^* = \mathcal{G}_V(R_\rho),\ R_\rho \in \mathcal{F}_\rho\}.$$

   We note that $\mathcal{F}_\delta$ has a strictly feasible point.

3. We define

   $$\mathcal{F}_\sigma := \{\mathcal{Z}_V(R_\delta) \in \mathbb{H}_+^k : R_\delta \in \mathcal{F}_\delta\}.$$

   Applying Lemma 6.3.3 to the set $\mathcal{F}_\sigma$, we obtain the matrix $V_\sigma \in \mathbb{C}^{k\times k_\delta}$. We choose V_σ with orthornormal columns satisfying

   $$\operatorname{range}(V_\sigma) = \operatorname{range}(\mathcal{Z}_V(I)).$$

After **FR** on δ, σ, we write the objective function in (6.3.1) using Lemma 6.3.2, followed by the orthonormality of V_δ and V_σ:

$$\begin{array}{rcl} \operatorname{trace}(\delta\log\delta - \sigma\log\sigma) & = & \operatorname{trace}\left(V_\delta R_\delta V_\delta^* \log\left(V_\delta R_\delta V_\delta^*\right)\right) - \operatorname{trace}\left(V_\delta R_\delta V_\delta^* \log\left(V_\delta R_\delta V_\delta^*\right)\right) \\ & = & \operatorname{trace}(R_\delta \log R_\delta) - \operatorname{trace}(R_\sigma \log R_\sigma). \end{array}$$

We highlight that, by Lemma 6.2.6, the order of R_δ and the order R_σ are not the same in general, i.e., $k \geq k_\sigma \geq k_\delta$. Moreover, $\operatorname{range}(V_\sigma) \supseteq \operatorname{range}(V_\delta)$. The **FR** performed on the variables δ, σ may yield $k_\delta < k_\sigma$. Hence the two trace operations for R_δ and R_σ are used individually.

Using the facial range vectors V_δ and V_δ, we define

$$\mathcal{V}_\delta(R_\delta) := V_\delta R_\delta V_\delta^* \quad\text{and}\quad \mathcal{V}_\sigma(R_\sigma) := V_\sigma R_\sigma V_\sigma^*.$$

We rewrite (6.3.1) to obtain the model below.

$$\begin{array}{cl} \min\limits_{R_\rho, R_\delta, R_\sigma} & \operatorname{trace}(R_\delta \log R_\delta) - \operatorname{trace}(R_\sigma \log R_\sigma) \\ \text{subject to} & P_{\bar{m}}\Gamma_V(R_\rho) = P_{\bar{m}}\gamma \\ & \mathcal{V}_\delta(R_\delta) = \mathcal{G}_V(R_\rho) \\ & \mathcal{V}_\sigma(R_\sigma) = \mathcal{Z}_V(R_\delta) \\ & R_\rho \in \mathbb{H}_+^{n_\rho},\ R_\delta \in \mathbb{H}_+^{k_\delta},\ R_\sigma \in \mathbb{H}_+^{k_\sigma}. \end{array} \tag{6.3.2}$$

In Theorem 6.3.4 and Theorem 6.3.5 below, we simplify the last two equalities (6.3.2) by appropriate rotations.

Theorem 6.3.4. *Let $R_\rho \in \mathbb{H}_+^{n_\rho}$ and $R_\delta \in \mathbb{H}_+^{k_\delta}$. Define $\mathcal{G}_{UV}(\cdot) := V_\delta^* \mathcal{G}_V(\cdot) V_\delta$. Then*

$$\mathcal{V}_\delta(R_\delta) = \mathcal{G}_V(R_\rho) \iff R_\delta = \mathcal{G}_{UV}(R_\rho).$$

Proof. Let P be a matrix that completes a unitary matrix $U = \begin{bmatrix} V_\delta & P \end{bmatrix}$. We rotate the equality $\mathcal{V}_\delta(R_\delta) = \mathcal{G}_V(R_\rho)$ to obtain

$$U^* \mathcal{V}_\delta(R_\delta) U = U^* \mathcal{G}_V(R_\rho) U. \tag{6.3.3}$$

Since $U^* V_\delta = \begin{bmatrix} I_{k_\delta} \\ 0 \end{bmatrix}$, the left-hand-side of (6.3.3) is equal to

$$U^* \mathcal{V}_\delta(R_\delta) U = \begin{bmatrix} R_\delta & 0 \\ 0 & 0 \end{bmatrix}.$$

Since the facial range vector V_δ holds $\text{range}(V_\delta) = \text{range}(\mathcal{G}_V(I))$, we have $P^* \mathcal{G}_V = 0$. Therefore the right-hand-side of (6.3.3) becomes

$$U^* \mathcal{G}_V(R_\rho) U = \begin{bmatrix} V_\delta^* \\ P^* \end{bmatrix} \mathcal{G}_V(R_\rho) \begin{bmatrix} V_\delta & P \end{bmatrix} = \begin{bmatrix} V_\delta^* \mathcal{G}_V(R_\rho) V_\delta & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{G}_{UV}(R_\rho) & 0 \\ 0 & 0 \end{bmatrix}.$$

□

A similar results follows for the constraint $\mathcal{V}_\sigma(R_\sigma) = \mathcal{Z}_V(R_\delta)$.

Theorem 6.3.5. *Let $R_\delta \in \mathbb{H}_+^{k_\delta}$ and $R_\sigma \in \mathbb{H}_+^{k_\sigma}$. Define $\mathcal{Z}_{UV}(\cdot) := V_\sigma^* \mathcal{Z}_V(\cdot) V_\sigma$. Then*

$$\mathcal{V}_\sigma(R_\sigma) = \mathcal{Z}_V(R_\delta) \iff R_\sigma = \mathcal{Z}_{UV}(R_\delta).$$

Proof. From the orthonormal matrix V_σ, we complete the unitary matrix $U = \begin{bmatrix} V_\sigma & P \end{bmatrix}$ with a matrix P. Then following the same steps in the proof of Theorem 6.3.4 gives the statement. □

Theorems 6.3.4 and 6.3.5 result in the reductions of the number of linear equalities in (6.3.2). We note that $\mathcal{V}_\delta(R_\delta) = \mathcal{G}_V(R_\rho) \in \mathbb{H}^k$ whereas $R_\delta = \mathcal{G}_{UV}(R_\rho) \in \mathbb{H}^{k_\delta}$ and $k_\delta \le k$. This is not a surprise; these reductions correspond to the implicit redundancies of the equality system discussed in Lemma 3.1.1. Consequently, we obtain the model below:

$$\begin{array}{rl} \min\limits_{R_\rho, R_\delta, R_\sigma} & \text{trace}(R_\delta \log R_\delta) - \text{trace}(R_\sigma \log R_\sigma) \\ \text{subject to} & P_{\bar{m}} \Gamma_V(R_\rho) = P_{\bar{m}} \gamma \\ & R_\delta = \mathcal{G}_{UV}(R_\rho) \\ & R_\sigma = \mathcal{Z}_{UV}(R_\delta) \\ & R_\rho \in \mathbb{H}_+^{n_\rho},\ R_\delta \in \mathbb{H}_+^{k_\delta},\ R_\sigma \in \mathbb{H}_+^{k_\sigma}. \end{array} \tag{6.3.4}$$

6.3.3 Final Model

We now present the main reformulated model with a simplified notation and the derivatives. We define

$$\begin{array}{rclcl} \widehat{\mathcal{G}}(\cdot) & := & \mathcal{G}_{UV}(\cdot) & = & \sum_{j=1}^{k} (V_\delta^* K_j V_\rho)(\cdot)(V_\delta^* K_j V_\rho)^*, \\ \widehat{\mathcal{Z}}(\cdot) & := & \mathcal{Z}_{UV} \circ \mathcal{G}_{UV}(\cdot) & = & \sum_{i,j} (V_\sigma Z_i K_j V_\rho)(\cdot)(V_\sigma Z_i K_j V_\rho)^*. \end{array}$$

We then write the last two equality constraints in (6.3.4) with

$$R_\delta = \widehat{\mathcal{G}}(R_\rho), \quad R_\sigma = \widehat{\mathcal{Z}}(R_\rho).$$

For simplification, we relabel the variables as

$$\rho \leftarrow R_\rho, \;\; \delta \leftarrow R_\delta, \;\; \sigma \leftarrow R_\sigma.$$

Finally, substituting the variables into the objective function in (6.3.4), we obtain the final model below:

$$\begin{array}{rcll} p^* & = & \min\limits_{\rho} & f(\rho) := \operatorname{trace}\left(\widehat{\mathcal{G}}(\rho)\log\widehat{\mathcal{G}}(\rho)\right) - \operatorname{trace}\left(\widehat{\mathcal{Z}}(\rho)\log\widehat{\mathcal{Z}}(\rho)\right) \\ & & \text{subject to} & P_{\bar{m}}\Gamma_V(\rho) = P_{\bar{m}}\gamma \in \mathbb{R}^{\bar{m}} \\ & & & \rho \in \mathbb{H}_+^{n_\rho}. \end{array} \tag{6.3.5}$$

We discuss some properties of the final model (6.3.5). We point out that the model structures of (6.2.7) and (6.3.5) are the same; the objective functions are the compositions of quantum relative entropy and linear maps, and the constraint sets are spectrahedra. We display these reductions in the dimensions below:

	original model (6.2.7)	reformulated model (6.3.5)
variable	$\rho \in \mathbb{H}_+^n$	$\rho \in \mathbb{H}_+^{n_\rho}$
constraint linear map	$\Gamma : \mathbb{H}_+^n \to \mathbb{R}^m$	$P_{\bar{m}}\Gamma_V : \mathbb{H}_+^{n_\rho} \to \mathbb{R}^{\bar{m}}$
objective linear map	$\mathcal{G} : \mathbb{H}_+^n \to \mathbb{H}_+^k$	$\widehat{\mathcal{G}} : \mathbb{H}_+^{n_\rho} \to \mathbb{H}_+^{k_\delta}$
objective linear map	$\mathcal{Z} : \mathbb{H}_+^k \to \mathbb{H}_+^k$	$\widehat{\mathcal{Z}} : \mathbb{H}_+^{n_\rho} \to \mathbb{H}_+^{k_\sigma}$

Due to the **FR** and the rotations, $n_\rho \leq n$ and $k_\delta \leq k_\sigma \leq k$. We also acquire the important property

$$\rho \succ 0 \implies \widehat{\mathcal{G}}(\rho) \succ 0 \implies \widehat{\mathcal{Z}}(\rho) \succ 0. \tag{6.3.6}$$

The implications (6.3.6) hold since $\operatorname{relint}(A\mathcal{C}) = A\operatorname{relint}(\mathcal{C})$, where $\mathcal{C} \subseteq \mathbb{E}^n$ is a convex set and $A : \mathbb{E}^n \to \mathbb{E}^m$ is a linear map (see [139, Theorem 6.6].). Hence, having $\rho \succ 0$ allows the use of Corollary 6.2.8 to the objective function in (6.3.5) as we see in Theorem 6.3.6 below; we highlight that the implication (6.3.6) grants the differentiability of the objective function. Therefore, we can use the Gauss-Newton interior point method developed in Section 6.1.

Theorem 6.3.6. *Let $\rho \succ 0$. The gradient of f in (6.3.5) is*

$$\nabla f(\rho) = \widehat{\mathcal{G}}^*\left(\log(\widehat{\mathcal{G}}(\rho))\right) - \widehat{\mathcal{Z}}^*\left(\log(\widehat{\mathcal{Z}}(\rho))\right) + \widehat{\mathcal{G}}^*(I) - \widehat{\mathcal{Z}}^*(I).$$

The Hessian in the direction $\Delta\rho$ is

$$\nabla^2 f(\rho)\Delta\rho = \widehat{\mathcal{G}}^*\left(\log'(\widehat{\mathcal{G}}(\rho))(\widehat{\mathcal{G}}(\Delta\rho))\right) - \widehat{\mathcal{Z}}^*\left(\log'(\widehat{\mathcal{Z}}(\rho))(\widehat{\mathcal{Z}}(\Delta\rho))\right).$$

□

We provide a summary of the reformulation presented throughout Section 6.2 and Section 6.3

with the sequence below:

$$(6.2.7) \xrightarrow{(1)} (6.3.1) \xrightarrow{(2)} (6.3.2) \xrightarrow{(3)} (6.3.4) \xrightarrow{(4)} (6.3.5)$$

(1) variable substitutions;

(2) property of the map $\mathcal{Z}$ from Proposition 6.2.5 and **FR**;

(3) rotation of the constraints;

(4) substituting the constraint equalities back to the objective.

Although our reformulation is motivated by the use of interior point method, our reformulated model can be used to improve numerical performances of different algorithms. In Section 6.3.5 we show that the reformulated model indeed enhances the performance of the existing approach that uses the Frank-Wolfe algorithm.

6.3.4 Dual and Bounding

Computing a valid lower bound to the optimal value p^* is the main concern of (**QKD**). Suppose that M bits are used in a **QKD** protocol. Suppose that ℓ is a number of bits that we can use among M bits to establish a shared secret. The optimal value p^* provides the information on the ratio between M and ℓ. Let ϕ be a lower bound to the optimal value of (**QKD**). Then $\phi \leq \frac{\ell}{M}$ implies $\phi \cdot M \leq \ell$. A valid lower bound ϕ provides the following information; given M bits, at least $\phi \cdot M$ secure bits for establishing a secret key can be extracted. In this section we present how to obtain upper and lower bounds to the optimal value p^* of the model (6.3.5). Although acquiring a valid lower bound to p^* is sufficient, a good upper bound to p^* provides the information on how close the lower bound is to the optimal value.

Upper Bound Computation We obtain upper bounds by finding a feasible point. If we achieve the zero primal residual, i.e., $P_{\bar{m}}\Gamma_V(\hat{\rho}) = P_{\bar{m}}\gamma$, for some $\hat{\rho}$, then we evaluate the objective function at $\hat{\rho}$ to obtain an upper bound. Since our algorithm is an infeasible-start interior point method, the primal residual is not always 0. In this case we project the current point ρ_c to the affine constraint set, i.e.,

$$\hat{\rho} = \rho_c - (P_{\bar{m}}\Gamma_V)^\dagger (P_{\bar{m}}\Gamma_V(\rho_c) - P_{\bar{m}}\gamma) = \operatorname{argmin}_\rho \left\{ \|\rho - \rho_c\|^2 \ : \ P_{\bar{m}}\Gamma_V(\rho) = P_{\bar{m}}\gamma \right\}.$$

Here, $\dagger$ is the Moore-Penrose generalized inverse. If $\hat{\rho}$ is positive semidefinite, we then obtain an upper bound $f(\hat{\rho}) \geq p^*$.

In our numerical test, we obtain valid upper bounds starting in the early iterations. We obtain exact primal feasibility resulting from taking step length of one as soon as possible resulting in the zero primal residual. Subsequent iterations maintain zero primal residual (see Theorem 6.1.2.) and we improve upper bounds every iteration.

Lower Bound Computation We obtain a valid lower bound by employing the duality theory. An approach for computing lower bounds to (6.2.1) is proposed by [156] by using weak duality. They obtain lower bounds by a two-stage implementation. They first obtain a near optimal solution found by the Frank-Wolfe method and then compute a lower bound by using the subgradient inequality at a near optimal solution. The linearized problem originating from the subgradient inequality is solved by CVX package and a lower bound is obtained by weak duality.

We first show that our construction of the dual, and the strong duality of our reformulated model. For simplicity, we use $\Gamma_V(\rho) = \gamma_V$ to mean $P_{\bar{m}}\Gamma_V(\rho) = P_{\bar{m}}\gamma$, i.e., we omit the projection $P_{\bar{m}}$.

Theorem 6.3.7. *Let $\mathcal{L}$ be the Lagrangian for (6.3.5), i.e., $\mathcal{L}(\rho, y) = f(\rho) + \langle y, \Gamma_V(\rho) - \gamma_V\rangle$. Then the Lagrangian dual of (6.3.5) is*

$$d^* = \max_{Z \succeq 0, y} \left(\min_{\rho} \mathcal{L}(\rho, y) - \langle Z, \rho\rangle \right).$$

Furthermore, strong duality holds for (6.3.5), i.e., $p^* = d^*$ and d^* is attained for some $(y, Z) \in \mathbb{R}^{\bar{m}} \times \mathbb{H}^{n_\rho}_+$.

Proof. Strong duality holds due to the **FR** process, i.e., the model (6.3.5) has a strictly feasible point. □

We continue with the Lagrangian dual to (6.3.5) presented in Theorem 6.3.7. We note that we maintain $\rho \succ 0$ throughout our execution. Hence we can evaluate the gradient $\nabla f(\rho)$ and the following result follows.

Proposition 6.3.8. *Consider the problem (6.3.5). Let ρ_c, y_c be a primal-dual iterate. Let*

$$\bar{Z} = \nabla f(\rho_c) + \Gamma_V^*(y_c).$$

1. *If $\bar{Z} \succeq 0$, then a lower bound for problem (6.3.5) is*

$$p^* \geq f(\rho_c) + \langle y_c, \Gamma_V(\rho_c) - \gamma_V\rangle - \langle \rho_c, \bar{Z}\rangle.$$

2. *Suppose that $\bar{Z} \not\succeq 0$, i.e., $\lambda_{\min}(\bar{Z}) < 0$ and there exists w such that $\Gamma_V^*(w) \succ 0$. Let*

$$\bar{\alpha} = \operatorname{argmin}_\alpha\{\alpha : \bar{Z} + \alpha\Gamma_V^*(w) \succeq 0\}. \tag{6.3.7}$$

Then a lower bound to problem (6.3.5) is

$$p^* \geq f(\rho_c) + \langle y_c, \Gamma_V(\rho_c) - \gamma_V\rangle - \langle \rho_c, \bar{Z} + \bar{\alpha}\Gamma_V^*(w)\rangle.$$

Proof. Recall, from Theorem 6.3.7, that

$$p^* = d^* = \max_{Z \succeq 0, y} \min_{\rho \succeq 0} \{f(\rho) + \langle y, \Gamma_V(\rho) - \gamma_V\rangle - \langle Z, \rho\rangle\}.$$

Let $\rho_c \succ 0, y_c$ be given. If ρ_c and y_c give rise to $\bar{Z}$ satisfying

$$\bar{Z} \succeq 0, \;\; \nabla f(\rho_c) + \Gamma_V^*(y_c) - \bar{Z} = 0,$$

then $\rho_c = \operatorname{argmin}_{\rho \succeq 0} \mathcal{L}(\rho, y_c) - \langle \bar{Z}, \rho\rangle$. In other words, we found a dual feasible point that minimizes the dual functional. Hence, we obtain the lower bound in Item 1.

Given ρ_c, y_c, we now suppose they give rise to $\bar{Z} \not\succeq 0$. Suppose further that $\Gamma_V^*(w) \succ 0$ for some w. Then, there exists $\bar{\alpha} > 0$ such that $\bar{Z} + \Gamma_V^*(\bar{\alpha} w) \succeq 0$. We choose $\bar{\alpha}$ as given in (6.3.7). We then see that y_c and $\bar{Z} + \Gamma_V^*(\bar{\alpha} w)$ minimize the dual functional since the first-order optimality condition of the dual functional yields

$$\nabla f(\rho_c) + \Gamma_V^*(y_c + \bar{\alpha} w) - Z = \nabla f(\rho_c) + \Gamma_V^*(y_c) + \Gamma_V^*(\bar{\alpha} w) - Z = \bar{Z} + \Gamma_V^*(\bar{\alpha} w) - Z = 0.$$

□

Proposition 6.3.8 has the assumption $\bar{Z} \succeq 0$ or the existence of w that yields $\Gamma_V^*(w) \succ 0$. We can always guarantee $\Gamma_V^*(w) \succ 0$ for some w for (**QKD**). We recall that the feasible region of the original model (6.2.1) has the unit trace property, $\operatorname{trace}(\rho) = \langle I, \rho \rangle = 1$. The unit trace property is preserved after **FR**, since we choose V that has orthonormal columns, i.e., $V^*V = I$. Hence, we can set $\Gamma_V^*(w) = I$, for some w. Thus we can always find dual feasible points that minimize the dual functional efficiently; see Corollary 6.3.9.

Corollary 6.3.9. *Let $\rho_c \succ 0, y_c$ be primal-dual iterate for the problem* (6.3.5) *and let $\bar{Z} = \nabla f(\rho_c) + \Gamma_V^*(y_c)$. Then a lower bound to problem* (6.3.5) *is*

$$p^* \geq f(\rho_c) + \langle y_c, \Gamma_V(\rho_c) - \gamma_V \rangle - \langle \rho_c, \hat{Z} \rangle,$$

where

$$\hat{Z} = \bar{Z} + \left| \min\{0, \lambda_{\min}(\bar{Z})\} \right| I.$$

6.3.5 Numerical Result

In this section we examine the comparative performance among three algorithms; the Gauss-Newton method, the Frank-Wolfe method and cvxquad. The Gauss-Newton method refers to the algorithm developed throughout this chapter. The Frank-Wolfe method refers to the algorithm developed in [156] and cvxquad is developed in [59] that uses the semidefinite approximations of the matrix logarithm. We use Table 6.3.1 to present detailed reports on some selected instances[3].

For the instances corresponds to the DMCV protocol, we used the tolerance $\epsilon = 10^{-9}$ and the tolerance $\epsilon = 10^{-12}$ was used for the remaining instances. The maximum number of iteration was set to 80 for the Gauss-Newton method.

Problem Data			**Gauss-Newton**		**Frank-Wolfe with FR**		**Frank-Wolfe w/o FR**		**cvxquad with FR**	
protocol	**parameter**	**size**	**gap**	**time**	**gap**	**time**	**gap**	**time**	**gap**	**time**
ebBB84	(0.50,0.05)	(4,16)	5.98e-13	0.40	1.01e-04	92.49	1.17e-04	93.05	5.46e-01	214.02
ebBB84	(0.90,0.07)	(4,16)	1.42e-12	0.20	2.71e-04	91.26	2.75e-04	94.49	7.39e-01	177.64
pmBB84	(0.50,0.05)	(8,32)	5.51e-13	0.23	1.12e-04	1.38	6.47e-04	1.91	5.26e-01	158.64
pmBB84	(0.90,0.07)	(8,32)	5.13e-13	0.17	7.31e-05	1.29	6.25e-04	38.65	6.84e-01	233.43
mdiBB84	(0.50,0.05)	(48,96)	1.14e-12	1.09	4.99e-05	104.31	5.22e-04	134.05	1.82e-01	557.08
mdiBB84	(0.90,0.07)	(48,96)	2.96e-13	0.96	2.04e-04	106.61	2.85e-03	126.62	4.57e-01	537.52
TFQKD	(0.80,100.00,0.70)	(12,24)	1.15e-12	0.79	2.60e-09	1.21	1.57e-03	124.48	n/a	0.01
TFQKD	(0.90,200.00,0.70)	(12,24)	1.04e-12	0.44	3.98e-09	1.13	1.68e-04	2.25	n/a	0.00
DMCV	(10.00,60.00,0.05,0.35)	(44,176)	2.71e-09	507.83	4.35e-06	467.41	3.57e-06	657.08	n/a	0.01
DMCV	(11.00,120.00,0.05,0.35)	(48,192)	3.24e-09	700.46	2.35e-06	194.62	2.15e-06	283.06	n/a	0.01
dprBB84	(1.00,0.08,30.00)	(12,48)	4.92e-13	1.19	3.85e-06	96.74	9.43e-05	141.38	⋆⋆	118.81
dprBB84	(2.00,0.14,30.00)	(24,96)	1.04e-12	11.76	5.71e-06	17.66	5.38e-06	34.60	⋆⋆	106.24
dprBB84	(3.00,0.10,30.00)	(36,144)	4.96e-13	63.26	6.48e-04	7.38	2.08e-02	29.00	⋆⋆	582.64
dprBB84	(4.00,0.12,30.00)	(48,192)	3.80e-13	330.39	4.42e-05	13.78	9.79e-04	175.39	⋆⋆	3303.23

Table 6.3.1: Numerical Report from Three Algorithms

[3] The instances are tested with MATLAB version 2021a using Dell PowerEdge R640 Two Intel Xeon Gold 6244 8-core 3.6 GHz (Cascade Lake) with 192 Gigabyte memory.

In Table 6.3.1 **Problem Data** refers to the data used to generate the instances. **Gauss-Newton** refers to the Gauss-Newton method. **Frank-Wolfe** refers to the Frank-Wolfe algorithm used in [156] and we use 'with **FR** (w/o **FR**, resp.)' to indicate that the model is solved with **FR** (without **FR**, resp.). The header **cvxquad with FR** refers to the algorithm provided by [59] with **FR** reformulation. If a certain algorithm fails to give a reasonable answer within a reasonable amount of time, we give a '$\star\star$' flag in the gap followed by the time taken to obtain the error message. We use 'n/a' to indicate the instances for which cvxquad is not applicable due to the size differences in the images under $\widehat{\mathcal{G}}$ and $\widehat{\mathcal{Z}}$ due to **FR**.

The following provides details for the remaining headers in Table 6.3.1.

1. **protocol**: the protocol name; we refer to [92, Appendix C] for the details of the protocols;
2. **parameter**: the parameters used for testing; we refer to [92, Appendix C] for the ordering of the parameters;
3. **size**: the size (n, k) of original problem; n, k are defined in (6.2.4);
4. **gap**: the relative gap between the bestub and bestlb;

$$\frac{\text{bestub - bestlb}}{1 + \frac{|\text{bestub}|+|\text{bestlb}|}{2}}. \tag{6.3.8}$$

5. **time**: time taken in seconds.

We make some discussions on the formula (6.3.8). The best upper bound from the Gauss-Newton algorithm is used for all instances for 'bestub' in (6.3.8). The Gauss-Newton algorithm computes the lower bounds as presented in Proposition 6.3.8. The Frank-Wolfe algorithm presented in [156] obtains the lower bound by a linearization technique near the optimal. As presented in [59], cvxquad uses the semidefinite approximations of the matrix logarithm. The lower bounds from cvxquad can be larger than the theoretical optimal values. Therefore, we adopt the lower bound strategy used in [156] for cvxquad.

We now discuss the results in Table 6.3.1. Comparing the two columns **gap** and **time** among the different methods, we see that the Gauss-Newton method outperforms other algorithms in both producing good relative gaps and the running time. For example, comparing **Gauss-Newton** and **Frank-Wolfe with FR**, the gaps and running times from **Gauss-Newton** are competitive. There are three instances that **Gauss-Newton** took longer time. We emphasize that the gap values with **Gauss-Newton** illustrate much higher accuracy.

We now illustrate that the reformulation strategy via **FR** contributes to superior algorithmic performances. For the columns **Frank-Wolfe with FR** and **Frank-Wolfe w/o FR** in Table 6.3.1, the **FR** reformulation contributes to not only giving tighter gaps but also reducing the running time significantly. We now consider the column corresponding to **cvxquad with FR** in Table 6.3.1. We see that the algorithm fails (marked with '$\star\star$') with some instances due to the memory shortage. Facial reduction indeed contributes to the reduction on the problem sizes. For example, for pmBB84 with the parameter setting $(0.5, 0.05)$, we reduce the size $(n, m) = (8, 21)$ to $(n_\rho, \bar{m}) = (4, 8)$; for mdiBB84 with the parameter setting $(0.5, 0.05)$, we reduce the size $(n, m) = (48, 305)$ to $(n_\rho, \bar{m}) = (12, 34)$.

We often get problems where the reduced density operator constraint yields the complete **FR**. However, we should be aware of the cases where the exposing vector given by the reduced density operator provides the maximal exposing vector for the entire constraint set $\{\rho : \Gamma(\rho) = \gamma\}$.

The approach [156] uses the perturbation in order to avoid the issues with the singular matrices. More specifically, in order to evaluate the derivative, [156] perturbs the points by adding a small multiple of the identity. Our **FR** approach removes the need for the perturbations owing to the implications (6.3.6).

Chapter 7

Conclusions and Open Questions

7.1 Conclusions

In this thesis we showed that facial reduction, **FR**, arises in many applicatins both as a result of failure of strict feasibility, as well nonsmoothness in the objective function. We have shown how to recognize where **FR** is needed and how to preprocess both **SDP** and **DNN** relaxations to obtain regularized, simplified problems. In addition, **FR** and singularity degrees help in understanding instabilities in models. Facial reduction, as a preprocessing mechanism, was introduced in various forms. In the theoretical aspect, facial reduction resulted in enhancing the model qualities that appear in various contexts of fields of study. We recognized diverse circumstances that implicit redundancies emerge. In the practical aspect, facial reduction enhanced performances of many different classes of algorithms such as the simplex method, the splitting methods and the interior point methods. Below, we summarize this thesis categorized by topics.

Two-Step Facial Reduction and Implicit Loss of Surjectivity We addressed the impact of the absence of strict feasibility in **SDP** and **LP** in the theoretical and computational aspects. In addition to the known notion of singularity degree, we introduced two new notions of singularity: the max-singularity degree, and the implicit problem singularity. We shed light on the main difficulties that arose with the implicit redundant constraints. This led to the view of the two-step facial reduction, and the discussion on the importance of removing implicit redundant constraints.

For the area of **SDP**, we observed the instability issues by exploiting the properties of **FR** with respect to the affine subspace. The Barvinok-Pataki bound guarantees the existence of a point X satisfying $t(\operatorname{rank}(X)) \leq m$. We improved this bound

$$t(r) \leq \min\left\{t(n - \mathbf{maxsd}(\mathcal{F})),\ m - \mathbf{ips}(\mathcal{F})\right\} \leq m.$$

The knowledge of the strengthened bound can help obtain low rank solutions in many applications. For example, the strengthened bound can be used for reducing the variable dimensions in nonlinear methods for solving **SDP**s [25]. Having this knowledge can help with low rank projections on the cone $\mathbb{S}^n_+$ that arise in the splitting methods such as **ADMM** or **PRSM**.

For the area of **LP**, we further made many interesting observations both in theoretical and practical aspects. Even though strict feasibility is not a necessary assumption to establish strong duality, we emphasized that ensuring strict feasibility should be part of preprocessing for linear programming, otherwise a problem may conceal implicit singularities. For the theory, we proved using the implicit redundancies that every **BFS** is degenerate in the absence of strict feasibility. Moreover, the implicit redundancies resulted in ill-conditioning in the system used for finding search directions in interior point methods such as the self-dual embedding. We also developed a preprocessing method for detecting variables fixed at 0, resulting in promoting stability. We provided an efficient preprocessing step for **FR** that can be directly concatenated to the phase-I of the two-phase simplex method. We have presented various numerical experiments that convey the importance of preprocessing for strict feasibility. Our numerics for **LP** illustrated the instability using the accuracy of optimality conditions as well as the effect of perturbations for the two most popular classes of algorithms, i.e., the simplex and interior point methods. This was illustrated on random problem, as well as instances from the NETLIB data set. The ill-conditioning arising from lack of strict feasibility highlighted the fact that free variables are generally not treated properly in the literature as splitting them into two results in making the dual ill-posed.

A Restricted Dual Peaceman-Rachford Splitting Method for Solving DNN Relaxation of Binary Quadratic Problems We presented a straightforward derivation of the **DNN** relaxation of the binary quadratic problem, **BQP**, with the unit row-sum constraint. Given a **BQP**, we derived a facially reduced **SDP** relaxation. We then identified some redundant constraints to the **SDP** relaxation of the **BQP** to complete the **DNN** relaxation. We also exploited the set of dual optimal multipliers to obtain prior knowledge and provided customized dual updates in the algorithm.

The **FR** provided a natural splitting of the variables and the splitting method was an excellent fit for employing splitting methods. Given constraints that are difficult to engage simultaneously, we distributed the constraints into two simpler subproblems to solve them efficiently. The splitting of the subproblems led to incorporating redundant constraints to the model that are not redundant in the individual subproblems.

The natural splitting provided by **FR** together with the known dual optimal elements led us to developing the restricted dual Peaceman-Rachford splitting method. We derived the algorithm by making connection to the monotone operator theory. Using this variant of splitting method, we exhibited numerical experiments with the two classes of NP-hard real-world problems, the protein side-chain positioning problem and the quadratic assignment problem. We illustrated the efficiency of our approach with the numerical experiments.

Gauss-Newton Framework for Solving Nonlinear SDP over Hermitian matrices We presented an interior point method framework for solving a **SDP** over the set of Hermitian matrices. We used the Gauss-Newton method for finding the points satisfying the first-order optimality conditions by forming the over-determined nonlinear least squares problem. We exploited the structure of the Jacobian system and reduced the computation cost for finding the search directions followed by the stable back substitution steps.

We then applied the framework to a robust numerical method for finding provable lower bounds for the convex optimization problem for computing the key rate for the **QKD** in the presence of an eavesdropper. We used the novel **FR** technique for not only applicable to the constraint set but also to the objective function. This is is done by regularizing the constraint set and the objective function. The conventional **FR** is performed in order to improve the characteristics of the constraint set. We showed that the **FR** technique can be extended to improve the feature of the objective function, e.g., differentiability.

This led to a robust numerical method for finding provable lower bounds for the key rate computation for **QKD**. Our empirical evidence illustrated significant improvements in solver running time and accuracy over previous methods. Our approach showed a competitive numerical performances that outperforms the available methods in the literature. We solved many problems close to machine accuracy and provide a theoretically provable accurate lower bounds.

7.2 Future Directions and Open Questions

7.2.1 Preprocessing for LP

The dual simplex method is a popular choice for solving linear programs. We also have seen that the failure of dual strict feasibility results in degeneracy problems. And redundant constraints have been shown in the literature to poorly affect algorithms [46]. Identifying redundant constraints is a nontrivial operation [31]. This motivates doing **FR** on both the primal and the dual problems. A few questions arise. Is it better to perform **FR** on the dual first than the primal? If the first use of Algorithm 4.2.2 does not guarantee strict feasibility on $\mathcal{F}$, do we continue with **FR** applied to the dual and focus on the primal again? That is, do we alternate the preprocessing steps between the primal and the dual? What is the best approach for guaranteeing the primal-dual strict feasibility?

Algorithm 4.2.2 is an extension of the usual phase-I of the two-phase simplex method. Hence, it would be beneficial to catch information for **FR** during this phase as well. Can phase-I of the two-phase simplex method reveal anything about strict feasibility or an exposing vector?

After **FR** done to the dual problem (i.e., identified slack variables that are fixed at 0), we noticed in (4.3.10) that there is a set of constraints that become redundant. These redundant equalities take place among the inequality constraints that are implicitly equality constraints, i.e., the inequalities that correspond to $\operatorname{supp}(w)$; see (4.3.8). Once we discard the redundant constraints, some equality constraints remain in the dual system. These remaining constraints lead some primal variables to be free. This is interesting since **FR** to the primal does not alter the structure of the dual constraint system. Hence, careful analysis on these relationships is necessary.

Many **LP** instances contain various forms of constraints, e.g., inequality constraints, bounded variables, free variables and so on. Although we can transform these instances into the standard form equivalently, it would be interesting to make the process more efficient to directly work with instances with various forms of constraints. In Appendix B.2, we outline the process illustrated in Algorithm 4.2.2 applicable to a general feasible region in the form

$$\mathcal{H} := \left\{ x = \big(x_1; x_2; x_3; x_4\big) \in \mathbb{R}^n : \begin{array}{ll} A_{\text{eq}}x = b_{\text{eq}}, & A_{\text{ineq}}x \le b_{\text{ineq}} \\ \hat{\ell} \le x_1, & x_2 \le \hat{u} \\ \bar{\ell} \le x_3 \le \bar{u}, & x_4 \text{ free} \end{array} \right\}, \tag{7.2.1}$$

where the data dimensions are $A_{\text{eq}} \in \mathbb{R}^{m_{\text{eq}} \times n}$, $A_{\text{ineq}} \in \mathbb{R}^{m_{\text{ineq}} \times n}$, $\hat{\ell} \in \mathbb{R}^{n_{x_1}}$, $\hat{u} \in \mathbb{R}^{n_{x_2}}$ and $\bar{\ell}, \bar{u} \in \mathbb{R}^{n_{x_3}}$.

7.2.2 Preprocessing for SDP

We saw throughout this thesis that facial reduction is a successful process that gives both the stability and the reduction on the problem size. Facial reduction for many of the successful applications is achieved by *exploiting* analytic expressions of exposing vector tailored to problem structure. However, **FR** for an arbitrary spectrahedron can be expensive. In such cases, we generally rely on available interior point method software to solve the auxiliary problem (2.3.4). Computing a reliable exposing vector within machine accuracy is a challenging task, see e.g., [134, Section 4.5] and [38, Section 4.4]. Nevertheless, it would be interesting to introduce a general framework for the **FR** algorithm tailored to compute exposing vectors.

Let $P \in \mathbb{S}^n_{++}$ and $\alpha > 1$. Consider the following problem motivated by (2.3.4):

$$(\mathcal{P}_{\mathbf{FR}}) \quad p^*_{\mathbf{FR}} := \min_y \{\langle b, y\rangle \ : \ \mathcal{A}^*y \succeq 0, \ \langle P, \mathcal{A}^*y\rangle = \alpha\}. \tag{7.2.2}$$

The dual $(\mathcal{D}_{\mathbf{FR}})$ of $(\mathcal{P}_{\mathbf{FR}})$ is

$$(\mathcal{D}_{\mathbf{FR}}) \quad d^*_{\mathbf{FR}} := \max_{\lambda, W} \{\alpha\lambda \ : \ \mathcal{A}(W) + \lambda\mathcal{A}(P) = b, \ W \succeq 0\}. \tag{7.2.3}$$

Then the Guass-Newton method can be used to solve the primal-dual pair (7.2.2) and (7.2.3). We include some properties of the primal-dual pair and outline the Gauss-Newton framework tailored for this pair in Appendix B.3.

For an instance where strict feasibility is known to fail, we may consider solving the alternative problem below:

$$\max_y \ \{\operatorname{trace}(\mathcal{A}^*(y)) : \langle b, y\rangle = 0, \mathcal{A}^*(y) \succeq 0\}. \tag{7.2.4}$$

We note that (7.2.4) promotes high rank for the exposing vector, which is a desirable property. We also note that we do not need to include the hyperplane $\langle P, \mathcal{A}^*y\rangle = \alpha$ to (7.2.4) that appears in (7.2.2) that is used to prevent the zero exposing vector.

We discussed the implicit problem singularity, **ips**, for the primal feasible set $\mathcal{F}$ by counting the number of implicit redundant equality constraints. The dual feasible set

$$\mathcal{F}_{\mathcal{D}} = \{C - \mathcal{A}^*(y) \in \mathbb{S}^n_+ : y \in \mathbb{R}^m\}$$

is in a conic form, and hence the counting the number of equality constraints does not seem to directly apply to $\mathcal{F}_{\mathcal{D}}$. Hence, it raises the following question; how do we define an analogue of **ips** for $\mathcal{F}_{\mathcal{D}}$? We can rewrite the dual feasible set using a null-space representation

$$\mathcal{F}_{\mathcal{D}} = \{X \in \mathbb{S}^n_+ : \mathcal{N}(X) = d\}, \text{ for some linear map } \mathcal{N} : \mathbb{S}^n \to \mathbb{R}^{n-m}, \, d \in \mathbb{R}^{n-m},$$

and cast the discussion of the implicit redundant constraints to the equality system $\mathcal{N}(X) = d$. However, an in-depth discussion on the analogue of **ips** tailored to the dual feasible set $\mathcal{F}_{\mathcal{D}}$ is necessary for the future development in this area.

The use of simplex method for preprocessing for **LP** led us to the reliable computation of an exposing vector. There is a generalized definition of the basic feasible solution for spectrahedra and a simplex-type method that can be applied to the **SDP** has been proposed, see [102,127,163,164]. It would be interesting to extend the simplex-type preprocessing method to **SDP** to resolve the accuracy issues carried by the interior point methods and to use it for strict feasibility testing.

7.2.3 Singularities on SDP and DNN Relaxations

We have observed that many **SDP** relaxations of hard combinatorial problems are known to fail strict feasibility. When solving the relaxations, low rank solutions are desired as they provide better approximations for the underlying problem. With the new strengthened bound, we obtain an immediate improvement for these classes of problems. This observation leads to an interesting question. Does it help to model a problem that has large implicit problem singularities or large max-singularity degree so that this results in low rank optimal solutions?

For two closed convex cones $\mathcal{K}_1, \mathcal{K}_2$, the equality below holds:

$$(\mathcal{K}_1 \cap \mathcal{K}_2)^* = \operatorname{cl}(K_1^* + \mathcal{K}_2^*).$$

If one of the cones is polyhedral, the relation below holds:

$$(\mathcal{K}_1 \cap \mathcal{K}_2)^* = K_1^* + \mathcal{K}_2^*.$$

We recall that the exposing vector $Z = \mathcal{A}^*(y)$ for $\mathcal{F}$ is a member of the dual cone $(\mathbb{S}^n_+)^*$. We let **sd**(**SDP**) (**sd**(**DNN**), resp.) denote the singularity degree of the **SDP** problems (**DNN** problems, resp.). Since $(\mathbb{S}^n_+)^* \subseteq (\mathbb{S}^n_+)^* + (\mathbb{R}^{n\times n}_+)^*$, we conclude that any exposing vector for **SDP** is an exposing vector for **DNN** and thus

$$\textbf{maxsd}(\textbf{DNN}) \leq \textbf{maxsd}(\textbf{SDP}).$$

How different are the **ips**, **sd** for problems with the **DNN** cone rather than the $\mathbb{S}^n_+$?

We recall Lemma 5.3.1, Corollary 5.3.2 and their implications. Adding one equality constraint (optimal plane) sets many of the elements of the variable Y to be 0, i.e., many of the inequalities of the type $0 \leq Y_{i,j} \leq 1$ become redundant. We also have seen in Example 3.2.13 that adding the optimal plane gives rise to a tighter bound on the rank of an optimal solution. This observation raises an interesting question. Does adding the optimal plane help achieving low rank optimal solutions since the optimal set contains a very large implicit problem singularities?

7.2.4 Extension of the Gauss-Newton Framework to Various Constraints

We have developed the Gauss-Newton interior point method for solving key rate computation for quantum key distribution. We developed the algorithm that is applicable to the standard spectrehedron. While our framework covers many interesting **QKD** protocols, there are scenarios where inequality constraints are needed, e.g., [70]. Hence, the need arises for extending our framework to the models that contain additional inequality constraints. It is interesting and important to address possible numerical instabilities introduced by those inequality constraints.

References

[1] T. Akutsu. Np-hardness results for protein side-chain packing. *Genome Informatics*, 8:180–186, 1997. 85

[2] F. Alizadeh, J-P.A. Haeberly, and M.L. Overton. A new primal-dual interior-point method for semidefinite programming. In J.G. Lewis, editor, *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 113–117. SIAM, 1994. 20, 102

[3] E. Althaus, O. Kohlbacher, H.-P. Lenhof, and P. Mauller. A combinatorial approach to protein docking with flexible side chains. *Journal of computational biology*, 9(4):597–612, 2002. 85

[4] E.D. Andersen. Finding all linearly dependent rows in large-scale linear programming. *Optimization methods & software*, 6(3):219–227, 1995. 28

[5] A.F. Anjos and J.B. Lasserre, editors. *Handbook on Semidefinite, Conic and Polynomial Optimization*. International Series in Operations Research & Management Science. Springer-Verlag, 2011. 29

[6] M.F. Anjos and H. Wolkowicz. Geometry of semidefinite max-cut relaxations via matrix ranks. *J. Comb. Optim.*, 6(3):237–270, 2002. New approaches for hard discrete optimization (Waterloo, ON, 2001). 29

[7] D. Bahadur, T. Akutsu, E. Tomita, and T. Seki. Protein side-chain packing problem: A maximum edge-weight clique algorithmic approach. *Journal of bioinformatics and computational biology*, 3 1:103–26, 2004. 85

[8] A. Barvinok. A remark on the rank of positive semidefinite matrices subject to affine constraints. *Discrete Comput. Geom.*, 25(1):23–31, 2001. 23, 28, 29

[9] A. Barvinok. *A course in convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2002. 29

[10] A.I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete Comput. Geom.*, 13(2):189–202, 1995. 29

[11] H.H. Bauschke and P.L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011. With a foreword by Hédy Attouch. 18, 75, 77

[12] J.E. Beasley, editor. *Advances in linear and integer programming*, volume 4 of *Oxford Lecture Series in Mathematics and its Applications*. The Clarendon Press, Oxford University Press, New York, 1996. Oxford Science Publications. 102

[13] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2009. 61

[14] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Oper. Res. Lett.*, 25(1):1–13, 1999. 61

[15] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997. 54

[16] R.K. Bhati and A. Rasool. Quadratic assignment problem and its relevance to the real world: A survey. *International Journal of Computer Applications*, 96(9):42–47, 2014. 91

[17] G. Birkoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuman Rev.*, Ser. A:147–151, 1946. 42, 91

[18] R.E. Bixby. Solving real-world linear programs: a decade and more of progress. *Oper. Res.*, 50(1):3–15, 2002. 50th anniversary issue of Operations Research. 37

[19] R.G. Bland. New finite pivoting rules for the simplex method. *Math. Oper. Res.*, 2(2):103–107, 1977. 37

[20] R.I. Boţ, E.R. Csetnek, and D. Meier. Variable metric ADMM for solving variational inequalities with monotone operators over affine sets. In *Splitting algorithms, modern operator theory, and applications*, pages 91–112. Springer, Cham, [2019] ©2019. 77

[21] J.M. Borwein and H. Wolkowicz. Characterization of optimality for the abstract convex program with finite-dimensional range. *J. Austral. Math. Soc. Ser. A*, 30(4):390–411, 1980/81. 1, 12

[22] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 1, 12, 13

[23] M.J. Bower, F.E. Cohen, and R.L. Dunbrack. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *Journal of Molecular Biology*, 267(5):1268–1282, 1997. 85

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Machine Learning*, 3(1):1–122, 2011. 18, 77

[25] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2, Ser. B):329–357, 2003. Computational semidefinite and second order cone programming: the state of the art. 29, 34, 124

[26] S. Burer and R.D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Math. Program.*, 103(3, Ser. A):427–444, 2005. 29, 34

[27] F. Burkowski, Y-L. Cheung, and H. Wolkowicz. Efficient use of semidefinite programming for selection of rotamers in protein conformations. *INFORMS J. Comput.*, 26(4):748–766, 2014. 12, 15, 66, 68, 70, 71, 74, 85, 88

[28] F. Burkowski, H. Im, and H. Wolkowicz. A Peaceman-Rachford splitting method for the protein side-chain positioning problem. Technical report, University of Waterloo, Waterloo, Ontario, 2022. arxiv.org/abs/2009.01450,21. 12, 15, 18, 29, 70

[29] F.J. Burkowski. *Computational and Visualization Techniques for Structural Bioinformatics Using Chimera*. Chapman & Hall/CRC mathematical and computational biology series. Chapman and Hall/CRC, London, 2015. 86, 87

[30] A.A. Canutescu, A.A. Shelenkov, and R.L. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein science*, 12(9):2001–2014, 2003. 85, 86, 89

[31] R.J. Caron, A. Boneh, and S. Boneh. Redundancy. In *Advances in sensitivity analysis and parametric programming*, volume 6 of *Internat. Ser. Oper. Res. Management Sci.*, pages 13.1–13.41. Kluwer Acad. Publ., Boston, MA, 1997. 126

[32] C. Cartis and N. Gould. Finding a point in the relative interior of a polyhedron. *Council for the Central Laboratory of the Research Councils*, pages 1–59, 2006. 28, 64

[33] E. Çela. *The quadratic assignment problem*, volume 1 of *Combinatorial Optimization*. Kluwer Academic Publishers, Dordrecht, 1998. Theory and algorithms. 91

[34] R. Chandrasekaran, Santosh N. Kabadi, and Katta G. Murty. Some NP-complete problems in linear programming. *Oper. Res. Lett.*, 1(3):101–104, 1981/82. 44

[35] A. Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20:160–170, 1952. 37, 50

[36] B. Chazelle, C. Kingsford, and M. Singh. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS J. Comput.*, 16(4):380–392, 2004. 85, 88

[37] Y. Chen and X. Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011. 83

[38] Y.-L. Cheung. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice*. PhD thesis, University of Waterloo, 2013. 10, 54, 64, 67, 127

[39] V. Chvátal. *Linear programming*. A Series of Books in the Mathematical Sciences. W. H. Freeman and Company, New York, 1983. 27

[40] C.W. Commander. *A survey of the quadratic assignment problem, with applications*. PhD thesis, University of Florida, 2003. PhD Thesis. 91

[41] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1963. 37

[42] G.B. Dantzig, A. Orden, and P. Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific J. Math.*, 5:183–195, 1955. 37

[43] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Englewood Cliffs, NJ, 1983. 18, 20

[44] J.E. Dennis Jr. and H. Wolkowicz. Sizing and least-change secant methods. *SIAM J. Numer. Anal.*, 30(5):1291–1314, 1993. 107

[45] J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature (London)*, 356(6369):539–542, 1992. 85

[46] A. Deza, E. Nematollahi, R. Peyghami, and T. Terlaky. The central path visits all the vertices of the Klee-Minty cube. *Optim. Methods Softw.*, 21(5):851–865, 2006. 126

[47] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002. 57

[48] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. *SIAM J. Optim.*, 27(4):2301–2331, 2017. 12, 15, 70

[49] D. Drusvyatskiy, G. Li, and H. Wolkowicz. A note on alternating projections for ill-posed semidefinite feasibility problems. *Math. Program.*, 162(1-2, Ser. A):537–548, 2017. 22

[50] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. 11, 13, 16, 25, 51, 99, 146

[51] R.L. Dunbrack, Jr. and M. Karplus. Backbone-dependent rotamer library for proteins application to side-chain prediction. *Journal of Molecular Biology*, 230(2):543–574, March 1993. 86

[52] M. Dür, B. Jargalsaikhan, and G. Still. The Slater condition is generic in linear conic programming. Technical report, University of Trier, Trier, Germany, 2012. 61

[53] J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(3, Ser. A):293–318, 1992. 77

[54] J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. *Pac. J. Optim.*, 11(4):619–644, 2015. 78

[55] A.N. Elshafei. Hospital layout as a quadratic assignment problem. *Operations Research Quarterly*, 28:167–179, 1977. 91

[56] M. Epelman and R.M. Freund. Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Math. Program.*, 88(3, Ser. A):451–485, 2000. 25

[57] O. Eriksson, Y. Zhou, and A. Elofsson. Side chain-positioning as an integer programming problem. In *Algorithms in bioinformatics (Århus, 2001)*, volume 2149 of *Lecture Notes in Comput. Sci.*, pages 128–141. Springer, Berlin, 2001. 85

[58] H. Fawzi, J. Gouveia, P.A. Parrilo, R.Z. Robinson, and R.R. Thomas. Positive semidefinite rank. *Math. Program.*, 153(1, Ser. B):133–177, 2015. 29

[59] H. Fawzi, J. Saunderson, and P.A. Parrilo. Semidefinite approximations of the matrix logarithm. *Foundations of Computational Mathematics*, 2018. Package cvxquad at https://github.com/hfawzi/cvxquad. 121, 122

[60] A.V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, 1983. 51

[61] A. Forsgren, P.E. Gill, and E. Wong. Primal and dual active-set methods for convex quadratic programming. *Mathematical programming*, 159(1-2):469–508, 2015. 63

[62] R. Freund, R. Roundy, and M. Todd. Identifying the set of always-active constraints in a system of linear inequalities by a single linear program. 02 1985. 63

[63] R.M. Freund and F. Ordonez. On an extension of condition number theory to nonconic convex optimization. *Mathematics of operations research*, 30(1):173–194, 2005. 25, 27

[64] R.M. Freund and J.R. Vera. Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system. Technical report, MIT, Cambridge, MA, 1997. 25, 27

[65] D. Gabay. Chapter ix applications of the method of multipliers to variational inequalities. *Studies in Mathematics and its Applications*, 15(C):299–331, 1983. cited By 308. 77, 78

[66] T. Gal, editor. *Degeneracy in optimization problems*. Baltzer Science Publishers BV, Bussum, 1993. Ann. Oper. Res. **46**/47 (1993), no. 1-4. 37

[67] J. Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Mathematical programming*, 12(1):136–138, 1977. 50

[68] J. Gauvin. Degeneracy, normality, stability in mathematical programming. In *Recent developments in optimization (Dijon, 1994)*, volume 429 of *Lecture Notes in Econom. and Math. Systems*, pages 136–141. Springer, Berlin, 1995. 42

[69] A.M. Geoffrion and G.W. Graves. Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach. *Operations Research*, 24:595–610, 1976. 91

[70] I. George, J. Lin, and N. Lütkenhaus. Numerical calculations of the finite key rate for general quantum key distribution protocols. *Phys. Rev. Research*, 3:013274, Mar 2021. 100, 128

[71] R. Glowinski and A. Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér.*, 9(R-2):41–76, 1975. 18

[72] R. Glowinski and S.J. Osher, editors. *Splitting methods in communication, imaging, science, and engineering*. Scientific Computation. Springer, Cham, 2016. 18, 75, 77

[73] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995. 35, 94

[74] A.J. Goldman and A.W. Tucker. Theory of linear programming. In *Linear inequalities and related systems*, pages 53–97. Princeton University Press, Princeton, N.J., 1956. Annals of Mathematics Studies, no. 38. 51

[75] R.F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal*, 66(5):1335 – 1340, 1994. 87

[76] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, 3^{nd} edition, 1996. 8

[77] M. Gonzalez-Lima, H. Wei, and H. Wolkowicz. A stable primal-dual approach for linear programming under nondegeneracy assumptions. *Comput. Optim. Appl.*, 44(2):213–247, 2009. 19, 37, 106

[78] N. Gould and J. Scott. A note on performance profiles for benchmarking software. *ACM transactions on mathematical software*, 43(2):1–5, 2016. 57

[79] N. Graham, H. Hu, H. Im, X. Li, and H. Wolkowicz. A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP. *INFORMS J. Comput.*, 34(4):2125–2143, 2022. 12, 15, 18, 29, 70, 81

[80] L. Grippo, L. Palagi, and V. Piccialli. Necessary and sufficient global optimality conditions for NLP reformulations of linear SDP problems. *J. Global Optim.*, 44(3):339–348, 2009. 29, 34

[81] Y. Gu, B. Jiang, and D. Han. A semi-proximal-based strictly contractive Peaceman-Rachford splitting method. *arXiv e-prints*, page arXiv:1506.02221, Jun 2015. 81

[82] B. Guenin, J. Könemann, and L. Tunçel. *A Gentle Introduction to Optimization.* Cambridge University Press, 2014. 40

[83] O. Güler, D. Den Hertog, C. Roos, T. Terlaky, and T. Tsuchiya. Degeneracy in interior point methods for linear programming: a survey. *Ann. Oper. Res.*, 46/47(1-4):107–138, 1993. Degeneracy in optimization problems. 37, 52, 55

[84] J.A.J. Hall and K.I.M. McKinnon. The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling. *Math. Program.*, 100(1, Ser. B):133–150, 2004. 37

[85] B. He, H. Liu, Z. Wang, and X. Yuan. A strictly contractive Peaceman–Rachford splitting method for convex programming. *SIAM Journal on Optimization*, 24(3):1011–1040, 2014. xii, 75, 76

[86] D.R Heffley. Assigning runners to a relay team. In *Optimal strategies in sports*, volume 5, pages 169–171. North Holland Amsterdam, 1977. 91

[87] C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361, 1996. 20

[88] N.J. Higham. *Functions of Matrices: Theory and Computation.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. 108, 110

[89] R.D. Hill and S.R. Waters. On the cone of positive semidefinite matrices. *Linear Algebra Appl.*, 90:81–88, 1987. 112

[90] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Grundlehren Text Editions. Springer-Verlag, Berlin, 2001. Abridged version of ıt Convex analysis and minimization algorithms. I [Springer, Berlin, 1993; MR1261420 (95m:90001)] and ıt II [ibid.; MR1295240 (95m:90002)]. 75, 78

[91] L. Holm and C. Sander. Database algorithm for generating protein backbone and side-chain co-ordinates from a c it trace : Application to model building and detection of co-ordinate errors. *Journal of Molecular Biology*, 218(1):183–194, 1991. 85

[92] H. Hu, H. Im, J. Lin, N. Lütkenhaus, and H. Wolkowicz. Robust interior point method for quantum key distribution rate computation. *Quantum*, 6:792–840, 2022. 122

[93] S. Huang and H. Wolkowicz. Low-rank matrix completion using nuclear norm minimization and facial reduction. *J. Global Optim.*, 72(1):5–26, 2018. 12

[94] X. Huang. Preprocessing and postprocessing in linear optimization. Master's thesis, McMaster University, 2004. 64

[95] S.E. Karisch, F. Rendl, and J. Clausen. solving graph bisection problems with semidefinite programming. Technical Report Report DIKU-Tr-97/9, University of Copenhagen, Dept. of Computer Science, 1997. 35

[96] C.L Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics (Oxford, England)*, 21(7):1028–1039, 2005. 85

[97] V.V. Kovacevic-Vujcic and A.D. Miroslav. Stabilization of interior-point methods for linear programming. *Comput. Optim. Appl.*, 14(3):331–346, 1999. 52

[98] J. Krarup and P.M. Pruzan. Computer-aided layout design. In *Mathematical programming in use*, pages 75–94. Springer, 1978. 91

[99] S. Kruk, M. Muramatsu, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. The Gauss-Newton direction in semidefinite programming. *Optim. Methods Softw.*, 15(1):1–28, 2001. 19, 20, 100

[100] S. Kruk and H. Wolkowicz. Convergence of a short-step primal-dual algorithm based on the Gauss-Newton direction. *J. Appl. Math.*, (10):517–534, 2003. 100

[101] N. Kishore Kumar and J. Schneider. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*, 65(11):2212–2244, 2017. 29

[102] J.B. Lasserre. Linear programming with positive semi-definite matrices. *MPE*, 2:499–521, 1996. 127

[103] V. Laudet and H. Gronemeyer. 3 - ligand binding. In V. Laudet and H. Gronemeyer, editors, *The Nuclear Receptor FactsBook*, Factsbook, pages 37 – 41. Academic Press, London, 2002. 84

[104] F. Leditzky, D. Leung, V. Siddhu, G. Smith, and J.A. Smolin. The platypus of the quantum channel zoo, 2022. 100

[105] C. Lee. Predicting protein mutant energetics by self-consistent ensemble optimization. *Journal of Molecular Biology*, 236(3):918–939, 1994. 85

[106] Y.T. Lee and S.S. Vempala. Tutorial on the robust interior point method. *CoRR*, abs/2108.04734, 2021. 55

[107] A. Lemon, A. Man-Cho So, and Y. Ye. Low-rank semidefinite programming: Theory and applications. *Foundations and Trends® in Optimization*, 2(1-2):1–156, 2016. 29, 31

[108] X. Li, T.K. Pong, H. Sun, and H. Wolkowicz. A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem. *Comput. Optim. Appl.*, 78(3):853–891, 2021. 66, 68, 70

[109] X. Li and X. Yuan. A proximal strictly contractive Peaceman-Rachford splitting method for convex programming with applications to imaging. *SIAM J. Imaging Sci.*, 8(2):1332–1365, 2015. 81

[110] X-B Li, F. Burkowski, and H. Wolkowicz. Semidefinite facial reduction and rigid cluster interpolation in protein structure elastic network models. IEEE BIBM 2016, Waterloo, Ontario, 2016. submitted for refereed conference, Dec. 1, 2016, 10 pages, unpublished. 29

[111] J. Lin. Security proofs for quantum key distribution protocols by numerical approaches, 2017. 110

[112] J. Lin. *Security Analysis of Quantum Key Distribution: Methods and Applications*. PhD thesis, 2021. 100

[113] P.L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979. 77

[114] L.L Looger, M.A Dwyer, J.J Smith, and H.W Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature (London)*, 423(6936):185–190, 2003. 84

[115] J. Mareček and M. Takáč. A low-rank coordinate-descent algorithm for semidefinite programming relaxations of optimal power flow. *Optim. Methods Softw.*, 32(4):849–871, 2017. 29

[116] N.A. Marze, S.S. Roy-Burman, W. Sheffler, and J.J. Gray. Efficient flexible backbone protein-protein docking for challenging targets. *Computer applications in the biosciences*, 34(20):3461–3469, 2018. 84

[117] N. Megiddo. A note on degeneracy in linear programming. *Math. Programming*, 35(3):365–367, 1986. 37

[118] N. Megiddo and R. Chandrasekaran. On the ϵ-perturbation method for avoiding degeneracy. *Oper. Res. Lett.*, 8(6):305–308, 1989. 50

[119] L. Minghui and G. Pataki. Exact duality in semidefinite programming based on elementary reformulations. *SIAM J. Optim.*, 25(3):1441–1454, 2015. 15, 22, 33

[120] D. Molzahn, B. Lesieutre, and C. DeMarco. A sufficient condition for global optimality of solutions to the optimal power flow problem. *IEEE transactions on power systems*, 29(2):978–979, 2014. 29

[121] R.D.C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM J. Optim.*, 7(3):663–678, 1997. 20, 102

[122] M.A. Nielsen and I.L. Chuang. *Quantum computation and quantum information.* Cambridge University Press, Cambridge, 2000. 107, 108

[123] D.E. Oliveira, H. Wolkowicz, and Y. Xu. ADMM for the SDP relaxation of the QAP. *Math. Program. Comput.*, 10(4):631–658, 2018. 12, 15, 18, 29, 66, 70, 93, 94, 95

[124] P. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: a survey and recent developments. In P.M. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems (New Brunswick, NJ, 1993)*, pages 1–42. Amer. Math. Soc., Providence, RI, 1994. 91

[125] P. Pardalos and H. Wolkowicz, editors. *Quadratic assignment and related problems.* American Mathematical Society, Providence, RI, 1994. Papers from the workshop held at Rutgers University, New Brunswick, New Jersey, May 20–21, 1993. 91

[126] P.M. Pardalos and S.A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Global Optim.*, 1(1):15–22, 1991. 69

[127] G. Pataki. Cone-LP's and semidefinite programs: Geometry and a simplex-type method. In Cunningham W. H., McCormick S., and Queyranne M., editors, *Integer programming and combinatorial optimization: 5th international IPCO conference, Vancouver June 3–5, 1996*, pages 162–174. 1996. 127

[128] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358, 1998. 23, 28, 29, 39

[129] G. Pataki. Strong duality in conic linear programming: facial reduction and extended duals. In David Bailey, Heinz H. Bauschke, Frank Garvan, Michel Thera, Jon D. Vanderwerff, and Henry Wolkowicz, editors, *Computational and analytical mathematics*, volume 50 of *Springer Proc. Math. Stat.*, pages 613–634. Springer, New York, 2013. 12

[130] G. Pataki. Characterizing bad semidefinite programs: normal forms and short proofs. *SIAM Rev.*, 61(4):839–859, 2019. 22

[131] J. Peña and V. Roshchina. A data-independent distance to infeasibility for linear conic systems. *SIAM J. Optim.*, 30(2):1049–1066, 2020. 25

[132] D.W. Peaceman and H.H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955. 77

[133] F. Permenter, H. Friberg, and E. Andersen. Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach. *SIAM J. Optim.*, 27(3):1257–1282, 2017. 12, 64

[134] F.N. Permenter. *Reduction methods in semidefinite and conic optimization.* PhD thesis, Massachusetts Institute of Technology, 2017. 13, 127

[135] M.V. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM J. Optim.*, 7(3):641–662, 1997. 12

[136] J. Renegar. Some perturbation theory for linear programming. *Math. Programming*, 65(1, Ser. A):73–91, 1994. 25

[137] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM J. Optim.*, 5(3):506–524, 1995. 25, 27

[138] R. T. Rockafellar. On the maximal monotonicity of subdifferential mappings. *Pacific J. Math.*, 33:209–216, 1970. 78

[139] R.T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970. 10, 81, 84, 118

[140] D. M. Ryan and M. R. Osborne. On the solution of highly degenerate linear programmes. *Math. Program.*, 41:385–392, 1988. 37

[141] R. Samudrala and J. Moult. Determinants of side chain conformational preferences in protein structures. *Protein engineering*, 11(11):991–997, 1998. 85

[142] L. Schork and J. Gondzio. Rank revealing Gaussian elimination by the maximum volume concept. *Linear Algebra Appl.*, 592:1–19, 2020. 28

[143] P.S. Shenkin, H. Farid, and J.S. Fetrow. Prediction and evaluation of side-chain conformations for protein backbone structures. *Proteins: Structure, Function, and Bioinformatics*, 26(3):323–352, 1996. 85

[144] S. Sremac. *Error bounds and singularity degree in semidefinite programming*. PhD thesis, University of Waterloo, 2019. 10, 13, 15, 23, 31

[145] S. Sremac, H.J. Woerdeman, and H. Wolkowicz. Error bounds and singularity degree in semidefinite programming. *SIAM J. Optim.*, 31(1):812–836, 2021. 15, 22, 25, 31

[146] J.F. Sturm. Error bounds for linear matrix inequalities. *SIAM J. Optim.*, 10(4):1228–1248 (electronic), 2000. 12, 15, 22, 25

[147] J. Telgen. Identifying redundant constraints and implicit equalities in systems of linear constraints. *Management Sci.*, 29(10):1209–1222, 1983. 63

[148] T. Terlaky and S.Z. Zhang. Pivot rules for linear programming: a survey on recent theoretical developments. *Ann. Oper. Res.*, 46/47(1-4):203–233, 1993. Degeneracy in optimization problems. 37

[149] K.C. Toh, M.J. Todd, and R.H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11/12(1-4):545–581, 1999. Interior point methods. 55

[150] L. Tunçel. On the Slater condition for the SDP relaxations of nonconvex sets. *Oper. Res. Lett.*, 29(4):181–186, 2001. 68, 69

[151] L. Tunçel. *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*, volume 27 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI, 2010. 12

[152] I. Ugi, J. Bauer, J. Brandt, J. Friedrich, J. Gasteiger, C. Jochum, and W. Schubert. Neue anwendungsgebiete für computer in der chemie. *Angewandte Chemie*, 91(2):99–111, 1979. 91

[153] J. von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. In *Contributions to the theory of games, vol. 2*, Annals of Mathematics Studies, no. 28, pages 5–12. Princeton University Press, Princeton, N. J., 1953. 42, 91

[154] C. Wang, P. Bradley, and D. Baker. Protein-protein docking with backbone flexibility. *Journal of molecular biology*, 373(2):503–519, 2007. 84

[155] J. Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. 100, 108

[156] A. Winick, N. Lütkenhaus, and P.J. Coles. Reliable numerical key rates for quantum key distribution. *Quantum*, 2:77, Jul 2018. 107, 108, 119, 121, 122, 123

[157] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27(3):382–398, 1959. 63

[158] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 20, 29

[159] S. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, 1996. 49, 50, 52, 105

[160] J. Xu and B. Berger. Fast and accurate algorithms for protein side-chain packing. *Journal of the ACM (JACM)*, 53(4):533–557, 2006. 85, 86

[161] L. Yang, D. Sun, and K.-C. Toh. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.*, 7(3):331–366, 2015. 74

[162] Y. Ye, M.J. Todd, and S. Mizuno. An $\mathcal{O}(\sqrt{n}L)$–iteration homogeneous and self–dual linear programming algorithm. *Math. Oper. Res.*, 19:53–67, 1994. 53

[163] V. Zhadan. Two-phase dual simplex method for linear semidefinite optimization. *Proceedings of the VIII International Conference on Optimization and Applications*, pages 591–597, 2017. 127

[164] V.G. Zhadan. A variant of the dual simplex method for a linear semidefinite programming problem. *Tr. Inst. Mat. Mekh.*, 22(3):90–100, 2016. 127

[165] Y. Zhang. On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. *SIAM J. Optim.*, 8:365–386, 1998. 20, 102

[166] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. Semidefinite Programming and Interior-point Approaches for Combinatorial Optimization Problems (Fields Institute, Toronto, ON, 1996). 12, 15, 24, 66, 67, 68, 70, 71, 93, 94

[167] Y. Zhu, G. Pataki, and Q. Tran-Dinh. Sieve-SDP: a simple facial reduction algorithm to preprocess semidefinite programs. *Math. Program. Comput.*, 11(3):503–586, 2019. 15

Appendix A

Supplementary Numerics

A.1 Numerical Results on SCP

Problem Data				Numerical Results			Timing	
#	name	p	n_0	lbd	ubd	rel-gap	iter	time(sec)
1	1AIE	26	34	-46.95892	-46.95892	1.04802e-15	200	0.10
2	2ERL	34	103	55.33285	55.33284	1.17985e-12	200	5.85
3	1CBN	37	112	-40.42751	-40.42751	1.68402e-14	300	7.77
4	1RB9	41	105	-76.96501	-76.96501	7.11964e-13	1000	26.39
5	1BX7	41	99	16.96026	16.96026	5.21525e-12	300	7.25
6	2FDN	42	51	-59.43091	-59.43092	3.71094e-14	200	0.04
7	1MOF	46	94	-79.05580	-79.05580	3.52629e-12	200	4.03
8	1CTF	47	74	-97.18893	-97.18893	4.64633e-13	200	2.81
9	1NKD	50	199	-51.78466	-51.78466	4.40639e-12	2680	192.65
10	2IGD	50	126	-78.50608	-78.50608	5.39611e-15	500	14.67
11	2SN3	53	112	-5.56818	-5.56818	6.73872e-13	700	16.77
12	1MSI	54	112	-87.46958	-87.46958	1.72043e-13	700	19.39
13	1AHO	54	140	24.66925	24.66925	4.19224e-14	1500	56.22
14	1COR	60	131	15.58314	15.58314	4.58637e-12	1000	32.31
15	1CTJ	61	258	-103.32705	-103.32705	1.64217e-12	1872	162.80
16	1RZL	65	121	17.26470	17.26470	1.22992e-11	2468	68.52
17	1TIF	66	614	-155.17859	-155.17859	4.69196e-14	1000	350.89
18	1BDO	69	221	-136.29933	-136.29933	8.93377e-15	1000	75.06
19	1OPD	70	112	-139.64632	-139.64632	1.18233e-13	300	5.98
20	1VQB	75	406	-96.94940	-96.94940	4.34568e-14	900	147.36
21	1IUZ	75	221	-150.88238	-150.88238	1.25791e-14	3200	227.45
22	1ABA	76	376	-137.59962	-137.59963	9.05546e-15	600	88.43
23	1FNA	76	131	-172.01313	-172.01313	3.64100e-14	800	23.32
24	1CYO	78	220	-75.36668	-75.36668	1.36739e-14	700	48.50
25	1FUS	79	302	-4.66627	-4.66627	1.11145e-12	3000	312.35
26	2MCM	80	123	-135.14024	-135.14024	8.30816e-13	400	10.30
27	1SVY	80	147	-141.92437	-141.92437	6.21219e-13	400	14.51
28	1A68	81	424	-178.12555	-178.12555	2.54581e-15	1500	249.80
29	1YCC	84	223	-79.21270	-79.21270	2.11079e-12	955	66.26
30	2ACY	84	580	-146.32254	-146.32254	1.06468e-14	7800	2175.04
31	1BM8	85	687	-119.54537	-119.54537	2.02428e-14	1300	509.88
32	1BKF	89	339	-170.80514	-170.80514	1.60935e-14	1000	117.73
33	3CYR	91	137	-144.06405	-144.06405	2.48290e-12	1900	52.09
34	3VUB	92	544	-229.38312	-229.38312	7.41813e-16	1400	349.67
35	1JER	96	462	-120.78401	-120.78400	1.15131e-12	3232	633.90
36	2HBG	97	275	-178.42210	-178.42210	2.70839e-13	500	42.98
37	1POA	97	470	278.08280	278.08280	2.02964e-12	5463	1099.55
38	1C52	99	256	-223.31096	-223.31096	2.41281e-15	2700	203.46
39	2A0B	99	642	-161.45228	-161.45228	1.75494e-16	5200	1800.90
40	2TGI	100	355	-14.03554	-14.03554	2.46249e-13	1300	153.95

Table A.1.1: Computation results on selected PDB instances up to 100 amino acids

Problem Data				**Numerical Results**			**Timing**	
#	**name**	$\boldsymbol{p}$	$\boldsymbol{n_0}$	**lbd**	**ubd**	**rel-gap**	**iter**	**time(sec)**
41	3NUL	101	285	-154.87542	-154.87542	1.28046e-15	2300	307.34
42	1WHI	101	298	-247.13457	-247.13457	6.94375e-14	1500	199.52
43	1PDO	104	453	-188.29848	-188.29848	9.10541e-12	5754	1456.33
44	3LZT	105	530	-48.81821	-48.81821	8.48591e-13	1100	300.50
45	1DHN	105	519	-133.77464	-133.77464	1.35468e-13	2000	535.83
46	1KUH	106	580	-155.56590	-155.56590	2.18536e-15	2296	743.57
47	1ECA	108	655	-169.74717	-169.74717	1.66944e-16	25200	12563.89
48	1BFG	108	410	-191.73261	-191.73262	8.54577e-14	900	210.84
49	1RIE	108	930	-117.91809	-117.91809	1.57208e-14	20200	17809.01
50	2SAK	111	214	-239.86975	-239.86975	1.08995e-12	500	37.26
51	1BGF	112	1180	-239.65571	-239.65571	1.52549e-13	56400	71503.54
52	2END	118	707	-8.22833	-8.22833	1.08596e-12	16100	8511.24
53	2SNS	119	634	620.86546	620.86546	1.79304e-14	6900	3082.12
54	1BD8	121	347	-219.12419	-219.12419	9.42666e-12	4970	760.81
55	1NPK	122	709	-205.56059	-205.56059	6.77231e-13	59075	31212.37
56	1A6M	124	613	-55.41007	-55.41008	4.93096e-14	22800	7608.82
57	2RN2	127	830	-198.37189	-198.37189	1.41057e-13	6073	4053.13
58	1RCF	130	733	-86.59895	-86.59775	1.38011e-05	100000	56927.20
59	1LCL	131	1246	-217.16433	-217.16433	2.53317e-14	3800	4821.11
60	2CPL	132	819	-284.97180	-284.97180	9.75693e-15	5900	3329.39
61	1VHH	133	844	-21.33604	-21.33604	3.59566e-14	3200	1843.96
62	1BJ7	135	917	-64.37915	-64.37915	5.69493e-14	11300	8946.94
63	119L	136	970	-234.21535	-234.21535	8.01617e-14	34200	30890.87
64	1RA9	136	1018	-185.07235	-185.07235	5.13076e-14	4400	4839.16
65	1L58	137	962	-285.60167	-285.60167	1.31131e-14	15600	13812.60
66	2ILK	142	708	-121.02712	-121.02712	1.82770e-13	4700	2750.13
67	1KOE	144	710	-13.87537	-13.87537	1.27269e-11	4124	2490.08
68	1HA1	146	538	-213.93793	-213.93793	1.44469e-13	3700	1229.31
69	1CEX	146	415	174.95279	174.95279	2.40438e-11	11447	2426.49
70	1CV8	146	730	-213.13554	-213.13554	3.28738e-13	5600	3442.13
71	153L	149	846	-170.13061	-170.13061	3.03488e-13	2100	1554.46
72	1BS9	150	935	103.16569	103.16569	1.31052e-13	2500	1736.57
73	2PTH	151	1198	-190.97344	-190.97344	1.39085e-13	1900	2233.17
74	1XNB	151	1233	-147.30040	-147.30040	2.69217e-15	13300	16562.76
75	1AQB	152	713	29.24537	29.24537	9.30418e-14	39300	17795.39
76	1LBU	152	1225	38.14603	38.14603	1.91397e-13	9900	11673.18
77	1KID	153	653	-351.91160	-351.91160	2.90337e-15	6600	2607.24
78	1CHD	154	489	-164.21510	-164.21510	3.27846e-14	19300	4097.50
79	1AMM	158	1480	-288.62671	-288.62671	2.75245e-15	3300	5793.13
80	2ENG	162	867	82.01797	82.01797	1.33295e-13	14200	8284.65
81	1G3P	165	921	-70.30769	-70.30769	6.66312e-14	7000	4469.99
82	1THV	167	902	5.12749	5.12749	4.63732e-12	4200	2637.88
83	1PPN	170	1259	-56.69346	-56.69346	1.23365e-13	11589	14139.22
84	1IAB	173	775	321.20652	321.20652	2.04964e-14	26500	13017.74
85	1DIN	175	1110	-264.73564	-264.73548	5.84356e-07	100000	93357.26
86	2AYH	176	1269	8428.18154	6089367.83709	1.99447e+00	100000	135879.29
87	1ZIN	177	853	-353.00431	-353.00431	3.18384e-14	23800	13742.52
88	1BYI	177	818	-242.78881	-242.78881	2.33646e-14	2400	1298.65
89	2BAA	178	1165	-43.77265	-43.77265	1.95480e-12	4600	4785.88
90	1A7S	179	524	-239.78218	-239.78218	1.00542e-14	1200	284.88
91	1WAB	183	1063	-317.46713	-317.46713	9.40337e-14	8500	7357.75
92	1MUN	185	1047	-378.01261	-378.01261	1.15635e-14	9500	7883.00
93	1LST	192	946	-244.76861	-244.76861	1.28627e-14	32300	21374.44
94	1GCI	194	1052	-205.63185	-205.63185	2.79899e-14	10300	8885.03
95	3CLA	198	857	-26.72768	-26.72768	9.89051e-14	3900	2287.99

Table A.1.2: Computation results on selected PDB instances up to 200 amino acids

Problem Data				**Numerical Results**			**Timing**	
#	**name**	$\boldsymbol{p}$	$\boldsymbol{n_0}$	**lbd**	**ubd**	**rel-gap**	**iter**	**time(sec)**
96	1AL3	201	1077	119.66598	119.66598	3.39407e-14	12500	10188.87
97	1ARB	202	1466	-61.52823	-61.52823	3.41363e-14	8900	14632.82
98	1XJO	202	776	-171.92443	-171.92443	8.24179e-15	3700	1455.50
99	1NLS	203	1060	-297.73578	-297.73578	5.33677e-15	2500	1976.08
100	1MRJ	208	1178	-295.13711	-295.13711	1.70740e-13	2300	2149.63
101	1OAA	208	854	-317.83422	-317.83422	1.44174e-12	3842	1823.52
102	2DRI	210	906	-398.45564	-398.45564	2.56465e-15	6200	3225.99
103	2CBA	223	1018	-86.52145	-86.52145	5.34000e-14	3400	2407.24
104	2POR	224	1304	-83.22221	-83.22221	5.55044e-14	6700	8044.39
105	3SEB	224	1412	77.15838	77.15852	1.84867e-06	100000	137194.81
106	1MLA	227	1322	-484.10542	-484.10542	1.68910e-14	62900	75257.79
107	1DCS	232	1170	-342.68600	-342.68600	1.39133e-14	8000	7459.07
108	1AKO	234	1387	-244.65691	-244.65691	1.18251e-14	7400	9809.00
109	1PDA	239	891	-423.50226	-423.50226	4.96037e-15	9100	4520.68
110	1EZM	239	1497	-217.36581	-217.36581	3.49620e-13	2300	3919.92
111	1C3D	243	1679	-400.69876	-400.69876	1.04846e-14	22100	134094.53
112	1RHS	244	1973	-341.20443	-341.20443	1.41400e-14	7300	62136.57
113	8ABP	245	1743	-273.90715	-273.90716	2.27865e-15	9000	59868.98
114	1CVL	246	910	-537.04249	-537.04249	2.11494e-16	14800	7522.51
115	1RYC	248	1831	-202.60568	-202.60568	4.81378e-14	15200	84674.22
116	1MRP	248	1648	-350.97062	-350.97062	1.39088e-14	11000	34303.23
117	1IXH	252	1134	-289.75241	-289.75241	4.11267e-14	1300	1087.30
118	1FNC	253	1940	-310.60999	-310.60999	6.54656e-13	34321	292924.91
119	1TCA	255	1062	-422.15387	-422.15387	4.24994e-14	8700	6424.87
120	1SBP	256	1704	-271.08838	-271.08838	3.59996e-14	40000	156330.60
121	2CTC	264	1536	-213.88596	-213.88596	2.17419e-14	15100	43642.85
122	1PGS	265	2190	-16.14049	-16.14049	2.28785e-12	21300	269611.15
123	1MSK	271	1798	-162.51007	-162.50978	1.77573e-06	100000	771330.61
124	1BG6	271	784	-452.62383	-452.62383	3.13620e-15	12700	4935.11
125	1ARU	271	939	-314.40612	-314.40589	7.15908e-07	100000	53858.54
126	1A8E	274	1096	-249.85499	-249.85499	3.58741e-14	96500	78746.74
127	1AXN	278	2343	-300.34291	-300.34291	7.55789e-15	12500	207625.02
128	1TAG	279	1330	-253.22167	-253.22167	1.68029e-14	4300	5038.43
129	1ADS	280	1560	733.91439	733.91440	1.39319e-13	18273	65301.22
130	3PTE	284	2006	161.17216	161.17216	5.09815e-15	13500	59169.60
131	1CEM	292	2400	-24.20196	-24.20196	3.85446e-14	7000	47701.70

Table A.1.3: Computation results on selected PDB instances up to 300 amino acids

A.2 Numerical Results on QAP

Problem Data			**Numerical Results**					**Timing**			
#	**name**	**true-opt**	**lbd**	**ubd**	**rel.gap**	**rel.opt.gap**	**rel.gap**$^{\mathbf{A}}$	**iter**	**iter**$^{\mathbf{A}}$	**time**	**time**$^{\mathbf{A}}$
47	chr22a	6156	6156	6156	**0**	0	0.02	4700	40000	120.34	937.29
48	chr22b	6194	6194	6194	**0**	0	0.11	6900	39300	184.47	965.52
49	chr25a	3796	3796	3796	**0**	0	0	3300	35600	155.01	1380.29
50	esc32a	130	104	168	**46.89**	22.13	106.07	9100	18200	1404.41	2591.16
51	esc32b	168	132	200	**40.84**	23.92	83.00	12000	4000	1875.72	573.07
52	esc32c	642	616	644	**4.44**	4.13	27.43	5400	1700	881.03	256.19
53	esc32d	200	192	208	**7.98**	4.07	54.37	4300	1400	670.15	202.91
54	esc32e	2	2	20	156.52	0	141.18	10700	3000	1653.70	435.27
55	esc32g	6	6	6	**0**	0	26.67	400	900	63.96	135.61
56	esc32h	438	426	448	**5.03**	2.77	33.46	12300	11300	1895.42	1638.85
57	kra30a	88900	86838	94750	**8.71**	2.35	16.50	9900	3700	1110.91	404.34
58	kra30b	91420	87858	100200	**13.13**	3.97	27.87	12800	4900	1461.35	537.19
59	kra32	88700	85776	92800	**7.87**	3.35	35.29	11100	4100	1720.85	626.41
60	nug21	2438	2382	2546	**6.65**	2.32	12.36	10500	5600	245.97	116.02
61	nug22	3596	3530	3678	**4.11**	1.85	12.76	11100	7400	296.19	195.65
62	nug24	3488	3402	3744	**9.57**	2.50	16.25	10800	4300	412.96	160.04
63	nug25	3744	3626	3798	**4.63**	3.20	15.37	11600	7500	528.03	343.50
64	nug27	5234	5130	5364	**4.46**	2.01	17.08	11000	8400	756.87	552.20
65	nug28	5166	5026	5466	**8.39**	2.75	18.55	10900	7200	854.10	536.63
66	nug30	6124	5950	6530	**9.29**	2.88	19.83	13000	8800	1424.95	908.56
67	ste36a	9526	9260	10204	**9.70**	2.83	42.28	24200	27300	7469.48	7694.55
68	ste36b	15852	15668	18770	**18.01**	1.17	82.03	25800	40000	7770.51	11593.78
69	ste36c	8239110	8134756	8302154	**2.04**	1.27	36.15	40000	40000	11854.97	11466.03
70	tai25a	1167256	1096658	1264590	**14.22**	6.24	20.56	1900	800	86.44	33.82
71	tai30a	1818146	1706872	1970990	**14.36**	6.31	15.21	4700	1400	514.10	143.77
72	tai35a	2422002	2216648	2672342	**18.64**	8.85	22.34	3000	1500	760.22	353.29
73	tai40a	3139370	2843314	3461270	**19.60**	9.90	23.43	5700	2200	2928.65	1118.45
74	tho30	149936	143576	166336	**14.69**	4.33	24.33	18300	7400	2016.71	803.19
75	tho40*	240516	226522	256890	**12.56**	5.99	26.25	16400	12200	8052.26	6229.43

Table A.2.1: QAPLIB instances of medium size

Problem Data			**Numerical Results**					**Timing**			
#	**name**	**true-opt**	**lbd**	**ubd**	**rel.gap**	**rel.opt.gap**	**rel.gap**$^{\mathbf{A}}$	**iter**	**iter**$^{\mathbf{A}}$	**time**	**time**$^{\mathbf{A}}$
76	esc64a	116	98	210	**72.49**	16.74	75.71	1100	1200	8837.48	9706.63
77	sko42	15812	15336	16270	**5.91**	3.06	17.24	11500	10700	7532.99	7039.99
78	sko49	23386	22654	24246	**6.79**	3.18	16.59	18900	16900	31388.90	27587.39
79	sko56	34458	33390	36598	**9.17**	3.15	16.60	13100	15100	46346.63	53804.58
80	sko64	48498	47022	50316	**6.77**	3.09	15.54	14300	21100	113747.66	164377.85
81	tai50a	4938796	4390982	5467512	**21.84**	11.74	25.79	2300	3300	4252.22	6026.14
82	tai60a	7205962	6326350	7915088	**22.31**	13.00	26.03	3400	5100	18191.52	26989.50
83	tai64c	1855928	1811366	1901250	**4.84**	2.43	38.79	2500	2400	20382.20	19268.53
84	wil50	48816	48126	50382	**4.58**	1.42	9.38	10100	11000	19151.38	20487.55

Table A.2.2: QAPLIB instances of large size

Appendix B

Supplementary Proofs and Notes

B.1 The Adjoints of $\mathcal{M}_\rho$ and $\mathcal{M}_Z$

We present an explicit representation of the linearized system for (6.1.3). We first present some preliminary results regarding the adjoint.

We define some mappings that are used in this section. We define the *symmetrization linear map,* $\mathcal{S}$, as $\mathcal{S}(M) = (M + M^*)/2$. The *skew-symmetrization linear map,* $\mathcal{SK}$, is $\mathcal{SK}(M) = (M - M^*)/2$. Any matrix $M \in \mathbb{C}^{n\times n}$ can be decomposed by

$$M = \frac{1}{2}(M + M^*) + \frac{1}{2}(M - M^*) = \mathcal{S}(M) + \mathcal{SK}(M).$$

Lemma B.1.1 (adjoint of $\mathcal{W}(R) := WR$)**.** *Given $W \in \mathbb{C}^{n\times n}$, define the (left matrix multiplication) linear map $\mathcal{W} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ by $\mathcal{W}(R) = WR$. Then the adjoint $\mathcal{W}^* : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is defined by*

$$\mathcal{W}^*(M) = \Re(W)^T\Re(M) + \Im(W)^T\Im(M) + i\left(\Re(W)^T\Im(M) - \Im(W)^T\Re(M)\right). \tag{B.1.1}$$

If $W \in \mathbb{H}^n$ and $\mathcal{W} : \mathbb{H}^n \to \mathbb{C}^{n\times n}$, then the adjoint $\mathcal{W}^* : \mathbb{C}^{n\times n} \to \mathbb{H}^n$ is defined by

$$\mathcal{W}^*(M) = \mathcal{S}\left[\Re(W)\Re(M) - \Im(W)\Im(M)\right] + i\,\mathcal{SK}\left[\Im(W)\Re(M) + \Re(W)\Im(M)\right]. \tag{B.1.2}$$

Proof. Let $M \in \mathbb{C}^{n\times n}$. By (2.1.2), we have

$$\langle \mathcal{W}(R), M\rangle = \Re(\langle \mathcal{W}(R), M\rangle) = \operatorname{trace}\left(\Re(\mathcal{W}(R))^T\Re(M)\right) + \operatorname{trace}\left(\Im(\mathcal{W}(R))^T\Im(M)\right). \tag{B.1.3}$$

We note that

$$\begin{array}{rcl} \mathcal{W}(R) = WR &=& (\Re(W) + i\,\Im(W))(\Re(R) + i\,\Im(R)) \\ &=& \Re(W)\Re(R) - \Im(W)\Im(R) + i\Re(W)\Im(R) + i\,\Im(W)\Re(R). \end{array}$$

Hence, the first term in (B.1.3) becomes

$$\begin{array}{rcl} \langle \Re(\mathcal{W}(R)), \Re(M)\rangle &=& \langle \Re(W)\Re(R) - \Im(W)\Im(R), \Re(M)\rangle \\ &=& \operatorname{trace}\left(\Re(R)^T\Re(W)^T\Re(M)\right) - \operatorname{trace}\left(\Im(R)^T\Im(W)^T\Re(M)\right) \\ &=& \left\langle \Re(R), \Re(W)^T\Re(M)\right\rangle - \left\langle \Im(R), \Im(W)^T\Re(M)\right\rangle, \end{array}$$

and the second term in (B.1.3) follows similarly:

$$\begin{array}{rcl}\langle \Im(\mathcal{W}(R)), \Im(M)\rangle & = & \langle \Re(W)\,\Im(R) + \Im(W)\Re(R), \Im(M)\rangle \\ & = & \operatorname{trace}\left(\Im(R)^T\Re(W)^T\,\Im(M)\right) + \operatorname{trace}\left(\Re(R)^T\,\Im(W)^T\,\Im(M)\right) \\ & = & \left\langle \Im(R), \Re(W)^T\,\Im(M)\right\rangle + \left\langle \Re(R), \Im(W)^T\,\Im(M)\right\rangle .\end{array}$$

Thus, by linearity, we have

$$\left\langle \Re(R), \Re(W)^T\,\Re(M) + \Im(W)^T\,\Im(M)\right\rangle + \left\langle \Im(R), -\,\Im(W)^T\,\Re(M) + \Re(W)^T\,\Im(M)\right\rangle . \qquad \text{(B.1.4)}$$

By (2.1.2),

$$\begin{array}{rcl}\Re(\mathcal{W}^*(M)) & = & \Re(W)^T\,\Re(M) + \Im(W)^T\,\Im(M) \\ \Im(\mathcal{W}^*(M)) & = & \Re(W)^T\,\Im(M) - \Im(W)^T\,\Re(M),\end{array}$$

and (B.1.1) follows.

Now suppose that $\mathcal{W}$ is a linear map from $\mathbb{H}^n$ and let $R \in \mathbb{H}^n$. Then

$$\Re(W), \Re(\mathcal{W}(R)) \in \mathbb{S}^n, \quad \text{and} \quad \Im(W), \Im(\mathcal{W}(R)) \in \mathbb{S}^n_{\text{skew}}.$$

Let $M \in \mathbb{C}^{n\times n}$. We note that

$$\begin{array}{rcll}\langle \Re(R), \Re(W)^T\,\Re(M)\rangle & = & \langle \Re(R), \Re(W)\,\Re(M)\rangle & \text{since } \Re(W) \in \mathbb{S}^n \\ & = & \langle \Re(R), \Re(M)^T\,\Re(W)^T\rangle & \text{since } \Re(R) \in \mathbb{S}^n,\end{array}$$

and

$$\begin{array}{rcll}\langle \Re(R), \Im(W)^T\,\Im(M)\rangle & = & -\langle \Re(R), \Im(W)\,\Im(M)\rangle & \text{since } \Im(W) \in \mathbb{S}^n_{\text{skew}} \\ & = & -\langle \Re(R), \Im(M)^T\,\Im(W)^T\rangle & \text{since } \Re(R) \in \mathbb{S}^n.\end{array}$$

Hence the first term in (B.1.4) is equal to

$$\left\langle \Re(R), \Re(W)^T\,\Re(M) + \Im(W)^T\,\Im(M)\right\rangle = \langle \Re(R), \mathcal{S}\left(\Re(W)\,\Re(M) - \Im(W)\,\Im(M)\right)\rangle .$$

Now for the second term in (B.1.4), we observe that

$$\begin{array}{rcll}\left\langle \Im(R), -\,\Im(W)^T\,\Re(M)\right\rangle & = & \langle \Im(R), \Im(W)\,\Re(M)\rangle & \text{since } \Im(W) \in \mathbb{S}^n_{\text{skew}} \\ & = & \left\langle \Im(R), -\Re(W)^T\,\Im(M)^T\right\rangle & \text{since } \Im(R) \in \mathbb{S}^n_{\text{skew}},\end{array}$$

and

$$\begin{array}{rcll}\left\langle \Im(R), \Re(W)^T\,\Im(M)\right\rangle & = & \langle \Im(R), \Re(W)\,\Re(M)\rangle & \text{since } \Re(W) \in \mathbb{S}^n \\ & = & \left\langle \Im(R), -\,\Re(M)^T\,\Re(W)^T\right\rangle & \text{since } \Im(R) \in \mathbb{S}^n_{\text{skew}}.\end{array}$$

Hence the second term in (B.1.4) is equal to

$$\left\langle \Im(R), -\,\Im(W)^T\,\Re(M) + \Re(W)^T\,\Im(M)\right\rangle = \langle \Im(R), \mathcal{SK}\left(\Im(W)\,\Re(M) + \Re(W)\,\Im(M)\right)\rangle .$$

□

Lemma B.1.2 (adjoint of $\rho(S) = S\rho$)**.** *Given $\rho \in \mathbb{C}^{n\times n}$, define the (right matrix multiplication) linear map $\rho : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ by $\rho(S) = S\rho$. Then the adjoint $\rho^* : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is defined by*

$$\rho^*(M) = \Re(M)\Re(\rho)^T + \Im(M)\,\Im(\rho)^T + i\left(-\Re(M)\,\Im(\rho)^T + \Im(M)\Re(\rho)^T\right). \qquad \text{(B.1.5)}$$

If $\rho \in \mathbb{H}^n$ and $\rho : \mathbb{H}^n \to \mathbb{C}^{n\times n}$, then the adjoint $\rho^* : \mathbb{C}^{n\times n} \to \mathbb{H}^n$ is defined by

$$\rho^*(M) = \mathcal{S}\left[\Re(M)\Re(\rho) - \Im(M)\,\Im(\rho)\right] + i\,\mathcal{S}\mathcal{K}\left[\Re(M)\,\Im(\rho) + \Im(M)\Re(\rho)\right]. \tag{B.1.6}$$

B.2 FR for a General Polyhedral Set

In this section we provide the auxiliary system for computing an exposing vector of the set $\mathcal{H}$ defined in (7.2.1) and outline the **FR** process for $\mathcal{H}$. We introduce nonnegative slack variables $s_{\text{ineq}} \in \mathbb{R}^{m_{\text{ineq}}}_+$, $s_1 \in \mathbb{R}^{n_{x1}}_+$, $s_2 \in \mathbb{R}^{n_{x2}}_+$, $s_3, s_4 \in \mathbb{R}^{n_{x3}}_+$ to transform the constraints in (7.2.1) as below:

$$A_{\text{eq}}x = b_{\text{eq}},\ A_{\text{ineq}}x + s_{\text{ineq}} = b_{\text{ineq}},\ x_1 - s_1 = \hat{\ell},\ x_2 + s_2 = \hat{u},\ x_3 - s_3 = \bar{\ell},\ x_3 + s_4 = \bar{u}.$$

We can represent these equalities in a compact form

$$L := \left\{(x;s) : \bar{A}(x;s) = \bar{b},\ x = (x_1;x_2;x_3;x_4),\ s = (s_{\text{ineq}};s_1;s_2;s_3;s_4)\right\},$$

where

$$\bar{A} := \begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & s_{\text{ineq}} & s_1 & s_2 & s_3 & s_4 \end{array} \\ \left[\begin{array}{ccccccccc} [& A_{\text{eq}} & &] & & & & & \\ [& A_{\text{ineq}} & &] & I & & & & \\ I & & & & & -I & & & \\ & I & & & & & I & & \\ & & I & & & & & -I & \\ & & I & & & & & & I \end{array}\right] \end{array},\quad \bar{b} := \begin{pmatrix} b_{\text{eq}} \\ b_{\text{ineq}} \\ \hat{\ell} \\ \hat{u} \\ \bar{\ell} \\ \bar{u} \end{pmatrix} \in \mathbb{R}^{m_{\text{eq}}+m_{\text{ineq}}+n_{x_1}+n_{x_2}+2n_{x_3}}.$$

We let

$$\mathcal{K} := \mathbb{R}^n \oplus \mathbb{R}^{m_{\text{ineq}}}_+ \oplus \mathbb{R}^{n_{x1}}_+ \oplus \mathbb{R}^{n_{x2}}_+ \oplus \mathbb{R}^{n_{x3}}_+ \oplus \mathbb{R}^{n_{x3}}_+.$$

Then

$$\mathcal{H} \text{ fails strict feasibility} \iff L \cap \operatorname{int}(\mathcal{K}) = \emptyset.$$

Then by the hyperplance separation theorem (see [50, Section 3.1]) we can deduce that there exists a vector y such that

$$\bar{A}^T y \in \mathcal{K}^* \setminus \{0\},\ \langle \bar{b}, y\rangle = 0,\ y = (y_{\text{eq}};y_{\text{ineq}};y_1;y_2;y_3;y_4). \tag{B.2.1}$$

We restate the system (B.2.1) as below:

$$\begin{array}{l} {A_{\text{eq}}}^T y_{\text{eq}} + {A_{\text{ineq}}}^T y_{\text{ineq}} + \begin{pmatrix} 0 \\ -y_1 \\ y_2 \\ -y_3 + y_4 \end{pmatrix} = 0_n, \\ \langle b_{\text{eq}}, y_{\text{eq}}\rangle + \langle b_{\text{ineq}}, y_{\text{ineq}}\rangle + \langle -\hat{\ell}, y_1\rangle + \langle \hat{u}, y_2\rangle + \langle -\bar{\ell}, y_3\rangle + \langle \bar{u}, y_4\rangle = 0, \\ y_{\text{eq}} \text{ free},\ (y_{\text{ineq}};y_1;y_2;y_3;y_4) \in \mathbb{R}^{m_{\text{ineq}}+n_{x_1}+n_{x_2}+2n_{x_3}}_+ \setminus \{0\}. \end{array} \tag{B.2.2}$$

We note that the system (B.2.2) is a generalization of the system (2.3.4) (after reducing to a polyhedral cone) and the system (4.3.6) in Lemma 4.3.3. For some i, where i is an index associated with slack variables, we solve the subproblem of the type

$$p_i^* = \max\{s_i : (x;s) \in L \cap \mathcal{K}\},$$

where p_i^* indicates if s_i is a variable fixed at 0 or not. If $p_i^* = 0$, the dual feasibility provides a solution of the system (B.2.2) and thus provides an exposing vector for $\mathcal{H}$, a certificate indicating failure of strict feasibility.

B.3 The FR Algorithm via GN Method

We first observe some properties of the primal-dual pair (7.2.2) and (7.2.3). The role of $P \succ 0$ in (7.2.2) is to prevent the solution exposing vector $\mathcal{A}^*(y)$ from being 0. It is unclear if strong duality holds for the primal-dual pair. We show in Lemma B.3.1 that there always exists $P \succ 0$ so that (7.2.3) holds the Slater constraint qualification.

Lemma B.3.1. *Let $\mathcal{F}$ be feasible. Then the followings hold.*

1. *There exists $P \succ 0$ that leads $(\mathcal{D}_{\boldsymbol{FR}})$ contain a Slater point.*
2. *If $\{A_i\}_{i=1}^m$ contains a positive definite or negative definite matrix, the Slater condition holds for $(\mathcal{P}_{\boldsymbol{FR}})$.*

Proof. 1. Let $\bar{W} \succeq 0$ be any feasible point to $\mathcal{F}$. Let

$$\bar{W} = \begin{bmatrix} V & U \end{bmatrix} \begin{bmatrix} D_V & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V & U \end{bmatrix}^T$$

be the spectral decomposition of $\bar{W}$. We let $\bar{T} = UD_UU^T \succeq 0$, where $D_U \succ 0$. Consider $\hat{W} = 2\bar{W} + 2\bar{T}$ and $P = 2\bar{T} + \bar{W}$. Clearly, both $\hat{W}$ and P are positive definite. We note that

$$\mathcal{A}(\hat{W}) - 1 \cdot \mathcal{A}(P) = \mathcal{A}(\hat{W} - P) = \mathcal{A}(\bar{W}) = b.$$

Hence, (7.2.3) contains a strictly feasible point.

2. Without loss of generality, let A_1 be positive definite. Then choosing $\bar{y} = (\alpha/\langle A_1, P\rangle, 0, \dots, 0)^T$ gives the desired property. If A_1 is negative definite, we may replace $\langle A_1, X\rangle = b_1$ with $\langle -A_1, X\rangle = -b_1$ and apply the above.

□

A consequence of Item 1 of Lemma B.3.1 us that $p^*_{\mathbf{FR}} = d^*_{\mathbf{FR}} = 0$ and the exposing vector $\mathcal{A}^*(y)$ is *attained*. Item 2 of Lemma B.3.1 is applicable to the case where the variables are restricted to be a density matrix. In this case, we have the strong duality for both the primal and the dual.

In addition to the stopping conditions discussed in Section 6.1.3, we can include additional conditions for early termination. Suppose that the dual feasibility is achieved with a positive dual objective value. Then the primal objective value is never 0 hence it certifies that $\mathcal{F}$ has a strictly feasible point. Similarly, if the primal feasibility is achieved with a negative primal objective value, then the problem is infeasible. In this case, the vector $\mathcal{A}^*(y)$ provides a certificate of infeasibility.

Optimality Conditions We subtract the slack variable $S \in \mathbb{S}^n_+$ to (7.2.2) and obtain the perturbed optimality conditions the primal-dual pair $(\mathcal{P}_{\mathbf{FR}})$ and $(\mathcal{D}_{\mathbf{FR}})$, with $W, S \succ 0, \mu > 0$:

$$0 = F_\mu(y, \lambda, W) = \begin{bmatrix} F^d_\mu \\ F^{p,1}_\mu \\ F^{p,2}_\mu \\ F^c_\mu \end{bmatrix} = \begin{array}{ll} \mathcal{A}(W) + \lambda \mathcal{A}(P) - b & \text{dual feasibility} \\ \text{trace}(\mathcal{A}^* y) - \alpha & \text{primal feasibility 1} \\ \mathcal{A}^* y - S & \text{primal feasibility 2} \\ WS - \mu I & \text{complementary slackness.} \end{array} \tag{B.3.1}$$

We may substitute equality $\mathcal{A}^*(y) = S$ into the complementarity equation. However, the interior point method requires $\mathcal{A}^*(y)$ to be positive definite, and $\mathcal{A}^* y \succ 0$ might not be possible depending on the data given. Hence, we work with (B.3.1) directly *without eliminating* primal feasibility 1 $(\mathcal{A}^*(y) - S)$.

Jacobian System We obtain the null-space representations for the dual feasibility

$$F^d_\mu(W, \lambda) = \mathcal{A}(W) + \lambda \mathcal{A}(P) - b = \begin{bmatrix} \mathcal{N}_W \\ \mathcal{N}_\lambda \end{bmatrix} v + \begin{bmatrix} \hat{W} \\ \hat{\lambda} \end{bmatrix} - \begin{bmatrix} W \\ \lambda \end{bmatrix} = \begin{bmatrix} F^{d,1}_\mu \\ F^{d,2}_\mu \end{bmatrix} \begin{array}{l} \text{(dual feasibility 1)} \\ \text{(dual feasibility 2)} \end{array}$$

as well as for the primal feasibility

$$\begin{bmatrix} F^{p,1}_\mu \\ F^{p,2}_\mu \end{bmatrix} = \begin{bmatrix} \text{trace}(P\mathcal{A}^*(y)) - \alpha \\ \mathcal{A}^* y - S \end{bmatrix} = \begin{bmatrix} \mathcal{N}_y \\ \mathcal{N}_S \end{bmatrix} u + \begin{bmatrix} \hat{y} \\ \hat{S} \end{bmatrix} - \begin{bmatrix} y \\ S \end{bmatrix} \begin{array}{l} \text{(primal feasibility 1)} \\ \text{(primal feasibility 2).} \end{array}$$

Consequently, the perturbed optimality conditions become

$$0 = \begin{bmatrix} F^{d,1}_\mu \\ F^{d,2}_\mu \\ F^{p,1}_\mu \\ F^{p,2}_\mu \\ F^c_\mu \end{bmatrix} = \begin{bmatrix} \mathcal{N}_W v + \hat{W} - W \\ \mathcal{N}_\lambda v + \hat{\lambda} - \lambda \\ \mathcal{N}_y u + \hat{y} - y \\ \mathcal{N}_S u + \hat{S} - S \\ WS - \mu I \end{bmatrix}. \tag{B.3.2}$$

We obtain the Gauss-Newton search direction by solving the system

$$F'_\mu d_{GN} = \begin{bmatrix} \mathcal{N}_W \Delta v - \Delta W \\ \mathcal{N}_\lambda \Delta v - \Delta \lambda \\ \mathcal{N}_y \Delta u - \Delta y \\ \mathcal{N}_S \Delta u - \Delta S \\ \Delta W S + W(\Delta S) \end{bmatrix} = - \begin{bmatrix} F^{d,1}_\mu \\ F^{d,2}_\mu \\ F^{p,1}_\mu \\ F^{p,2}_\mu \\ F^c_\mu \end{bmatrix}.$$

Projected Gauss-Newton Direction We note that the first four blocks give

$$\Delta W = \mathcal{N}_W \Delta v + F^{d,1}_\mu, \ \Delta \lambda = \mathcal{N}_\lambda \Delta v + F^{d,2}_\mu, \ \Delta y = \mathcal{N}_y \Delta u + F^{p,1}_\mu, \ \Delta S = \mathcal{N}_S \Delta u + F^{p,2}_\mu. \tag{B.3.3}$$

We use $(\Delta W, \Delta S)$ to make substitutions into the last block

$$\begin{array}{rcl} (F_\mu^c)' d_{GN} & = & \Delta W(S) + W(\Delta S) \\ & = & (\mathcal{N}_W \Delta v + F_\mu^{d,1})S + W(\mathcal{N}_S \Delta u + F_\mu^{p,2}) \\ & = & (\mathcal{N}_W \Delta v)S + (F_\mu^{d,1})S + W(\mathcal{N}_S \Delta u) + W F_\mu^{p,2} \\ & = & -F_\mu^c. \end{array}$$

Rearranging the terms, we solve the following system

$$(\mathcal{N}_W \Delta v)S + W(\mathcal{N}_S \Delta u) = -F_\mu^c - (F_\mu^{d,1})S - W(F_\mu^{p,2}). \tag{B.3.4}$$

and backsolve to complete the remaining components for $(\Delta W, \Delta\lambda, \Delta y, \Delta S)$ using $(\Delta v, \Delta u)$; see (B.3.3). We then update the iterate with these directions

$$y \leftarrow y + \alpha_y \Delta y, \;\; \lambda \leftarrow \lambda + \alpha_\lambda \Delta\lambda, \;\; W \leftarrow W + \alpha_W \Delta W, \;\; S \leftarrow S + \alpha_S \Delta S.$$

The stepsizes $\alpha_y, \alpha_\lambda, \alpha_W, \alpha_S$ are chosen to maintain sufficient positive definiteness of W and S.

Index