A Blueprint for Semidefinite Relaxations of Binary-Constrained Quadratic Programs

Computing tight bounds on NP-hard problems using ADMM

by

Naomi Graham

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics & Optimization

Waterloo, Ontario, Canada, 2020

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

This thesis looks at the solution techniques of two NP-hard, large scale problems, recently presented in [7, 16] , i.e., the *quadratic assignment problem,* ***QAP***, and the *side chain positioning,* ***SCP***, problem. We summarize the approaches from [7, 16] and look at the two problems in a unified way using a *binary-constrained quadratic program,* ***BCQP***.

We show how to obtain upper and lower bounds on their optimal values by formulating the semidefinite programming (**SDP**) relaxation and applying the Alternating Direction Method of Multipliers (**ADMM**) algorithm to solve it.

Both problems, seemingly different on the surface, can we solved using a totally analogous approach. For this reason we were motivated to examine why this is the case. As a result, we have found that both problems can be expressed as an integer quadratic program with a binary linear constraint. We have therefore decided to derive the process of obtaining the **SDP** relaxation and computing bounds via **ADMM** for the more general problem of which **SCP** and **QAP** are shown to be particular cases. In doing this, we can provide a motivation for every step of the process by examining the shared structure of these two problems. Lastly, we show by nothing that the feasible set of the **SDP** is contained a proper face of the **SDP** cone, we obtain a splitting of the primal variables in a natural way, providing a good setting for **ADMM**.

By unifying the two problems under the umbrella of the **BCQP** , we better understand why the method is so successful for these two problems and obtain a blueprint for applying **ADMM** to similar combinatorial optimization problems.

Acknowledgements

I would like to thank my advisor Henry Wolkowicz for giving me the opportunity to study in the exciting area of convex optimization. Doing research with someone with his level expertise in this field has been an incredible learning experience that I feel deeply privileged to have had.

I would next like to thank the dear friends I have made during my time in Waterloo. I am blown away by the number of brilliant and kind people I have met in this department and truly believe it is a place unlike any other. I would especially like to thank my office mates for supplying me with a steady stream of encouragement, laughs, and most importantly, coffee.

Finally, I would like to thank my research collaborators with whom it has been an absolute pleasure to work with. I am particularly grateful to Haesol Im, whose advice and support made much of the work in this thesis possible.

Table of Contents

Chapter 1

Introduction

Let $\mathbb{S}^n$ denote the space on $n \times n$ symmetric matrices. A matrix $X \in \mathbb{S}^n$ is called *positive semidefinite* if $x^T X x \geq 0$ for all $x \in \mathbb{R}^n$. The set of $n \times n$ positive semidefinite matrices is denoted by $\mathbb{S}^n_+$. We also use the notation $X \succeq 0$ to denote that X is positive semidefinite when the dimension is clear.

Let $\mathcal{A}$ be a linear map from $\mathbb{S}^n \to \mathbb{R}^m$ and let $b \in \mathbb{R}^m$. Let $C, X \in \mathbb{R}^{n \times n}$ and let $\langle C, X \rangle = \text{trace}(C^T X)$ denote the trace inner product of C and X. A semidefinite program is an optimization problem of the form:

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0. \end{array} \tag{1.0.1}$$

Semidefinite programming (**SDP**) is useful in forming relaxations of hard combinatorial problems, a notable example being the famous approximation algorithm for MAX-CUT by Goemans and Williamson [15]. This thesis looks at the semidefinite programming techniques for relaxations of two NP-hard combinatorial problems recently presented in [7, 16]: the *Quadratic Assignment Problem,* ***QAP***, and the *Side Chain Positioning,* ***SCP***, problem. We summarize the computational techniques used in [7, 16] by viewing the two problems as cases of a more general class of problems which we call the *binary-constrained quadratic program,* ***BCQP***.

We introduce the **BCQP** now, and provide further details on its associated graph theoretic problem are presented in Chapter 3. The **BCQP** is a particular case of an *integer quadratic program,* ***IQP***. An **IQP** is an optimization problem of the form:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \in \{0, 1\}^n, \end{array} \tag{1.0.2}$$

for a given linear transformation $A : \mathbb{R}^n \to \mathbb{R}^m$, a vector $b \in \mathbb{R}^m$, and some real symmetric

matrix $Q \in \mathbb{S}_n$. The **BCQP** is a particular instance of an **IQP** (1.0.2) wherein the *constraint matrix* A is a *binary matrix*, meaning that all entries of A are either 0 or 1, and where $b \in \mathbb{R}^m$ is the all ones vector which we denote $\bar{e}_m$, where the subscript m gives the size of the vector.

Quadratic integer programming is known to be NP-hard [9], but in some cases, by forming the **SDP** relaxation, we obtain a convex program which can be solved by interior point methods [2], or first order methods, such as Alternating Direction Method of Multipliers (**ADMM**) [24].

A standard approach to forming the **SDP** relaxation of a binary **IQP** (1.0.2) is via a direct lifting strategy (see [2, 23]). We give a rough summary of this process applied to a general **IQP** for context, and demonstrate this process applied to the **BCQP** in detail in Chapter 3. To obtain an **SDP** relaxation for (1.0.2), we first form an equivalent problem by lifting the variable $x \in \mathbb{R}^n$ to the space of $(n+1) \times (n+1)$ symmetric matrices, which we denote by $\mathbb{S}^{n+1}$. We define the matrix:

$$Y_x := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \in \mathbb{S}^{n+1}, \quad (x \in \mathbb{R}^n),$$

and reformulate the constraints on x in (1.0.2) into constraints on this higher dimensional variable Y_x. We show in Section 3.3.1 that by making the appropriate choice of linear map $\mathcal{A}$, vector $d \in \mathbb{R}^m$, and matrix $C \in \mathbb{S}^{n+1}$, that it is possible to obtain an equivalent *lifted problem* of the form

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & \langle C, Y_x \rangle \\ \text{s.t.} & \mathcal{A}(Y_x) = d. \end{array} \tag{1.0.3}$$

The problem above is the reformulated version of (1.0.2). To obtain an **SDP** relaxation of (1.0.3), we relax the rank-one constraint by replacing Y_x with $Y \in \mathbb{S}^{n+1}_+$ to obtain an **SDP** relaxation of (1.0.2). The resulting **SDP** has the general form:

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n+1}} & \langle C, Y \rangle \\ \text{s.t.} & \mathcal{A}(Y) = d \\ & Y \succeq 0. \end{array} \tag{1.0.4}$$

Further refining of the model is possible as well, i.e., we may also add the constraint of elementwise nonnegativity on Y, which we denote by $Y \geq 0$. We then call the resulting **SDP** relaxations a *doubly non-negative,* ***DNN*** relaxation, since the variable Y is nonnegative both in the sense that $Y \succeq 0$ *and* $Y \geq 0$.

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n+1}} & \langle C, Y \rangle \\ \text{s.t.} & \mathcal{A}(Y) = d \\ & Y \succeq 0 \\ & Y \geq 0. \end{array} \tag{1.0.5}$$

One notable advantage gained in using **ADMM** over interior point methods is that non-

negativity constraints are no longer computationally expensive to compute, making the **DNN** relaxation more tractable [24].

It is often the case that the feasible set of (1.0.4) is contained is a proper face (see Section 2.3 for definitions of face and proper face) of the **SDP** cone. We can rephrase this as: all feasible Y satisfy $Y = VRV^T$, where range$(Y) \subseteq$ range(V). This leads to an equivalent formulation of (1.0.5), the optimization problem to which we will apply **ADMM**:

$$\begin{aligned} \inf_{Y \in \mathbb{S}^{n+1}, R \in \mathbb{S}^k} \quad & \langle C, Y \rangle \\ \text{s.t.} \quad & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}. \end{aligned} \tag{1.0.6}$$

We use constraints from (1.0.4) along with the relationship $Y = VRV^T$ to define sets $\mathcal{R}$ and $\mathcal{Y}$. To better understand the motivation for constructing this model, it is necessary to consider the algorithm that we apply to solve it. In [5] Boyd et al. define the general setting for which we can apply **ADMM** to be problems of the form:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^k} \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = c, \end{aligned} \tag{1.0.7}$$

where $f \colon \mathbb{R}^n \to \mathbb{R}$ and $g \colon \mathbb{R}^k \to \mathbb{R}$ are convex functions, A, B are linear transformations from $\mathbb{R}^n \to \mathbb{R}^m$ and $\mathbb{R}^k \to \mathbb{R}^m$, respectively and $c \in \mathbb{R}^m$. We call this model a *split model* to emphasize that the variable is split into two variables, x and y. This is needed because each iteration of **ADMM**solves minimization sub-problems over each of the primal variables in an alternating fashion (see Chapter 4 for the definition of the iterates). It is interesting to consider the semidefinite setting (1.0.6), particularly where the splitting arises from the face $Y = VRV^T$, since many semidefinite relaxations of combinatorial problems can be formulated as (1.0.5). A notable difference in (1.0.6) from (1.0.7) is the use set constraints $Y \in \mathcal{Y}$, $R \in \mathcal{R}$, which we take into account in our analysis in Chapter 4.

1.1 Notation

1.1.1 Matrix and vector definitions

Since we make frequent use of vectors and matrices with zeros and ones in different positions we now define conventions for notation throughout this thesis. Further definitions are given in the preliminaries Chapter 2.

- For any matrix X, we write X_{ij} to denote the element in the i-th row and j-th column of X

- For any vector x, we write x_i to denote the i-th coordinate of a vector x.
- For any $m \in \mathbb{N}$, $\bar{e}_m$ denotes the all-ones vector in $\mathbb{R}^m$.
- We write $\bar{E}$ to denote the matrix of all ones, in other words, $\bar{E} = \bar{e}\bar{e}^T$.
- The notation e_m denotes the standard basis vector with 1 in the m-th coordinate.
- We write E_{00} to denote the matrix $e_0 e_0^T$.
- I denotes the identity matrix.

1.1.2 Euclidean spaces

$\mathbb{E}$ (or $\mathbb{E}^n$ to specify the dimension) denotes general Euclidean space. Two important Euclidean spaces we use are $\mathbb{R}^n$, the real vector space of dimension n, and $\mathbb{S}^n$, the space of $n \times n$ symmetric matrices. $\mathbb{S}^n$ is equipped with the standard trace inner product:

$$\langle X, Y \rangle := \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Y_{ij} = \operatorname{trace}(XY),$$

and the Frobenius norm,

$$\|X\|_F := \sqrt{\langle X, X \rangle}.$$

1.1.3 Some linear maps

We now define some important linear maps which are used repeatedly throughout this thesis.

Definition 1.1.1 (arrow map)**.**

$$\operatorname{arrow} : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} \quad \operatorname{arrow}(X) = \begin{pmatrix} X_{00} \\ -X_{01} + X_{11} \\ -X_{02} + X_{22} \\ \vdots \\ -X_{0n} + X_{nn} \end{pmatrix} \tag{1.1.1}$$

Definition 1.1.2.

$$\operatorname{Diag} : \mathbb{R}^n \to \mathbb{S}^n \quad \operatorname{Diag}(x) = \begin{bmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & x_n \end{bmatrix}$$

Definition 1.1.3.

$$\operatorname{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n} \quad \operatorname{diag}(X) = \begin{pmatrix} X_{11} \\ X_{22} \\ \vdots \\ X_{nn} \end{pmatrix}$$

Definition 1.1.4.

$$\operatorname{vec} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n^2} \quad \operatorname{vec}(X) = \begin{pmatrix} X_{11} \\ \vdots \\ X_{1n} \\ \vdots \\ X_{n1} \\ \vdots \\ X_{nn} \end{pmatrix}$$

Definition 1.1.5.

$$\operatorname{Mat} : \mathbb{R}^{nm} \to \mathbb{R}^{m\times n} \quad \operatorname{Mat}(x) = \begin{bmatrix} x_1 & x_{m+1} & \dots & x_{m(n-1)} \\ x_2 & x_{m+2} & \dots & x_{m(n-1)+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{2m} & \dots & x_{mn} \end{bmatrix}$$

Definition 1.1.6. *The gangster operator is a linear map from $\mathbb{S}^{n+1}$ to $\mathbb{S}^{n+1}$ defined by:*

$$\mathcal{G}_J : \mathbb{S}^{n+1} \to \mathbb{S}^{n+1}, \quad (\mathcal{G}_J(Y))_{ij} = \begin{cases} Y_{ij} & \text{if} (i,j) \in J \text{ or } (j,i) \in J, \\ 0 & \text{otherwise}, \end{cases} \tag{1.1.2}$$

where J is an index set in $\{0, \dots n\} \times \{0, \dots, n\}$ which we refer to as the gangster index.

Chapter 2

Preliminaries

This chapter is dedicated to preliminary results which fall under two categories. The first category consists of selected definitions from convex analysis and optimization, covering the required background for understanding the results presented in this thesis. The second category consists of preliminary results that facilitate the longer proofs in later chapters.

2.1 Definitions from convex analysis

Definition 2.1.1 (convex set)**.** *[18, Definition 1.1.1] A set $C \subseteq \mathbb{E}$ is convex if*

$$x, y \in C,\ \lambda \in [0,1] \implies \lambda x + (1-\lambda)y \in C.$$

Definition 2.1.2 (relative interior)**.** *[18, Definition 2.1.1] The relative interior of a set S is define to be the interior of S in the affine hull of S.*

$$\mathrm{ri}(S) = \{x \in S \colon \exists \epsilon > 0, B(x,\epsilon) \cap \mathrm{aff}(S) \subseteq S\},$$

where $B(x,\epsilon)$ denotes the open ball of radius ϵ centered at x.

Definition 2.1.3 (Minkowski sum)**.** *[18, Proposition 1.2.4] For any sets $C_1, C_2 \in \mathbb{E}$, the direct sum, or Minkowski sum, of C_1 and C_2, denoted by $C_1 + C_2$, is the following set:*

$$C_1 + C_2 := \{x = x_1 + x_2 \colon x_1 \in C_1, x_2 \in C_2\}.$$

More generally, for any real numbers α_1, α_2, we define the set

$$\alpha_1 C_1 + \alpha_2 C_2 := \{x = \alpha_1 x_1 + \alpha_2 x_2 \colon x_1 \in C_1, x_2 \in C_2\}.$$

For a concise definition of a convex cone, we cite the following theorem from [27].

Definition 2.1.4 (convex cone)**.** *[27, Theorem 2.6]A set $K \subseteq \mathbb{E}$ is a convex cone if and only if*

$$K + K = K$$
$$\forall \lambda \geq 0,\ \lambda K = K.$$

Definition 2.1.5 (Kronecker product)**.** *[20, Definition 4.2.1] Let $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{p\times q}$. The Kronecker product of A and B, denoted $A \otimes B$, is defined by:*

$$A \otimes B := \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix} \in \mathbb{R}^{mp\times nq}$$

Definition 2.1.6 (Normal cone)**.** *[18, Definition 5.2.3] Let $C \in \mathbb{E}^n$ be a convex set and let $x \in C$. The normal cone to C at x, $N_C(x)$, is the set of directions $s \in \mathbb{E}^n$ that are normal to C at $x \in C$. In other words:*

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

Example 2.1.7. *Let S be a subspace of $\mathbb{E}^n$. Then for any $x \in S$, the normal cone of S at x is the orthogonal complement of S denoted $S^\perp$.*

Proof. Let $x \in S$. By Definition 2.1.6, and using the fact that S is a subspace, we get

$$\begin{aligned} N_S(x) &= \{s \in \mathbb{E}^n : \langle y - x, s\rangle \leq 0\ \forall y \in S\} \\ &= \{s \in \mathbb{E}^n : \langle z, s\rangle \leq 0\ \forall z \in S\} \\ &= \{s \in \mathbb{E}^n : \langle z, s\rangle = 0\ \forall z \in S\} \\ &= S^\perp. \end{aligned}$$

□

Definition 2.1.8 (adjoint of a linear map)**.** *Let $\mathcal{A}\colon \mathbb{E}^n \to \mathbb{E}^m$ denote a linear map. The adjoint of $\mathcal{A}$ is the unique linear map $\mathcal{A}^*\colon \mathbb{E}^m \to \mathbb{E}^n$ satisfying:*

$$\langle \mathcal{A}(x), y\rangle = \langle x, \mathcal{A}^*(y)\rangle,\ \ \forall x \in \mathbb{E}^n,\ \forall y \in \mathbb{E}^m.$$

Proposition 2.1.9. *[3, Proposition 6.46] Let $\mathcal{A}\colon \mathbb{E}^n \to \mathbb{E}^m$ denote a linear map. Let $\mathcal{A}^*\colon \mathbb{E}^m \to \mathbb{E}^n$ denote the adjoint of $\mathcal{A}$. Then*

$$\text{null}(\mathcal{A}) = \text{range}(\mathcal{A}^*)^\perp$$

Corollary 2.1.10. *Let $\mathcal{A}\colon \mathbb{E}^n \to \mathbb{E}^m$ denote a linear map. Let $\mathcal{A}^*\colon \mathbb{E}^m \to \mathbb{E}^n$ denote the adjoint of $\mathcal{A}$. Let $S := \text{null}(\mathcal{A}) = \{x \in \mathbb{E}^n \colon \mathcal{A}(x) = 0\}$ denote the nullspace. Let $\bar{x} \in S$. Then*

$$N_S(\bar{x}) = \text{range}(\mathcal{A}^*).$$

Proof. By Proposition 2.1.9, $S^\perp = \text{range}(\mathcal{A}^*)$. Since, by Example 2.1.7 $S^\perp = N_S(x)$ by for any $x \in S$, we have that $N_S(\bar{x}) = \text{range}(\mathcal{A}^*)$. □

2.2 The cone of positive (semi)definite matrices

Theorem 2.2.1. *[31, Theorem 1.8] For every $X \in \mathbb{S}^n$, there exists orthogonal $Q \in \mathbb{R}^{n\times n}$ such that:*

$$X = Q\,\text{diag}(\lambda(X))Q^T.$$

Definition 2.2.2 (orthogonal spectral decomposition)**.** *For any $X \in \mathbb{S}^n$, we write the spectral decomposition of X,*

$$X = U\Lambda U^T = \sum_{i=1}^{n} \lambda_i u_i u_i^T,$$

where (u_i, λ_i) are the eigenpairs of X, and $\{u_1, \ldots, u_n\}$ are an orthonormal set of eigenvectors. Then U is an orthogonal matrix comprised of these eigenvectors of X, and Λ is the diagonal matrix of eigenvalues of X.

$$\Lambda = \begin{bmatrix} \lambda_1(X) & 0 & \ldots & 0 \\ 0 & \lambda_2(X) & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & \lambda_n(X) \end{bmatrix}, \quad \lambda_1(X) \geq \lambda_2(X) \geq \ldots \geq \lambda_n(X).$$

Definition 2.2.3. *[19] Let $X \in \mathbb{S}^n$ and let $S \subset \{1, \ldots, n\}$. A principal submatrix of X, denoted by X_S, is the matrix formed by deleting the ith row and column from X for every $i \notin S$.*

Theorem 2.2.4. *[31, Proposition 1.10]Let $X \in \mathbb{S}^n$. The following are equivalent:*

1. *$X \in \mathbb{S}^n_+$;*
2. *$x^T X x \geq 0,\ \forall x \in \mathbb{R}^n$;*
3. *$\lambda_i(X) \geq 0\ \forall i = 1, \ldots, n$;*
4. *For any $S \subseteq \{1, \ldots, n\}$ Any square principal submatrix X_S has non-negative determinant;*

5. $\langle X, Y \rangle \geq 0 \ \forall \ Y \in \mathbb{S}^n_+$.

Similarly, the following theorem characterizes $\mathbb{S}^n_{++}$, the set of $n \times n$ positive definite matrices. We $X \succ 0$ to mean that X is positive definite.

Theorem 2.2.5. *[31, Proposition 1.11]*

1. $\mathbb{S}^n_{++} = \text{int}(\mathbb{S}^n_+)$.
2. *Let $X \in \mathbb{S}^n$. The following are equivalent:*
   (a) $X \in \mathbb{S}^n_{++}$;
   (b) $x^T X x > 0, \ \forall x \neq 0 \in \mathbb{R}^n$;
   (c) $\lambda_i(X) > 0 \ \forall i = 1, \ldots n$;
   (d) $X \in \mathbb{S}^n_+$ *and* $\text{rank}(X) = n$.

2.3 Faces of the positive semidefinite cone

In this section we define the notion of a face of a convex set, and pay special attention to faces of the cone of positive semidefinite matrices. We will show that there a various was to represent a the faces of the positive semidefinite cone. We also define the minimal face of a convex cone, and prove a theorem which relates the notion of the minimal face $\mathcal{F}$ of $\mathbb{S}^n_+$ containing a convex set to the elements of the relative interior of $\mathcal{F}$.

Definition 2.3.1 (face of a convex set)**.** *Let $C \subseteq \mathbb{E}$ be a convex set. The convex set $F \subseteq C$ is called a face of C, denoted $F \trianglelefteq C$, if for every $x \in F$ and every $y, z \in C$ such that x is contained in the open interval (y, z), we also have $y, z \in F$.*

Definition 2.3.2 (face of a convex cone)**.** *[12, Definition 2.2.1] Let $K \subseteq \mathbb{E}$ be a convex cone. A convex cone $F \subseteq K$ is called a face of K, and denoted $F \trianglelefteq K$, if*

$$x, y \in K, x + y \in F \implies x, y \in F.$$

Definition 2.3.3 (proper face)**.** *A face F of a convex cone K is called a* proper face *of K, denoted $F \triangleleft K$, if $F \neq K$ and $F \neq \emptyset$.*

The following proposition shows that we have a correspondence between faces of $\mathbb{S}^n_+$ and subpaces of $\mathbb{R}^n$.

Proposition 2.3.4. *[25, Theorem 3.7.1] A set is a face of $\mathbb{S}^n_+$ if and only if it is of the form*

$$\left\{ X \in \mathbb{S}^n_+ : \ \text{range}(X) \subseteq L \right\}$$

for some linear subspace L.

We now give a similar proposition which also gives a definition for the relative interior if a face of $\mathbb{S}^n_+$.

Proposition 2.3.5. *[12, Example 2.2.3] For any k dimensional linear subspace S of $\mathbb{R}^n$, the set,*

$$\mathcal{F}_S = \{X \in \mathbb{S}^n_+ \colon \operatorname{range}(X) \subseteq S\}$$

is a face of $\mathbb{S}^n_+$. Conversely, for any face $\mathcal{F}$ of $\mathbb{S}^n_+$ there exists a linear subspace S of $\mathbb{R}^n$ such that $\mathcal{F} = \{X \in \mathbb{S}^n_+ \colon \operatorname{range}(X) \subseteq S\}$. Moreover, the relative interior of $\mathcal{F}$ is

$$\begin{aligned}\operatorname{ri}(\mathcal{F}) &= \{X \in \mathbb{S}^n_+ \colon \operatorname{range}(X) = S\} \\ &= \{X \in \mathbb{S}^n_+ \colon \operatorname{range}(X) = \operatorname{range}(\hat{X})\},\end{aligned}$$

for any $\hat{X} \in \operatorname{ri}(\mathcal{F})$.

Proposition 2.3.6. *[12, Example 2.2.3] Let S be an r-dimensional linear subspace of $\mathbb{R}^n$ and let*

$$\mathcal{F}_S = \{X \in \mathbb{S}^n_+ \colon \operatorname{range}(X) \subseteq S\}.$$

Then $\mathcal{F}_S$ is a face of $\mathbb{S}^n_+$. Moreover, for any $V \in \mathbb{R}^{n\times r}$ such that $\operatorname{range}(V) = S$,

$$\mathcal{F}_S = V\mathbb{S}^r_+V^T.$$

Definition 2.3.7 (minimal face)**.** *Let K be a convex cone and let $C \subseteq K$ be a convex set. Then the minimal face $\mathcal{F}$ of K containing C, denoted $\operatorname{face}(C,K)$, is defined to be the smallest face of K containing C, in other words,*

$$\operatorname{face}(C,K) = \bigcap_{\substack{C\subseteq\mathcal{F}\\ \mathcal{F}\trianglelefteq K}} \mathcal{F}.$$

Proposition 2.3.8 (minimal face alternate characterization [12])**.** *Let K be a convex cone. If $C \subseteq K$ is convex, then $\operatorname{face}(C,K)$ is equal to the smallest face of K intersecting the relative interior of C. In particular, for any $x \in \operatorname{ri}(C)$, $\operatorname{face}(C,K) = \operatorname{face}(x,K)$.*

2.4 Optimality conditions for convex split problems

In this section we consider problems of the form:

$$\begin{aligned}\min_{X\in\mathbb{S}^{n_1},Y\in\mathbb{S}^{n_2}} \quad & f(X,Y) \\ \text{s.t.} \quad & \mathcal{A}_1(X) + \mathcal{A}_2(Y) = 0 \\ & X \in \mathcal{X} \\ & Y \in \mathcal{Y},\end{aligned} \tag{2.4.1}$$

where $f\colon \mathbb{S}^n \to \mathbb{R}$ is a differentiable convex function, and $\mathcal{A}_1\colon \mathbb{S}^n_1 \to \mathbb{R}^m$ and $\mathcal{A}_2\colon \mathbb{S}^n_2 \to \mathbb{R}^m$ are linear maps. The two sets $\mathcal{X} \subseteq \mathbb{S}^{n_1}$ and $\mathcal{Y} \subseteq \mathbb{S}^{n_2}$ are closed convex sets with $n_1+n_2 = n$.

Theorem 2.4.1 (Rockafellar-Pshenichnyi lemma)**.** *[26, Theorem 2.1] Let $C \subseteq \mathbb{E}$ be a convex set containing a point $\bar{x}$ and let $f : C \subseteq \mathbb{E} \to \mathbb{R}$ be a convex function. Then*

$$\bar{x} \in \operatorname{argmin}_{x\in C} f(x) \iff 0 \in \partial f(\bar{x}) + N_C(\bar{x}). \tag{2.4.2}$$

If the objective function f is differentiable, then (2.4.2) becomes:

$$\bar{x} \in \operatorname{argmin}_{x\in C} f(x) \iff -\nabla f(\bar{x}) \in N_C(\bar{x}). \tag{2.4.3}$$

We can apply Theorem 2.4.1 to obtain optimality conditions for a split model (2.4.1). The linear constraints give rise to the linear manifold defined by:

$$C := \{(X,Y) \in \mathbb{S}^n \colon \mathcal{A}_1(X) + \mathcal{A}_2(Y) = 0\}.$$

Therefore, if we let $\mathcal{K} := C \cap (\mathcal{X} \times \mathcal{Y})$, then (2.4.2) is equivalent to:

$$\begin{array}{ll} \min\limits_{(X,Y)\in\mathbb{S}^n} & f(X,Y) \\ \text{s.t.} & (X,Y) \in \mathcal{K}. \end{array} \tag{2.4.4}$$

Thus we can apply Theorem 2.4.1 and obtain:

$$(\bar{X},\bar{Y}) \in \operatorname{argmin}_{(X,Y)\in\mathcal{K}} f(X,Y) \iff -\nabla f(\bar{X},\bar{Y}) \in N_{\mathcal{K}}(\bar{X},\bar{Y}).$$

So far, this does not tell us much the optimal solutions to (2.4.1). We now prove a result that says: if we can decompose the normal cone of $\mathcal{K}$ into the normal cones of C and $\mathcal{X} \times \mathcal{Y}$, then we obtain a more useful characterization of optimality. The usefulness will become apparent in later chapters when we define the dual program to (1.0.6) and justify the convergence of **ADMM** in Section 4.1.3.

Corollary 2.4.2. *Let $f\colon \mathbb{S}^n \to \mathbb{R}$ be a differentiable convex function, let $\mathcal{A}_1\colon \mathbb{S}^n_1 \to \mathbb{R}^m$ and $\mathcal{A}_2\colon \mathbb{S}^n_2 \to \mathbb{R}^m$ be linear maps, and let $\mathcal{X} \subseteq \mathbb{S}^{n_1}$ and $\mathcal{Y} \subseteq \mathbb{S}^{n_2}$ be closed convex sets where $n_1 + n_2 = n$. Define the sets*

$$\begin{aligned} C &:= \{(X,Y) \in \mathbb{S}^n \colon \mathcal{A}_1(X) + \mathcal{A}_2(Y) = 0\}, \\ \mathcal{K} &:= C \cap (\mathcal{X} \times \mathcal{Y}). \end{aligned}$$

Then, assuming $\bar{X} \in \mathcal{X}$ and $\bar{Y} \in \mathcal{Y}$ are such that:

$$N_{\mathcal{K}}(\bar{X},\bar{Y}) = N_C(\bar{X},\bar{Y}) + (N_{\mathcal{X}}(\bar{X}) \times N_{\mathcal{Y}}(\bar{Y})) \tag{2.4.5}$$

holds, then:

$$(\bar{X},\bar{Y}) \in \operatorname{argmin}_{(X,Y)\in\mathcal{K}} f(X,Y) \iff \begin{cases} -\nabla_X f(\bar{X},\bar{Y}) \in \operatorname{range}(\mathcal{A}_1^*) + N_{\mathcal{X}}(\bar{X}) \\ -\nabla_Y f(\bar{X},\bar{Y}) \in \operatorname{range}(\mathcal{A}_2^*) + N_{\mathcal{Y}}(\bar{Y}). \end{cases} \tag{2.4.6}$$

Proof. Assume (2.4.5) holds. Then, by Theorem 2.4.1,

$$\begin{aligned} (\bar{X},\bar{Y}) \in \operatorname{argmin}_{(X,Y)\in K} f(X,Y) &\iff -\nabla f(\bar{X},\bar{Y}) \in N_K(\bar{X},\bar{Y}) \\ &\iff -\nabla f(\bar{X},\bar{Y}) \in N_A(\bar{X},\bar{Y}) + \left(N_{\mathcal{X}}(\bar{X}) \times N_{\mathcal{Y}}(\bar{Y})\right). \end{aligned}$$

Let $\mathcal{A}(X,Y) = \mathcal{A}_1(X) + \mathcal{A}_2(Y)$. By corollary 2.1.10 we have that

$$N_C(\bar{X},\bar{Y}) = \operatorname{range}\mathcal{A}^*. \tag{2.4.7}$$

Let $w \in \mathbb{R}^m$. We have $\langle \mathcal{A}(X,Y), w\rangle = \langle (X,Y), \mathcal{A}^*(w)\rangle$. We also have

$$\begin{aligned} \langle \mathcal{A}(X,Y), w\rangle &= \langle \mathcal{A}_1(X) + \mathcal{A}_2(Y), w\rangle \\ &= \langle \mathcal{A}_1(X), w\rangle + \langle \mathcal{A}_2(X), w\rangle \\ &= \langle x, \mathcal{A}_1^*(w)\rangle + \langle Y, \mathcal{A}_2^*(W)\rangle \\ &= \langle (X,Y), (\mathcal{A}_1^*(W), \mathcal{A}_2^*(W))\rangle. \end{aligned}$$

Therefore $\mathcal{A}^*(w) = (\mathcal{A}_1^*(w), \mathcal{A}_1^*(w)) \ \ \forall w \in \mathbb{R}^m$, which further implies that

$$\operatorname{range}\mathcal{A}^* = \operatorname{range}\mathcal{A}_1^* \times \operatorname{range}\mathcal{A}_2^*.$$

Combing this result with (2.4.7), we have

$$N_C(\bar{X},\bar{Y}) = \operatorname{range}\mathcal{A}_1^* \times \operatorname{range}\mathcal{A}_2^*.$$

Finally since $\nabla f(X,Y) = (\nabla_X f(X,Y), \nabla_Y f(X,Y))$,

$$0 \in \nabla f(\bar{X},\bar{Y}) + N_K(\bar{X},\bar{Y}) \iff \begin{cases} -\nabla_X f(\bar{X},\bar{Y}) \in \operatorname{range}(\mathcal{A}_1^*) + N_{\mathcal{X}}(\bar{X}) \\ -\nabla_Y f(\bar{X},\bar{Y}) \in \operatorname{range}(\mathcal{A}_2^*) + N_{\mathcal{Y}}(\bar{Y}). \end{cases}$$

□

The natural question to ask now is: when does (2.4.5) hold? For this, we present the following result from Rockafellar's *Convex Analysis* [27].

Theorem 2.4.3 (Corollary 23.8.1 [27])**.** *Let $C_1, \dots C_m$ be convex sets whose relative interiors have a point in common. Then the normal cone to $C_1 \cap C_2 \dots \cap C_m$ at any given point x in $C_1 \cap C_2 \dots \cap C_m$ is $K_1 + \cdots + K_m$ where K_i is the normal cone to C_i at x. If certain of the sets, say $C_1, \dots, C_k$, are polyhedral, the conclusion holds if merely the sets $C_1, \dots, C_k$ and $\operatorname{ri} C_{k+1}, \dots, \operatorname{ri} C_m$ have a point in common.*

Chapter 3

Binary-constrained quadratic program

3.1 Problem definition

Let $x \in \mathbb{R}^n$, $A \in \mathbb{S}^n$, and let $H \in \mathbb{R}^{p \times n}$ be a matrix with all entries $H_{ij} \in \{0, 1\}$. Recall that we denote the p-vector of all ones by $\bar{e}_p \in \mathbb{R}^p$. Consider the following class of binary-constrained quadratic programs (**BCQP**):

$$\begin{array}{ll} \min & x^T A x \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in \{0,1\}^n. \end{array} \tag{3.1.1}$$

It is helpful to view each of the rows of H as a indicator vector over a subset of $\{1, \dots n\}$. This gives is a one-to-one correspondence between H and a collection of subsets $\mathcal{V}_1, \dots \mathcal{V}_p$ of $\{1, \dots, n\}$. Formally, for any binary matrix H, we can write H row-wise as:

$$H = \begin{bmatrix} — & (e^{\mathcal{V}_1})^T & — \\ & \vdots & \\ — & (e^{\mathcal{V}_p})^T & — \end{bmatrix} \in \mathbb{R}^{p \times n}, \tag{3.1.2}$$

where the rows of H are indicator vectors in the sense that for $k = 1, \dots, p$,

$$e^{\mathcal{V}_k} \in \mathbb{R}^n \ : \ \forall i \in \mathcal{V}, \ e_i := \begin{cases} 1 \text{ if } i \in \mathcal{V}_k \\ 0 \text{ otherwise.} \end{cases}$$

Observe that for any $x \in \{0, 1\}^n$, the constraint $Hx = \bar{e}_p$ forces $x_i = 1$ for *exactly* one $i \in \mathcal{V}_k$, for all k. Furthermore, the objective $x^T A x = \sum_{i,j=1}^n A_{ij} x_i x_j$, sums up all values A_{ij} for which $x_i = x_j = 1$.

3.2 Exact hitting set

In order to see what type of problem the **BCQP** is modelling, we relate it a known problem called the hitting set problem. In [14], Rawitz, Even and Shahar define the notion of a hitting set to be the following:

Definition 3.2.1 (hitting set)**.** *Let $\mathcal{X}$ be a set of elements and let $\mathcal{C}$ be a collection of subsets $S_1, S_2, \ldots, S_n$ of $\mathcal{X}$. A hitting set of $(\mathcal{X}, \mathcal{C})$ is a set $\mathcal{U} \subseteq \mathcal{X}$ such that $S_i \cap \mathcal{U} \geq 1$ $\forall i \in \{1, \ldots n\}$.*

Suppose now that we replace the requirement that $S_i \cap \mathcal{U} \geq 1$ $\forall i \in \{1, \ldots n\}$ with the stricter requirement of $S_i \cap \mathcal{U} = 1$ $\forall i \in \{1, \ldots n\}$. We then call such a set $\mathcal{U}$ an *exact hitting set* because $\mathcal{X}$ "hits" each set S_i *exactly* once. The precise definition is as follows.

Definition 3.2.2 (exact hitting set)**.** *Let $\mathcal{X}$ be a set of elements and let $\mathcal{C}$ be a collection of subsets $S_1, S_2, \ldots, S_n$ of $\mathcal{X}$. A hitting set of $(\mathcal{X}, \mathcal{C})$ is a set $\mathcal{U} \subseteq \mathcal{X}$ such that $S_i \cap \mathcal{U} = 1$ $\forall i \in \{1, \ldots n\}$.*

In [17], Guruswami and Zhou use the term *exact hitting set* to refer to the problem of finding an exact hitting set of maximal cardinality, and thus they don't require that every subset S_i intersects with the hitting set $\mathcal{U}$. Rather they look for a set $\mathcal{U}$ that intersects the greatest number of subsets S_i, while not intersecting any such subset more than once. On the other hand, though we have not encountered a precise definition of it in the literature so far, we instead consider the problem of finding a *minimal* exact hitting set where the notion of minimality is defined with respect to a cost function that is quadratic in the number of elements of $\mathcal{X}$ (which we will define shortly). We can phrase this problem more precisely by letting the set of elements $\mathcal{X}$ correspond to vertices on a graph, and adding up the cost of the weighted edges incident to the hitting set vertices. We call this version the quadratic exact hitting set problem (**QEHSP**) and we define it now. To define the **QEHSP**, we consider an undirected weighted graph, $G = (\mathcal{V}, \mathcal{E})$ along with a collection of subsets $\mathcal{V}_1, \ldots \mathcal{V}_p \subseteq \mathcal{V}$. The vertex set $\mathcal{V}$ corresponds to the set of elements for which we would like to find an exact hitting set and let $|\mathcal{V}| = n$. The edge set, $\mathcal{E}$, consists of all unordered pairs (i, j) (occasionally denoted e_{ij} to emphasize that it corresponds an edge) for which i and j are not members of the same V_k (for any k), with the exception of *self-loops,* e_{ii}, which we include in the edge set for all i. The self-loops are included just as a technicality, and we explain this later in Remark 3.2.3. For every edge e_{ij}, there is a an associated weight that we denote by weight(e_{ij}). The **QEHSP** then is the problem of finding a vertex set $\mathcal{U} \subseteq \mathcal{V}$ that satisfies $\mathcal{V}_k \cap \mathcal{U} = 1$ $\forall k \in \{1, \ldots, p\}$ and that minimizes the sum

$$\sum_{i,j \in U} \text{weight}(e_{ij}).$$

To see that this problem is modeled as the **BCQP** in a natural way, we simply interpret

any feasible solution x as the *indicator vector* of some subset $\mathcal{U}$ of $\mathcal{V}$. In other words,

$$x_{\mathcal{U}} \in \mathbb{R}^{|\mathcal{V}|} \ : \ \forall i \in \mathcal{V}, \ x_i := \begin{cases} 1 \text{ if and only if } i \in \mathcal{U}; \\ 0 \text{ otherwise.} \end{cases}$$

This way, there is a one-to-one correspondence between a feasible $x = x_{\mathcal{U}}$ and an exact hitting set $\mathcal{U} \subseteq \mathcal{V}$. Furthermore, since $x \in \{0,1\}^n$, the constraint

$$\sum_{i \in \mathcal{V}_k} x_i = 1, \ \forall k \in \{1, \ldots, p\} \tag{3.2.1}$$

enforces the condition x is an indicator vector for an exact hitting set. Thus, the **QEHSP** is then modeled by the following integer program:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & \sum\limits_{i,j \in \mathcal{U}} \text{weight}(e_{ij}) \\ \text{s.t.} & \sum\limits_{i \in \mathcal{V}_k} x_i = 1, \ \forall k \in \{1, \ldots, p\} \\ & x \in \{0,1\}^n. \end{array} \tag{3.2.2}$$

To write this as a **BCQP** , we first recall that there is also a one-to-one correspondence between the collection of subsets $\mathcal{V}_1, \ldots, \mathcal{V}_p$ and the binary matrix $H \in \mathbb{R}^{p \times n}$, where H is defined as in (3.1.2). Therefore, we immediately have

$$\sum_{u \in V_k} x_u = 1 \ \forall k \in \{1, \ldots, p\} \iff Hx = \bar{e}_p.$$

Now, we construct the corresponding *weight matrix* $W \in \mathbb{R}^{n \times n}$:

$$W_{ij} = \begin{cases} \frac{1}{2} \text{weight}(e_{ij}) & \forall i, j \in \mathcal{V}, \ i \neq j \\ \text{weight}(e_{ii}) & \forall i, j \in \mathcal{V}, \ i = j. \end{cases}$$

By this definition, the sum of the pairwise edge-weights between selected vertices (i.e., vertices in $\mathcal{U}$) is given by the quadratic objective function $q(x)$ defined below. Letting $x = x_{\mathcal{U}}$,

$$q(x) := x^T W x = \sum_{i,j=1}^{n} W_{ij} x_i x_j = \sum_{i,j \in \mathcal{U}} \text{weight}(e_{ij}). \tag{3.2.3}$$

Hence the problem (3.2.2) is equivalent to the **BCQP**:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & x^T W x \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in \{0,1\}^n. \end{array}$$

Remark 3.2.3. *By allowing for self-loops in the graph, the objective function will always*

include the value $\text{weight}(e_{uu})$ *of the self loop* e_{uu} *for every* u *in* $\mathcal{U}$*. It is useful to observe that this is just equivalent to adding a linear term* $c^T x$ *to* $q(x)$*. Due to* $x \in \{0,1\}^n$ *we have*

$$q(x) = x^T W x + c^T x = x^T (W + \text{Diag}(c))x.$$

Thus without loss of generality, we can also associate a cost c_i *to every vertex* $i \in \mathcal{V}$ *by letting it equal to the weight of the self-loop edge* e_{ii} *for every* $i \in \mathcal{V}$*.*

We now give two examples of problems which can be modeled by the **BCQP**, and for which the use of semidefinite programming with **ADMM** has been successful [7, 16]. Though these problems appear different on the surface, when viewed as particular cases of the **BCQP**, we can unify the approaches used in [16] and [7], and treat them as applications of a more general method. Generalizing the methods used in [7, 16] to work on a more general problem requires analyzing what are the core traits of these problems that make it possible to apply these methods. This is the intended outcome of this thesis.

3.2.1 Side chain positioning

In Chapter 6 we look at the *Side Chain Positioning,* ***SCP*** problem in detail. For now we give a brief preview of its formulation as a **BCQP**. We will show that the **SCP** is an instance of (3.1.1), where the weight matrix is denoted by $E \in \mathbb{R}^{n \times p}$ and E_{uv} gives an energy value between elements u and v. The graph $G = (\mathcal{V}, \mathcal{E})$, as well as the subsets $\mathcal{V}_1, \ldots, \mathcal{V}_p \subseteq \mathcal{V}$, are given by the problem (see Chapter 6 for definitions) and we define the binary matrix H in the usual way:

$$H = \begin{bmatrix} — & (e^{\mathcal{V}_1})^T & — \\ & \vdots & \\ — & (e^{\mathcal{V}_p})^T & — \end{bmatrix}.$$

Let $n = |\mathcal{V}|$, then we get the **SCP**:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \sum_{u,v \in \mathcal{V}} E_{uv} x_u x_v \\ \text{s.t.} \quad & Hx = \bar{e}_p \\ & x \in \{0,1\}^n. \end{aligned} \tag{3.2.4}$$

The **SCP** differs from the general **BCQP** in that it has the additional structure that the collection of subsets $\mathcal{V}_1, \ldots, \mathcal{V}_p$ form a partition of $\mathcal{V}$, which is not a required assumption for the **BCQP**, and is in fact not true for our next example, the **QAP**.

3.2.2 Quadratic Assignment Problem

The **QAP**, a well-known problem in combinatorial optimization, which we detail in Chapter 5, can also be modeled as a **BCQP**. The **QAP** is an optimization problem defined

over the space of permuation matrices (see Definition 5.1.1). We can rewrite the problem by vectorizing the permuatition matrix X by defining $x = \text{vec}(X) \in \mathbb{R}^{n^2}$. Using some simple algebraic manipulations we show in Chapter 5 that we can model the **QAP** as the following **BCQP**.

$$\begin{aligned} \min_{x \in \mathbb{R}^{n^2}} \quad & x^T(B^T \otimes A)x - 2c^T x \\ \text{s.t.} \quad & \begin{bmatrix} e^T \otimes I \\ I \otimes e^T \end{bmatrix} x = \bar{e}_{2n} \\ & x \in \{0,1\}^{n^2}. \end{aligned} \tag{3.2.5}$$

On the surface, the **QAP** does not seem like it should be an instance of a **QEHSP**, but since it satisfies **BCQP** formulation, we now know that it is. In this case, the **QAP** corresponds to a **QEHSP** on a graph $G = (\mathcal{V}, \mathcal{E})$ where $|\mathcal{V}| = n^2$, and the subsets of $\mathcal{V}$ (which we can obtain from the rows of the binary matrix in (3.2.5)) are *not* a partition of $\mathcal{V}$.

3.2.3 Complexity

We now discuss the overall complexity of the **BCQP**. Recall that a problem is NP-complete if it is both in NP and it is an NP-hard problem. A problem is NP-hard if there is a polynomial time reduction to the problem from another known NP-complete problem. It is well known [11] that the **QAP** contains the Traveling Salesman Problem as a special case. It is also well known that there does not exist a polynomial time approximation scheme for the **TSP** unless $P = NP$, due to the fact that such a scheme would be used to solve the NP-complete Hamiltonian cycle problem in polynomial time, thereby proving $P = NP$. Specifically, we have the following theorem:

Theorem 3.2.4 ([33, Theorem 2.9]). *For any $\alpha > 1$, there does not exist an α-approximation for the Traveling Salesman Problem on n cities, provided $P \neq NP$. In fact, the existence of a $\mathcal{O}(2^n)$-approximation algorithm for the TSP would similarly imply that $P = NP$.*

As the **QAP** is an instance of the **BCQP** , we have proven here that the **BCQP** is both NP-hard and inapproximable. Although the **SCP** problem seems like a simpler case instance of the **BCQP** , it is also NP-hard [1] *and* hard to approximate. Chazelle proves in [10] that it is NP-complete to approximate the solution to the **SCP** within a factor of cn, where c is a constant and n is the number of vertices of the corresponding graph.

In regards to feasibility, checking whether or not the **BCQP** is feasible is clearly in NP. As we saw earlier, the problem of finding a feasible solution to the **BCQP** is equivalent to the problem finding an exact hitting set. One can reformulate the exact hitting set as a *exact satisfiability problem* (**XSAT**), (see [29].) Furthermore, **XSAT** was shown to be NP-Complete [29] [28]. Thus we conclude that finding a feasible solution to the general **BCQP** is an NP-complete problem. However, for the **QAP** and **SCP** problem, we obtain the existence of a feasible solution for free due to their inherent structure. This point implies that the **BCQP** is too general to properly capture the shared structure of the **QAP** and the **SCP**. We will see that there are further traits that the **SCP** and the **QAP** share with makes

them amenable to solving via their **SDP** relaxations. Identifying the minimal structure is needed on H in order to have success with the proposed method is an interesting question, and we expect that the work presented in this thesis will help to narrow it down.

3.3 The doubly non-negative relaxation

We now move on to formulating the *doubly non-negative,* ***DNN*** relaxation of (3.1.1). Recall that the first step is to reformulate (3.1.1) in terms of a larger, symmetric, rank-one matrix. We do not prescribe an algorithm for this reformulation, but rather provide derivations that justify why the constraints used in [7, 16] apply more generally to the **BCQP** reformulation. Since there are many valid reformulations, the results in this section serve as a blueprint for using semidefinite programming with **ADMM** for any problem that can be formulated as a **BCQP**.

3.3.1 Reformulation

Recall the general format for the **BCQP**:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & x^TWx \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in \{0,1\}^n. \end{array} \tag{3.3.1}$$

In this section we rewrite (3.3.1) as an equivalent program (P_{BCQP}) over the space of rank one symmetric matrices. We propose the following program as a reformulation of (3.3.1):

$$(P_{BCQP}) \qquad \begin{array}{ll} \min\limits_{Y\in\mathbb{S}^{(n+1)}} & \langle \widehat{W}, Y\rangle \\ \text{s.t.} & \text{arrow}(Y) = e_0 \\ & KY = 0 \\ & \text{rank}(Y) = 1, \end{array} \tag{3.3.2}$$

with

$$\widehat{W} := \begin{bmatrix} 0 & 0 \\ 0 & W \end{bmatrix} \quad \text{and} \quad K := \begin{bmatrix} p & -\bar{e}_p^T H \\ -H^T\bar{e}_p & H^TH \end{bmatrix}. \tag{3.3.3}$$

We let $\mathcal{F}_{BCQP}$ denote the feasible set of (3.3.2). We claim that this program is equivalent to (3.3.1), despite the fact that there is no clear dependency on x. However, the constraint $\text{rank}(Y) = 1$ creates an implicit dependency on x in the sense that:

$$Y \text{ is feasible for } (P_{BCQP}) \iff Y = \begin{pmatrix} 1 \\ x \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}^T \text{ where } x \text{ is feasible for (3.1.1).}$$

To prove this, we first prove two preliminary propositions. We begin by showing that the arrow constraint forces all rank one solutions of the **SDP** relaxation to be integral.

Proposition 3.3.1. *Let* $Y \in \mathbb{S}^{n+1}$ *and* $\operatorname{rank}(Y) = 1$. *Then*

$$\operatorname{arrow}(Y) = e_0 \iff Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, \text{ with } x \in \{0,1\}^n.$$

Proof. Since Y is rank-one and symmetric, $Y = yy^T$ where $y \in \mathbb{R}^{n+1}$. We denote the first component of y by y_0, and the remaining components by $\bar{y}$,

$$y = \begin{pmatrix} y_0 \\ \bar{y} \end{pmatrix}, \quad y_0 \in \mathbb{R}, \bar{y} \in \mathbb{R}^n.$$

Since $\operatorname{arrow}(Y) = e_0$, we have $Y_{00} = 1$, thus $y_0^2 = 1$. Also by $\operatorname{arrow}(Y) = e_0$, we have $\bar{y} \circ \bar{y} = y_0 \bar{y}$. If $y_0 = 1$ then $\bar{y} \in \{0,1\}^n$. Thus we have

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, \quad \text{with } x \in \{0,1\}^n.$$

If $y_0 = -1$ then $\bar{y} \in \{0,-1\}^n$ and

$$Y = \begin{pmatrix} -1 \\ -x \end{pmatrix} \begin{pmatrix} -1 \\ -x \end{pmatrix}^T = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, \quad \text{with } x \in \{0,1\}^n.$$

For the other direction, suppose

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, \text{ with } x \in \{0,1\}^n.$$

Then

$$Y = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} \text{ with } x \in \{0,1\}^n.$$

Since $x \in \{0,1\}^n$, $x = x \circ x$. Thus

$$\operatorname{arrow}(Y) = \begin{pmatrix} 1 \\ x \circ x - x \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = e_0.$$

□

Proposition 3.3.2. *Let* $Y_x = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$. *Then* $Hx = e \iff KY_x = 0$.

Proof. The following are equivalent:

$$\begin{aligned}
Hx &= \bar{e}_p \\
\begin{bmatrix} -\bar{e}_p & H \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} &= 0 \\
\begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -\bar{e}_p & H \end{bmatrix}^T \begin{bmatrix} -\bar{e}_p & H \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} &= 0 \\
\operatorname{tr}\left(\begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -\bar{e}_p & H \end{bmatrix}^T \begin{bmatrix} -\bar{e}_p & H \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \right) &= 0 \\
\operatorname{tr}\left(\begin{bmatrix} \bar{e}_p^T \bar{e}_p & -\bar{e}_p^T H \\ -H^T \bar{e}_p & H^T H \end{bmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \right) &= 0 \\
\operatorname{tr}\left(\begin{bmatrix} p & -\bar{e}_p^T H \\ -H^T \bar{e}_p & H^T H \end{bmatrix} \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} \right) &= 0 \\
\operatorname{trace} KY_x &= 0.
\end{aligned}$$

Note that $K = \begin{bmatrix} p & -\bar{e}_p^T H \\ -H^T \bar{e}_p & H^T H \end{bmatrix} = \begin{bmatrix} -\bar{e}_p^T \\ H^T \end{bmatrix} \begin{bmatrix} -\bar{e}_p^T \\ H^T \end{bmatrix}^T$, so K must be positive semidefinite. Since both $K \succeq 0$ and $Y \succeq 0$, we have

$$\operatorname{trace} KY_x = 0 \iff KY_x = 0.$$

□

We can now prove that the lifted problem (3.3.2) is indeed equivalent to (3.3.1).

Theorem 3.3.3. *Y is feasible for (3.3.2) if and only if*

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \text{ where } x \text{ is feasible for (3.1.1).}$$

Proof. Suppose Y is feasible for (3.3.2). Then $\operatorname{arrow}(Y) = e_0$ and $\operatorname{rank}(Y) = 1$, therefore $Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$ with $x \in \{0,1\}^n$, by Proposition 3.3.1. By Proposition 3.3.2, $Hx = e$. Therefore the forward direction holds. Now suppose that $Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$ and x is feasible for (3.1.1). By Proposition 3.3.2, $KY = 0$. Clearly, $\operatorname{rank}(Y) = 1$, therefore $\operatorname{arrow}(Y) = e_0$ by Proposition 3.3.1. □

Finally, we see that (3.3.1) is equivalent to (3.3.2) since we can rewrite the objective function in terms of Y_x.

$$q(x) = x^T W x = \operatorname{trace}(x^T W x) = \operatorname{trace}(W x x^T) = \operatorname{trace}(\widehat{W} Y_x) = \langle \widehat{W}, Y_x \rangle$$

Since an equivalent program is equally difficult, we must consider a relaxation of (P_{BCQP}) obtained by relaxing the rank-one constraint on Y.

3.3.2 Relaxing the rank-one constraint

Since (3.3.2) is a reformulation of (3.1.1), it remains NP-hard. To form the **SDP** relaxation, we relax the rank-one constraint on Y into a positive semidefinite constraint $Y \succeq 0$. Recall that we obtain a **DNN** relaxation by including elementwise nonnegativity as a constraint. We let (P_{DNN}) denote the following doubly nonnegative relaxation:

$$(P_{DNN}) \qquad \begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n+1}} & \langle \widehat{W}, Y \rangle \\ \text{s.t.} & \operatorname{arrow}(Y) = e_0 \\ & KY = 0 \\ & 0 \leq Y \leq 1 \\ & Y \succeq 0. \end{array} \tag{3.3.4}$$

In addition to the elementwise nonnegativity constraint, we proceed as in [7, 16, 24] and further impose the constraint $Y \leq 1$. Note that if Y is a solution to (3.3.2), the diagonal of Y is made up of x_i^2 where $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$. Therefore, we know that $Y_{ii} \leq 1$ for all i. This fact, along with the fact that Y is positive semidefinite, implies that $Y \leq 1$. To see why this is, we recall Theorem 2.2.4 Item 4. If there exist coordinates (i, j) such that $Y_{ij} \geq 1$, then necessarily, $Y_{ji} \geq 1$ since Y is symmetric, and furthermore, the principal submatrix $X_{\mathcal{I}}$, generated by removing all the columns and rows with indices in $\mathcal{I} = \{0, 1, \ldots, n\} \setminus \{i, j\}$ will have negative determinant, violating Theorem 2.2.4 Item 4.

We denote the feasible set of the **DNN** relaxation by $\mathcal{F}_{SDP}$, i.e.,

$$\mathcal{F}_{SDP} := \left\{ Y \in \mathbb{S}^{n+1}_{+} \colon \operatorname{arrow}(Y) = e_0, KY = 0, 0 \leq Y \leq 1. \right\}. \tag{3.3.5}$$

By relaxing the rank-one constraint we have that $\mathcal{F}_{BCQP} \subseteq \mathcal{F}_{SDP}$. Ultimately we are interested in solutions to (3.3.2), which are exactly the rank one feasible points of $\mathcal{F}_{SDP}$. We have the relationship:

$$\mathcal{F}_{BCQP} = \mathcal{F}_{SDP} \cap \left\{ Y \in \mathbb{S}^{n+1}_{+} \colon \operatorname{rank}(Y) = 1 \right\}.$$

Thus, if we obtain a rank one optimal solution for (3.3.4) it must also be an optimal solution for (3.3.2). We may also add constraints which are redundant to (3.3.1) and maintain this property, possibly obtaining an even stronger relaxation. We use this idea in the next sections to obtain two additional constraints: the gangster constraint and the trace constraint.

3.3.3 Gangster constraint

The *gangster constraint* is a constraint that "shoots holes" in a matrix in the sense that it constrains elements of the matrix to be equal to zero. This constraint is used for both the **QAP** and **SCP** in previous work, however, for each of the two problems, the indices of the entries constrained to be zero (the *gangster index*) differ. In this section we present an original derivation for the gangster constraint that has the added benefit of showing that there exists a general formula for this constraint which depends entirely on matrix H in (3.1.1).

Recall the definition of the gangster operator Definition 1.1.6. We let E_{00} denote a $(n+1) \times (n+1)$ matrix with all entries equal to zero aside from the top left $(0,0)$ entry which is equal to one. Given a two dimensional index set $J \subseteq \{1,\dots,n+1\} \times \{1,\dots,n+1\}$, the so-called gangster constraint applied to $Y \in S^{n+1}$ is the following:

$$G_J(Y) = E_{00}.$$

In [34], Zhao et. al define the gangster index for the **QAP** by first forming a block matrix from the lifting of x to Y_x. As we will see in Chapter 5, the variable x in (3.2.5) is the vectorization of a matrix $X \in \mathbb{R}^{n\times n}$. By letting x^i denote the ith column of X, we can write:

$$x = \begin{pmatrix} 1 \\ x^1 \\ \vdots \\ x^n \end{pmatrix}, \quad x^i \in \mathbb{R}^n \ \forall i \in \{1,\dots n\}.$$

Then the lifted variable Y_X is given by:

$$Y_X = \begin{pmatrix} 1 \\ x^1 \\ \vdots \\ x^n \end{pmatrix} \begin{pmatrix} 1 \\ x^1 \\ \vdots \\ x^n \end{pmatrix}^T = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix},$$

with xx^T made up of n^2 block matrices given by $x^i x^{jT}$ for all $i,j \in \{1,\dots,n\}$. It was observed in [34] that for all x feasible for the lifting of the **QAP**, the diagonal elements of the off diagonal blocks of xx^T are all equal to zero, and the off-diagonal elements of the diagonal blocks of xx^T are also all equal to zero. In the case of the **SCP**, it was found that the off-diagonal elements of the diagonal blocks are always equal to zero [6]. Instead of looking at the specific problem at hand to determine what entries of xx^T can be constrained to zero, we now show that given a **BCQP** in the form (3.1.1), we need to simply use the constraint matrix H to get the appropriate gangster indices.

Theorem 3.3.4. *Let x be feasible for (3.1.1). Let D_H be a diagonal matrix with entries equal to the diagonal entries of H^TH. Then*

$$(H^TH - D_H) \circ xx^T = 0. \tag{3.3.6}$$

Proof. The following are equivalent.

$$\begin{aligned}
Hx &= \bar{e}_p \\
H^THx &= H^T\bar{e}_p \\
H^THx - H^T\bar{e}_p x &= H^T\bar{e}_p - H^T\bar{e}_p x \\
(H^TH - D_H)x &= D_H(e - x) \\
(H^TH - D_H)xx^T &= D_H(ex^T - xx^T) \\
\operatorname{trace}\left((H^TH - D_H)xx^T\right) &= \operatorname{trace}\left(D_H(ex^T - xx^T)\right) \\
&\left(= \sum_{i=1}^{n} (D_H)_{ii} \underbrace{(x_i - x_i^2)}_{=0}\right) \\
\operatorname{trace}\left((H^TH - D_H)xx^T\right) &= 0 \\
(H^TH - D_H) \circ xx^T &= 0.
\end{aligned}$$

Note $(H^TH - D_H)$ and xx^T are both symmetric with all non-negative entries. The last equality follows from the fact that

$$\operatorname{trace}(AB) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij},$$

for square symmetric matrices A,B of order n. When all terms in the a sum are non-negative, they must all be zero if their sum is zero. That is,

$$\operatorname{trace}(AB) = 0 \implies A \circ B = 0.$$

Hence the last equality holds. □

When we consider the binary constraint $Hx = \bar{e}_p$, in particular, the corresponding graphical interpretation presented in Section 3.2 as an exact hitting set, we see that Theorem 3.3.4 makes intuitive sense. (3.3.6) is enforcing the condition that x_i and x_j cannot simultaneous be equal to 1 if i and j are in the same set $\mathcal{V}_k$, in other words, we cannot select two elements from the same subset $\mathcal{V}_k$.

We now use (3.3.6) to formulate the gangster constraint. First, we construct indices from the non-diagonal nonzero entries of H^TH. Then, as Y is one column and one row larger than xx^T, we extend the index set to contain also the top left entry of Y. This gives the gangster index. To avoid shifting all the indices over, we do this by starting the index from $(0, 0)$.

Definition 3.3.5 (Gangster Index)**.** *Let H be as in the **BCQP** (3.1.1) and let*

$$\bar{J} := \{(i,j) \colon i \neq j \text{ and } (H^TH)_{ij} \neq 0,\ i,j \in \{1,\dots,n\}\}. \tag{3.3.7}$$

The gangster set, J, is defined by

$$J = \{0,0\} \cup \bar{J}$$

Corollary 3.3.6. *Let Y_x be feasible for (3.3.2). Then $G_J(Y) = E_{00}$.*

Proof. This fact follows directly from Theorem 3.3.4 and Theorem 3.3.3. □

The gangster constraint enforces two things, first that the upper $(0,0)$ element of Y is one; and second that $Y_{ij} = 0$ for all $(i,j) \in \bar{J}$. With the addition of the gangster constraint we have the updated **DNN** relaxation of (3.3.2):

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n+1}} & \langle \widehat{W}, Y \rangle \\ \text{s.t.} & \text{arrow}(Y) = e_0 \\ & G_J(Y) = E_{00} \\ & KY = 0 \\ & 0 \leq Y \leq 1 \\ & Y \succeq 0. \end{array} \tag{3.3.8}$$

3.4 Refining the model

In this section we take a closer look at the structure of the feasible set of (3.3.8). The purpose of this section is to give proofs of the relationship between the constraints in the model which are not always apparent a priori. In particular, we find that the presence of the gangster constraint renders the arrow constraint redundant (Theorem 3.4.3).

We also provide a condition on the original problem (3.3.1) under which we can determine that the trace of all feasible Y is equal to a constant $t+1$ (Proposition 3.4.5) . Although this shows that trace $Y = t+1$ is redundant to the model, we include it in the model as it strengthens the **ADMM** subproblems [16]. The final outcome of this section is an equivalent model to (3.3.8) formed by removing the arrow constraint and imposing the trace constraint. The resulting model is the general version of the **DNN** relaxation of the **QAP** use in [16] and the **DNN** relaxation of the **SCP** problem in [7].

3.4.1 Gangster and arrow constraints

We will see that there is significant structure imposed by the constraint $G_J(Y) = E_{00}$.

Proposition 3.4.1. *Let K be as defined in (3.3.3) and let $Y \in \mathbb{S}^{n+1}_+$. Then*

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \\ \text{arrow}(Y) = e_0 \end{cases} \implies \sum_{i \in V^k} Y_{ii} = 1 \ \forall k \in \{1, \ldots, p\}.$$

Proof. Define $D = \begin{bmatrix} -e_p & H \end{bmatrix}$ and recall that $K = D^T D$. Since $\operatorname{null} K = \operatorname{null} D$, $KY = 0 \iff DY = 0$. Therefore we have

$$0 = DY = \left[\begin{array}{c|c} -1 & -e_{V_1}^T- \\ \vdots & \vdots \\ -1 & -e_{V_i}^T- \\ \vdots & \vdots \\ -1 & -e_{V_n}^T- \end{array}\right] \begin{bmatrix} Y_{00} & Y_{01} & \dots & Y_{0n} \\ Y_{10} & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \dots & Y_{nn} \end{bmatrix} \tag{3.4.1}$$

Expanding the first column of DY (3.4.1) gives:

$$\begin{array}{rll} 0 &= -Y_{00} + e_{V_k}^T Y_{[1:n]0} & \forall k \in \{1, \dots, p\} \\ \iff 1 &= \sum_{i \in V^k} Y_{i0} & \forall k \in \{1, \dots, p\} \\ \iff 1 &= \sum_{i \in V^k} Y_{0i} & \forall k \in \{1, \dots, p\} \text{ by symmetry of } Y, \\ \iff 1 &= \sum_{i \in V^k} Y_{ii} & \forall k \in \{1, \dots, p\} \text{ by } \operatorname{arrow}(Y) = e_0. \end{array}$$

□

Lemma 3.4.2. *Let $Y \in \mathbb{S}^{n+1}$ and suppose $G_J(Y) = E_{00}$ holds. Then*

$$\sum_{i \in V_k} Y_{ij} = Y_{jj} \quad \forall k \in \{1, \dots, p\} \tag{3.4.2}$$

Proof. If $G_J(Y) = E_{00}$, then by definition of J, $Y_{ij} = 0$ if $i, j \in V^k$ and $i \neq j$. Therefore,

$$\sum_{i \in V_k} Y_{ij} = Y_{jj}.$$

□

The next theorem shows the surprising fact that the arrow constraint of the model becomes redundant when we add the gangster constraint. This means, in particular, that we can restate theorem above without the assumption that $\operatorname{arrow}(Y) = e_0$.

Theorem 3.4.3. *Let K be as defined in (3.3.3) and let $Y \in \mathbb{S}_+^{n+1}$. Then*

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \end{cases} \implies \operatorname{arrow}(Y) = e_0.$$

Proof. If we expand (3.4.1) elementwise, we get, for every $i, j \in \{1, \dots, n\}$,

$$0 = (DY)_{ij} = -Y_{0j} + e_{V_i}^T Y_{[1:n]j} = -Y_{0j} + \sum_{k \in V_i} Y_{kj} = -Y_{0j} + Y_{jj}$$

The last equality holds by Lemma 3.4.2. Therefore $Y_{0j} = Y_{jj}$ $\forall j \in \{1, \dots, n\}$. And since $Y_{00} = 1$, we have $\operatorname{arrow}(Y) = e_0$. □

The redundancy of the arrow constraint allows us to now strengthen Proposition 3.4.1.

Corollary 3.4.4. *Let K be as defined in (3.3.3) and let $Y \in \mathbb{S}_+^{n+1}$. Then*

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \end{cases} \implies \sum_{i \in V^k} Y_{ii} = 1 \ \forall k \in \{1, \ldots, p\}.$$

Proof. By Proposition 3.4.1, we have

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \\ \text{arrow}(Y) = e_0 \end{cases} \implies \sum_{i \in V^k} Y_{ii} = 1 \ \forall k \in \{1, \ldots, p\}.$$

However, Theorem 3.4.3 tell us that

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \end{cases} \implies \text{arrow}(Y) = e_0.$$

Therefore, we conclude that

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \end{cases} \implies \sum_{i \in V^k} Y_{ii} = 1 \ \forall k \in \{1, \ldots, p\}.$$

□

We have shown that $\text{arrow}(Y) = e_0$ is implied by $KY = 0$ and $G_J(Y) = e_0$. This means that we can remove the arrow constraint from (3.3.8) with no change to the feasible set. We also showed that all feasible solutions to (3.3.8) satisfy $\sum_{i \in \mathcal{V}_k} Y_{ii} = 1$ for all $k = 1, \ldots, p$. We will use this fact in the next section to obtain an additional constraint on Y.

3.4.2 Trace constraint

We now give a condition on (3.3.1) under which we can add a trace constraint on Y into the model. Both of our examples, the **QAP** and the **SCP** problem, satisfy this condition. But for a general **BCQP**, this may not be satisfied. This is to say that not all feasible solutions to (3.1.1) have the same trace. Let $G = (V, E)$ be the input graph to (3.1.1) and let $\mathcal{V}_1, \ldots, \mathcal{V}_p$ denote the given vertex subsets. Suppose that these subsets satisfy the

following:

$$\text{There exists a subcollection } \{\tilde{\mathcal{V}}_1 \dots \tilde{\mathcal{V}}_t\} \subseteq \{\mathcal{V}_1, \dots, \mathcal{V}_p\} \text{ such that } \{\tilde{\mathcal{V}}_1 \dots \tilde{\mathcal{V}}_t\} \text{ is a partition of } \mathcal{V} = \{1, \dots, n\}. \tag{3.4.3}$$

Proposition 3.4.5. *Suppose there exists a subcollection* $\{\tilde{\mathcal{V}}_1 \dots \tilde{\mathcal{V}}_t\} \subseteq \{\mathcal{V}_1, \dots, \mathcal{V}_p\}$ *such that* $\{\tilde{\mathcal{V}}_1 \dots \tilde{\mathcal{V}}_t\}$ *is a partition of* $\mathcal{V} = \{1, \dots, n\}$*. Then*

$$\begin{cases} KY = 0 \\ G_J(Y) = E_{00} \end{cases} \implies \operatorname{trace}(Y) = t + 1.$$

Proof. We apply Lemma 3.4.2 and note that

$$\operatorname{trace}(Y) = Y_{00} + \sum_{i=1}^{n} Y_{ii} = 1 + \sum_{k=1}^{t} \sum_{i \in \tilde{V}_k} Y_{ii} = 1 + t.$$

□

We now establish an additional lemma which uses the same logic as Proposition 3.4.5, but is phrased in such a way that it will be useful later on when for the computation of upper bounds in Section 4.2.3.

Lemma 3.4.6. *Suppose that the* ***BCQP*** *satisfies the condition* (3.4.3) *and let* x *be feasible for the* ***BCQP****. Then* $x^T x = t$*.*

Proof. $x^T x = \sum_{i=1}^{n} (x_i)^2 = \sum_{i=1}^{n} x_i \sum_{k=1}^{t} \sum_{i \in \tilde{V}^k} x_i = \sum_{k=1}^{t} = t.$ □

Without the trace constraint (i.e., if the **BCQP** does not satisfy (3.4.3)) we obtain the following **DNN** relaxation:

$$\begin{aligned} \min_{Y \in \mathbb{S}^{n+1}} \quad & \langle \widehat{W}, Y \rangle \\ \text{s.t.} \quad & G_J(Y) = E_{00} \\ & KY = 0 \\ & 0 \leq Y \leq 1 \\ & Y \succeq 0. \end{aligned} \tag{3.4.4}$$

If the problem satisfies the condition (3.4.3), we can add the redundant trace constraint to

the model:

$$\begin{array}{ll} \min\limits_{Y\in\mathbb{S}^{n+1}} & \langle \widehat{W}, Y\rangle \\ \text{s.t.} & G_J(Y) = E_{00} \\ & KY = 0 \\ & 0 \le Y \le 1 \\ & \operatorname{trace} Y = t+1 \\ & Y \succeq 0. \end{array} \tag{3.4.5}$$

Remark 3.4.7. *For simplicity, for the remainder of this thesis we consider model with the trace constraint* (3.4.5).

Since the arrow constraint was redundant, we may remove it without changing the feasible set. Furthermore, since the trace constraint is also redundant, we can add it without any change to the feasible set. Thus the model (3.4.5) is equivalent to (3.3.8). Since we will refer to it often in subsequent chapters, we now give a name to the feasible set of (3.4.5):

$$\mathcal{F}_{DNN} = \{Y \in \mathbb{S}^{n+1}_+ \colon G_J(Y) = E_{00}, KY = 0, 0 \le Y \le 1, \operatorname{trace} Y = t+1\}. \tag{3.4.6}$$

3.5 The split model

We now construct the split model on which we will apply **ADMM**. The model presented in this chapter is a generalized version of the model used for the **QAP** in [16] and for the **SCP** in [7]. Deriving this model for the **BCQP** unifies the techniques used in [7, 16], potentially opening the door to apply this technique for other problems provided that the problem can be formulated as a **BCQP**. We define the sets:

$$\mathcal{R} := \left\{R \in \mathbb{S}^{n+1-r}_+ \colon \operatorname{trace} R = t+1\right\}, \tag{3.5.1}$$

$$\mathcal{Y} := \left\{Y \in \mathbb{S}^{n+1} \colon G_J(Y) = E_{00}, 0 \le Y \le 1\right\}, \tag{3.5.2}$$

where $r := \operatorname{rank}(K)$. We let $V \in \mathbb{R}^{n+1\times(n+1-r)}$ be a matrix whose columns form a basis of $\operatorname{null}(K)$, which has dimension $n+1-r$. In other words, V is a full rank matrix such that $\operatorname{range}(V) = \operatorname{null}(K)$. We also assume that V has orthonormal columns, i.e., $V^TV = I$. The split **SDP** relaxation (P_V) is then defined to be the following:

$$(P_V) \qquad \begin{array}{ll} \min & \langle \widehat{W}, Y\rangle \\ \text{s.t.} & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}. \end{array} \tag{3.5.3}$$

We call (3.5.3) the split model because we have written the problem in terms of two separate primal variables, R and Y, which are constrained to be in separate sets $\mathcal{R}$, and $\mathcal{Y}$. This choice will becoming significant later when we use **ADMM** to solve (3.5.3).

To show that (3.5.3) is equivalent to (3.4.5), we simply require the following lemma and proposition. The following lemma justifies applying the trace constraint to R.

Lemma 3.5.1. *Let $V \in R^{n\times k}$ and let $R \in \mathbb{R}^{k\times k}$, for some arbitray $n, k \in \mathbb{N}$ and suppose that $V^TV = I$. Then* $\text{trace}(VRV^T) = \text{trace}(R)$

.

Proof.

$$\text{trace}(Y) = \text{trace}(VRV^T) = \text{trace}(V^TVR) = \text{trace}(R).$$

□

Now, for completeness we establish the equivalence of (3.4.4) and (3.5.3) by showing that there is a one-to-one correspondence between feasible solutions to (3.4.4) and (3.5.3).

Proposition 3.5.2. *Let $V \in \mathbb{R}^{(n+1)\times(n+1-r)}$ be a full rank matrix that satisfies $V^TV = I$ and* $\text{range}(V) = \text{null}(K)$. *Then*

$$Y \in \mathcal{F}_{DNN} \iff Y = VRV^T \text{ and } R \in \mathcal{R},\, Y \in \mathcal{Y}.$$

Proof. Observe that $KY = 0 \iff \text{range}(Y) \subseteq \text{null}(K)$. Since K is a subspace of dimension $(n+1-r)$, by Proposition 2.3.4 it determines a face of $\mathbb{S}^{n+1}_n$, given by

$$\mathcal{F}_K := \left\{Y \in \mathbb{S}^{n+1}_+ \colon \text{range}(Y) \subseteq \text{null}(K)\right\}.$$

In other words,

$$KY = 0 \iff Y \in \mathcal{F}_K \trianglelefteq \mathbb{S}^{n+1}_+.$$

Since $V \in \mathbb{R}^{(n+1)\times(n+1-r)}$ and $\text{range}(V) = \text{null}(K)$, by Proposition 2.3.6 we have

$$\mathcal{F}_K = V\mathbb{S}^{n+1-r}_+V^T.$$

The rest of the statement is trivial, since the constraints defining $\mathcal{Y}$ are identical to the constraints that define $\mathcal{F}_{DNN}$, with the exception of the trace constraint, which is covered by Lemma 3.5.1.

□

We can see that there are many possible ways to define $\mathcal{R}$ and $\mathcal{Y}$ so that (3.5.3) is equivalent to (3.4.5). Since Y and R are linked through the linear equation $Y = VRV^T$, we can take any affine constraint on Y and formulate an equivalent affine constraint on R, and add this constraint to the set $\mathcal{R}$ without changing the feasible set.

Ultimately the choice constraints in $\mathcal{R}$ and $\mathcal{Y}$ impact the way in which the iterates of **ADMM** are computed, as we will see in detail in Chapter 4. However, it is not clear whether adding a greater number of redundant constraints to the model will help or hinder the performance of **ADMM**. This raises an interesting question of whether it is always

better to have more constraints, or whether we can determine at what point adding more constraints will negatively affect the behaviour of **ADMM**. So far this is not established and is an area to be further researched. For now, our choice of $\mathcal{R}$ and $\mathcal{Y}$ follows the model used in practice in [7, 16].

Remark 3.5.3. *Note that a matrix V satisfying the hypothesis of Proposition 3.5.2 always exists. Previous work in [6, 34] constructs an explicit formula for V using the structure of K. If such an explicit formula is hard to find, one can always construct V satisfying the required properties by implementing a QR factorization. Recall that $K = D^T D$ and therefore* $\text{null}(D) = \text{null}\, K$. *Thus by taking the QR factorization of D^T, we can find a V with* $\text{range}(V) = \text{null}(D)$. *The QR factorization as defined in [30] involves writing D^T in terms the matrices Q_1, Q_2 and R:*

$$D^T = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

$$Q_1 \in \mathbb{R}^{p \times p} \quad Q_2 \in \mathbb{R}^{p \times n+1-p}.$$

Assuming D^T is of full rank r, the columns of Q_2 form an orthonormal basis for $\text{null}(D)$ *[30]. Therefore, we can take $V = Q_2$.*

Chapter 4

Applying ADMM to the BCQP

In this chapter, we define the iterates for **ADMM** applied to the split model (3.5.3) derived in Section 3.5. We address the assumptions required for the convergence of **ADMM** and provide a simple condition under which they are satisfied. We define the dual problem to (3.5.3) and show that under the provided condition, the primal and dual optimal values are equal, and the dual optimal value is attained. We then summarize computational approaches from [7, 16] for explicitly computing the iterates of **ADMM** efficiently, as well as strategies for rounding output of **ADMM** to obtain upper and lower bounds on the optimal solution to (3.1.1).

4.1 Algorithm definition

Recall that K is as defined in (3.3.3) and $r := \dim \operatorname{null}(K)$. Futhermore, we assume that V is some full rank matrix in $\mathbb{R}^{(n+1)\times(n+1-r)}$ such that $\operatorname{range}(V) = \operatorname{null}(K)$ and $V^TV = I$. Recall from section 3.5 that the sets $\mathcal{R}$ and $\mathcal{Y}$ are defined as follows:

$$\begin{aligned}\mathcal{R} &:= \left\{R \in \mathbb{S}_+^{n+1-r} \colon \operatorname{trace} R = t+1\right\},\\ \mathcal{Y} &:= \left\{Y \in \mathbb{S}^{n+1} \colon G_J(Y) = E_{00}, 0 \le Y \le 1\right\}.\end{aligned}$$

Lastly, recall that we obtained the following split model

$$\begin{array}{ll}\min & \langle \widehat{W}, Y\rangle \\ \text{s.t.} & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}\end{array} \tag{4.1.1}$$

as a relaxation for the original problem (3.1.1). This is the model on which we will apply **ADMM**. The **ADMM** algorithm works by iteratively updating the primal variables Y^k and R^k, as well as a dual iterate which we denote by Z^k. Since we have not yet defined the dual problem, we do this now.

4.1.1 Dual problem

As always, we assume that the columns of the matrix $V \in \mathbb{R}^{n+1-r \times n+1}$ form a basis of null(K) and that $V^T V = I$. Recall the primal problem

$$(P_V) \quad p^* = \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \{\langle \widehat{W}, Y \rangle \colon Y = VRV^T\}. \tag{4.1.2}$$

The Lagrangian $\mathcal{L}(R, Y, Z) \colon \mathcal{R} \times \mathcal{Y} \times \mathbb{S}^{n+1} \to \mathbb{R}$: for (4.1.2) is defined by:

$$\mathcal{L}(R, Y, Z) = \langle \widehat{W}, Y \rangle + \langle Z, Y - VRV^T \rangle. \tag{4.1.3}$$

Note that the *set constraints* $R \in \mathcal{R}$ and $Y \in \mathcal{Y}$ are not treated as constraints, but rather define the domain of the objective function of (4.1.2). This subtle difference changes how we define the Lagrangian and consequently the Lagrangian dual problem. In particular, we do not assign Lagrange multipliers to the constraints of $\mathcal{R}$ and $\mathcal{Y}$. The dual function arises by minimizing the Lagrangian with respect to (R, Y) over $\mathcal{R} \times \mathcal{Y}$:

$$g(Z) = \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \mathcal{L}(R, Y, Z).$$

Finally, the Lagrangian dual problem of (4.1.2) is

$$(D_V) \quad d^* = \sup_Z g(Z). \tag{4.1.4}$$

To define that **ADMM** iterates, we require the notion of an augmented Lagrangian, which is defined for our problem (4.1.2) as the following function:

$$\mathcal{L}_A(R, Y, Z) := \langle \widehat{W}, Y \rangle + \langle Z, Y - VRV^T \rangle + \frac{\beta}{2} \left\| Y - VRV^T \right\|_F^2, \tag{4.1.5}$$

where $\beta > 0$ is a penalty parameter. Finally, the **ADMM** algorithm applied to (4.1.1) is defined by the iterates:

$$R^{k+1} = \operatorname{argmin}_{R \in \mathcal{R}} \mathcal{L}_A(R, Y^k, Z^k) \tag{4.1.6}$$

$$Y^{k+1} = \operatorname{argmin}_{Y \in \mathcal{Y}} \mathcal{L}_A(R^{k+1}, Y, Z^k) \tag{4.1.7}$$

$$Z^{k+1} = Z^k + \beta(Y^{k+1} - \widehat{V} R^{k+1} \widehat{V}^T) \tag{4.1.8}$$

Definition 4.1.1 (Strong Duality)**.** *Consider the primal-dual pair $(P_V), (D_V)$ defined in equations (4.1.2) and (4.1.4) respectively. We say that strong duality holds for $(P_V), (D_V)$ the dual optimal value is attained, i.e., there exists Z^* such that $g(Z^*) = \max_Z g(Z) = d^*$, and furthermore, we have $p^* = d^*$.*

4.1.2 Assumptions for convergence

In [5] Boyd, Parikh, Chu, Peleato and Eckstein give a survey on the history, theory and applications of Alternating Direction Method of Multipliers (**ADMM**). They show that the following three sequences converge, under assumption that we will list next.

- Residual convergence: $Y^k - VR^kV^T \to 0$ as $k \to \infty$.
- Objective convergence: $\langle \widehat{W}, Y^k \rangle \to p^*$ as $k \to \infty$.
- Dual variable convergence: $Z^k \to Z^*$ as $k \to \infty$.

The proof convergence of the sequences above in [5] relies on the following assumptions.

1. The objective function is closed, proper and convex;
2. The Lagrangian has a *saddle point*, i.e., there exists a point $(R^*, Y^*, Z^*) \in \mathcal{R} \times \mathcal{Y} \times \mathbb{S}^{n+1}$ such that

$$\mathcal{L}(R^*, Y^*, Z) \leq \mathcal{L}(R^*, Y^*, Z^*) \leq \mathcal{L}(R, Y, Z^*) \ \forall R \in \mathcal{R}, Y \in \mathcal{Y}, Z. \tag{4.1.9}$$

Note that item 1 is a fairly weak condition, which is clearly satisfied for our model as $\langle \widehat{W}, Y \rangle$ is a linear function. In contrast, we will dedicate the entire next section addressing item 2.

4.1.3 Normal cone intersection property

In this section we make use of the preliminary results established in Section 2.4 to show that the existence of a feasible $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$ is sufficient in guaranteeing that item 2 is holds.

Let V be as defined for the primal **SDP** (4.1.2). We define the set

$$\mathcal{N} := \{(R, Y) \in \mathbb{S}^{n+1-r} \times \mathbb{S}^{n+1} : Y - VRV^T = 0\} \subseteq \left(\mathbb{S}^{n+1-r} \times \mathbb{S}^{n+1}\right).$$

Note that $\mathcal{R} \subseteq \mathbb{S}^{n+1-r}$ and $\mathcal{Y} \subseteq \mathbb{S}^{n+1}$. In order to write the feasible region of (4.1.2) as an intersection of sets, we first extend the sets $\mathcal{R}$ and $\mathcal{Y}$ so that they are subsets of $(\mathbb{S}^{n+1-r} \times \mathbb{S}^{n+1})$. We define the notation that, for any $n \in \mathbb{N}$, 0^n denotes the set of all-zero symmetric matrices of size n. Then we have the extended versions of $\mathcal{R}$ and $\mathcal{Y}$ given by:

$$\bar{\mathcal{R}} = \mathcal{R} \times 0^{n+1} \subseteq \left(\mathbb{S}^{n+1-r} \times \mathbb{S}^{n+1}\right),$$
$$\bar{\mathcal{Y}} = 0^{n+1-r} \times \mathcal{Y} \subseteq \left(\mathbb{S}^{n+1-r} \times \mathbb{S}^{n+1}\right).$$

Observe that $\mathcal{K} := \mathcal{N} \cap \mathcal{R} \times \mathcal{Y} = \mathcal{N} \cap \bar{\mathcal{R}} \cap \bar{\mathcal{Y}}$. Furthermore, since $\mathcal{Y}$ is a polyhedral set, $\bar{\mathcal{Y}}$ also polyhedral. Likewise, $\bar{\mathcal{R}}$ is a closed convex set and $\text{ri}(\bar{\mathcal{R}}) = \text{ri}(\mathcal{R})$. Our goal is to show that the normal cone condition (2.4.5) holds for all feasible points (R, Y), in other words that

$$N_{\mathcal{K}}(R, Y) = N_{\mathcal{N}}(R, Y) + (N_{\mathcal{R}}(R) \times N_{\mathcal{Y}}(Y)),$$

for all $(R, Y) \in \mathcal{K}$. We note that this is equivalent to

$$N_{\mathcal{K}} = N_{\mathcal{N}}(R, Y) + N_{\bar{\mathcal{R}}}(R, Y) + N_{\bar{\mathcal{Y}}}(R, Y) \ \forall (R, Y) \in \mathcal{K}. \tag{4.1.10}$$

By Theorem 2.4.3, (4.1.10) holds if $\mathcal{N} \cap \bar{\mathcal{Y}} \cap \text{ri}(\bar{\mathcal{R}})$ is nonempty, in other words if there exists $(R, Y) \in \mathcal{N}$ such that $Y \in \mathcal{Y}$ and $R \in \text{ri}(\mathcal{R})$. The following lemma states that we can show that $\mathcal{N} \cap \bar{\mathcal{Y}} \cap \text{ri}(\bar{\mathcal{R}})$ is nonempty simply by finding that there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$.

Lemma 4.1.2. *Suppose there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$. Then there exists $\hat{R}$ such that $(\hat{R}, \hat{Y})$ is feasible for (3.5.3) and $\hat{R} \in \text{ri}(\mathcal{R})$.*

Proof. Recall that in the model of (3.5.3), we assume that $V \in \mathbb{R}^{(n+1)\times(n+1-r)}$ is a full rank matrix such that $\text{range}(V) = \text{null}(K)$. Since $\hat{Y} \in \mathcal{F}_{DNN}$, then by Proposition 3.5.2, $\hat{Y} = V\hat{R}V^T$ for some $\hat{R} \in \mathbb{S}^{n+1-r}_+$. We can see by the definitions of $\mathcal{R}$, $\mathcal{Y}$, and $\mathcal{F}_{DNN}$, that the pair $(\hat{R}, \hat{Y})$ is feasible for (3.5.3). We claim that the relative interior of $\mathcal{R}$ corresponds to the positive definite matrices in $\mathcal{R}$, i.e.,

$$\text{ri}(\mathcal{R}) = \{R \in \mathbb{S}^{n+1-r} \colon \text{trace}(R) = t + 1, R \succ 0\}.$$

Let $\mathcal{Z} = \{R \in \mathbb{S}^{n+1-r} \colon \text{trace}(R) = t + 1\}$. Then since the trace is linear transformation, $\mathcal{Z}$ is an affine set. Therefore $\text{aff}(\mathcal{R}) \subseteq \mathcal{Z}$. Furthermore, since $\mathcal{Z} \subseteq \mathcal{R}$, we must have $\mathcal{Z} \subseteq \text{aff}(\mathcal{R})$, which gives $\text{aff}(\mathcal{R}) = \mathcal{Z}$. By recalling the definition of relative interior,

$$\begin{aligned}\text{ri}(\mathcal{R}) &= \{R \in \mathcal{R} \colon \exists \epsilon > 0 \text{ such that } B(R, \epsilon) \cap \text{aff}(\mathcal{R}) \subseteq \mathcal{R}\} \\ &= \{R \in \mathcal{R} \colon \exists \epsilon > 0 \text{ such that } B(R, \epsilon) \cap \mathcal{Z} \subseteq \mathcal{R}\}.\end{aligned}$$

Since $\mathcal{Z}$ is open, for any $R \in \mathcal{Z}$ there exists $\epsilon_0 > 0$ such that $B(x, \epsilon_0) \subseteq \mathcal{Z}$. Thus for any $R \in \mathcal{R}$, if $\exists \epsilon > 0$ such that $B(R, \epsilon) \in \mathbb{S}^{n+1-r}_+$, we can take $\epsilon_{\min} = \min(\epsilon, \epsilon_0)$ so that $B(R, \epsilon_{\min}) \subseteq \mathcal{Z} \cap \mathbb{S}^{n+1-r}_+ = \mathcal{R}$. Therefore,

$$\begin{aligned}\text{ri}(\mathcal{R}) &= \{R \in \mathcal{R} \colon \exists \epsilon > 0 \text{ such that } B(R, \epsilon) \subseteq \mathbb{S}^{n+1-r}_+\} \\ &= \{R \in \mathcal{R} \colon R \in \text{int}(\mathbb{S}^{n+1-r}_+)\} \\ &= \{R \in \mathcal{R} \colon R \in \mathbb{S}^{n+1-r}_{++}\}.\end{aligned}$$

The last line holds by Theorem 2.2.5, item 1.

We now show that given $\text{rank}(\hat{Y}) = n + 1 - r$ and $\hat{Y} = V\hat{R}V^T$, we have $\hat{R} \succ 0$. We already know that $\text{rank}(R) \leq n + 1 - r$ because $R \in \mathbb{S}^{n+1-r}_+$. Therefore, by Theorem 2.2.5, item 2d, to show that $\hat{R} \succ 0$, it suffices to show that $\text{rank}(R) \geq n + 1 - r$.

Since $\hat{Y}$ is obtained by multiplying $\hat{R}$ by matrices V and V^T on either side, the rank of $\hat{Y}$ can be at most the rank of $\hat{R}$. Thus

$$n + 1 - r = \text{rank}(\hat{Y}) \leq \text{rank}(\hat{R}),$$

as desired. Therefore, we conclude that $(\hat{R}, \hat{Y})$ is feasible for (3.5.3) and $\hat{R} \in \text{ri}(\mathcal{R})$. □

Remark 4.1.3. *We note that by Proposition 2.3.8, the condition $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$ is equivalent to saying that the face defined by $KY = 0$ is in fact the minimal face of $\mathbb{S}^{n+1}_+$ containing the feasible set $\mathcal{F}_{DNN}$.*

4.1.4 Optimality conditions

Due to the fact that the objective function of (4.1.2) is continuous and the feasible set is closed, we know that an optimal solution to (4.1.2) always exists. We now show that if there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$, then for any optimal solution to the primal problem (4.1.2), there exists a dual optimal solution with optimal value equal to the primal optimal value. A crucial step in the proof uses the normal cone intersection/sum property (4.1.10) from the previous section.

Proposition 4.1.4. *Suppose there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\text{rank}(\hat{Y}) = \dim(\text{null}(K))$. Then (R, Y) is optimal for (4.1.2) if and only if there exists Z satisfying:*

$$V^T Z V \in N_{\mathcal{R}}(R), \tag{4.1.11}$$

$$-\widehat{W} \in Z + N_{\mathcal{Y}}(Y). \tag{4.1.12}$$

Proof. Let $\mathcal{A}$ be the linear map defined by

$$\mathcal{A}(R, Y) = VRV^T - Y.$$

Then $\mathcal{A}(R, Y) = \mathcal{A}_1(R) + \mathcal{A}_2(Y)$ where,

$$\mathcal{A}_1(R) = VRV^T,$$
$$\mathcal{A}_2(Y) = -Y.$$

Recall that by definition, the adjoint $\mathcal{A}^*$ of $\mathcal{A}$ is the unique linear map satisfying

$$\langle \mathcal{A}(R, Y), Z \rangle = \langle (R, Y), \mathcal{A}^*(Z) \rangle.$$

By observing that

$$\langle \mathcal{A}(R, Y), Z \rangle = \langle VRV^T - Y, Z \rangle = \langle R, V^T Z V \rangle - \langle Y, Z \rangle = \langle (R, Y), (V^T Z V, -Z) \rangle,$$

we get

$$\mathcal{A}^*(Z) = (V^T Z V, -Z) = (\mathcal{A}_1^*(Z), \mathcal{A}_2^*(Z)).$$

As before, let $\mathcal{N} = \{(R, Y) : Y = VRV^T\}$ and let $\mathcal{K} = \mathcal{N} \cap \mathcal{R} \times \mathcal{Y}$. Then, by Corollary 2.1.10,

$$N_{\mathcal{N}}(R, Y) = \operatorname{range} \mathcal{A}^* = \operatorname{range}(\mathcal{A}_1^*) \times \operatorname{range}(\mathcal{A}_2^*).$$

By assumption, there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\operatorname{rank}(\hat{Y}) = \dim(\operatorname{null}(K))$. Therefore, by Lemma 4.1.2 and Theorem 2.4.3, we have that

$$N_{\mathcal{K}}(R, Y) = N_{\mathcal{N}}(R, Y) + (N_{\mathcal{R}}(R) \times N_{\mathcal{Y}}(Y))$$

holds for all $(R, Y) \in \mathcal{K}$. Therefore, by Corollary 2.4.2, we have that $(R, Y) \in \mathcal{K}$ is optimal for (4.1.2) if and only if

$$\begin{cases} -\nabla_R \langle \widehat{W}, Y \rangle & \in \operatorname{range}(\mathcal{A}_1^*) + N_{\mathcal{R}}(\bar{R}); \\ -\nabla_Y \langle \widehat{W}, Y \rangle & \in \operatorname{range}(\mathcal{A}_2^*) + N_{\mathcal{Y}}(\bar{Y}), \end{cases}$$

which is equivalent to saying that there exists Z such that

$$\begin{aligned} V^T Z V &\in N_{\mathcal{R}}(R), \\ -\widehat{W} &\in Z + N_{\mathcal{Y}}(Y). \end{aligned}$$

□

4.1.5 Lagrangian saddle point

We now use the optimality conditions proved in Section 4.1.4 to show that if there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\operatorname{rank}(\hat{Y}) = \dim(\operatorname{null}(K))$ then the Lagrangian has a saddle point, i.e. item 2 is satisfied. Recall that we already know that an optimal R, Y exists, and furthermore, there exists Z such that R, Y, Z satisfy (4.1.11). We now show that this R, Y, Z is also a saddle point of the Lagrangian.

Proposition 4.1.5. *Suppose there exists $\hat{Y} \in \mathcal{F}_{DNN}$ such that $\operatorname{rank}(\hat{Y}) = \dim(\operatorname{null}(K))$. If $(\bar{R}, \bar{Y})$ is optimal for (4.1.2), then there exists $\bar{Z}$ such that $(\bar{R}, \bar{Y}, \bar{Z})$ is a saddle point for the Lagrangian (4.1.3).*

Proof. Let $(\bar{R}, \bar{Y})$ be optimal for (4.1.2). By Proposition 4.1.4, there exists $\bar{Z}$ satisfying:

$$V^T \bar{Z} V \in N_{\mathcal{R}}(\bar{R}), \tag{4.1.13}$$

$$-\widehat{W} \in \bar{Z} + N_{\mathcal{Y}}(\bar{Y}). \tag{4.1.14}$$

We compute the gradients:

$$\begin{aligned} \nabla_R \mathcal{L}(R, \bar{Y}, \bar{Z}) &= -V^T \bar{Z} V, \\ \nabla_Y \mathcal{L}(\bar{R}, Y, \bar{Z}) &= \widehat{W} + \bar{Z}. \end{aligned} \tag{4.1.15}$$

Thus, (4.1.13) is equivalent to:

$$\begin{cases} 0 \in \nabla_R \mathcal{L}(\bar{R}, \bar{Y}, \bar{Z}) + N_{\mathcal{R}}(\bar{R}) \\ 0 \in \nabla_Y \mathcal{L}(\bar{R}, \bar{Y}, \bar{Z}) + \mathcal{N}_{\mathcal{Y}}(\bar{Y}). \end{cases} \tag{4.1.16}$$

By Theorem 2.4.1, (4.1.16) is equivalent to

$$(\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \mathcal{L}(R, Y, \bar{Z}).$$

Next, we note that since $\nabla_Z \mathcal{L}(\bar{R}, \bar{Y}, Z) = \bar{Y} - V\bar{R}V^T$,

$$\begin{aligned} \bar{Y} - V\bar{R}V^T &\iff \nabla_Z \mathcal{L}(\bar{R}, \bar{Y}, Z) = 0 \\ &\iff \bar{Z} \in \operatorname{argmax}_Z \mathcal{L}(\bar{R}, \bar{Y}, Z). \end{aligned}$$

This proves that that $(\bar{R}, \bar{Y}, \bar{Z})$ is a saddle point of the Lagrangian. □

Finally we use the the existence of a saddle point of the Lagrangian to conclude that strong duality holds.

Lemma 4.1.6. *If the Lagrangian (4.1.3) has a saddle point $(\bar{R}, \bar{Y}, \bar{Z})$, then strong duality holds. Moreover, $(\bar{R}, \bar{Y}, \bar{Z})$ is primal-dual optimal.*

Proof. Let $(\bar{R}, \bar{Y}, \bar{Z})$ be a saddle point. Then,

$$\bar{Z} \in \operatorname{argmax}_Z \mathcal{L}(\bar{R}, \bar{Y}, Z) \text{ and } (\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \mathcal{L}(R, Y, \bar{Z}).$$

Note that $\mathcal{L}(\bar{R}, \bar{Y}, Z)$ is concave in Z. Thus we can apply Theorem 2.4.1 to get:

$$\begin{aligned} &\bar{Z} \in \operatorname{argmax}_Z \mathcal{L}(\bar{R}, \bar{Y}, Z) \\ \iff &\nabla_Z \mathcal{L}(\bar{R}, \bar{Y}, \bar{Z}) = 0 \\ \iff &\bar{Y} = V\bar{R}V^T. \end{aligned}$$

Since $(\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \mathcal{L}(R, Y, \bar{Z})$ and $\bar{Y} = V\bar{R}V^T$,

$$\begin{aligned} &(\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \{\mathcal{L}(R, Y, \bar{Z}) \colon \bar{Y} = V\bar{R}V^T\} \\ \iff &(\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \{\langle \widehat{W}, Y \rangle \colon \bar{Y} = V\bar{R}V^T\} \\ \iff &\mathcal{L}(\bar{R}, \bar{Y}, \bar{Z}) = p^*. \end{aligned}$$

Therefore, $\bar{R}, \bar{Y}$ is optimal for the primal problem (4.1.2). Furthermore

$$\begin{aligned} &(\bar{R}, \bar{Y}) \in \operatorname{argmin}_{R \in \mathcal{R}, Y \in \mathcal{Y}} \mathcal{L}(R, Y, \bar{Z}) \\ \iff &g(\bar{Z}) = \mathcal{L}(\bar{R}, \bar{Y}, \bar{Z}) = \langle \widehat{W}, \bar{Y} \rangle + \langle \bar{Z}, \bar{Y} - V\bar{R}V^T \rangle = \langle \widehat{W}, \bar{Y} \rangle = p^*. \end{aligned}$$

Thus, since weak duality holds, $g(\bar{Z}) = p^*$ implies that $\bar{Z} \in \operatorname{argmax}_Z g(Z)$. In other words, $\bar{Z}$ is optimal for the dual problem, and, $d^* = g(\bar{Z}) = p^*$. □

In this section so far we have shown that if we can determine the existence of $Y \in \mathcal{F}_{DNN}$ such that $\operatorname{range}(Y) = \operatorname{null}(K)$, then the Lagrangian has a saddle point, strong duality holds, and also the conditions:

$$\begin{aligned} V^T Z V &\in N_{\mathcal{R}}(R), \\ -\widehat{W} - Z &\in N_{\mathcal{Y}}(Y), \\ Y = VRV^T \quad Y \in \mathcal{Y},\, R &\in \mathcal{R}. \end{aligned}$$

are necessary and sufficient for optimality. But this raises the question: when does such a Y exist? We have not yet addressed how to check this. The examples of the **QAP** and the **SCP** problem are structured in such a way that we can check this by constructing such a feasible Y with $\operatorname{rank}(Y) = \dim(\operatorname{null} K))$. By definition of K, we know that the null space of K is determined by the null space of $\begin{bmatrix} -\bar{e}_p & H \end{bmatrix}$. We are interested then in the question of whether the may be some more general property of H which lead us to conclude the existence of a sufficiently high rank feasible Y. We don't have the answer to this question, but we investigate it further in the next section.

4.1.6 Alternative condition for strong duality

We rephrase this condition in the following theorem in hopes that it will help make it easier to answer. Much of the observations made in the proof come from [12, Section 6.3].

Theorem 4.1.7. *Let*

$$\mathcal{F}_B := \{x \in \mathbb{R}^n \colon Hx = e, x \in \{0,1\}^n\} \quad \mathcal{H} := \{x \in \mathbb{R}^n \colon Hx = e\}$$

If $\operatorname{aff}(F_B) = \mathcal{H}$ *then there exists* $\hat{Y}$ *feasible such that* $\operatorname{range}(\hat{Y}) = \operatorname{null}(K)$. *In particular, strong duality holds for* (3.5.3).

Proof. Suppose that $\operatorname{aff}(F_B) = \mathcal{H}$, and let $d := \dim \mathcal{H} = \dim \operatorname{aff}(F_B)$. Then there exists $x_1, \ldots, x_{d+1}$ affinely independent vectors in F_B. Therefore the vectors $\begin{pmatrix} 1 \\ x_1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ x_{d+1} \end{pmatrix}$ are affinely independent. Then let $\hat{Y}$ be the barycentre:

$$\hat{Y} = \frac{1}{d+1} \sum_{i=1}^{d+1} \begin{pmatrix} 1 \\ x_i \end{pmatrix} \begin{pmatrix} 1 \\ x_i \end{pmatrix}^T.$$

Therefore $\operatorname{rank}(\hat{Y}) = d + 1$ and $\hat{Y} \in \mathcal{F}_{DNN}$. Recall that $\mathcal{F}_V$ is defined to be

$$\begin{aligned} \mathcal{F}_V &= \{X \in \mathbb{S}^{n+1}_+ \colon \operatorname{range}(X) \subseteq \operatorname{null}(K)\} \\ &= \{X \in \mathbb{S}^{n+1}_+ \colon \operatorname{range}(X) \subseteq \operatorname{range}(V)\}. \end{aligned}$$

We already have range$(\hat{Y}) \subseteq$ null(K). So it remains to show that null$(K) = d+1$. This is true since dim null(K) = dim null$(\begin{bmatrix} -\bar{e}_p & H \end{bmatrix})$ = dim $\mathcal{H} + 1 = d + 1$. □

Therefore we can determine that strong duality holds for (3.5.3) if we can determine that the dimension of integer polytope $\mathcal{F}_B$ is the same as the dimension of the subspace $\mathcal{H}$. We show in later chapters that strong duality holds the **QAP** and the **SCP** by giving an explicit formula for a feasible Y with range(Y) = null(K). However, it is not clear if these are simply special cases of the **BCQP** or if there are some addition assumptions we can place on the **BCQP** to show that such a Y will exist.

4.2 Computational strategies

We now look at the computational aspect of finding explicit iterates for (4.1.6) and (4.1.7) and using the output $(R^{out}, Y^{out}, Z^{out})$ to obtain upper and lower bounds on the optimal value of the original problem (3.1.1). These strategies are found in previous work, specifically in [6, 7, 16, 24] for the **QAP** and the **SCP** problem. We present these strategies for the general **BCQP** in order to highlight how the choice of constraints in the model (3.5.3) is related to, and motivated by, the computability of the resulting **ADMM** subproblems.

4.2.1 R subproblem

We refer to the problem argmin$_{R\in\mathcal{R}} \mathcal{L}_A(R, Y^k, Z^k)$ as the R *subproblem* and to the problem argmin$_{Y\in\mathcal{Y}} \mathcal{L}_A(R^{k+1}, Y, Z^k)$ as the Y *subproblem*. These subproblems can be solved explicitly. Here we assume $\widehat{V}^T\widehat{V} = I$. We begin by rearranging terms and competing the square:

$$\begin{aligned} R^{k+1} &= \text{argmin}_{R\in\mathcal{R}}\, \mathcal{L}_A(R, Y^k, Z^k) \\ &= \text{argmin}_{R\in\mathcal{R}}\, -\langle Z, Y\widehat{V}R\widehat{V}^T\rangle + \frac{\beta}{2}\left\|Y - \widehat{V}R\widehat{V}^T\right\|_F^2 \\ &= \text{argmin}_{R\in\mathcal{R}}\, \frac{\beta}{2}\left\|R - \widehat{V}(Y^k + \frac{1}{\beta}Z^k)\widehat{V}\right\|_F^2 \qquad (4.2.1) \\ &= \mathcal{P}_{\mathcal{R}}(\widehat{V}^T(Y^k + \frac{1}{\beta}Z^k)\widehat{V}). \end{aligned}$$

The equality at (4.2.1) holds by the assumption that $\widehat{V}^T\widehat{V} = I$. Let $\widehat{X} := \widehat{V}^T(Y^k + \frac{1}{\beta}Z^k)\widehat{V} \in \mathbb{S}^{n+1-r}$ with the spectral decomposition

$$\widehat{X} = Q\Lambda Q^T.$$

If there is no trace constraint in $\mathcal{R}$, it is possible to obtain $\mathcal{P}_{\mathcal{R}}(\widehat{X})$ by projecting $\widehat{X}$ onto the positive semidefinite cone [6, 24] using Eckart-Young Theorem [13]. Otherwise, as in [7, 16],

$\mathcal{P}_{\mathcal{R}}(\widehat{X})$ is obtained by projecting diag(Λ) onto the *unit simplex,* Δ defined by:

$$\Delta := \{\lambda \in \mathbb{R}_+^{n+1-p} \colon \sum_{i=1}^{n+1-p} \lambda_i = n+1\}.$$

Explicitly, [16] computes the solution to R the subproblem (4.1.6) by computing:

$$R^{k+1} = Q\mathcal{P}_{\Delta}(\text{diag}(\Lambda))Q^T.$$

Where Λ is the diagonal matrix of eigenvalues of $\widehat{V}^T(Y^k + \frac{1}{\beta}Z^k)\widehat{V}$.

4.2.2 Y subproblem

We show how to obtain an explicit formula for the Y-update by examining the Y subproblem. Again by completing the square we obtain a projection, this time, onto the set $\mathcal{Y}$.

$$\begin{aligned} Y^{k+1} &= \text{argmin}_{Y\in\mathcal{Y}}\, \mathcal{L}_A(R^{k+1}, Y, Z^k) \\ &= \text{argmin}_{Y\in\mathcal{Y}}\, \frac{\beta}{2}\left\| Y - \left(\widehat{V}R^{k+1}\widehat{V}^T - \frac{1}{\beta}(\widehat{W} + Z^k)\right)\right\|_F^2 \\ &= \mathcal{P}_{\mathcal{Y}}(\widehat{V}R^{k+1}\widehat{V}^T - \frac{1}{\beta}(\widehat{W} + Z^k)) \end{aligned}$$

Now we redefine $\widehat{X} = \widehat{V}R^{k+1}\widehat{V}^T - \frac{1}{\beta}(\widehat{W} + Z^k)$. $\mathcal{P}_{\mathcal{Y}}(\widehat{X})$ denotes the projection of X onto the set $\mathcal{Y}$. Then by definition of the set $\mathcal{Y}$, we get:

$$\left(Y^{k+1}\right)_{ij} = \left(P_{\mathcal{Y}}(\widehat{X})\right)_{ij} = \begin{cases} 1 \text{ if } i = j = 0, \\ 0 \text{ if } (i,j) \in \bar{J}, \\ \min\{1, \max\{X_{ij}, 0\}\} \text{ otherwise.} z \end{cases}$$

4.2.3 Upper bound

Let $(Y^{out}, R^{out}, Z^{out})$ be the output from **ADMM** . A strategy used successfully [24] [6] [16] for obtaining an upper bound on the optimal solution is to take the second through last entries of the first column of Y^{out} to obtain a vector $x^{out} \in \mathbb{R}^n$ that satisfies $Hx^{out} = \bar{e}_p$ and $x^{out} \in [0,1]$. Next it is necessary to obtain an integral solution x^* from the fractional solution x^{out} so that x^* is feasible for (3.1.1). If we let $c = x^{out}$, we can find the nearest integral solution by solving the following problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & \|x - c\|^2 \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in \{0,1\}^n. \end{array} \tag{4.2.2}$$

Here we must make the assumption that our problem satisfies the condition (3.4.3). Recall that if this condition holds then by lemma 3.4.6, every feasible solution x satisfies $x^T x = t$. Thus we can replace the quadratic objective with a linear objective by:

$$\|x - c\|^2 = -2c^T x + \|c\|^2 + t.$$

Therefore we get an equivalent linear program:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & -c^T x \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in [0,1]^n. \end{array} \tag{4.2.3}$$

In order to guarantee that a solution to (4.2.2) will return a solution to (4.2.3) we must have that the set of extreme points of the set $\{x \in \mathbb{R}^n \colon Hx = e, x \in [0,1]^n\}$ are integral. If H is totally unimodular, as is the case for the **QAP** and the **SCP** problem, then this holds. Thus, if H is totally unimodular, then (4.2.3) is equivalent to (4.2.2). To obtain a feasible solution to (4.2.2) it suffices to run the simplex method on (4.2.3).

4.2.4 Lower bound

As (4.1.2) is a relaxation (3.3.1), finding the optimal solution of (4.1.2) to a very high accuracy will yield a lower bound on the optimal value of (3.3.1). However, using **ADMM** to obtain a high accuracy solution can be time consuming [5]. Instead, we solve (4.1.2) to moderate accuracy and, taking advantage of the fact that strong duality holds, and that the dual iterate converges to the optimum dual multiplier, we may use a suboptimal dual output variable Z^{out} to obtain a lower bound. This requires simply computing the value of $g(Z^{out})$. We can facilitate this computation using the method from [16, 22]. This method uses a new functional $h(Z)$ to obtain an equivalent dual problem to (4.1.4). We will first require a small lemma which we prove now.

Lemma 4.2.1. *Let $A \in \mathbb{S}^n$. Then*

$$\max_{B \in \mathbb{S}^n_+ \, \mathrm{tr}(B) = k} \langle A, B \rangle = k\lambda_{\max}(A).$$

Proof. Let $A = UD_AU^T$ be the spectral decomposition of A.

$$\begin{aligned}
\max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} \langle A, B\rangle &= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} \langle D_A, B\rangle && (4.2.4)\\
&= \sum_{i=1}^{n} \lambda_i(A) B_{ii} && (4.2.5)\\
&\leq \sum_{i=1}^{n} \lambda_{\max}(A) B_{ii}\\
&= \lambda_{\max} \mathrm{tr}(B)\\
&= k\lambda_{\max}(A)
\end{aligned}$$

Note that (4.2.4) holds because

$$\begin{aligned}
\max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} \langle A, B\rangle &= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} \langle UD_AU^T, B\rangle\\
&= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} tr(UD_AU^TB)\\
&= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} tr(D_AU^TBU),
\end{aligned}$$

and $B \succeq 0$ if and only if $U^TBU \succeq 0$, hence,

$$\begin{aligned}
\max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} tr(D_AU^TBU) &= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} tr(D_AB)\\
&= \max_{B\in\mathbb{S}_+^n \ \mathrm{tr}(B)=k} \langle D_A, B\rangle.
\end{aligned}$$

Lastly, (4.2.5) holds because B is positive semidefinite, thus all diagonal entries of B are non-negative. □

Proposition 4.2.2. *Let*

$$h(Z) := \min_{Y\in\mathcal{Y}} \langle \widehat{W}, Y\rangle + (n+1)\lambda_{max}(V^TZV) \quad d_Z^* := \max_Z h(Z).$$

Then

$$d_Z^* \text{ is attained and } d_Z^* = d^*.$$

Proof.

$$\begin{aligned} d^* &= \max_Z \min_{R\in\mathcal{R}, Y\in\mathcal{Y}} \left\{ \langle \widehat{W}, Y\rangle + \langle Z, Y - VRV^T \right\} \\ &= \max_Z \left\{ \min_{Y\in\mathcal{Y}} \left\{ \langle \widehat{W}, Y\rangle + \langle Z, Y\rangle \right\} + \min_{R\in\mathcal{R}} \left\{ \langle V^T Z V, -R\rangle \right\} \right\} \\ &= \max_Z \left\{ \min_{Y\in\mathcal{Y}} \left\{ \langle \widehat{W}, Y\rangle + \langle Z, Y\rangle \right\} - \max_{R\in\mathcal{R}} \left\{ \langle V^T Z V, R\rangle \right\} \right\} \\ &= \max_Z \left\{ \min_{Y\in\mathcal{Y}} \left\{ \langle \widehat{W}, Y\rangle + \langle Z, Y\rangle \right\} - (t+1)\lambda_{\max}(V^T Z V) \right\} \qquad (4.2.6) \\ &= d_Z^* \end{aligned}$$

Equality at (4.2.6) holds as a consequence of Lemma 4.2.1. □

By this Proposition 4.2.2, we have that $h(Z^{out}) = g(Z^{out}) \leq p^*$. It's clear now that to obtain a lower bound on p^* we simply take the output dual variable Z^{out} and compute:

$$h(Z^{out}) = \min_{Y\in\mathcal{Y}} \left\{ \langle \widehat{W}, Y\rangle + \langle Z^{out}, Y\rangle \right\} - (t+1)\lambda_{\max}(V^T Z^{out} V).$$

Computing $(t+1)\lambda_{\max}(V^T Z^{out} V)$ is just a matter of finding the largest eigenvalue of $V^T Z^{out} V$. As for solving the minimization problem $\min_{Y\in\mathcal{Y}} \langle \widehat{W} + Z^{out}, Y\rangle$, we take advantage of the nice structure of the set $\mathcal{Y}$ and the fact that this minimization problem is separable into scalar minimization problems: By the definition of the inner product,

$$\langle \widehat{W} + Z^{out}, Y\rangle = \sum_{ij} (\widehat{W} + Z^{out})_{ij} Y_{ij}.$$

Therefore

$$\begin{aligned} \min_{Y\in\mathcal{Y}} \langle \widehat{W} + Z^{out}, Y\rangle &= \min_{Y\in\mathcal{Y}} \sum_{ij} (\widehat{W} + Z^{out})_{ij} Y_{ij} \\ &= \sum_{ij\in J^c} \min_{0\leq Y_{ij}\leq 1} (\widehat{W} + Z^{out})_{ij} Y_{ij} + \sum_{ij\in J^c} \min_{Y_{ij}=0} (\widehat{W} + Z^{out})_{ij} Y_{ij}. \end{aligned}$$

We obtain from inspection that the minimizer Y^* is:

$$(Y^*)_{ij} = \begin{cases} 0 & \text{if } (i,j) \in J, \\ 1 & \text{if } (i,j) \notin J \text{ and } (\widehat{W} + Z^{out})_{ij} < 0, \\ 0 & \text{if } (i,j) \notin J \text{ and } (\widehat{W} + Z^{out})_{ij} \geq 0. \end{cases}$$

Chapter 5

The Quadratic Assignment Problem

5.1 Background

The Quadratic Assignment Problem (**QAP**) is a combinatorial optimization problem that is of fundamental importance in optimization. As discussed in Section 3.2.3, the Traveling Salesman Problem is a particular case of the **QAP**. This means that the **QAP** not only NP-hard, but also is NP-hard to approximate, assuming that P $\neq$ NP. Before defining the **QAP**, we get some preliminary definitions out of the way.

Definition 5.1.1 (*permutation, π*)**.** *Let $\mathcal{I} = \{1, ..., n\}$. A permutation of the set $\mathcal{I}$ is a bijective map $\pi : \mathcal{I} \to \mathcal{I}$.*

Remark 5.1.2. *We denote the set of all permutations on n elements by $\mathcal{S}_n$.*

Definition 5.1.3 (doubly stochastic matrices)**.** *The set of doubly stochastic matrices is defined as the following:*

$$\mathcal{D} := \left\{ X \in \mathbb{R}^{n \times n} : Xe = e, X^T e = e, X_{ij} \geq 0 \ \forall i, j \in \{1, ...n\} \right\}.$$

Definition 5.1.4 (binary matrices)**.** *The set of $n \times n$ matrices with all entries in $\{0, 1\}$ is denoted by:*

$$\mathcal{Z} := \left\{ X \in \mathbb{R}^{n \times n} : X_{ij} \in \{0.1\}, \forall i, j \in \{1, ...n\} \right\}.$$

Definition 5.1.5 ([4, 32])**.** *The set of permutation matrices Π is defined as $\Pi := \mathcal{D} \cap \mathcal{Z}$.*

The **QAP** was first introduced in 1957 by Koopmans and Beckman to model the *facility location problem* [21]. In the facility location problem, we have a set of n *locations* and n *facilities*. All locations are at some given fixed distance apart. We may imagine that the

set of locations represents a collection of holes that are already dug in the ground, and ready to have a building constructed in its place. We then also have n *facilities* and a associated *flow* between every pair of facilities. We imagine these facilities to be buildings and for each building, we must decide which location (among the n given locations) is best suited for its construction. The flows can represent, for example, expected foot traffic, or any similar quantity that we would like to minimize. We can imagine that two buildings serve some function that results in there being a lot of people traveling in between them. Thus, placing two such facilities further apart results in greater total travel, which is clearly undesirable. As such, we prefer to place facilities which have a high flow between them, closer together, and vice versa. To do so optimally, we must minimize:

$$\sum_{\text{location pairs } (i,j)} \text{distance}(i,j) \times \text{flow}(\pi(i), \pi(j)),$$

over all possible choices of permutation π. A permutation is equivalent to an assignment of facilities to locations. That is to say, we place the facility $\pi(i)$ at the location i, for every $i \in \{1, \ldots, n\}$. Then, for every pair of locations (i, j), we take the distance between them and multiply it by the flow between two facilities $\pi(i)$ and $\pi(j)$. The cost of placing a single facility at a specific location will depend on the location of all other facilities, so it is necessary to sum up costs in a pairwise manner as above. Intuitively, it makes sense that such a problem is not amenable to a greedy algorithm and that this problem possesses many local minima, making it highly non-convex and thus very difficult to solve.

Most discussion on the subject of the **QAP** uses the language of the facility location problem, even when we not dealing with the problem of physically assigning facilities to locations. Strictly speaking, the **QAP** is the more abstract problem of finding a minimum-cost pairing between elements of two equally-sized sets, given some *flow* values for all pairs of elements in the first set, and some *distance* values between all pairs of elements in the second set. The *flow* and *distance* values are fundamentally no different, just as it is arbitrary to decide which set corresponds to the facilities and which corresponds to the locations. Using the words location, facility, flow and distance merely allows us to refer to these two separate sets and the weights of their respective pairs in a clear and unambiguous way.

The commonly used formulation of **QAP** is called the Koopmans-Beckman formulation [8], which we define now. Suppose we have the following input data:

- a_{ij}, *the flow between facilities* i *and* j, $\forall i, j \in \{1, \ldots n\}$,
- b_{ij}, the *distance between locations* i *and* j, $\forall i, j \in \{1, \ldots n\}$.

The Koopmans-Beckman **QAP** is then following optimization problem:

$$\min_{\pi \in \mathcal{S}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi(i)\pi(j)} b_{ij}. \tag{5.1.1}$$

A slightly more general formulation allows for a linear cost term, cost $c_{\pi(i)i}$, which represents the cost associated to placing facility $\pi(i)$ at location i:

$$\min_{\pi \in \mathcal{S}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi(i)\pi(j)} b_{ij} + 2 \sum_{i=1}^{n} c_{\pi(i)i}. \tag{5.1.2}$$

Given an instance of the **QAP** of the form (5.1.2), we can define its equivalent *trace formulation*. To show that this works, we must first define the notion of a *permutation matrix associated to* π.

Definition 5.1.6. *Let $\pi \in \mathcal{S}_n$. We define the permutation associated to π as the $n \times n$ matrix that satisfies:*

$$(X_\pi)_{ij} = \begin{cases} 1 & \text{if } \pi(j) = i \\ 0 & \text{otherwise.} \end{cases}$$

Remark 5.1.7. *We denote the set of $n \times n$ permutation matrices by Π_n.*

Lemma 5.1.8. *Let $\pi \in \mathcal{S}_n$ and let X_π be the associated permutation matrix. Then we obtain the permutation matrix X_π by permuting the columns of an identity matrix according to π. In other words, let $e_1, \ldots, e_n$ denote the standard basis vectors, then*

$$I = \begin{bmatrix} e_1^T & \ldots & e_n^T \end{bmatrix} \quad X_\pi = \begin{bmatrix} e_{\pi(1)} & \ldots & e_{\pi(n)} \end{bmatrix}.$$

Proof. This follows directly from Definition 5.1.6. □

Given an instance of the **QAP** in the form of (5.1.2). We let A, B, C be symmetric matrices such that $A_{ij} = a_{ij}$, $B_{ij} = b_{ij}$, and $C_{ij} = -c_{ij}$. We obtain the following *trace formulation*

$$(QAP) \qquad \min_{X \in \Pi_n} \langle AXB - 2C, X \rangle, \tag{5.1.3}$$

where a feasible solution X is interpreted as:

$$X_{ij} = \begin{cases} 1 & \text{if facility } i \text{ is to be built at location } j, \\ 0 & \text{otherwise.} \end{cases}$$

The connection between (5.1.2) and (5.1.3) is straightforward if we apply the following lemma:

Lemma 5.1.9. *Let $\pi \in \mathcal{S}_n$ and let X be the associated permutation matrix. Let $A \in \mathbb{S}^n$. Then*

$$(X^T A X)_{ij} = A_{\pi(i)\pi(j)}.$$

Proof. To proof follows from these two facts:

1. multiplying A on the right by X rearranges the columns,

2. multiplying A on the left by X^T rearranges the rows.

Item 1 and Item 2 follow from straightforward matrix multiplication. □

Applying Lemma 5.1.9 we see the equivalence between (5.1.2) and (5.1.3).

$$\begin{aligned}\langle AXB - 2C, X\rangle &= \operatorname{trace}(X^T AXB) - 2\operatorname{trace}(X^T C)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{\pi(i)\pi(j)}B_{ji} - 2\sum_{i=1}^{n} C_{\pi(i)i}\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{\pi(i)\pi(j)}b_{ij} + 2\sum_{i=1}^{n} c_{\pi(i)i}\end{aligned}$$

Although the Koopmans-Beckman formulation is more transparent in terms of what optimization problem we are solving, the trace formulation is easier to manipulate. For this reason we will use the trace formulation going forward.

5.2 SDP relaxation of the QAP

To obtain the **SDP** relaxation for the **QAP** , we will first show that it can be written as a **BCQP**. As a result, it suffices to apply the theory established in Chapter 3 to obtain the **SDP** model and the **ADMM** implementation. In this section we show that all necessary assumptions are satisfied and we obtain the resulting algorithm by applying the results in Chapter 3. The results will be the same as previous work applying **ADMM** to the **DNN** relaxation of the **QAP** in [24] and [16], thus we do this simply to confirm that the process works. In Chapter 6 we do the same again fo the **SCP** problem. An implementation is done in

5.2.1 The QAP as a BCQP

Let $A, B, C \in \mathbb{S}^n$ be given, and let Π_n denote the set of $n \times n$ permutation matrices. Recall the trace formulation of the **QAP**is:

$$(QAP) \qquad \min_{X\in\Pi_n} \langle AXB - 2C, X\rangle. \tag{5.2.1}$$

We obtain a more explicit form for the constraint $X \in \Pi_n$ by recalling that $\Pi = \mathcal{D} \cap \mathcal{Z}$ [4, 32].Thus we can rewrite (5.2.1) to obtain:

$$\begin{array}{ll} \min\limits_{X \in \mathbb{R}^{n \times n}} & \langle AXB - 2C, X \rangle \\ \text{s.t.} & Xe = e \\ & X^T e = e \\ & X \circ X - X = 0. \end{array} \tag{5.2.2}$$

To write this in the form of (3.3.1) we simply take advantage of the property of the kronecker product:

Lemma 5.2.1. *Let A, B, X be real matrices with the appropriate compatible dimensions. Then*

$$\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X).$$

Proposition 5.2.2. *Let A, B, X be as in (5.2.2) and let*

$$x = \operatorname{vec}(X) \in \mathbb{R}^{n^2}, \quad c = \operatorname{vec}(C) \in \mathbb{R}^{n^2}.$$

Then

$$\langle AXB - 2C, X \rangle = x^T (B^T \otimes A)x - 2c^T x.$$

Proof.

$$\begin{aligned} \langle AXB - 2C, X \rangle &= \langle AXB, X \rangle - 2\langle C, X \rangle \\ &= \langle \operatorname{vec}(AXB), \operatorname{vec}(X) \rangle - 2\langle \operatorname{vec}(C), \operatorname{vec}(X) \rangle \\ &= \langle B^T \otimes A\operatorname{vec}(X), \operatorname{vec}(X) \rangle - 2\langle \operatorname{vec}(C), \operatorname{vec}(X) \rangle \\ &= \operatorname{trace} B^T \otimes A\operatorname{vec}(X)\operatorname{vec}(X)^T - \operatorname{trace} \operatorname{vec}(C)\operatorname{vec}(X)^T \\ &= x^T (B^T \otimes A)x - 2c^T x \end{aligned}$$

□

Proposition 5.2.3. *Let $X \in \mathbb{R}^{n \times n}$, let $x := \operatorname{vec}(X)$ and let*

$$H := \begin{bmatrix} e^T \otimes I \\ I \otimes e^T \end{bmatrix} \in \mathbb{R}^{n^2 \times 2n}. \tag{5.2.3}$$

Then

$$\begin{cases} Xe = e \\ X^T e = e \\ X \circ X - X = 0 \end{cases} \iff \begin{cases} Hx = \bar{e}_{2n} \\ x \in \{0, 1\}^{n^2}. \end{cases}$$

Proof. By Lemma 5.2.1 the following are equivalent:

$$\begin{aligned} Xe &= e \\ IXe &= e \\ (e^T \otimes I)\text{vec}(X) &= \text{vec}(e) \\ (e^T \otimes I)x &= e. \end{aligned}$$

Similarly, the following are also equivalent:

$$\begin{aligned} X^T e &= e \\ e^T X I &= e^T \\ (I^T \otimes e^T)\text{vec}(X) &= \text{vec}(e^T) \\ (I \otimes e^T)x &= e. \end{aligned}$$

Lastly, $X \circ X - X = 0$ if and only if all entries of X are in $\{0,1\}$. Equivalently, $x \in \{0,1\}^{n^2}$. □

By Proposition 5.2.2 and Proposition 5.2.3, we obtain following **BCQP** model below which is equivalent to (5.2.1):

$$\begin{aligned} \min_{x} \quad & x^T(B^T \otimes A)x - 2c^T x \\ \text{s.t.} \quad & Hx = \bar{e}_{2n} \\ & x \in \{0,1\}^{n^2}. \end{aligned} \tag{5.2.4}$$

5.2.2 Rank one reformulation of the QAP

Now that it is established that the **QAP** is an instance of a **BCQP** , we can obtain the **SDP** relaxation in the analogous way to Section 3.3.2. We first write the lifted reformulation, which requires that we define the following matrices:

$$L := \begin{bmatrix} 0 & -(\text{vec}(C)^T) \\ -\text{vec}(C) & B \otimes A \end{bmatrix},$$

$$K = \begin{bmatrix} n^2 & \bar{e}_{n^2}^T H \\ H^T \bar{e}_{n^2} & H^T H \end{bmatrix} = \begin{bmatrix} \bar{e}_{n^2} & H \end{bmatrix}^T \begin{bmatrix} \bar{e}_{n^2} & H \end{bmatrix}.$$

Applying the process detailed in Section 3.3.1, we linearize the objective and rewrite the constraints on x as constraints on Y. We obtain the following lifted reformulation of (5.2.1):

$$\begin{aligned} \min_{Y \in \mathbb{S}^{n^2+1}} \quad & \langle L, Y \rangle \\ \text{s.t.} \quad & KY = 0 \\ & \text{arrow}(Y) = e_0 \\ & \text{rank}(Y) = 1. \end{aligned} \tag{5.2.5}$$

Then, as in Section 3.3.2 our initial **DNN** relaxation for (5.2.5) is obtained by relaxing the rank one constaint and adding the cutting planes $0 \leq Y \leq 1$. We obtain:

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n^2+1}} & \langle L, Y \rangle \\ \text{s.t.} & \text{arrow}(Y) = e_0 \\ & KY = 0 \\ & Y \succeq 0 \\ & 0 \leq Y \leq 1. \end{array} \tag{5.2.6}$$

5.2.3 Gangster constraint for the QAP

As in Section 3.3.3 we will modify the model by removing the arrow constraint, adding the gangster constraint as well as the trace constraint. In order to obtain an expression for the gangster indices J we simply apply Theorem 3.3.4. Recall by the definition of H we have:

$$\begin{aligned} H^T H &= \begin{bmatrix} (e^T \otimes I)^T & (I \otimes e^T)^T \end{bmatrix} \begin{bmatrix} e^T \otimes I \\ I \otimes e^T \end{bmatrix} \\ &= (e \otimes I)(e^T \otimes I) + (I \otimes e)(I \otimes e^T) \\ &= ee^T \otimes I + I \otimes ee^T. \end{aligned} \tag{5.2.7}$$

To see the elegant structure of this set, we write (5.2.7) explicitly:

$$ee^T \otimes I = \begin{bmatrix} I & I & \dots & I \\ I & I & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \dots & I \end{bmatrix} \quad I \otimes ee^T = \begin{bmatrix} E & 0 & \dots & 0 \\ 0 & E & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & E \end{bmatrix}.$$

Since $\text{Diag}(H^T H) = 2I$, then by Theorem 3.3.4 the gangster indices correspond to the indices of the nonzero entires of the matrix:

$$H^T H - 2I = \begin{bmatrix} \bar{E} - I & I & \dots & I \\ I & \bar{E} - I & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \dots & \bar{E} - I \end{bmatrix}.$$

We see now that $\bar{J}$ is equal to the index of the following elements:

- The off-diagonal elements of the n diagonal n-blocks.
- The diagonal elements of the off-diagonal n-blocks.

5.2.4 Trace constraint for the QAP

It is apparent from the structure of (5.2.3) that (5.2.4) satisfies the condition (3.4.3). We can apply Proposition 3.4.5 to get that $\operatorname{trace} Y = n + 1$. Furthermore, as in Section 3.4.1, the arrow constraint is redundant and we chose to remove it. Thus we get the equivalent refined model:

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n^2+1}} & \langle L, Y \rangle \\ \text{s.t.} & G_J(Y) = E_{00} \\ & KY = 0 \\ & \operatorname{trace} Y = n + 1 \\ & 0 \leq Y \leq 1 \\ & Y \succeq 0. \end{array} \tag{5.2.8}$$

The next and final step is to derive the split **SDP** model, the one on which we apply **ADMM**.

5.2.5 Split model for the QAP

Analogously to section 3.5 we define the sets:

$$\begin{aligned} \mathcal{R} &:= \{R \in \mathbb{S}_+^{n-p} \,:\, \operatorname{trace}(R) = t + 1\}, \\ \mathcal{Y} &:= \{Y \in \mathbb{S}_+^{n+1} \,:\, 0 \leq Y \leq 1, G_J(Y) = e_0\}, \end{aligned} \tag{5.2.9}$$

and let V be a matrix such that the columns of V form an orthonormal basis for $\operatorname{null}(K)$. In [34], Zhao et al. propose a choice of basis for $\operatorname{null}(K)$. They let

$$V_0 = \begin{bmatrix} I_{n-1} \\ -e_{n-1}^T \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}, \tag{5.2.10}$$

$$V = \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n} e_{n^2} & V_0 \otimes V_0 \end{array}\right] \in \mathbb{R}^{(n^2+1) \times (n-1)^2+1}. \tag{5.2.11}$$

We use the fact prove in [34] that $\operatorname{rank}(V) = (n-1)^2+1$ and $\operatorname{range}(V) = \operatorname{null}(K)$. Using QR decomposition V, we can obtain $\widehat{V}$ that satisfies $\widehat{V}^T\widehat{V} = I$ with the same dimensions and range as V. Thus we assume going forward that $\widehat{V}$ is some matrix with $\operatorname{rank}(\widehat{V}) = \operatorname{null} K$ and $\widehat{V}^T\widehat{V} = I$. We then have the split model:

$$\begin{array}{ll} \min\limits_{R,Y} & \langle L, Y \rangle \\ \text{s.t.} & Y = \widehat{V} R \widehat{V}^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}. \end{array} \tag{5.2.12}$$

We define the optimal value of the problem (5.2.12):

$$p^*_{QAP} = \min_{R\in\mathcal{R},Y\in\mathcal{Y}}\{\langle L,Y\rangle \colon Y = \widehat{V}R\widehat{V}^T\}. \tag{5.2.13}$$

The Lagrangian $\mathcal{L}(R,Y,Z)\colon \mathcal{R}\times\mathcal{Y}\times\mathbb{S}^{n^2+1}\to\mathbb{R}$: for (5.2.13) is defined by:

$$\mathcal{L}(R,Y,Z) = \langle L,Y\rangle + \langle Z, Y - \widehat{V}R\widehat{V}^T\rangle. \tag{5.2.14}$$

The Lagrangian dual function is defined by:

$$g(Z) = \min_{R\in\mathcal{R},Y\in\mathcal{Y}} \mathcal{L}(R,Y,Z),$$

Thus we obtain the Lagrangian dual problem of (5.2.13):

$$d^*_{QAP} = \sup_Z g(Z). \tag{5.2.15}$$

5.2.6 Strong duality

Let $\widehat{V}$ be as defined in the previous section, let $\mathcal{F}_{\widehat{V}} = \widehat{V}\mathbb{S}^{(n-1)^2+1}_+\widehat{V}^T$, and define the barycenter:

$$\hat{Y} := \frac{1}{n!}\sum_{X\in\Pi_n}\begin{pmatrix}1\\ \mathrm{vec}(X)\end{pmatrix}\begin{pmatrix}1\\ \mathrm{vec}(X)\end{pmatrix}^T. \tag{5.2.16}$$

By construction every rank-one term in the sum is feasible for (5.2.5), thus it is also feasible for the relaxation. Since the cardinality of Π_n is $n!$, $\hat{Y}$ is a convex combination of feasible points for (5.2.5). As the feasible set in convex, we conclude that $\hat{Y}$ is feasible for (5.2.8). This implies $\hat{Y}$ satisfies $K\hat{Y} = 0$, and consequently, $\hat{Y}\in\mathcal{F}_{\widehat{V}}$. Thus, if it holds that

$$\mathrm{rank}(\hat{Y}) \le (n-1)^2 + 1 = \mathrm{rank}(\widehat{V}).$$

Furthermore, it is proven in [34] that $\mathrm{rank}(\hat{Y}) = (n-1)^2+1$. Thus by Proposition 2.3.5 $\hat{Y}\in\mathrm{ri}(\mathcal{F}_V)$. By Proposition 4.1.5, this implies that a saddle point of (5.2.14) exists, and furthermore (by lemma 4.1.6) that strong duality holds, i.e.,

$$d^*_{QAP} = \max_Z \min_{R\in\mathcal{R},Y\in\mathcal{Y}}\langle L,Y\rangle + \langle Z, Y - \widehat{V}R\widehat{V}^T\rangle \quad \text{and} \quad p^*_{QAP} = d^*_{QAP}. \tag{5.2.17}$$

5.3 ADMM for the QAP

In this section we define the **ADMM** iterates for the model (5.2.13) and briefly discuss how to use the output of **ADMM** to obtain upper and lower bounds on the optimal solution to

(5.2.1). Applying **ADMM** to the **DNN** relaxation of the **QAP** was first done in [24], and more recently in [16] authors applied a variant of **ADMM**, called Peaceman-Rachford to solve the same model as in [24] but with the additional trace constraint. For simplicity, we present the **ADMM** iterates and not Peaceman-Rachford, however, the strategy used in [16] for finding upper and lower bounds still applies. As in Chapter 4, let $\beta > 0$ be a parameter and define the augmented Lagrangian by:

$$\mathcal{L}_A(R, Y, Z) := \langle L, Y \rangle + \langle Z, Y - \widehat{V} R \widehat{V}^T \rangle + \frac{\beta}{2} \left\| Y - \widehat{V} R \widehat{V}^T \right\|_F^2. \tag{5.3.1}$$

Then we have the following updates for the primal and dual iterates respectively:

$$\begin{aligned} R^{k+1} &= \operatorname{argmin}_{R \in \mathcal{R}} \mathcal{L}_A(R, Y^k, Z^k), \\ Y^{k+1} &= \operatorname{argmin}_{Y \in \mathcal{Y}} \mathcal{L}_A(R^{k+1}, Y, Z^k), \\ Z^{k+1} &= Z^k + \beta(Y^{k+1} - \widehat{V} R^{k+1} \widehat{V}^T). \end{aligned}$$

By section 4.2.1 and section 4.2.2, these are solved explicitly by computing the projections:

$$\begin{aligned} R^{k+1} &= \mathcal{P}_{\mathcal{R}}(\widehat{V}^T(Y^k + \frac{1}{\beta} Z^k)\widehat{V}), \\ Y^{k+1} &= \mathcal{P}_{\mathcal{Y}}(\widehat{V} R^{k+1} \widehat{V}^T - \frac{1}{\beta}(L + Z^k)), \\ Z^{k+1} &= Z^k + \beta(Y^{k+1} - \widehat{V} R^{k+1} \widehat{V}^T). \end{aligned}$$

Upper and lower bounds on the optimal value of (5.2.1) are computed using the method described in and section 4.2.3. A variation in the upper bound computation from [16] uses the spectral decomposition of the output and perturbs the eigenvalues before solving the LP (4.2.3). Specifically, they take the spectral decompositon of Y^{out}:

$$Y^{out} = \sum_{i=1}^{r} \lambda_i v_i v_i^T,$$

and let $\xi \in \mathbb{R}^r$ be a random vector with entries in the interval $(0, 1)$ and then define

$$x^{out} = \sum_{i=1}^{r} \xi_i \lambda_i v_i.$$

Finally, we obtain an upper bound by solving the LP in (4.2.3) with $c = x^{out}$. This process is repeated a number of times, and the best result, i.e. the feasible solution to (5.2.4) corresponding to the lowest objective value is returned.

Chapter 6

Side Chain Positioning problem

6.1 SDP relaxation of the SCP problem

In this chapter we present a second application of the semidefinite programming for binary constrained quadratic programs: the *Side Chain Positioning,* ***SCP*** problem. The work of using **ADMM** for the **SDP** relaxation of the **SCP** problem was done in [7] and we follow a very similar approach, borrowing many of their techniques, while also noting that these techniques are not dissimilar from the approach used for the **QAP**. Implementation details and numerical results can be found in [7]. The work presented in [7] is not a direct contribution of this thesis but rather is included to show that the application of the techniques in [16] developed for QAP generalizes to other NP-hard problems.

We now describe the **SCP** problem and how it arises in Molecular Biology. We then show that it is an instance of a **BCQP**, and as a result we can directly apply the method defined in Chapter 3 to obtain the split model, and then use the methods outlined in Chapter 4 to compute **ADMM** iterates and obtain bounds on the optimal solution.

6.1.1 Protein folding biology

The **SCP** problem arises in molecular biology, in particular, in the task of protein structure prediction. Protein structure prediction is the process of determining the three-dimensional shape of a protein based on its known two-dimensional sequence of amino acids. In what follows, we briefly describe the necessary background for understanding protein-structure-prediction and define the **SCP** problem. We subsequently focus on defining the **SCP** on a graph and modeling it as a **BCQP** . For further information on the biology of protein folding see [10, Sect.1.1] and the references therein.

A protein molecule is made up of a sequence of amino acids connected by peptide bonds. Each amino acid in the chain is formed of a *central carbon atom* which is attached to an *amino group*, a *carboxyl group*, and a *side chain*. The amino acids are connected in such a way that the carboxyl group of one amino acid forms a peptide bond with the amino

group of the next amino-acid. This alternating sequence forms what is know as the *protein backbone*. We consider the setting wherein the backbone is fixed, but the side chains of each amino acid may take on various positions in space. These different possible positions of a side chain are called *rotamers*. By determining the position of every side chain we obtain the shape of protein molecule, know as the *protein conformation*.

In practice, protein side chains are observed to occupy a only a discrete number of positions in space. Thus, the side-chain positioning problem (**SCP**) amounts to the selection of a *single* rotamer, from a discrete set of possible rotamers, for each amino side chain in the protein molecule. In order to determine the true protein conformation, we rely on the assumption that the side chains will be positioned in a way such that the protein molecule's total *energy* is minimized. The total energy of a protein molecule is defined to be the sum of the pairwise enegry between atoms. The positions of side chains that achieves this minimum is called *global minimum energy conformation* (GMEC).

The energy between two atoms can be calculated based off some energy function which we do not discuss here. Instead we note that the energy will depend on, among other things, the distance between the two atoms. Thus it is reasonable that different rotatmers will result in different pairwise energy values between two side chain atoms. For our purposes we do not look at the energy function but rather assume that we are given a fixed energy value for each pair or rotamers corresponding to different side chains.

In addition to the energy between two rotamers, there exists an energy between the backbone and selected rotamer, sometimes referred to as the *self-energy* of the rotamer. Hence the energy of a proposed protein conformation obtained summing up over all pairs of side chain atoms, the energy between their respective rotamer pairs, in addition to summing up the self-energies of the selected rotamers.

6.1.2 Graphical representation

As in [1, 6, 10] we express the **SCP** problem as an optimization problem on a graph. We remark that this corresponds exactly to the formulation of the **BCQP** as **QEHSP**.

For a given protein molecule, we can extract the following information:

- p, the number of side chain atoms;
- S_i, the set of rotamers for the ith side chain, for $i \in \{1, \ldots, p\}$;
- m_i, the number of rotamers for the ith side chain, $m_i = |S_i| \ \forall i \in \{1, \ldots, p\}$;
- n_0, the sum of the rotamer set sizes, $n_0 = \sum_{i=1}^{p} m_i$;
- For every pair of rotamers (u, v), $u \in S_i$, $v \in S_j$ with $i, j \in \{1, \ldots n_0\}$, we have the energy value $w_{uv} = w_{vu}$;
- For every rotamer u, we have the self energy given by w_{uu}.

Then we construct a weighted undirected graph $G = (\mathcal{V}, \mathcal{E})$ in the following way. We label the rotamers using the numbers in $\{1, \ldots, n_0\}$, and let the vertices of the graph, $\mathcal{V}$, correspond to the set of all rotamers. In particular we let $\mathcal{V} = \mathcal{V}_1 \cap \cdots \cap \mathcal{V}_p$ where

$$\begin{aligned} \mathcal{V}_1 &= \{1, \ldots, m_1\} \\ \mathcal{V}_2 &= \{m_1 + 1, \ldots, m_1 + m_2\} \\ &\vdots \\ \mathcal{V}_p &= \left\{\textstyle\sum_{i=1}^{p-1} m_i + 1, \ldots, \sum_{i=1}^{p} m_i\right\}. \end{aligned}$$

The edge set $\mathcal{E}$ is comprised of pairs of rotamers (u, v) belonging to different side chains, and the corresponding edge-weight is given by the energy, i.e., $\text{weight}(e_{uv}) = w_{uv}$. Thus we have just defined an undirected, weighted p-partite graph.

The total energy of a protein molecule conformation is given by the sum of the pairwise energies between rotamers. Thus we are looking to pick one vertex from every $\mathcal{V}_i$ in a way that minimizes the sum of the weights of the edges between selected vertices. Note that this is no different from the **QEHSP** presented in Chapter 3. As such, we apply the same process to obtain the corresponding **BCQP**.

6.1.3 The SCP problem as a BCQP

To begin, we define the following *energy matrix,* $E \in \mathbb{R}^{n_0 \times n_0}$:

$$\begin{aligned} E_{ij} &:= w_{ij} \quad \forall (i,j) \in \mathcal{E} \; i \neq j, \\ E_{ij} &:= 0 \quad \forall (i,j) \notin \mathcal{E} \; i \neq j, \\ E_{ii} &:= w_{ii} \quad \forall i \in V. \end{aligned} \tag{6.1.1}$$

For any $U \subseteq V$, we define the indicator vector $x \in \{0,1\}^{n_0}$ by:

$$x_u = \begin{cases} 1, & \text{if } u \in U, \\ 0, & \text{if } u \notin U. \end{cases}$$

The **SCP** problem is then be formulated as the following integer problem:

$$\begin{aligned} \min_{x} \quad & \sum_{(i,j) \in \mathcal{E}} E_{ij} x_i x_j \\ \text{s.t.} \quad & \sum_{i \in \mathcal{V}_k} x_i = 1 \quad \forall k = 1, \ldots, p \\ & x \in \{0,1\}^{n_0}. \end{aligned} \tag{6.1.2}$$

To obtain the corresponding **BCQP** , we define the binary constraints matrix:

$$H = \begin{bmatrix} \bar{e}_{m_1}^T & 0 & \dots & 0 \\ 0 & \bar{e}_{m_2}^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{e}_{m_p}^T \end{bmatrix} \in \mathbb{R}^{p \times n_0}.$$

Then we obtain the equivalent **BCQP** :

$$\begin{array}{ll} \min\limits_{x} & x^T E x \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in \{0,1\}^{n_0}. \end{array} \tag{6.1.3}$$

6.1.4 Lifted reformulation of SCP problem

In this section we obtain the lifted reformulation of (6.1.3). Let

$$\hat{E} := \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \quad K := \begin{bmatrix} p & -\bar{e}_p^T A \\ -A^T \bar{e}_p & A^T A \end{bmatrix}.$$

Applying the process detailed in Section 3.3.1, we get that (6.1.3) is equivalent to (6.1.4).

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n_0+1}} & \langle \hat{E}, Y \rangle \\ \text{s.t.} & KY = 0 \\ & \text{arrow}(Y) = e_0 \\ & \text{rank}(Y) = 1. \end{array} \tag{6.1.4}$$

Then, as in Section 3.3.2 our initial **DNN** relaxation for (6.1.4) is obtained by relaxing the rank one constraint and adding the cutting planes $0 \leq Y \leq 1$. We obtain:

$$\begin{array}{ll} \min\limits_{Y \in \mathbb{S}^{n_0+1}} & \langle \hat{E}, Y \rangle \\ \text{s.t.} & \text{arrow}(Y) = e_0 \\ & KY = 0 \\ & Y \succeq 0 \\ & 0 \leq Y \leq 1. \end{array} \tag{6.1.5}$$

6.1.5 Gangster and trace constraint for the SCP problem

To obtain the final model we just need to find the trace of all feasible Y and apply Theorem 3.3.4 to obtain the gangster index. Recall that the gangster index set J is defined by $J = (0,0) \cup \bar{J}$ where $\bar{J}$ is the set of indices for the nonzero entries of $(A^T A - I)$. Note the

structure of A^TA:

$$A^TA = \begin{bmatrix} \bar{E}_{m_1} & 0 & \cdots & 0 \\ 0 & \bar{E}_{m_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{E}_{m_p}. \end{bmatrix}.$$

Since $\text{Diag}(A^TA) = I$, the gangster indices correspond to the indices of the nonzero entires of the matrix $A^TA - I$.

$$A^TA - I = \begin{bmatrix} \bar{E}_{m_1} - I_{m_1} & 0 & \cdots & 0 \\ 0 & \bar{E}_{m_2} - I_{m_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{E}_{m_p} - I_{m_p}. \end{bmatrix}.$$

We see now that $\bar{J}$ is equal to the index of the following element the off-diagonal elements of the p diagonal m_i-blocks. Next we can easily obtain that trace $Y = p+1$. This is because by the problem definition in Section 6.1.3, all $\mathcal{V}_1 \ldots \mathcal{V}_p$ form a partition of V. Thus (3.4.3) is satisfied and applying Proposition 3.4.5 gives $\text{trace}(Y) = p + 1$.

6.2 Split model for the SCP problem

To obtain the split model, we note the $\text{rank}(K) = n_0 + 1 - p$ [6]. Thus we define a matrix $\hat{V}$ whose columns form an othonormal basis for $\text{null}(K)$. In [6], Burkowski et al. construct the following matrices. For $k \geq 2$,

$$B_k := \begin{bmatrix} I_{k-1} \\ -\bar{e}_{k-1}^T \end{bmatrix}.$$

If $k = 1$, $B_k = 0 \in \mathbb{R}$. They also define the matrix:

$$W = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ e_{m_1} & B_{m_1} & 0 & \cdots & 0 \\ e_{m_1} & 0 & B_{m_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{m_p} & 0 & 0 & \cdots & B_{m_p} \end{bmatrix} \in \mathbb{R}^{(n_0+1)\times(n_0+1-p)}$$

where the columns of W are linearly independent and are in the null space of K [6].

In implementation (see [7]), the QR decomposition of W in Matlab can be used to obtain an orthogonal matrix $\widehat{V}$ whose columns form a basis for $\text{null}(K)$. Thus we obtain

the relaxation of (6.1.3) that is used in [7]:

$$\begin{aligned} \min_{R,Y} \quad & \langle \hat{E}, Y \rangle \\ & Y = \widehat{V} R \widehat{V}^T \\ & R \in \mathcal{R} \\ & Y \in \mathcal{Y}, \end{aligned} \tag{6.2.1}$$

where $\mathcal{R}$ and $\mathcal{Y}$ are defined by

$$\begin{aligned} \mathcal{R} &:= \left\{ R \in \mathbb{S}_+^{n_0-p} \colon \ \mathrm{trace}(R) = p+1 \right\}, \\ \mathcal{Y} &:= \left\{ Y \in \mathbb{S}^{n_0+1} \colon 0 \leq Y \leq 1, G_J(Y) = E_{00} \right\}. \end{aligned} \tag{6.2.2}$$

6.3 Strong duality for the split model

To see that $\widehat{V}$ corresponds to the minimal face, it suffices to find $\widehat{Y}$ feasible for (6.2.1) such that $\mathrm{rank}(\widehat{Y}) = n_0+1-p$. Surprisingly, we can easily construct such a $\hat{Y}$ by using the matrix W above. We Let w^i denote the ith column of W. We observe that $w^i + w^1 \in \{0,1\}^{n_0+1}$ for all $i = 2, \dots n_0+1$. Thus, we define a new set of linearly independent vectors by

$$\begin{aligned} v^1 &= w^1 \\ v^i &= w^i \ \forall i \in \{2, \dots, n_0+1\}. \end{aligned}$$

Then the v^i are linearly independent and we can write

$$v_i = \begin{pmatrix} 1 \\ \tilde{v}^i \end{pmatrix},$$

where the $\tilde{v}^i$ satisfy for all $i \in \{1, \dots n_0+1-p\}$:

- $\tilde{v}^i \in \{0,1\}^{n_0}$;
- $H\tilde{v}^i = \bar{e}_p$.

Thus each $\tilde{v}^i$ is feasible for (6.1.3). Thus $\begin{pmatrix} 1 \\ \tilde{v}^i \end{pmatrix} \begin{pmatrix} 1 \\ \tilde{v}^i \end{pmatrix}^T$ is feasible for (6.1.4). Thus,

$$\hat{Y} = \frac{1}{n_0+1-p} \sum_{i=1}^{n_0+1-p} \begin{pmatrix} 1 \\ \tilde{v}^i \end{pmatrix} \begin{pmatrix} 1 \\ \tilde{v}^i \end{pmatrix}^T$$

is feasible for (6.2.1) and $\mathrm{rank}(\hat{Y}) = n_0+1-p$. Thus by Proposition 4.1.5, we have that the Lagrangian has a saddle point and by lemma 4.1.6, strong duality holds. Thus we can apply **ADMM** to the model (6.2.1). We do this in the next section.

6.4 ADMM for the SCP problem

6.4.1 Defining the iterates

To obtain the **ADMM** iterates, it suffices to use the iterates from Chapter 4, and tailor them to the split model (6.2.1). Similarly to the work done for the **QAP** in [16], we can alternatively apply the variant of **ADMM** called Peaceman-Rachford. This work is done in [7]. The calculations of the iterates, as well as the upper and lower bounds remains the same, and we present them in the following sections.

We now define the Augmented Lagrangian for the problem (6.2.1):

$$\mathcal{L}_A(R, Y, Z) := \langle \widehat{E}, Y \rangle + \langle Z, Y - \widehat{V} R \widehat{V}^T \rangle + \frac{\beta}{2} \left\| Y - \widehat{V} R \widehat{V}^T \right\|_F^2. \tag{6.4.1}$$

The iterates are:

$$\begin{aligned} R^{k+1} &= \operatorname{argmin}_{R \in \mathcal{R}} \mathcal{L}_A \left(R, Y^k, Z^k \right) \\ &= \mathcal{P}_{\mathcal{R}} \left(\widehat{V}^T \left(Y^k + \frac{1}{\beta} Z^k \right) \widehat{V} \right), \\ Y^{k+1} &= \operatorname{argmin}_{Y \in \mathcal{Y}} \mathcal{L}_A \left(R^{k+1}, Y, Z^k \right) \\ &= \mathcal{P}_{\mathcal{Y}} \left(\widehat{V} R^{k+1} \widehat{V}^T - \frac{1}{\beta} (\widehat{E} + Z^k) \right), \\ Z^{k+1} &= Z^k + \beta \left(Y^{k+1} - \widehat{V} R^{k+1} \widehat{V}^T \right). \end{aligned}$$

6.4.2 Upper bound computation

We apply the approach outlined in Section 4.2.3 for obtaining an upper bound. This general approach was first implemented for the **SCP** problem in [6] and then again in [7]. Let Y^{out} be the output Y-iterate of the **ADMM** algorithm. To obtain an upper bound on the optimal solution of (6.1.3) it suffices to find a feasible solution to (6.1.3). Note that if $\operatorname{rank}(Y^{out}) = 1$, we are done. Otherwise, we obtain a rank-one approximation to Y^{out}, which will have the form $Y^c := \begin{pmatrix} 1 \\ c \end{pmatrix} \begin{pmatrix} 1 \\ c \end{pmatrix}^T$. Approaches for obtaining c are given in [6] and [7]. To obtain a feasible solution for (6.1.3) we find the closest feasible x to c [6]. This means that we must solve :

$$\min_x \|x - c\| \text{ s.t. } Ax = \bar{e}_p, \ x \in \{0,1\}^{n_0}. \tag{6.4.2}$$

From section 4.2.3, recall that this reduces to solving the LP:

$$\begin{array}{ll} \min & -c^T x \\ \text{s.t.} & Hx = \bar{e}_p \\ & x \in [0,1]^n. \end{array} \tag{6.4.3}$$

Although we could run the simplex algorithm on this, as noted in [6], a solution to (6.4.3) just returns a greedy solution in the sense for the every rotamer set $\mathcal{V}_k$, the subvector x^k will have a one in the index i_k such that c_{i_k} has the greatest value.

6.4.3 Lower bound computation

We now present the lower bound computation. This is a direct application of section 4.2.4. We let h be the dual functional defined in Proposition 4.2.2. Thus by Proposition 4.2.2 we get a lower bound on on the optimal value of (6.1.3) by simply computing:

$$h(Z^{out}) = \langle \hat{E}, Y^* \rangle + \langle Z^{out}, Y^* \rangle - (t+1)\lambda_{\max}(V^T Z^{out} V) \leq d^*$$

where

$$(Y^*)_{ij} = \begin{cases} 0 & \text{if } (i,j) \in J \\ 1 & \text{if } (i,j) \notin J \text{ and } (\widehat{W} + Z^{out})_{ij} < 0 \\ 0 & \text{if } (i,j) \notin J \text{ and } (\widehat{W} + Z^{out})_{ij} \geq 0. \end{cases}$$

We have demonstrated that the computation of the **ADMM** iterates, as well as the bounding techniques for the **SCP** problem are nearly identical to those of the **QAP**, and therefore, as done in [7], we can directly apply the approach from [16]. Generalizing the two problems to a **BCQP** reveals why this is the case and allows us to re-use this technique for future problems.

Chapter 7

Conclusions and further notes

In this thesis we aimed to generalize the method of obtaining tight bounds on the **BCQP** based off of the success of the methods in [6,7,24,34]. In the process of developing this theory, we came across steps which required some further assumptions that were not present in the original **BCQP** formulation. For example, we found that property (3.4.3) holds for both the **SCP** problem and the **QAP**. We used this property to obtain the trace constraint, and also in our computation of the upper bound in section 4.2.3. Arguably, not having this property is not an immediate deal breaker as we could simply drop the trace constraint and find a new method of computing the upper bound. We found that the constraint $KY = 0$ restricts the feasible set to a face of the positive semidefinite cone. We showed that if the face determined by $KY = 0$ was in fact the *minimal face* of the positive semidefinite cone containing the feasible set, then strong duality holds, and the afrementionned convergence results for **ADMM** hold.

However, we did not find that this would hold for the general **BCQP** formulation of K. Instead we proved this individually for both problems by finding a feasible Y with $\text{rank}(Y) = n + 1 - \text{rank}(K)$. The fact that it holds for the **QAP** and **SCP** problem is encouraging, but these are also very structured problems, and there are no guarantees (yet) that this will exist for the general **BCQP** . We noted that for both the **QAP** and the **SCP** problem, the constraints matrix H is totally unimodular. A first question is whether the total unimodularity plays a role in having the affine hull of the feasible set $\mathcal{F}_{BCQP}$ be equal to $\{x \in \mathbb{R}^n \colon Hx = \bar{e}_p\}$. Perhaps this is not a sufficient enough assumption but we have not yet determined a counter example.

Without better understanding of this question, we will have to continue treating instances of the **BCQP** based on their specific structure. However, by following the work done in this thesis, we no longer have to start from scratch. We now have a blueprint for how to obtain the **SDP** model and its iterates, and we know what assumptions we need to check in order to apply this approach.

References

[1] T. Akutsu. NP-hardness results for protein side-chain packing. *Genome Informatics*, 8:180–186, 1997. 17, 55

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5:13–51, 1995. 2

[3] S. Axler. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer New York, 1997. 7

[4] G. Birkhoff. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A.*, 5:147–151, 1946. 44, 48

[5] 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. 3, 33, 41

[6] 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. 22, 30, 39, 40, 55, 58, 60, 61, 62

[7] F. Burkowski, J. Im, and H. Wolkowicz. A peaceman-rachford splitting method for the protein side-chain positioning problem, 2020. iii, 1, 16, 18, 21, 24, 28, 30, 31, 39, 54, 58, 59, 60, 61, 62

[8] E. Çela. *The quadratic assignment problem*, volume 1 of *Combinatorial Optimization*. Kluwer Academic Publishers, Dordrecht, 1998. Theory and algorithms. 45

[9] W. Chaovalitwongse, I. Androulakis., and P. Pardalos. Quadratic integer programming: Complexity and equivalent forms. 01 2008. 2

[10] 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. 17, 54, 55

[11] E. de Klerk, D.V. Pasechnik, and R. Sotirov. On semidefinite programming relaxations of the traveling salesman problem. *SIAM Journal on Optimization*, 19(4):1559–1573, Jan 2009. 17

[12] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. 9, 10, 38

[13] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrica*, 1(3):211–218, 1936. 39

[14] D. Rawitz G. Even and S. Shahar. Hitting sets when the vc-dimension is small. *Information Processing Letters*, 95(2):358 – 362, 2005. 14

[15] M.X. Goemans and D.P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. on Discrete Mathematics*, 7, 1994. 1

[16] N. Graham, H. Hu, H. Im, X. Li, and H. Wolkowicz. A restricted dual Peaceman-Rachford splitting method for QAP. Technical report, University of Waterloo, Waterloo, Ontario, 2020. 29 pages, research report. iii, 1, 16, 18, 21, 24, 28, 30, 31, 39, 40, 41, 47, 53, 54, 60, 61

[17] V. Guruswami. and Y. Zhou. Tight bounds on the approximability of almost-satisfiable horn sat and exact hitting set. *Theory of Computing*, 8(11):239–267, 2012. 14

[18] 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)]. 6, 7

[19] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, New York, 1985. 8

[20] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original. 7

[21] T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957. 44

[22] 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. Technical report, University of Waterloo, Waterloo, Ontario, 2019. 40 pages, research report. 41

[23] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991. 2

[24] D.E. Oliveira, H. Wolkowicz, and Y. Xu. ADMM for the SDP relaxation of the QAP. *Math. Program. Comput.*, 10(4):631–658, 2018. 2, 3, 21, 39, 40, 47, 53, 62

[25] G. Pataki. Geometry of Semidefinite Programming. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook OF Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000. 9

[26] B.N. Pshenichnyi. *Necessary conditions for an extremum*, volume 4 of *Translated from the Russian by Karol Makowski. Translation edited by Lucien W. Neustadt. Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1971. 11

[27] R.T. Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks. 6, 7, 12

[28] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978. 17

[29] T. Schmidt. *Computational Complexity of SAT, XSAT and NAE-SAT for linear and mixed Horn CNF formulas*. PhD thesis, Universität zu Köln, 2010. 17

[30] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997. 30

[31] L. Tunçel. *Polyhedral and semidefinite programming methods in combinatorial optimization*, volume 27 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI; Fields Institute for Research in Mathematical Sciences, Toronto, ON, 2010. 8, 9

[32] J. von Neumann. Discussion of a maximum problem. In A.H. Taub, editor, *Jon von Neumann, Collected Works*, pages 89–95. Pergamon, New York, 1963. 44, 48

[33] D.P. Williamson and D.B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, USA, 1st edition, 2011. 17

[34] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. volume 2, pages 71–109. 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Toronto, ON, 1996). 22, 30, 51, 52, 62

Index