LINEAR AND NONLINEAR SEMIDEFINITE RELAXATIONS OF SOME NP-HARD PROBLEMS

BY

TAO ZHU

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Assistant Professor Jiming Peng, Chair
Associate Professor Angelia Nedich
Associate Professor Carolyn L. Beck
Assistant Professor Alexandra Kolla
ABSTRACT

Semidefinite relaxation (SDR) is a powerful tool to estimate bounds and obtain approximate solutions for NP-hard problems. This thesis introduces and studies several novel linear and nonlinear semidefinite relaxation models for some NP-hard problems. We first study the semidefinite relaxation of Quadratic Assignment Problem (QAP) based on matrix splitting. We characterize an optimal subset of all valid matrix splittings and propose a method to find them by solving a tractable auxiliary problem. A new matrix splitting scheme called sum-matrix splitting is also proposed and its numerical performance is evaluated.

We next consider the so-called Worst-case Linear Optimization (WCLO) problem which has applications in systemic risk estimation and stochastic optimization. We show that WCLO is NP-hard and a coarse linear SDR is presented. An iterative procedure is introduced to sequentially refine the coarse SDR model and it is shown that the sequence of refined models converge to a nonlinear semidefinite relaxation (NLSDR) model. We then propose a bisection algorithm to solve the NLSDR in polynomial time. Our preliminary numerical results show that the NLSDR can provide very tight bounds, even the exact global solution, for WCLO.

Motivated by the NLSDR model, we introduce a new class of relaxation called conditionally quasi-convex relaxation (CQCR). The new CQCR model is obtained by augmenting the objective with a special kind of penalty function. The general CQCR model has an undetermined nonnegative parameter $\alpha$ and the CQCR model with $\alpha = 0$ (denoted by CQCR(0)) is the strongest of all CQCR models. We next propose an iterative procedure to approximately solve CQCR(0) and a bisection procedure to solve CQCR(0) under some assumption. Preliminary numerical experiments illustrate the proposed algorithms are effective and the CQCR(0) model outperforms classic relaxation models.
To my parents and my wife, for their love and support.
ACKNOWLEDGMENTS

This thesis would not be possible without the generous support of many people.

Firstly and foremost, I would like to express my greatest gratitude to my academic advisor Professor Jiming Peng. I feel extremely fortunate to have Professor Peng as my advisor. His encouragement, vision and expertise play the roles of fuel, lighthouse and technique in my exploratory journey of research. I also want to thank him for providing me the opportunities to outreach great researchers in academia, national labs and industry through conferences and internship programs which exposed me to many attractive research areas and allowed me to develop my own research interests. Apart from coursework and research, I also received many sincere advise from him such as to pay more attention to my English and to be a better communicator, to be more precise and formal with my writing and to have a good work-life balance.

I would like to sincerely thank my thesis committee members, Professor Jiming Peng, Professor Angelia Nedich, Professor Carolyn L. Beck and Professor Alexandra Kolla for their time, efforts, and valuable suggestions. I also want to thank Professor Hezhi Luo for his invaluable inputs and stimulating discussions for several results in this thesis.

I want to convey my appreciations to the great and dedicated professors at University of Illinios including Professor Uday Shanbhag, Professor Angelia Nedich, Professor Xin Chen, Professor Liming Feng, Professor Sheldon Jacobson, Professor Negar Kiyavash, Professor Feng Liang, Professor Jiawei Han, Professor R. Srikant, Professor Junho Song, Professor Jeff Erickson, Professor Chandra Chekuri and many others. I thank them for their inspiring lectures on several areas such as optimization, convex analysis, financial mathematics, probability theory, learning theory and
algorithms.

I am also indebted to the great staff members in the ISE department including Holly Michelle Kizer, Amy Summers, Debra Hilligoss, Barbara Bohlen and many others. I sincerely thank these people whose dedicated work have made my life a lot easier.

During my four years’ study at University of Illinois, I am fortunate to meet friends such as Fan Ye, Jingnan Chen, Zisheng Chen, Rui Yang, Fuyuan Wang, Tinghao Guo, Zhenyu Hu, Xiangyu Gao, Hao Jiang, Limeng Pan, Ning Quan, Ehsan Salimi, Dane Schiro, Alok Tiwari, Yunwen Xu, Seyed Farzad Yousefian, Helin Zhu and many others. I also wan to thank my roommate Xiangli Wang and Seyed Rasoul Etesami and my volleyball friends without whom my life at UIUC will be much more boring and unhealthy.

Finally, I would like to thank my father, Jianmin Zhu, my mother, Shanhong Luo and my wife, Xiaolan Zheng for their selfless love and support.
# TABLE OF CONTENTS

LIST OF TABLES ........................................ v

LIST OF FIGURES ....................................... ix

LIST OF NOTATIONS ...................................... x

CHAPTER 1  INTRODUCTION ................................ 1

1.1  Semidefinite Programming ......................... 1

1.2  Applications of SDP ................................. 2

1.3  Literature Review .................................... 5

1.4  Structure of The Thesis ............................. 7

CHAPTER 2  SEMIDEFINITE RELAXATION OF QAP USING NON-
REduNDANT MATRIX SPLITTING .......................... 9

2.1  Introduction ......................................... 9

2.2  Redundant and Non-redundant Matrix Splitting ... 12

2.3  Two Matrix Splitting Schemes based on the Minimal Trace Principle 20

2.4  SDRs of QAPs based on minimal trace matrix splitting .......... 30

2.5  Numerical Experiments ............................. 33

2.6  Conclusions ........................................ 36

CHAPTER 3  NONLINEAR SEMIDEFINITE RELAXATION OF WCLO  . 41

3.1  Introduction ......................................... 41

3.2  The Coarse SDO Relaxation of WCLO ................ 43

3.3  A Nonlinear SDO Relaxation for WCLO ............... 46

3.4  Applications ........................................ 55

3.5  Numerical results .................................... 58

3.6  Conclusions ........................................ 61
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Optimal solution ((\alpha, \beta)) of problem ((2.25))</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Improved bounds using non-redundant matrix splitting</td>
<td>35</td>
</tr>
<tr>
<td>2.3</td>
<td>Selected bounds for QAPs of small sizes ((n \leq 30)), computed using CVX with the SDPT3 solver.</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Selected bounds for QAPs of median sizes ((30 &lt; n \leq 70)), computed using CVX with the SDPT3 solver. 'NA' denote the solver failed due to out of memory.</td>
<td>38</td>
</tr>
<tr>
<td>2.5</td>
<td>Selected bounds for QAPs of large sizes ((n &gt; 70)), computed using SDPNAL. The symbol ((†)) means that the best solution is only a feasible solution. The computation time is reported in the format of hours:minutes:seconds.</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>New best known bounds for QAPLIB instances. In the table, “NA” means that the item is not available. Note that information for the last column of the table is obtained from [27].</td>
<td>40</td>
</tr>
<tr>
<td>3.1</td>
<td>Number of iterations to solve NLSDR model</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>Number of iterations to solve NLSDR model</td>
<td>60</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>3.1</td>
<td>Distribution of gaps obtained by applying SDR, NLSDR and AFFINE to randomly generated WCLO problems</td>
<td>62</td>
</tr>
<tr>
<td>3.2</td>
<td>Distribution of gaps obtained by applying SDR, NLSDR and AFFINE to randomly generated problems of (3.20)</td>
<td>63</td>
</tr>
<tr>
<td>4.1</td>
<td>Distributions of gaps obtained by solving the SDR-QCKP and CQCR-QCKP of 200 randomly generated QCKP</td>
<td>80</td>
</tr>
<tr>
<td>4.2</td>
<td>Distributions of gaps obtained by solving the SDR-ILP and CQCR-ILP of 200 randomly generated ILP problems</td>
<td>83</td>
</tr>
<tr>
<td>4.3</td>
<td>Distributions of gaps obtained by solving the SDR2-BoxQP and CQCR-BoxQP of 200 randomly generated BoxQP problems</td>
<td>85</td>
</tr>
</tbody>
</table>
LIST OF NOTATIONS

\( \mathbb{R}^n \) : \( n \)-dimensional Euclidean vector space;
\( \mathbb{R}^{n \times n} \) : space of \( n \times n \) real matrices;
\( A^T \) : transpose of \( A \in \mathbb{R}^{m \times n} \);
\( A_{ij} \) : \( ij \)-th element of \( A \in \mathbb{R}^{m \times n} \);
\( \mathcal{S}^n \) : space of \( n \times n \) real symmetric matrices;
\( \mathcal{S}^n_+ \) : space of \( n \times n \) symmetric positive semi-definite (PSD) matrices;
\( X \succeq 0 \) : \( X \in \mathcal{S}^n_+ \);
\( X \succeq 0 \) : \( X_{ij} \geq 0 \) for all \( i, j \);
\( \text{Tr}(A) = \sum_i A_{ii}, A \in \mathbb{R}^{n \times n} \);
\( e \) : vector of ones;
\( E = ee^T \);
\( \text{diag}(X) \) : vector obtained by extracting the diagonal of \( X \in \mathbb{R}^{n \times n} \);
\( \text{Diag}(x) \) : diagonal matrix with components of \( x \in \mathbb{R}^n \);
\( \text{vec}(A) = [A_{11}, A_{21}, ..., A_{n1}, A_{12}, A_{22}, ..., A_{nn}]^T \) for \( A \in \mathbb{R}^{n \times n} \);
\( \text{mat}(a) = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_{n+1} & a_{n+2} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n-n+1} & a_{2n-n+2} & \cdots & a_{nn} \end{bmatrix} \) for \( a \in \mathbb{R}^{n^2 \times 1} \).
CHAPTER 1
INTRODUCTION

1.1 Semidefinite Programming

Semidefinite Programming (SDP) refers to the class of optimization problems that can be expressed in the following standard form,

\[(SDP) \quad \min \ \text{Tr}(CX) \]
\[\text{s.t.} \quad \text{Tr}(A_i X) = b_i, \ i = 1, \ldots, m \]
\[X \succeq 0.\]

where \(C \in S^n, A_i \in S^n, i = 1, \ldots, m, b \in \mathbb{R}^n\) are given parameters for SDP. SDP can be considered as a generalized version of linear programming in which the element-wise nonnegativity constraints are replaced by the positive semidefinite (PSD) constraint. The dual of SDP is another SDP

\[(Dual\text{-}SDP) \quad \max \ b^T y \]
\[\text{s.t.} \quad C - \sum_i y_i A_i \succeq 0.\]

The primal and dual SDPs have zero duality gap if slater constraint qualification (see e.g., [35]) is satisfied.

It is well-known that SDP can be solved in polynomial time and there are many successful algorithms. For example, Nesterov and Nemirovskii [79] showed that SDP can be solved by sequential minimization techniques where the objective is augment-
ed by a suitable barrier term. One suitable barrier function for PSD constraint \( X \succeq 0 \) is \( -\log \det(X) \). The barrier function serves as a penalty as it goes to infinity if \( X \) approaches the boundary of PSD cone. And the barrier function \( -\log \det(X) \) is smooth, convex and self-concordant so that it can solved efficiently by Newton’s method. Readers may refer to the SDP handbook [98] for a review of these algorithms.

In the light of the development of efficient algorithms, effective solvers are also made available to researchers; These includes SDPT3 [97], SeDuMi [93], SDPNAL [105] and PENSDP [64]. This makes SDP an ideal computational model.

### 1.2 Applications of SDP

SDP has rich applications in combinatorial optimization [51, 52, 85, 104], system and control thoery [20], structural design [12, 11] and statistics [46]. These problems either by itself is modeled an SDP problem or use SDP as a tractable relaxation. The problems that use SDP relaxation are usually NP-hard thus cannot be solved in polynomial time unless P=NP. Solving the SDP relaxations can provide bounds for Branch-and-Bound procedure which is a common method for solving NP-hard problems. This thesis focuses on using SDP as a tractable relaxation model. As a preparation, we review two application examples of SDP relaxations in the remaining of this section.

#### 1.2.1 SDP relaxation of Max-Cut problem

Given a graph \( G = (V, E) \) and a weight matrix \( W \) associated with each edge in \( E \). The Max-Cut problem is the problem of finding a partition of \( V \) into \( S \) and \( V \setminus S \) such that the edges across the partition, or in the cut \( \delta(S) = \{(i, j) \in E | \{i, j\} \cap S = 1\} \), have maximum total weight. Define indicator variables \( x_i = 1 \) if \( i \in S \) and \( x_i = -1 \) if
\( i \in V \setminus S \), the Max-Cut problem is equivalent to the following optimization problem

\[
\text{(Max-Cut)} \quad \max_{x \in \{-1,1\}^n} \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - x_i x_j) = \max_{x \in \{-1,1\}^n} x^T Q x
\]

where \( Q = \frac{1}{4}(\text{Diag}(W e) - W) \). The Max-Cut problem is NP-hard. A more practical way to solve the Max-Cut problem is to solve its tractable relaxation and expect to obtain some 'insights' on the original problem.

By introducing new variables \( X \in \mathbb{R}^{n \times n} \), the Max-Cut problem can be rewritten as

\[
\begin{align*}
\text{max} & \quad \text{Tr}(Q X) \\
\text{s.t.} & \quad \text{diag}(X) = e, \ X = xx^T.
\end{align*}
\]

The above problem is linear except the constraints \( X = xx^T \). Replacing the constraints \( X = xx^T \) with constraints \( X \succeq 0 \), we then obtain a tractable SDP relaxation of the Max-Cut problem

\[
\begin{align*}
\text{max} & \quad \text{Tr}(Q X) \\
\text{s.t.} & \quad \text{diag}(X) = e, \ X \succeq 0.
\end{align*}
\] (1.1)

The trick of relaxation by replacing the quadratic term \( xx^T \) with a linear term \( X \) is referred as linearization and it is commonly used throughout this thesis. Sometimes, we also call it lifting because the searching space of the original problem \( \mathbb{R}^n \) is lifted to a higher dimensional space of the relaxation problem \( \mathbb{R}^{n \times n} \).

The relaxation by linearization is intimately related to the Lagrangian relaxation. To see this, consider the Lagrangian of the Max-Cut problem

\[
\mathcal{L}(\lambda) = \max x^T Q x + \sum_i \lambda_i (1 - x_i^2) = \max x^T (Q - \text{Diag}(\lambda)) x + e^T \lambda.
\]

Since \( x \) is free, \( \mathcal{L}(\lambda) \) is bounded if and only if \( Q - \text{Diag}(\lambda) \preceq 0 \) and we thus obtain
the following Lagrangian Dual

\[
\begin{align*}
\min & \quad e^T \lambda \\
\text{s.t.} & \quad \text{Diag}(\lambda) \succeq Q.
\end{align*}
\] (1.2)

Problem (1.2) is precisely the dual problem of problem (1.1).

Using the solution from (1.1), Goemans and Williamson [51] devised a randomized algorithm that can produce a cut whose expected weight is at least \(0.87856 \cdot \text{OPT}_{SDP}\) where \(\text{OPT}_{SDP}\) denotes the optimal value of problem (1.1).

1.2.2 SDP relaxation of QCQP problem

Quadratically Constrained Quadratic Programming (QCQP), refers to the problem in the following form

\[
\begin{align*}
\text{(QCQP)} \quad \max & \quad x^T Q_0 x + c_0^T x \\
\text{s.t.} & \quad x^T Q_i x + c_i^T x \leq d_i, \quad i = 1, \ldots, m.
\end{align*}
\]

QCQP is in general NP-hard and it is a fundamental problem in global optimization literature [68, 106].

Similar to Section 1.2.1, we apply the linearization technique to obtain an SDP relaxation of QCQP:

\[
\begin{align*}
\max & \quad \text{Tr}(Q_0 X) + c_0^T x \\
\text{s.t.} & \quad \text{Tr}(Q_i X) + c_i^T x \leq d_i, \quad i = 1, \ldots, m; \\
& \quad X - xx^T \succeq 0.
\end{align*}
\] (1.3)

Here we relax the constraints \(X = xx^T\) to \(X \succeq xx^T\) instead of \(X \succeq 0\) because \(x\) is involved in the objective and constraints and it is better to relate \(X\) to \(x\). The
constraints $X - xx^T \succeq 0$ is equivalent to the following linear PSD constraint

$$
\begin{bmatrix}
X & x \\
x^T & 1
\end{bmatrix} \succeq 0
$$

due to the Schur’s complement lemma.

**Lemma 1.2.1.** *(Schur’s Complement Lemma, see e.g., [102])* Let $X$ be a symmetric matrix with submatrices $A, B, C$ as follows,

$$
X = \begin{bmatrix}
A & B \\
B^T & C
\end{bmatrix}.
$$

If $C \succ 0$ then

$$
A - BC^{-1}B^T \succeq 0 \iff X \succeq 0.
$$

### 1.3 Literature Review

SDP relaxation can provide tight bounds for NP-hard problems. Anstreicher [3] compared various relaxations for QCQP and showed the SDP relaxation with some additional RLT cuts is provably stronger than many other tractable convex relaxation models. For some special type of problems, the SDP relaxation is so strong that it can provide exact optimal solution. For example, the SDP relaxation of the matrix completion problem [29] can be shown to be exact under certain condition (see e.g., [30, 29]). For several special cases of QCQP, it can be shown their SDP relaxations are exact as well [103, 101, 63, 7]. Even in the case that SDP relaxation is not exact, one may be able to bound the gap. For example, Goemans and Williamson [51] showed that when combined with a suitable rounding procedure, one can obtain a constant approximation to the Max-Cut problem based on its SDP relaxation. Similar results are reported for some special cases of QCQP [78, 100]. For QCQP with only quadratic forms, Nemirovskii et al [76] obtained an approximation bound in terms of the number of constraints in the underlying problem.
Given a NP-hard problem, there might be different ways to construct an SDP relaxation. For example, Peng et al. [82] proposed a general framework to construct SDP relaxation using matrix splitting for Quadratic Assignment Problem (QAP) [87]. Although the SDP relaxation models introduced in [82] is very competitive in terms of the computational cost and the quality of the bounds, there might exists even stronger models because of great flexibility in the choice of various matrix splitting schemes.

Although the SDP relaxation provides tight bounds for NP-hard problems, we are still interested in strengthening it. One way to strengthen convex relaxations is to add tractable constraints either by exploiting the structure of the original problems [74, 82] or by some rather-general scheme such as RLT [91] and triangle inequalities [24, 99]. Usually, exploiting the data structure requires a rich domain knowledge with respect to the original problem and the number of the cuts may grow fast. Another approach is to use lifting. In [70], Lovasz and Shrijver proposed a hierarchical lift-and-project procedure to tighten the SDP or LP relaxation for binary programs and established its convergence. Lasserre showed that a sequence of SDP relaxations constructed by multiple lifting converges to global optimal as the times of lifting goes to infinity [66]. However, in both of these lifting methods, the number of variables and constraints grows very fast and the relaxation problem may be very expensive to solve.

Majority of the studies on SDP relaxations is focused on using the linear convex SDP model [13, 18, 71, 77, 85, 92] because of its tractability. On the other hand, Copositive Programming [23] as another convex optimization model has become very popular for combinatorial optimization in recent years [36, 23, 39]. However, Copositive Programming itself is NP-hard thus intractable.

In this thesis, we introduce and study several novel semidefinite relaxation models for several selected NP-hard problems. These relaxation models can be considered as the extensions of classic linear semidefinite relaxation model. We showed that the classic linear semidefinite relaxation models can be strengthened by (i) using the parameters from an optimal subset; (ii) constructing conditionally quasi-convex relaxation model by augmenting the objective with a special type of penalty function.
In particular, we first study the semidefinite relaxation of Quadratic Assignment Problem (QAP) based on matrix splitting. We characterize an optimal subset of all valid matrix splitting schemes and propose a method to find them by solving a tractable auxiliary problem. We next consider the so-called Worst-case Linear Optimization (WCLO) problem which has applications in systemic risk estimation and stochastic optimization. We show that WCLO is NP-hard. Linear and Nonlinear semidefinite relaxations of WCLO are introduced and their relationship is studied. The nonlinear relaxation model of WCLO is generalized and a new class of relaxation called conditionally quasi-convex relaxation (CQCR) is introduced. General CQCR has an undetermined nonnegative parameter \( \alpha \) and the CQCR with \( \alpha = 0 \) is the strongest relaxation of all CQCR models. We propose two algorithms to tackle the CQCR model with \( \alpha = 0 \).

1.4 Structure of The Thesis

Chapter 2 introduces the non-redundant matrix splitting method and its application in deriving strong semidefinite relaxation for QAP. In Section 2.2, we first introduce the notion of redundant and non-redundant matrix splitting and show that the SDR based on a non-redundant PSD splitting can provide a stronger lower bound than a redundant one. The minimal trace principle is proposed to find a non-redundant matrix splitting scheme. In Section 2.3, we introduce the new sum-matrix splitting scheme and apply the minimal trace principle to general sum-matrix splitting and its special case - one-matrix splitting. We give conditions under which these two splitting schemes are non-redundant and compare the lower bounds derived from their corresponding SDRs. Numerical results on some large QAP instances from QAPLIB [27] are presented in Section 2.5.

Chapter 3 introduces a nonlinear semidefinite relaxation model for worst-case linear optimization. In Section 3.2, we describe the coarse SDR for WCLO, and discuss the lower and upper bounds obtained from the coarse SDR. In Section 3.3, we first introduce an iterative procedure to enhance the coarse SDR and show that the result-
ing series of SDRs will converge to a nonlinear SDO. Then, we propose a bi-section search method for solving the nonlinear SDO. In Section 3.4, we give two application examples of the WCLO model: the worst case estimation of the systemic risk [41] and two-stage adaptive optimization [15]. Numerical results are reported in Section 3.5.

Chapter 4 introduces a new class of relaxation model called the conditionally quasi-convex relaxation model. We first give the definition of conditionally quasi-convexity in Section 4.2. Section 4.3 discusses methods to solve the conditionally quasi-convex relaxation model. In Section 4.4, we show the NLSDR model introduced in Chapter 3 is actually a special case of the conditionally quasi-convex model. Preliminary numerical results are presented in Section 4.5
CHAPTER 2

SEMIDEFINITE RELAXATION OF QAP USING NON-REDUNDANT MATRIX SPLITTING

2.1 Introduction

Given matrices $A, B$, we consider the quadratic assignment problem (denoted by QAP) of the following form

$$
\min_{X \in \Pi} \text{Tr}(AXBX^T) \tag{2.1}
$$

where $\text{Tr}(\cdot)$ denotes the trace of a matrix and $\Pi$ is the set of permutation matrices. We assume that $A$ and $B$ are $n \times n$ symmetric matrices throughout this chapter. QAP was first introduced by Koopmans and Beckmann [65] for facility location and has applications in many areas such as chip design [34, 56], image analysis and processing [75, 95], and communications [8]. For more applications of QAP, we refer the reader to the survey paper [69].

It is well-known that QAP is NP-hard. Searching for the global solution of QAP usually involves the branch and bound (B&B) method. A crucial issue in the B&B method is how to compute strong lower bounds efficiently. Various relaxations and bounds for QAPs have been proposed in the literature. Roughly speaking, these bounds can be categorized into two groups. The first group includes several bounds that are not very strong but can be computed efficiently such as the well-known Gilmore-Lawler bound (GLB) [50, 67], the bound based on projection [54] (denoted by PB) and the bound based on convex quadratic programming (denoted by QPB) [54]. The second group contains strong bounds that require expensive computation such as the bounds derived from lifted integer linear programming [1, 2, 55].
Among all the relaxation models, we are particularly interested in the semidefinite relaxation models which can provide relatively stronger lower bounds compared with other relaxations based on linear and quadratic programming. A popular way to derive the SDRs of QAP is to relax the rank-1 matrix $\text{vec}(X)\text{vec}(X)^T$ to a $n^2 \times n^2$ positive semidefinite matrix with nonnegative elements, where $\text{vec}(X)$ denotes the $n^2$-dimensional vector obtained from $X$ by stacking its columns sequentially into a long vector. Though much progress has been obtained in solving the SDR based on the gram matrix $\text{vec}(X)\text{vec}(X)^T$ [86, 25, 37], the large number of $O(n^4)$ variables and constraints in these relaxations still make them formidable for medium size QAP instances with the current computation facilities. Recently, Ding and Wolkowicz [38] introduced a new SDR of QAP based on matrix lifting. The resulting SDR model has only $O(n^2)$ variables and constraints and thus can be solved using open source SDP solvers for QAPs of size $n \leq 30$, though it still remains a computational challenge for $n \geq 30$.

In a recent work [82], a new framework to derive cheap and strong SDR for QAP based on various matrix splitting schemes was introduced. It is shown that some relaxation models in [74, 82] can provide competitive bounds comparing with other relaxation models in the literature. However, since there is a lot of freedom in choosing the specific matrix splitting, it is unclear which splitting can lead to a stronger relaxation.

In this chapter, we attempt to address the issue of selecting a matrix splitting whose resulting relaxation model can provide a strong bound. Our first major contribution is to introduce a new notion of the so-called redundant and non-redundant positive semidefinite (PSD) matrix splitting and use the new notion to compare the bounds obtained by SDRs using different matrix splittings. For this, we first show that for any given redundant matrix splitting, there exists a corresponding non-redundant matrix splitting whose SDR can provide a stronger bound. To find such a non-redundant matrix splitting, we propose to solve some auxiliary SDP problems following the minimal trace principle\(^1\). In particular, we show that a straightforward

\(^1\)The minimal trace principle is chosen because, as shown in our analysis in Section 2.2, the final
application of the minimal trace principle leads to the so-called orthogonal matrix splitting introduced in [74, 82]. We also illustrate that for a given matrix, there may exist multiple non-redundant matrix splitting but we cannot find any dominant relationship between non-redundant matrix splittings.

Secondly, to further help to select a non-redundant splitting scheme whose corresponding SDR can be solved relatively efficiently, we consider two specific matrix splitting schemes based on the so-called one-matrix and the sum-matrix. We investigate the theoretical properties of the optimal solutions to the auxiliary SDP problems under these two circumstances and characterize when the derived matrix splitting schemes are non-redundant. We also compare the two lower bounds from the one-matrix and sum-matrix splitting schemes and show that, under certain conditions, the lower bound derived from the sum-matrix splitting is stronger.

Thirdly, based on the rank information at the optimal solution to the auxiliary problem, we present a new implementation of the relaxation model which leads to substantial improvement over the implementation in [82]. Numerical experiments show that the bound based on the new non-redundant matrix splitting schemes and implementation is very competitive with existing bounds including the bounds based on other matrix splitting schemes, and they can be computed more efficiently.

The chapter is organized as follows. In Section 2.2, we first introduce the notion of redundant and non-redundant matrix splitting and show that the SDR based on a non-redundant PSD splitting can provide a stronger lower bound than a redundant one. The minimal trace principle is proposed to find a non-redundant matrix splitting scheme. In particular, we show that a direct application of the minimal trace principle leads to the so-called orthogonal PSD matrix splitting introduced in [82]. In Section 2.3, we apply the minimal trace principle to the one-matrix and sum-matrix splitting whose corresponding SDRs are relatively easy to solve. We give conditions under which these two splitting schemes are non-redundant and compare the lower bounds derived from their corresponding SDRs. In Section 2.4, we present the SDR solution matrix following the minimal principle has the minimal rank and the rank information on the splitting matrix can be further used to reduce the memory requirement and simplify the relaxation model as discussed in Section 2.4.
models of QAPs based on the three matrix splitting schemes. Numerical results on some large QAP instances from QAPLIB [27] are presented in Section 2.5.

2.2 Redundant and Non-redundant Matrix Splitting

As shown in [82], there exist various matrix splitting schemes for a given matrix $B$ and it is unclear which splitting can lead to the strongest relaxation. In this section, we first introduce a new notion of the so-called redundant and non-redundant matrix splitting and show that for any given redundant matrix splitting, there exists another non-redundant matrix splitting that can provide a stronger relaxation. To find such a non-redundant positive semidefinite (PSD) splitting, we refer to the minimal trace principle. The relationship between the non-redundant matrix splitting based on minimal trace principle and the orthogonal PSD splitting schemes introduced in [82] will be discussed as well.

We start with the following definition from [82].

**Definition 2.2.1.** Given matrix $B$, we call matrix pair $(B_1, B_2)$ a PSD matrix splitting of $B$ if it satisfies

$$B = B_1 - B_2, \quad B_1, B_2 \succeq 0.$$  

In particular, if the additional constraint $B_1B_2 = 0$ is satisfied, then we call $(B_1, B_2)$ an orthogonal PSD splitting of $B$.

As pointed out in [82], there exist many PSD matrix splitting schemes. If a PSD splitting $(B_1, B_2)$ of matrix $B$ is available, then we can obtain the following basic SDR for QAPs:

$$\mu(B_1, B_2) = \min_{Y \in Y(B_1, B_2)} \text{Tr}(AY),$$

(2.2)

\footnote{For simplicity of discussion, in all the theoretical analysis of this work, we consider only the basic model (2.2) which is slightly different from the full SDR model to be described in Section 4. However, since in the full model we only add some convex constraints on the elements of $Y$, one can easily extend the results for the basic model to the full model.}
where the feasible set \( \mathcal{Y}(B_1, B_2) \) is defined by

\[
\mathcal{Y}(B_1, B_2) = \left\{ Y \in \mathbb{S}^n \middle| \exists Y_1, Y_2 \in \mathbb{S}^n_+, X \in \mathbb{R}^{n \times n} \text{ satisfying } Y = Y_1 - Y_2, \right. \\
\left. Y_1 - XB_1X^T \succeq 0, \quad Y_2 - XB_2X^T \succeq 0, \right. \\
\left. \text{diag}(Y_1) = X \text{diag}(B_1), \quad Y_1e = XB_1e, \right. \\
\left. \text{diag}(Y_2) = X \text{diag}(B_2), \quad Y_2e = XB_2e, \right. \\
\left. Xe = X^Te = e, \quad X \succeq 0 \right\}. \tag{2.3}
\]

We now introduce the following definition.

**Definition 2.2.2.** A PSD matrix splitting \((B_1, B_2)\) is said to be redundant (or non-redundant) if there exists (or does not exist) a nonzero matrix \(R \succeq 0\) satisfying

\[
B_1 - R \succeq 0, \quad B_2 - R \succeq 0.
\]

From the above definition, we immediately have

**Proposition 2.2.3.** Given a matrix \(B\). A PSD matrix splitting \((B_1, B_2)\) is non-redundant if and only if 0 is the optimal solution of the following SDP

\[
\begin{align*}
\max & \quad \text{Tr}(R) \\
\text{s.t.} & \quad B_1 - R \succeq 0, B_2 - R \succeq 0, R \succeq 0. \tag{2.4}
\end{align*}
\]

We next recall a well-known result regarding the doubly stochastic matrices. A real \(n \times n\) matrix \(M = (M_{ij})\) is doubly stochastic if the entries of \(M\) are non-negative, and each row and column of \(M\) sums to 1 [80]. The following result is from [80, Theorem 2, Birkhoff’s theorem].

**Lemma 2.2.4.** The set of \(n \times n\) doubly stochastic matrices is a convex set whose extreme points are the permutation matrices.

Based on Lemma 2.2.4, we first establish a result regarding a redundant PSD splitting.
Theorem 2.2.5. If a PSD matrix splitting \((B_1, B_2)\) of the matrix \(B\) is redundant with matrix \(R\), then we have

\[
\mathcal{Y}(B_1 - R, B_2 - R) \subseteq \mathcal{Y}(B_1, B_2),
\]

(2.6)

where \(\mathcal{Y}(\cdot)\) is the set as defined in (2.3).

Proof. Since the PSD matrix splitting \((B_1, B_2)\) is redundant, there exists nontrivial \(R \succeq 0 \in S^n\) such that

\[
B_1 - R \succeq 0, \quad B_2 - R \succeq 0.
\]

Clearly, \((B_1 - R, B_2 - R)\) is also a PSD splitting of \(B\).

Now let \(Y \in \mathcal{Y}(B_1 - R, B_2 - R)\), i.e., there exist \((Y_1, Y_2, X) \in S^n_+ \times S^n_+ \times \mathbb{R}^{n \times n}\) such that

\[
Y = Y_1 - Y_2, \quad Y_1 - X(B_1 - R)X^T \succeq 0, \quad Y_2 - X(B_2 - R)X^T \succeq 0, \quad \text{diag} (Y_1) = X \text{diag} (B_1 - R), \quad Y_1 e = X(B_1 - R)e, \quad \text{diag} (Y_2) = X \text{diag} (B_2 - R), \quad Y_2 e = X(B_2 - R)e, \quad X e = X^T e = e, \quad X \succeq 0.
\]

(2.7), (2.8), (2.9), (2.10), (2.11)

Since \(X\) is a \(n \times n\) doubly stochastic matrix, by Lemma 2.2.4, it can be expressed as a convex combination of permutation matrices. Let \(|\Pi|\) be the cardinality of the set of permutation matrices. Therefore, there exists \(\lambda_i \geq 0, \forall i = 1, \cdots, |\Pi|\) such that

\[
X = \sum_{i=1}^{[\Pi]} \lambda_i \hat{X}_i, \quad \sum_{i=1}^{[\Pi]} \lambda_i = 1, \quad \hat{X}_i \in \Pi, \quad \forall i = 1, \cdots, |\Pi|.
\]

(2.12)

Define

\[
Y_R = \sum_{i=1}^{[\Pi]} \lambda_i \hat{X}_i R \hat{X}_i^T.
\]

(2.13)
Since \( \hat{X}_i \in \Pi \) for all \( i \) and \( R \succeq 0 \), from (2.12) we have \( Y_R \succeq 0 \). Further, it is easy to verify that
\[
\text{diag}(Y_R) = \sum_{i=1}^{||\Pi||} \lambda_i \hat{X}_i \text{diag}(R) = X \text{diag}(R), \quad Y_Re = \sum_{i=1}^{||\Pi||} \lambda_i \hat{X}_i Re = XRe.
\]
(2.14)

It remains to show
\[
Y_R - XRX^T \succeq 0.
\]
(2.15)

Since \( R \succeq 0 \), we have \( R = V V^T \) for some \( V \in \mathbb{R}^n \). Therefore, for any \( d \in \mathbb{R}^n \), from (2.13) we obtain
\[
d^T XRX^T d = \left( V^T X^T d \right)^T \left( V^T X^T d \right) = \left\| V^T X^T d \right\|_2^2 = \left\| \sum_{i=1}^{||\Pi||} \lambda_i V^T \hat{X}_i^T d \right\|_2^2
\]
\[
\leq \sum_{i=1}^{||\Pi||} \lambda_i \left\| V^T \hat{X}_i^T d \right\|_2^2 = \sum_{i=1}^{||\Pi||} \lambda_i d^T \left( \hat{X}_i R \hat{X}_i^T \right) d = d^T Y_R d,
\]
where the inequality follows from the fact that the function \( \| \cdot \|_2^2 \) is convex and because \( \sum_{i=1}^{||\Pi||} \lambda_i = 1 \) and \( \lambda_i \geq 0, i = 1, \ldots, ||\Pi|| \). The above relation means that (2.15) holds true.

Now let us define
\[
\bar{Y}_1 = Y_1 + Y_R, \quad \bar{Y}_2 = Y_2 + Y_R.
\]

Since \( Y = \bar{Y}_1 - \bar{Y}_2 \), and
\[
\bar{Y}_1 - X B_1 X^T = Y_1 - X(B_1 - R)X^T + Y_R - XRX^T \succeq 0,
\]
\[
\bar{Y}_2 - X B_2 X^T = Y_1 - X(B_2 - R)X^T + Y_R - XRX^T \succeq 0,
\]
from (2.7)-(2.11) and (2.14)-(2.15) one can easily verify \( Y \in \mathcal{Y}(B_1, B_2) \). This proves (2.6).

From Theorem 2.2.5, we see that given a QAP with matrices \((A, B)\) and a PSD
matrix splitting \((B_1, B_2)\) of \(B\), if the matrix splitting \((B_1, B_2)\) is redundant, then we have

\[
\min_{Y \in \mathcal{Y}(B_1 - R, B_2 - R)} \text{Tr}(AY) \geq \min_{Y \in \mathcal{Y}(B_1, B_2)} \text{Tr}(AY). \tag{2.16}
\]

Therefore, in order to derive a strong lower bound, a non-redundant PSD matrix splitting should be used.

We next discuss how to find a non-redundant PSD matrix splitting for a given matrix \(B\). Inspired by Proposition 2.3, we consider the following auxiliary problem induced by the minimal trace principle:

\[
\text{(MTMS-PSD)} \quad \min \quad \text{Tr}(B_1) \quad \text{s.t.} \quad B_1 - B_2 = B, \quad B_1 \succeq 0, \quad B_2 \succeq 0.
\]

It is easy to see that the above problem is strictly feasible.

For a given matrix \(B\), let \(Q\) be an orthogonal matrix whose columns are the eigenvectors of the matrix \(B\) associated with the eigenvalues \(\{\lambda_1, \cdots, \lambda_n\}\), i.e., \(B = \sum_{i=1}^{n} \lambda_i q_i q_i^T\) where \(q_i\) is the \(i\)-th column of \(Q\). Let us define

\[
B^+ = \sum_{i: \lambda_i \geq 0} \lambda_i q_i q_i^T, \quad B^- = -\sum_{i: \lambda_i < 0} \lambda_i q_i q_i^T. \tag{2.17}
\]

The splitting \((B^+, B^-)\) is precisely the orthogonal PSD splitting introduced in [82]. Our next result establishes the equivalence between the optimal solution to the MTMS-PSD problem and the orthogonal PSD splitting.

**Theorem 2.2.6.** The optimal solution \((B_1^*, B_2^*)\) to the problem (MTMS-PSD) is given by \((B^+, B^-)\). Furthermore, the splitting \((B^+, B^-)\) is non-redundant.

**Proof.** Denote the optimal solution to the MTMS-PSD problem by \((B_1^*, B_2^*)\). We
first show \((B_1^*, B_2^*) = (B^+, B^-)\). Let \(P\) be the projection matrix defined by

\[
P = \sum_{i : \lambda_i \geq 0} q_i q_i^T.
\]

It follows immediately that

\[
\operatorname{Tr}(B_1^*) \geq \operatorname{Tr}(B_1^* P) = \operatorname{Tr}(B_1^* P^2) = \operatorname{Tr}(PB_1^* P) \geq \operatorname{Tr}(P(B_1^* - B_2^*) P) = \operatorname{Tr}(B^+),
\]

where the first inequality follows from the relation

\[
\operatorname{Tr}(B_1^* (I - P)) \geq 0.
\]

Here \(I\) denotes the identity matrix in \(\mathbb{R}^{n \times n}\). Similarly, one has

\[
\operatorname{Tr}(B_2^*) \geq \operatorname{Tr}(B_2^* (I - P)) = \operatorname{Tr}((I - P)B_2^* (I - P)) \geq \operatorname{Tr}(B^-).
\]

Therefore, we have

\[
\operatorname{Tr}(B_1^*) + \operatorname{Tr}(B_2^*) \geq \operatorname{Tr}(B^+) + \operatorname{Tr}(B^-),
\]

and the equality holds if and only if

\[
\operatorname{Tr}(B_1^* (I - P)) = 0, \quad \operatorname{Tr}(B_2^* P) = 0. \tag{2.18}
\]

Since all the matrices \(B_1^*, B_2^*, P\) and \(I - P\) are positive semi-definite. Relation (2.18) holds if and only if

\[
B_1^* = B_1^* P = P B_1^*, \quad B_2^* P = P B_2^* = 0. \tag{2.19}
\]

Since \(B_1^* - B_2^* = B = B^+ - B^-\), we thus have

\[
B_1^* = P B_1^* P = B^+, \quad B_2^* = -(I - P) B (I - P) = B^-.
\]
It remains to show that the matrix splitting \((B^+, B^-)\) is non-redundant. Suppose to the contrary that \((B^+, B^-)\) is a redundant splitting of \(B\), i.e., there exists \(R \neq 0 \succeq 0\) such that

\[
B_1 = B^+ - R \succeq 0, \quad B_2 = B^- - R \succeq 0, \quad B_1 - B_2 = B.
\]

Then we have

\[
\text{Tr}(B^+ B^-) = \text{Tr}((B_1 + R)(B_2 + R)) \geq \text{Tr}(B_1 B_2) + \text{Tr}(R^2) > 0,
\]

which contradicts to the relation \(\text{Tr}(B^+ B^-) = 0\). This finishes the proof of the theorem. \(\square\)

We remark that for the orthogonal PSD splitting \((B^+, B^-)\), it is easy to see that

\[
\text{Rank}(B) = \text{Rank}(B^+) + \text{Rank}(B^-).
\]

Since for any matrix splitting \(B = B_1 - B_2\), one has

\[
\text{Rank}(B) \leq \text{Rank}(B_1) + \text{Rank}(B_2).
\]

We thus have the following corollary.

**Corollary 2.2.7.** For any given matrix \(B\), the optimal solution to the MTMS-PSD problem is also optimal to the following rank minimization problem

\[
\begin{align*}
\text{min} & \quad \text{Rank}(B_1) + \text{Rank}(B_2) \\
\text{s.t.} & \quad B_1 - B_2 = B, \quad B_1, B_2 \succeq 0.
\end{align*}
\]

Since the rank of the splitting matrices will be further used to reduce the memory requirement and speed up the solving process for the relaxation model, the minimal rank solution is very appealing from both a theoretical and computational viewpoint.

Based on Theorem 2.2.6, for a given matrix \(B\), if its non-redundant splitting is unique, then the SDR based on the orthogonal PSD matrix splitting will be the
strongest among all the PSD matrix splittings. However, as one can see from the following example, the non-redundant PSD matrix splitting of a matrix might not be unique.

**Example 2.2.8.** Consider the matrix

\[
B = \begin{pmatrix}
0 & 1 & 2 \\
1 & 0 & 1 \\
2 & 1 & 0
\end{pmatrix}.
\]

By solving the MTMS-PSD problem with the above \(B\), we obtain the optimal solution as follows

\[
B_1^* = \begin{pmatrix}
1.0774 & 0.7887 & 1.0774 \\
0.7887 & 0.5774 & 0.7887 \\
1.0774 & 0.7887 & 1.0774
\end{pmatrix},
B_2^* = \begin{pmatrix}
1.0774 & -0.2113 & -0.9226 \\
-0.2113 & 0.5774 & -0.2113 \\
-0.9226 & -0.2113 & 1.0774
\end{pmatrix}.
\]

By Theorem 2.2.6, \((B_1^*, B_2^*)\) is an orthogonal and non-redundant PSD matrix splitting of \(B\). One can easily check that \(\text{Tr}(B_1^*B_2^*) = 0\).

Now, let us choose

\[
B_1 = \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{pmatrix},
B_2 = \begin{pmatrix}
1 & 0 & -1 \\
0 & 1 & 0 \\
-1 & 0 & 1
\end{pmatrix}.
\]

It is easy to see that \((B_1, B_2)\) is a PSD matrix splitting of \(B\). Since \(\text{Tr}(B_1B_2) \neq 0\), by Theorem 2.2.6, \((B_1, B_2)\) is not the optimal solution to the MTMS-PSD problem. We next show that \((B_1, B_2)\) is also non-redundant. Suppose to the contrary that \((B_1, B_2)\) is redundant, i.e., there exists \(R \neq 0 \succeq 0\) satisfying

\[
B_1 - R \succeq 0, \quad B_2 - R \succeq 0.
\]

Since \(B_1 = E\) and \(B_1 - R \succeq 0\), it must hold that \(R = \alpha E\) for some \(0 < \alpha \leq 1\). On the other hand, for any \(0 < \alpha \leq 1\), one can easily check that the matrix \(B_2 - \alpha E\) is
not positive semidefinite.

The non-uniqueness of the non-redundant PSD matrix splitting for a given matrix $B$ shown in the above example illustrates that it is nontrivial to find the strongest SDR based on matrix splitting. In the next section, we will discuss how to find other non-redundant matrix splitting schemes whose corresponding SDR is relatively easy to solve.

2.3 Two Matrix Splitting Schemes based on the Minimal Trace Principle

In this section, we first use the minimal trace principle to derive two matrix splitting schemes and characterize conditions under which the constructed matrix splitting is non-redundant. Then, we compare the lower bounds provided by the SDRs of QAPs based on these two matrix splitting schemes. We start by stating an assumptions regarding the QAPs throughout this section.

**Assumption 2.3.1.** At least one matrix ($A$ or $B$) in the underlying QAP has zeros on its diagonal.

It should be pointed out that the above assumption is quite reasonable and most QAP instances from the QAP library indeed satisfy such a condition. On the other hand, suppose both matrices $A$ and $B$ have nonzero diagonals. Let $d_a$ and $d_b$ denote the vectors consisting of the diagonal elements from $A$ and $B$, respectively. We thus can write $A$ and $B$ as $A = A_0 + \text{diag}(d_a)$, $B = B_0 + \text{diag}(d_b)$. It follows immediately that

$$
\text{Tr}(AXBX^T) = \text{Tr}(A_0XB_0X^T) + d_a^T X d_b
$$

The second term in the above relation reduces to a linear assignment problem which can be solved via its linear programming relaxation. For the ease of discussion, in the remaining part of this section, we assume that the matrix $A$ has zeros on its diagonal.
Under Assumption 2.3.1, we can easily show that for any diagonal matrix $D$, one has

$$\text{Tr}(AXBX^T) = \text{Tr}(AX(B - D)X^T) \quad \forall X \in \Pi. \quad (2.22)$$

That is, we can arbitrarily adjust the diagonal of $B$ without affecting the objective value of the QAP problem.

### 2.3.1 Minimal trace one-matrix splitting

Let us first consider a special case of the MTMS-PSD problem when $B_1 = tE$ for $t \geq 0$ where $E = ee^T$ is the all-1 matrix. In such a scenario, the MTMS-PSD problem reduces to the auxiliary SDP problem considered in [82]:

$$\min \alpha \quad (2.23)$$

s. t. $\alpha E - B \succeq 0$, $\alpha \geq 0$.

If the problem (2.23) is feasible, then the optimal solution $\alpha$ of problem (2.23) is used to split the matrix $B$ into the following form used in [74]:

$$B = \alpha E - (\alpha E - B).$$

However, as pointed out in [82], the problem (2.23) is in general infeasible. As a remedy for such an infeasibility issue, we propose to split the matrix $B - \beta I$ into the following form:

$$B - \beta I = \alpha E - B_2, \quad \text{with} \quad B_2 = \alpha E + \beta I - B \succeq 0, \quad (2.24)$$

where $\alpha \geq 0$ and $\beta$ are parameters to be identified. We call $(\alpha, \beta I)$ the one-matrix splitting of $B$. The above splitting is particularly attractive because the matrix $B_1 = \alpha E + \beta I$ is invariant under permutation. Therefore, it can substantially reduce the computational cost for solving the relaxed problem.
As in the previous section, in order to find a non-redundant matrix splitting for \( B \), we propose to solve the following auxiliary problem:

\[
\min \left\{ n(\alpha + \beta) : \alpha E + \beta I - B \succeq 0, \quad (\alpha, \beta) \in \mathbb{R}^2 \right\}, \tag{2.25}
\]

We next present an interesting result regarding problem (2.25) and the detailed proof of the theorem is given in Appendix A.

**Theorem 2.3.2.** Let \((\alpha, \beta)\) be the optimal solution of problem (2.25). If the matrix \( B \) is nonnegative, then \( \alpha > 0 \).

Theorem 2.3.2 states for any nonnegative matrix, we can always extract a positive scalar of the all one-matrix that is invariant for any permutation. Therefore, it is desirable to use the one-matrix PSD splitting. Our next theorem explores conditions under which the one-matrix splitting derived from problem (2.25) is non-redundant.

**Theorem 2.3.3.** Let \((\alpha, \beta)\) be the optimal solution of problem (2.25). Then the following statements hold:

(i) If \( B \) is nonnegative, then \((\alpha E, \alpha E + \beta I - B)\) is a non-redundant PSD matrix splitting of \( B - \beta I \);

(ii) If \( \beta > 0 \), then \((\alpha E + \beta I, \alpha E + \beta I - B)\) is a redundant PSD matrix splitting of \( B \).

**Proof.** We first consider statement (i). By Theorem 2.3.2, \( \alpha > 0 \). We have \( B_1 = \alpha E, B_2 = \alpha E + \beta I - B \). It is easy to see that \((B_1, B_2)\) is a PSD matrix splitting of \( B - \beta I \). We now prove that \((B_1, B_2)\) is non-redundant. Suppose to the contrary that \((B_1, B_2)\) is redundant, i.e., there exists nonzero matrix \( R \succeq 0 \) satisfying

\[
B - \beta I = (B_1 - R) - (B_2 - R), \quad B_1 - R \succeq 0, \quad B_2 - R \succeq 0.
\]

Since \( B_1 = \alpha E, \alpha > 0 \) and \( B_1 - R \succeq 0 \), it must hold that \( R = \tau E \) for some \( 0 < \tau \leq \alpha \). Therefore,

\[
B_2 - R = (\alpha - \tau)E + \beta I - B \succeq 0.
\]
This implies that \((\alpha - \tau, \beta)\) is a feasible solution of the problem (2.25). Note that \(n(\alpha - \tau + \beta) < n(\alpha + \beta)\), which contradicts the optimality of \((\alpha, \beta)\) with respect to problem (2.25). This proves statement (i).

Next we turn to statement (ii). Since \(\beta > 0\) and \(\alpha \geq 0\), we have \(\alpha E + \beta I > 0\). Because \(B = (\alpha E + \beta I) - (\alpha E + \beta I - B)\), and \(\alpha E + \beta I - B \succeq 0\), \((\alpha E + \beta I, \alpha E + \beta I - B)\) is a PSD splitting of \(B\). Let \(R = \eta(\alpha E + \beta I - B) \succeq 0\), and \(\eta \in (0, 1)\) is chosen such that 

\[
\lambda_{\max} (R) < \beta.
\]

It is easy to show that \(\alpha E + \beta I - R \succeq 0\) and \(\alpha E + \beta I - B - R \succeq 0\). Therefore, the matrix splitting \((\alpha E + \beta I, \alpha E + \beta I - B)\) is redundant. \(\square\)

Theorem 2.3.3 implies that if problem (2.25) has an optimal solution \((\alpha, \beta)\) with \(\beta > 0\), then the resulting SDR can be further improved by using a non-redundant PSD splitting of \(B\). When \(\beta \leq 0\), the one-matrix splitting might be a very good choice due to the simplicity of the resulting SDR model. We also point out that when \(B\) is the Hamming distance matrix of the hypercube in \(\mathbb{R}^m\), as proved in [74], the one-matrix splitting based on the minimal trace principle is also the orthogonal PSD splitting of \(B\). In such a case, the optimal solution to problem (2.25) is \((\alpha^*, \beta^*) = (\frac{m}{2}, 0)\).

In our experiments, we also observe that for some QAP instances such as Tai20b, Tai25b, Tai35b, Tai40b and Tai50b, problem (2.25) has an optimal solution with \(\beta < 0\) as listed in Table 2.1. We can further check from Table 2.3 and Table 2.4 of Section 2.5 that for these QAP instances, the SDR based on the one-matrix splitting scheme can provide a stronger lower bound than that based on the orthogonal PSD matrix splitting scheme.

2.3.2 Minimal trace sum-matrix splitting

In this subsection, we combine the minimal trace principle and the so-called sum-matrix to construct a non-redundant matrix splitting for a given matrix \(B\). First we recall the following definition [26].
Table 2.1: Optimal solution \((\alpha, \beta)\) of problem (2.25)

<table>
<thead>
<tr>
<th>Prob.</th>
<th>(\alpha)</th>
<th>(\beta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tai20b</td>
<td>420.4951</td>
<td>-2.5025</td>
</tr>
<tr>
<td>Tai25b</td>
<td>558.5071</td>
<td>-2.5238</td>
</tr>
<tr>
<td>Tai35b</td>
<td>584.1755</td>
<td>-2.0483</td>
</tr>
<tr>
<td>Tai40b</td>
<td>797.8480</td>
<td>-3.5505</td>
</tr>
<tr>
<td>Tai50b</td>
<td>969.1780</td>
<td>-5.2837</td>
</tr>
</tbody>
</table>

**Definition 2.3.4.** A matrix \(M\) is called a sum-matrix if

\[
M = ue^T + eu^T
\]  

for some \(u \in \mathbb{R}^n\). The sum-matrix has the following property:

\[
X(ue^T + eu^T)X^T = Xue^T + eu^TX^T, \quad \forall X \in \Omega,
\]  

where \(\Omega = \{X \in \mathbb{R}^{n \times n} \mid Xe = X^Te = e\}\).

Given a matrix \(B\), we can decompose it into two parts as \(B = \bar{B} + ue^T + eu^T\). Correspondingly, we have

\[
\text{Tr}(AXBX^T) = \text{Tr}(AX\bar{B}X^T) + 2e^TAXu.
\]

Using the above decomposition, we can reduce the contribution of the quadratic term in the objective function by moving the cost to a linear term.\(^3\) In what follows we use the sum-matrix to construct a non-redundant matrix splitting framework. Based on relation (2.22), we propose a splitting of the following form:

\[
B - D = ue^T + eu^T - B_2, \quad B_2 \succeq 0, \quad D = \text{Diag}(d),
\]  

\(^3\)We note that a similar approach (called the reduction method) has been used to improve the GLB and eigenvalue bound for QAPs with nonsymmetric matrices in the literature [26, 32, 40, 90]. One simple choice is \(u = \min(B_{\text{off}})\). For more details on the reduction method, we refer to Section 7.5.2 of [26].
where \(d \in \mathbb{R}^n\) and \(u \in \mathbb{R}^n\) are vectors to be found. We call a pair \((u, D)\) satisfying (2.28) a sum-matrix splitting of the matrix \(B\). Like in the last subsection, we use the minimal trace principle to find a non-redundant sum-matrix splitting of the matrix \(B\) as follows:

\[
\min_{u \in \mathbb{R}^n, d \in \mathbb{R}^n} e^T (2u + d) \\
\text{s. t.} \quad ue^T + eu^T + D - B \succeq 0, \quad D = \text{Diag}(d).
\]

It is easy to see that the above problem is strictly feasible. It should be pointed out that for simplicity of the model, we can also impose the constraint that \(D = \beta I\) for some parameter \(\beta \in \mathbb{R}\). In such a circumstance, the sum-matrix splitting \((u, \beta I)\) of \(B\) includes the one-matrix splitting \((\alpha, \beta I)\) as a special case where \(u = \frac{\alpha}{2} e\) for some \(\alpha \in \mathbb{R}\).

2.3.3 Relations between the lower bounds

In this subsection, we compare the two lower bounds provided by the SDRs based on the one-matrix splitting and the sum-matrix splitting described in the previous subsections.

Let \((\alpha, \beta I)\) and \((u, \beta I)\) be respectively the minimal trace one-matrix splitting and sum-matrix splitting of \(B\). Then we can derive the following two SDRs of QAPs

\[
\mu_1(\alpha, \beta I) = \min_{(X,Y) \in \mathcal{X}_{\alpha, \beta I}} \alpha \text{Tr}(AE) - \text{Tr}(AY),
\]

\[
\mu_2(u, \beta I) = \min_{(X,Y) \in \mathcal{Y}_{u, \beta I}} 2e^T AX u - \text{Tr}(AY),
\]
where

\[
\mathcal{Y}_{(\alpha, \beta I)} = \left\{ (X, Y_2) \in \mathbb{R}^{n \times n} \times \mathbb{S}^n \ \middle| \ \begin{array}{l}
Y_2 = X - X(\beta I - B)X^T \succeq 0, \\
\text{diag} (Y_2) = \alpha e + X \text{diag} (\beta I - B), \\
Y_2 e = nae + X(\beta I - B)e, \\
X e = X^T e = e, \quad X \geq 0
\end{array} \right\},
\]

\[
\mathcal{Y}_{(u, \beta I)} = \left\{ (X, Y_2) \in \mathbb{R}^{n \times n} \times \mathbb{S}^n \ \middle| \ \begin{array}{l}
Y_2 = X - X u - eu^T X^T + X(B - \beta I)X^T \succeq 0, \\
\text{diag} (Y_2) = 2X u - X \text{diag} (B - \beta I), \\
Y_2 e = nX u + eu^T e - X(B - \beta I)e, \\
X e = X^T e = e, \quad X \geq 0
\end{array} \right\}.
\]

Note that here we only deal with a special case of problem (2.29) by setting \( d = \beta e \) to get:

\[
\min_{u \in \mathbb{R}^n, \beta \in \mathbb{R}} \quad 2e^T u + n\beta \\
\text{subject to} \quad eu^T + eu^T + \beta I - B \succeq 0.
\] (2.32)

**Lemma 2.3.5.** Suppose that \((\bar{\alpha}, \bar{\beta})\) and \((\hat{u}, \hat{\beta})\) are the optimal solution of problems (2.25) and (2.32), respectively. Then \(\bar{\beta} \geq \hat{\beta}\).

**Proof.** Denote \((u, \beta)\) any feasible solution to problem (2.32). Define

\[
\alpha = \frac{2u e}{n}, \quad v = u - \frac{\alpha}{2} e.
\]

It follows immediately

\[
u = \frac{\alpha}{2} e + v, \quad v^T e = 0.
\]

Using the above notation, we can rewrite problem (2.32) as

\[
\min_{v \in \mathbb{R}^n, \beta \in \mathbb{R}} \quad n\alpha + n\beta \\
\text{subject to} \quad \alpha E + v e^T + e v^T + \beta I - B \succeq 0; \\
\quad e^T v = 0.
\] (2.33)
We next show that the optimal solution to the above problem can be obtained explicitly. Denote the optimal solution of problem (2.33) by \((\hat{\alpha}, \hat{\beta})\) and \(P = I - \frac{E}{n}\).

From the constraint \(e^T v = 0\) we obtain

\[
P(\hat{\alpha}E + ve^T + ev^T + \hat{\beta}I - B)P = \hat{\beta}P - PBP \succeq 0,
\]

which implies

\[
\hat{\beta} \geq \lambda_{\max}(PBP).
\]

Here \(\lambda_{\max}(PBP)\) denotes the largest eigenvalue of the matrix \(PBP\). Similarly, we have

\[
(I - P)(\hat{\alpha}E + ve^T + ev^T + \hat{\beta}I - B)(I - P) = \hat{\alpha}E + \frac{\hat{\beta}}{n}E - \frac{e^T Be}{n^2}E \succeq 0,
\]

which implies

\[
\hat{\alpha} + \frac{\hat{\beta}}{n} \geq \frac{e^T Be}{n^2}.
\]

It follows that

\[
\hat{\alpha} + \hat{\beta} \geq \frac{e^T Be}{n^2} + \frac{(n - 1)\hat{\beta}}{n} \geq \frac{e^T Be}{n^2} + \frac{(n - 1)\lambda_{\max}(PBP)}{n}.
\]

Now let us choose

\[
v = \frac{e^T Be}{n}e - Be, \quad \beta = \lambda_{\max}(PBP), \quad \alpha = \frac{e^T Be}{n^2} - \frac{\beta}{n}.
\]

One can easily verify that \((\alpha, \beta, v)\) satisfy all the constraints in problem (2.33). Therefore, we can conclude that at the optimal solution of problem (2.33), it must hold

\[
\hat{\alpha} = \frac{e^T Be}{n^2} - \frac{1}{n}\lambda_{\max}(PBP), \quad \hat{\beta} = \lambda_{\max}(PBP).
\]

On the other hand, if \((\bar{\alpha}, \bar{\beta})\) is the optimal solution to problem (2.25), then by
following a similar process, one can show that

\[ \bar{\beta} \geq \lambda_{\text{max}}(PBP) = \hat{\beta}. \]

The completes the proof of the lemma. \[\Box\]

Based on Lemma 2.3.5, we can establish the following result regarding the two lower bounds \( \mu_1(\alpha, \bar{\beta}I) \) and \( \mu_2(u, \hat{\beta}I) \).

**Theorem 2.3.6.** Assume that \((\alpha, \bar{\beta}I)\) and \((u, \hat{\beta}I)\) are the minimal trace one-matrix splitting and sum-matrix splitting of \( B \), respectively. Then we have

\[ \mu_2(u, \hat{\beta}I) \geq \mu_1(\alpha, \bar{\beta}I). \] (2.34)

*Proof.* Let \( Y \) be an optimal solution of the problem (2.31). Then there exists \( X \) such that

\[
\begin{align*}
Y - Xue^T - eu^TX^T - X(\hat{\beta}I - B)X^T &\succeq 0, \\
\text{diag}(Y) &= 2Xu + X\text{diag}(\hat{\beta}I - B), \\
ye &= nu + eu^Te + X(\hat{\beta}I - B)e, \\
x_e &= X^Te = e, \quad X \succeq 0,
\end{align*}
\] (2.35)

and

\[ \mu_2(u, \hat{\beta}I) = 2e^TAXu - \text{Tr}(AY). \] (2.36)

Since \( X \) is a \( n \times n \) doubly stochastic matrix, similar to the proof of Theorem 2.2.5, we can infer that there exist \( \lambda_i \geq 0, \forall i = 1, \ldots, |\Pi| \) such that

\[
X = \sum_{i=1}^{|\Pi|} \lambda_i \hat{X}_i, \quad \sum_{i=1}^{|\Pi|} \lambda_i = 1, \quad \hat{X}_i \in \Pi \quad \forall \; i,
\]
which further implies
\[ \text{diag}(Xue^T) = \sum_{i=1}^{\Pi} \lambda_i \text{diag}(\hat{X}_iue^T) = \sum_{i=1}^{\Pi} \lambda_i \hat{X}_i u = Xu. \]

The above relation, together with \( eu^T X^T = (Xue^T)^T \), yields
\[ \text{diag}(eu^T X^T) = \text{diag}(Xue^T) = Xu. \] (2.37)

Since \( X \) is a doubly stochastic matrix, we have
\[ I - XX^T \succeq 0. \] (2.38)

Let us define
\[ \hat{Y} = Y - Xue^T - eu^T X^T + \alpha E + (\bar{\beta} - \hat{\beta})I. \] (2.39)

It follows
\[ \hat{Y} - \alpha E - X(\bar{\beta}I - B)X^T = Y - Xue^T - eu^T X^T - X(\bar{\beta}I - B)X^T + (\bar{\beta} - \hat{\beta})(I - XX^T). \]

By Lemma 2.3.5, \( \bar{\beta} - \hat{\beta} \geq 0 \), it follows immediately from (2.35), (2.37) and (2.38) that
\[ \hat{Y} - \alpha E - X(\bar{\beta}I - B)X^T \succeq 0, \]
\[ \hat{Y}e = nae + X(\bar{\beta}I - B)e, \]
\[ \text{diag}(\hat{Y}) = \alpha e + X \text{diag}(\bar{\beta}I - B). \]

These, together with (2.35), imply that \( \hat{Y} \in \mathcal{Y}_*(\alpha, \beta I) \). Thus
\[ \alpha \text{Tr}(AE) - \text{Tr}(A\hat{Y}) \geq \mu_1(\alpha, \beta I). \] (2.40)
On the other hand, under Assumption 2.3.1, we can deduce from (2.39) that
\[ \alpha \text{Tr}(AE) - \text{Tr}(A\hat{Y}) = 2e^TAXu - \text{Tr}(AY), \]
which, together with (2.36) and (2.40), implies that
\[ \mu_2(u, \hat{\beta}I) \geq \mu_1(\alpha, \bar{\beta}I). \]

Theorem 2.3.6 shows that under certain conditions, the SDR (2.31) based on the sum-matrix splitting is at least as good as the SDR (2.30) based on the one-matrix splitting. However, in our numerical experiments, we have observed that in most cases, we have \( \mu_2(u, D) > \mu_1(\alpha, D). \) Theorem 2.3.6 provides a partial explanation for such a phenomenon. It should also be pointed out that, as illustrated by the numerical results in Section 2.5, there is no dominance relation between the bounds derived from the three different matrix splitting schemes described in this work.

2.4 SDRs of QAPs based on minimal trace matrix splitting

In this section, we present the SDR models of QAPs based on the three matrix splitting schemes discussed in Section 2.2 and 2.3 using the framework introduced in [82], which combine a technique to reduce the dimension of PSD constraints.

We first present the SDR model of QAP derived from the so-called orthogonal PSD matrix splitting (denoted by SDRMS-SVD) which can be derived by using the singular value decomposition (SVD) of \( B. \) Let \((B_1, B_2)\) be the orthogonal PSD splitting of \( B. \) By Corollary 2.2.7, we have \( \text{Rank}(B_1) + \text{Rank}(B_2) \leq n. \) Because \( B_1 \succeq 0 \) and \( B_2 \succeq 0, \) we have \( B_i = \hat{B}_i^T \hat{B}_i \) for some \( \hat{B}_i \in \mathbb{R}^{m_i \times n}, i = 1, 2. \) Based on the well-known Schur complement lemma, the quadratic PSD constraints
\[ Y_i - XB_iX^T \succeq 0, \quad i = 1, 2, \]
can equivalently be replaced by the PSD constraints of smaller scale

\[
\begin{pmatrix}
I_{m_i \times m_i} & \hat{B}_i X^T \\
X \hat{B}_i^T & Y_i
\end{pmatrix}_{(m_i+n) \times (m_i+n)} \succeq 0, \quad i = 1, 2. \tag{2.41}
\]

Note that we can write \( B_i = \hat{B}_i^T \hat{B}_i \) by setting \( \hat{B}_i = \Lambda_i \hat{V}_i^T \), where \( \Lambda_i \) is an \( m_i \times m_i \) diagonal matrix whose diagonal elements are the square roots of the non-zero eigenvalues of \( \hat{B}_i \), and \( \hat{V}_i \) is an \( n \times m_i \) matrix whose columns are the corresponding eigenvectors.

Note that the new SDRMS-SVD model is different from the SDRMS-SVD model in [82] in two aspects: First the SDP constraints on \( Y_1 \) and \( Y_2 \) are simplified (see (2.41)) by using the rank information of \( B_1 \) and \( B_2 \), respectively. Secondly, we add extra constraints on the matrix \( Y_1 + Y_2 \). For self-completeness, we describe the full model below:

(SDRMS-SVD) \( \min \) \( \text{Tr}(A(Y_1 - Y_2)) \)

s.t. \( \text{diag}(Y_1) = X \text{diag}(B_1) \), \( Y_1 e = X B_1 e; \)

\( \text{diag}(Y_2) = X \text{diag}(B_2) \), \( Y_2 e = X B_2 e; \)

\( (X \min(|B_1|_{\text{off}}))_{i} \leq [Y_1]_{i,j} \leq (X \max(|B_1|_{\text{off}}))_{i}, \quad \forall i \neq j \)

\( (X \min(|B_2|_{\text{off}}))_{i} \leq [Y_2]_{i,j} \leq (X \max(|B_2|_{\text{off}}))_{i}, \quad \forall i \neq j \)

\( (X \min(|B_1+B_2|_{\text{off}}))_{i} \leq [Y_1 + Y_2]_{i,j} \leq (X \max(|B_1+B_2|_{\text{off}}))_{i}, \quad \forall i \neq j \)

\( L_2(Y_1) \leq X L_2(B_1), X L_2(Y_2) \leq X L_2(B_2) \)

\( L_2(Y_1 - Y_2) \leq X L_2(B), X L_2(Y_1 + Y_2) \leq X L_2(B_1 + B_2) \)

\[
\begin{pmatrix}
I_{m_1 \times m_1} & \hat{B}_1 X^T \\
X \hat{B}_1 & Y_1
\end{pmatrix} \succeq 0, \quad \begin{pmatrix}
I_{m_2 \times m_2} & \hat{B}_2 X^T \\
X \hat{B}_2 & Y_2
\end{pmatrix} \succeq 0
\]

\( X \geq 0, \quad X e = X^T e = e. \)

Next, we present the SDR model of QAPs based on minimal trace sum-matrix splitting (denoted by SDRMS-SUM). Let \((u, D)\) be the optimal solution of prob-
lem (2.29). The minimal trace sum-matrix splitting of $B$ is given by

$$B_1 = ue^T + eu^T + D, \quad B_2 = ue^T + eu^T + D - B \succeq 0.$$ 

We observe that for all the tested examples, the rank of $B_2$ tends to be much smaller than $n$. This is not surprising since in (2.29), we are minimizing the trace of $B_2$, which is a proxy for minimizing the rank of $B_2$. Let $\tilde{B}_2$ be a $m \times n$ matrix satisfying $B_2 = \tilde{B}_2^T \tilde{B}_2$. The SDRMS-SUM model is defined as follows:

(SDRMS-SUM)

$$\begin{align*}
\min & \quad 2e^TAXu - \text{Tr}(AY_2) \\
\text{s.t.} & \quad \text{diag}(Y_2) = X \text{diag}(B_2), \quad Y_2e = XB_2e, \\
& \quad Y_1 = Xue^T + eu^TX^T + \text{diag}(Xd) \\
& \quad (X \min([B_2]_{\text{off}}))_i \leq [Y_2]_{i,j} \leq (X \max([B_2]_{\text{off}}))_i \quad \forall i \neq j \\
& \quad (X \min([B]_{\text{off}}))_i \leq [Y_1 - Y_2]_{i,j} \leq (X \max([B]_{\text{off}}))_i \quad \forall i \neq j \\
& \quad (X \min([B_1 + B_2]_{\text{off}}))_i \leq [Y_1 + Y_2]_{i,j} \leq (X \max([B_1 + B_2]_{\text{off}}))_i, \quad \forall i \neq j \\
& \quad \left( \begin{array}{c}
I_{m \times m} \\
X \tilde{B}_2^T \\
Y_2
\end{array} \right) \succeq 0, \quad \mathcal{L}_2(Y_2) \leq X \mathcal{L}_2(B_2) \\
& \quad \mathcal{L}_2(Y_1 - Y_2) \leq X \mathcal{L}_2(B), \quad \mathcal{L}_2(Y_1 + Y_2) \leq X \mathcal{L}_2(B_1 + B_2) \\
& \quad X \geq 0, \quad Xe = XT^Te = e.
\end{align*}$$

We note that it is possible to obtain a tighter lower bound based on the intersection of the feasible regions of (SDRMS-SVD) and (SDRMS-SUM) by considering the two different splittings $(Y_1^\text{SVD}, Y_2^\text{SVD})$ and $(Y_1^\text{sum}, Y_2^\text{sum})$ in (SDRMS-SVD) and (SDRMS-SUM), respectively. However, our numerical experience show that the tighter lower bound is usually only slightly better than the best lower bound obtained from (SDPRMS-SVD) and (SDRMS-SUM) individually but the computational cost is substantially larger than the total cost of solving (SDRMS-SVD) and (SDRMS-SUM) individually. Thus in this chapter, we shall not consider such a tighter lower bound.
Finally, the SDR of QAP based on minimal trace one-matrix splitting (denoted by SDRMS-ONE) discussed in the previous section is a variant of the models introduced in [74, 82]. Let $(\alpha, \beta)$ be the solution of problem (2.25). Then $B_2 = \alpha E + \beta I - B \succeq 0$. Again, we observed that for all the tested examples, $m = \text{Rank}(B_2) < n$. Let $\hat{B}_2$ be a $m \times n$ matrix such that $B_2 = \hat{B}_2^T \hat{B}_2$. The model is described below:

$$(\text{SDRMS-ONE}) \quad \min \quad \alpha \text{Tr}(AE) - \text{Tr}(AY)$$

subject to

- $\text{diag}(Y) = X\text{diag}(B_2)$,
- $Ye = XB_2e$
- $(X \min([B_2]_{off}))_i \leq [Y]_{i,j} \leq (X \max([B_2]_{off})), \ \forall i \neq j$
- $\begin{pmatrix} I_{m \times m} & \hat{B}_2 X^T \\ X \hat{B}_2^T & Y \end{pmatrix} \succeq 0$, \quad $\mathcal{L}_2(Y - \alpha E - \beta I) \leq X \mathcal{L}_2(B)$,
- $X \geq 0$, \quad $Xe = X^T e = e$.

### 2.5 Numerical Experiments

In this section, we report some numerical results of the three relaxation models based on non-redundant matrix splitting schemes given in the previous section on QAP instances from the QAP library [27]. For a comparison between the SDRMS-SVD model and other existing relaxation models we refer to our earlier work [82]. We remind the readers that while the SDRMS-SVD is almost identical to the F-SVD model used in [82], the SDRMS-ONE model in this work is different from the model used in [74] in terms of the parameters, the cuts and constraints. We also mention that adding extra constraints to the SDRMS-ONE model as in [74] will further improve the lower bound. However, in our experiments, we found that those improvements are usually associated with the particular instances and thus might not be substantial for generic QAPs. On the other hand, it may increase the computational cost of the resulting relaxation model. The SDRMS-SUM bound is derived by applying the relaxation framework introduced in [82] to the new sum-matrix splitting scheme. As QAP(A,B) is equivalent to QAP(B,A), we compute the lower bounds for both orderings of the $A, B$ matrices and report the stronger one.
only for all the three models.

In our experiments, all the problems were solved in MATLAB R2009b on a 3.33GHz Intel Core 2 Duo PC with 8GB memory. For QAPs of small and median sizes \( n \leq 70 \), the SDR problems were automatically generated by CVX 1.2 [53] and solved by the SDP solver SDPT3 [97] (see Table 2.3 and Table 2.4). In our numerical experiments, we observed that CVX would consume more than 50% of the total computation time just to generate the SDP data for the relaxation problem when \( n > 40 \), and the amount of computer memory required by CVX becomes prohibitively large when \( n > 70 \), in addition to taking excessively long computer time to solve the resulting SDP problem. Thus for large scale QAPs \( (n > 70) \), the SDR problems were solved by using the new SDP solver – SDPNAL [105], which is designed to solve large scale SDP problems to moderate accuracy, and the input SDP data was coded on our own in order to control its structure (see Table 2.5). We should emphasize that our own routine substantially cut down the time taken and the memory needed to generate the SDP data as compared to that consumed by CVX.

In our numerical experiments, SDPNAL usually stops with an approximate SDP solution where the maximum of the relative primal infeasibility, dual infeasibility and duality gap is in the order to \( 10^{-5} \) to \( 10^{-6} \). In such a case, we use the procedure described in [60] to find a rigorous lower bound for our relaxation model. In all the tables, the relative gap is computed by

\[
R_{\text{gap}} = 1 - \frac{\text{Lower bound}}{\text{Optimal or best known feasible objective value}}
\]

and the CPU time (in seconds) to compute the bound is listed under the column “CPU”. We use the boldface font to highlight the strongest of the three bounds. We note that for several medium and large instances, the SDRMS-SUM bounds have exceeded the best-known bounds reported in QAPLIB [27]. We list those bounds in a separate table (Table 2.6) for ease of reference.

For the SDRMS-ONE model, we also report the matrix splitting parameters \( \alpha, \beta \) in the tables. For the QAP instances that are associated with a Hamming or Manhattan distance matrix, the bounds computed by splitting the distance matrices are always
better than the one based on the non-distance matrix. This confirms the results in [74] from a different perspective (see Tables 2.3, 2.4 and 2.5). The SDRMS-ONE bounds may be stronger than the SDRMS-SVD if the matrix splitting parameter $\beta \leq 0$ because the matrix splitting $(\alpha E + \beta I, \alpha E + \beta I - B)$ is a non-redundant matrix splitting of matrix $B$ (see Tai20b, Tai25b in Table 2.3, Tai35b, Tai40b, Tai50b, Ste36c in Table 2.4 and Tai80b in Table 2.5). In some cases, the SDRMS-ONE bounds may be stronger than the SDRMS-SVD even if $\beta > 0$ (see Tai12b and Tai60b in Tables 2.3 and 2.4). This is because a new non-redundant matrix splitting $(\alpha E + \beta I - R, \alpha E + \beta I - B - R)$ can be obtained by solving the following SDP

$$\begin{align*}
\max & \quad \text{Tr}(R) \\
\text{s. t.} & \quad R \succeq 0, \quad \alpha E + \beta I - R \succeq 0, \\
& \quad \alpha E + \beta I - B - R \succeq 0.
\end{align*}$$

For the instances Tai12b and Tai60b, their SDRMS-ONE bounds can thus be improved by using the new non-redundant matrix splitting according to Theorem 2.2.5. But the improvements of the bounds are marginal as one can see from Table 2.2 because $\beta$ is very small (which shows the R-redundant matrix splitting is actually very “close” to its corresponding non-redundant matrix splitting).

For all the QAP instances tested, SDRMS-SUM bounds are always stronger than SDRMS-ONE (see Tables 2.3, 2.4 and 2.5). Theorem 2.3.6 provides an interesting explanation for such a phenomena. For most QAP instances, SDRMS-SUM bounds are stronger than the SDRMS-SVD bounds (see Tables 2.3 and 2.4). This is because the sum-matrix splitting is not only non-redundant, but also can reduce the contribution of the quadratic term in the objective function. One exception here is

<table>
<thead>
<tr>
<th></th>
<th>OPT/Feas</th>
<th>SDRMS-SVD</th>
<th>SDRMS-ONE</th>
<th>Improved bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tai12b</td>
<td>39464925</td>
<td>28523049</td>
<td>28863106</td>
<td>28957953</td>
</tr>
<tr>
<td>Tai60b</td>
<td>608215054</td>
<td>494775609</td>
<td>515731315</td>
<td>515943103</td>
</tr>
</tbody>
</table>

Table 2.2: Improved bounds using non-redundant matrix splitting
the TaiXXc instances (see Tai64c in Table 2.4) where the SDRMS-SVD bound is stronger than the SDRMS-SUM bound. This is possibly due to the fact that the matrix in the TaiXXc instances has a very specific sparse block structure and the orthogonal PSD matrix splitting can preserve such a desirable structure, while the sum matrix splitting fails to retain such a structure. The same reasoning can also be used to explain the SDRMS-ONE bounds for the TaiXXc instances (see the \(\alpha, \beta\) values for Tai64c in Table 2.4).

In terms of computation time, we note that for small and medium scale instances, the SDRMS-SVD model is the most expensive, while the SDRMS-ONE model is the cheapest. This is not surprising due to their model complexities. Overall, SDRMS-SUM is usually preferred to SDRMS-SVD and SDRMS-ONE considering the quality of the bounds and the complexity of the model.

In Table 2.4, we compared the cpu time of SDRMS-SVD model which takes advantage of the rank information at the optimal solution of the MTP auxiliary problem with the cpu time of the full SVD model used in [82]. We found the SDRMS-SVD model is much more efficient to compute.

Table 2.5 has shown that SDPNAL is an very effective SDP solver for large scale SDP problems. Although, it usually does not provide an accurate solution to the SDP problem as SDPT3 does, the accuracy level is good enough for our purpose of estimating the lower bounds.

### 2.6 Conclusions

In this chapter, we considered the issue of how to choose an appropriate matrix splitting scheme so that the resulting SDR for QAPs can provide a strong lower bound. To obtain such a desirable relaxation, we introduced the notion of redundant and non-redundant matrix splitting and showed that for every redundant splitting, there is a corresponding non-redundant splitting whose resulting SDR can provide a stronger bound. To find a non-redundant matrix splitting, we proposed to solve some auxiliary SDP problems. The properties of the optimal solutions to these SDP
<table>
<thead>
<tr>
<th></th>
<th>SDRMS-SUM</th>
<th></th>
<th>SDRMS-SVD</th>
<th></th>
<th>SDRMS-ONE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R_{gap}$</td>
<td>CPU</td>
<td>$R_{gap}$</td>
<td>CPU</td>
<td>$R_{gap}$</td>
<td>CPU</td>
</tr>
<tr>
<td>Chr12a</td>
<td>11.02%</td>
<td>8</td>
<td>11.41%</td>
<td>8</td>
<td>20.66%</td>
<td>4</td>
</tr>
<tr>
<td>Chr12b</td>
<td>24.65%</td>
<td>6</td>
<td>17.55%</td>
<td>6</td>
<td>30.03%</td>
<td>4</td>
</tr>
<tr>
<td>Chr12c</td>
<td>11.87%</td>
<td>6</td>
<td>12.46%</td>
<td>7</td>
<td>28.84%</td>
<td>4</td>
</tr>
<tr>
<td>Chr15a</td>
<td>24.80%</td>
<td>24</td>
<td>21.76%</td>
<td>11</td>
<td>32.50%</td>
<td>6</td>
</tr>
<tr>
<td>Chr15b</td>
<td>35.33%</td>
<td>10</td>
<td>34.39%</td>
<td>12</td>
<td>51.85%</td>
<td>6</td>
</tr>
<tr>
<td>Chr15c</td>
<td>7.31%</td>
<td>10</td>
<td>7.89%</td>
<td>12</td>
<td>18.88%</td>
<td>5</td>
</tr>
<tr>
<td>Had12</td>
<td>1.63%</td>
<td>6</td>
<td>2.90%</td>
<td>8</td>
<td>3.33%</td>
<td>4</td>
</tr>
<tr>
<td>Had14</td>
<td>1.69%</td>
<td>8</td>
<td>2.72%</td>
<td>11</td>
<td>3.05%</td>
<td>5</td>
</tr>
<tr>
<td>Had16</td>
<td>2.45%</td>
<td>11</td>
<td>3.09%</td>
<td>13</td>
<td>3.55%</td>
<td>6</td>
</tr>
<tr>
<td>Had18</td>
<td>2.43%</td>
<td>14</td>
<td>3.08%</td>
<td>18</td>
<td>3.53%</td>
<td>7</td>
</tr>
<tr>
<td>Had20</td>
<td>2.07%</td>
<td>18</td>
<td>2.86%</td>
<td>25</td>
<td>3.41%</td>
<td>10</td>
</tr>
<tr>
<td>Nug12</td>
<td>9.34%</td>
<td>6</td>
<td>10.03%</td>
<td>7</td>
<td>10.72%</td>
<td>4</td>
</tr>
<tr>
<td>Nug14</td>
<td>6.80%</td>
<td>8</td>
<td>7.10%</td>
<td>10</td>
<td>7.20%</td>
<td>5</td>
</tr>
<tr>
<td>Nug15</td>
<td>7.30%</td>
<td>10</td>
<td>7.57%</td>
<td>12</td>
<td>7.83%</td>
<td>6</td>
</tr>
<tr>
<td>Nug16a</td>
<td>7.02%</td>
<td>11</td>
<td>8.14%</td>
<td>14</td>
<td>8.39%</td>
<td>6</td>
</tr>
<tr>
<td>Nug16b</td>
<td>8.95%</td>
<td>11</td>
<td>9.52%</td>
<td>13</td>
<td>10.24%</td>
<td>6</td>
</tr>
<tr>
<td>Nug17</td>
<td>9.01%</td>
<td>13</td>
<td>9.24%</td>
<td>11</td>
<td>9.70%</td>
<td>7</td>
</tr>
<tr>
<td>Nug18</td>
<td>9.17%</td>
<td>15</td>
<td>9.38%</td>
<td>17</td>
<td>9.74%</td>
<td>8</td>
</tr>
<tr>
<td>Nug20</td>
<td>9.03%</td>
<td>19</td>
<td>9.30%</td>
<td>24</td>
<td>9.69%</td>
<td>10</td>
</tr>
<tr>
<td>Nug21</td>
<td>9.15%</td>
<td>22</td>
<td>9.60%</td>
<td>28</td>
<td>10.09%</td>
<td>11</td>
</tr>
<tr>
<td>Nug22</td>
<td>8.68%</td>
<td>26</td>
<td>9.15%</td>
<td>33</td>
<td>9.54%</td>
<td>12</td>
</tr>
<tr>
<td>Nug24</td>
<td>8.89%</td>
<td>32</td>
<td>9.29%</td>
<td>40</td>
<td>9.69%</td>
<td>15</td>
</tr>
<tr>
<td>Nug25</td>
<td>9.05%</td>
<td>36</td>
<td>9.19%</td>
<td>46</td>
<td>9.75%</td>
<td>17</td>
</tr>
<tr>
<td>Nug27</td>
<td>7.91%</td>
<td>48</td>
<td>8.41%</td>
<td>60</td>
<td>8.79%</td>
<td>20</td>
</tr>
<tr>
<td>Nug28</td>
<td>8.38%</td>
<td>52</td>
<td>8.73%</td>
<td>65</td>
<td>8.96%</td>
<td>22</td>
</tr>
<tr>
<td>Nug30</td>
<td>8.43%</td>
<td>67</td>
<td>8.54%</td>
<td>79</td>
<td>8.93%</td>
<td>28</td>
</tr>
<tr>
<td>Kra30a</td>
<td>16.78%</td>
<td>62</td>
<td>16.84%</td>
<td>82</td>
<td>17.72%</td>
<td>25</td>
</tr>
<tr>
<td>Kra30b</td>
<td>17.68%</td>
<td>63</td>
<td>17.76%</td>
<td>83</td>
<td>18.69%</td>
<td>22</td>
</tr>
<tr>
<td>Rou12</td>
<td>13.40%</td>
<td>6</td>
<td>11.16%</td>
<td>7</td>
<td>14.50%</td>
<td>4</td>
</tr>
<tr>
<td>Rou15</td>
<td>14.77%</td>
<td>10</td>
<td>13.20%</td>
<td>11</td>
<td>15.77%</td>
<td>5</td>
</tr>
<tr>
<td>Rou20</td>
<td>16.87%</td>
<td>20</td>
<td>16.02%</td>
<td>24</td>
<td>17.70%</td>
<td>10</td>
</tr>
<tr>
<td>Scr12</td>
<td>7.25%</td>
<td>6</td>
<td>7.85%</td>
<td>7</td>
<td>8.79%</td>
<td>4</td>
</tr>
<tr>
<td>Scr15</td>
<td>10.02%</td>
<td>9</td>
<td>10.53%</td>
<td>12</td>
<td>11.70%</td>
<td>6</td>
</tr>
<tr>
<td>Scr20</td>
<td>16.00%</td>
<td>20</td>
<td>16.39%</td>
<td>24</td>
<td>17.28%</td>
<td>10</td>
</tr>
<tr>
<td>Tai12b</td>
<td>8.61%</td>
<td>6</td>
<td>23.69%</td>
<td>7</td>
<td>23.71%</td>
<td>4</td>
</tr>
<tr>
<td>Tai20b</td>
<td>6.39%</td>
<td>20</td>
<td>46.16%</td>
<td>28</td>
<td>30.53%</td>
<td>11</td>
</tr>
<tr>
<td>Tai25b</td>
<td>15.15%</td>
<td>40</td>
<td>23.70%</td>
<td>54</td>
<td>20.58%</td>
<td>18</td>
</tr>
<tr>
<td>Tai30b</td>
<td>12.72%</td>
<td>68</td>
<td>14.68%</td>
<td>94</td>
<td>14.82%</td>
<td>31</td>
</tr>
<tr>
<td>Thou30</td>
<td>12.24%</td>
<td>67</td>
<td>13.09%</td>
<td>79</td>
<td>13.62%</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prob.</td>
<td>$\beta$</td>
<td></td>
<td>$\alpha$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3: Selected bounds for QAPs of small sizes ($n \leq 30$), computed using CVX with the SDPT3 solver.
Table 2.4: Selected bounds for QAPs of median sizes (30 < n ≤ 70), computed using CVX with the SDPT3 solver. 'NA' denote the solver failed due to out of memory.

<table>
<thead>
<tr>
<th>Prob</th>
<th>SDRMS-SUM</th>
<th>SDRMS-SVD</th>
<th>SVD-FULL</th>
<th>SDRMS-ONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R_{gap}$</td>
<td>$R_{gap}$</td>
<td>$R_{gap}$</td>
<td>$R_{gap}$</td>
</tr>
<tr>
<td>Kra32</td>
<td>18.49%</td>
<td>18.59%</td>
<td>19.27%</td>
<td>28</td>
</tr>
<tr>
<td>Sko42</td>
<td>7.58%</td>
<td>7.94%</td>
<td>7.94%</td>
<td>8.25%</td>
</tr>
<tr>
<td>Sko49</td>
<td>6.92%</td>
<td>7.35%</td>
<td>7.35%</td>
<td>7.68%</td>
</tr>
<tr>
<td>Sko56</td>
<td>6.87%</td>
<td>7.11%</td>
<td>7.11%</td>
<td>7.48%</td>
</tr>
<tr>
<td>Sko64</td>
<td>6.29%</td>
<td>6.61%</td>
<td>6.61%</td>
<td>6.78%</td>
</tr>
<tr>
<td>Ste36a</td>
<td>18.83%</td>
<td>19.34%</td>
<td>19.34%</td>
<td>19.60%</td>
</tr>
<tr>
<td>Ste36b</td>
<td>18.41%</td>
<td>23.71%</td>
<td>23.72%</td>
<td>55</td>
</tr>
<tr>
<td>Ste36c</td>
<td>14.33%</td>
<td>17.74%</td>
<td>17.74%</td>
<td>15.30%</td>
</tr>
<tr>
<td>Tai35b</td>
<td>13.78%</td>
<td>21.47%</td>
<td>21.47%</td>
<td>16.83%</td>
</tr>
<tr>
<td>Tai40b</td>
<td>11.13%</td>
<td>14.30%</td>
<td>14.30%</td>
<td>13.98%</td>
</tr>
<tr>
<td>Tai50b</td>
<td>13.61%</td>
<td>16.70%</td>
<td>16.70%</td>
<td>15.58%</td>
</tr>
<tr>
<td>Tai60b</td>
<td>10.18%</td>
<td>18.32%</td>
<td>18.32%</td>
<td>14.96%</td>
</tr>
<tr>
<td>Tai64c</td>
<td>60.92%</td>
<td>2.40%</td>
<td>2.40%</td>
<td>73.73%</td>
</tr>
<tr>
<td>Tho40</td>
<td>12.60%</td>
<td>13.20%</td>
<td>13.20%</td>
<td>13.53%</td>
</tr>
<tr>
<td>Wil50</td>
<td>3.92%</td>
<td>4.07%</td>
<td>4.07%</td>
<td>4.35%</td>
</tr>
</tbody>
</table>

problems were investigated. These explored properties not only help to select the matrix splitting scheme, but also lead to a more concise and effective implementation of the relaxation model.

A new SDR for QAPs based on the sum matrix and the minimal trace principle was derived. It was shown that in special cases, the new SDR can provide a stronger bound than the one from the one-matrix splitting. Numerical results also indicates that for most tested instances, the new SDR can provide stronger bounds than the those based on two other matrix splitting schemes.

On the other hand, we should point out that although in this chapter we have presented several ways to select a non-redundant PSD matrix splitting scheme to construct a strong SDR, it remains an open question on how to find the strongest SDR based on matrix splitting. Such a difficulty is possibly due to the multiplicity of the non-redundant splitting schemes. Even for the three selected splitting schemes, we could not find any dominance relationship among them. Further study is needed to address such an issue.
<table>
<thead>
<tr>
<th>problem</th>
<th>best solution</th>
<th>best lower bound based on</th>
<th>SDRMS-SUM $R_{gap}$ (%)</th>
<th>CPU</th>
<th>SDRMS-SVD $R_{gap}$ (%)</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>sko72</td>
<td>† 6.62560000 4</td>
<td>6.24190000 4</td>
<td><strong>5.79</strong></td>
<td>11:41</td>
<td>5.99</td>
<td>16:22</td>
</tr>
<tr>
<td>sko81</td>
<td>† 9.09980000 4</td>
<td>8.57920000 4</td>
<td><strong>5.72</strong></td>
<td>16:15</td>
<td>5.95</td>
<td>22:23</td>
</tr>
<tr>
<td>sko90</td>
<td>† 1.15534000 5</td>
<td>1.09251000 5</td>
<td><strong>5.44</strong></td>
<td>22:51</td>
<td>5.61</td>
<td>31:42</td>
</tr>
<tr>
<td>sko100a</td>
<td>† 1.52002000 5</td>
<td>1.44088000 5</td>
<td><strong>5.21</strong></td>
<td>31:09</td>
<td>5.39</td>
<td>42:34</td>
</tr>
<tr>
<td>sko100b</td>
<td>† 1.53890000 5</td>
<td>1.45645000 5</td>
<td><strong>5.48</strong></td>
<td>32:00</td>
<td>5.54</td>
<td>42:25</td>
</tr>
<tr>
<td>sko100c</td>
<td>† 1.47862000 5</td>
<td>1.40110000 5</td>
<td><strong>5.24</strong></td>
<td>13:52</td>
<td>5.53</td>
<td>42:34</td>
</tr>
<tr>
<td>sko100d</td>
<td>† 1.49576000 5</td>
<td>1.41513000 5</td>
<td><strong>5.39</strong></td>
<td>9:52</td>
<td>5.62</td>
<td>42:30</td>
</tr>
<tr>
<td>sko100e</td>
<td>† 1.49150000 5</td>
<td>1.41248000 5</td>
<td><strong>5.30</strong></td>
<td>12:39</td>
<td>5.57</td>
<td>40:55</td>
</tr>
<tr>
<td>sko100f</td>
<td>† 1.49036000 5</td>
<td>1.40848000 5</td>
<td><strong>5.49</strong></td>
<td>13:02</td>
<td>5.80</td>
<td>42:42</td>
</tr>
<tr>
<td>tai80b</td>
<td>† 8.18415043 8</td>
<td>7.23968001 8</td>
<td><strong>11.54</strong></td>
<td>17:55</td>
<td>16.26</td>
<td>25:36</td>
</tr>
<tr>
<td>tai100b</td>
<td>† 1.18599614 9</td>
<td>1.06455129 9</td>
<td><strong>10.24</strong></td>
<td>36:32</td>
<td>18.71</td>
<td>52:08</td>
</tr>
<tr>
<td>tai150b</td>
<td>† 4.98896643 8</td>
<td>4.42788590 8</td>
<td><strong>11.36</strong></td>
<td>2:44:55</td>
<td>11.71</td>
<td>2:32:29</td>
</tr>
<tr>
<td>tho150</td>
<td>† 8.13339800 6</td>
<td>7.61238100 6</td>
<td><strong>6.46</strong></td>
<td>32:59</td>
<td>6.69</td>
<td>2:26:48</td>
</tr>
<tr>
<td>wil100</td>
<td>† 2.73038000 5</td>
<td>2.64720000 5</td>
<td><strong>3.05</strong></td>
<td>30:44</td>
<td>3.14</td>
<td>42:48</td>
</tr>
<tr>
<td>tai256c</td>
<td>† 4.47592940 7</td>
<td>4.38491950 7</td>
<td><strong>73.55</strong></td>
<td>6:53:05</td>
<td><strong>2.03</strong></td>
<td>1:04:09</td>
</tr>
</tbody>
</table>

Table 2.5: Selected bounds for QAPs of large sizes ($n > 70$), computed using SDPNAL. The symbol (†) means that the best solution is only a feasible solution. The computation time is reported in the format of hours:minutes:seconds.
Table 2.6: New best known bounds for QAPLIB instances. In the table, “NA” means that the item is not available. Note that information for the last column of the table is obtained from [27].

<table>
<thead>
<tr>
<th>Instance</th>
<th>best feasible solution</th>
<th>best lower bound computed in this Chapter</th>
<th>$R_{\text{gap}}(%)$</th>
<th>CPU</th>
<th>previous best $R_{\text{gap}}(%)$</th>
<th>Source</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>sko90</td>
<td>† 1.15534000 5</td>
<td>1.09251000 5 (SUM)</td>
<td>5.44</td>
<td>22:51</td>
<td>6.10, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100a</td>
<td>† 1.52002000 5</td>
<td>1.44088000 5 (SUM)</td>
<td>5.21</td>
<td>31:09</td>
<td>6.14, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100b</td>
<td>† 1.53890000 5</td>
<td>1.45645000 5 (SUM)</td>
<td>5.48</td>
<td>32:00</td>
<td>6.51, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100c</td>
<td>† 1.47862000 5</td>
<td>1.40110000 5 (SUM)</td>
<td>5.24</td>
<td>13:52</td>
<td>5.73, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100d</td>
<td>† 1.49576000 5</td>
<td>1.41513000 5 (SUM)</td>
<td>5.39</td>
<td>9:52</td>
<td>6.47, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100e</td>
<td>† 1.49150000 5</td>
<td>1.41248000 5 (SUM)</td>
<td>5.30</td>
<td>12:39</td>
<td>6.07, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sko100f</td>
<td>† 1.49036000 5</td>
<td>1.40848000 5 (SUM)</td>
<td>5.49</td>
<td>13:02</td>
<td>6.43, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai35b</td>
<td>† 2.83315445 8</td>
<td>2.46180672 8 (SUM)</td>
<td>13.80</td>
<td>1:08</td>
<td>15.42, [25], 430914 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai40b</td>
<td>† 6.37250948 8</td>
<td>5.66070838 8 (SUM)</td>
<td>11.17</td>
<td>1:48</td>
<td>74.09, computed in 2005 [55], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai50b</td>
<td>† 4.58821517 8</td>
<td>3.98749304 8 (SUM)</td>
<td>13.67</td>
<td>3:30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai60b</td>
<td>† 6.08215054 8</td>
<td>5.45971917 8 (SUM)</td>
<td>10.23</td>
<td>7:01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai80b</td>
<td>† 8.18415043 8</td>
<td>7.23968001 8 (SUM)</td>
<td>11.54</td>
<td>17:55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai100b</td>
<td>† 1.18599614 9</td>
<td>1.06455129 9 (SUM)</td>
<td>10.24</td>
<td>36:32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai150b</td>
<td>† 4.98896643 8</td>
<td>4.42788590 8 (SUM)</td>
<td>11.36</td>
<td>2:44:55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tho150</td>
<td>† 8.13339800 6</td>
<td>7.61238100 6 (SUM)</td>
<td>6.46</td>
<td>32:59</td>
<td>6.32, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>will100</td>
<td>† 2.73038000 5</td>
<td>2.64720000 5 (SUM)</td>
<td>3.05</td>
<td>30:44</td>
<td>3.35, [62], NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tai256c</td>
<td>† 4.47592940 7</td>
<td>4.38491950 7 (SVD)</td>
<td>2.03</td>
<td>1:04:09</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 3

NONLINEAR SEMIDEFINITE RELAXATION OF WCLO

3.1 Introduction

Usually, optimization problem with uncertainty may remind the reader of stochastic programming [19] or robust optimization [9, 14] where decisions need to be made before uncertainty is observed. In practice, there exist cases when decisions can be made after uncertainty is observed. For example, the adjustable decisions (or “wait-and-see” decisions) in adjustable robust optimization [10] is one of such kind. In these cases, instead of finding optimal solution as in stochastic programming or robust optimization, we are interested in estimating the worst case of the optimal value. In this chapter, we consider the problem of estimating the worst-case linear optimization (WCLO) with uncertainties in the right-hand-side of the constraints defined by

\[
(WCLO) \max_{b \in \mathbf{U}} \min_{Ax \leq b} c^T x
\]

where \( \mathbf{U} \subseteq \mathbb{R}^l \) denote a single-ellipsoid uncertainty set defined below

\[
\mathbf{U} := \{ b = Qu + b_0 \mid \|u\|_2 \leq 1 \}.
\]

As we shall see later, such a problem arises naturally in the estimation of the systemic risk in finance [41], which can be modeled as WCLO where the right-hand side of the constraints is defined by the operating cash flow of a financial institute. Note that in practice, the operating cash flow is constantly changing and hard to
estimate precisely. An appropriate approach to deal with such a scenario is to use the WCLO model with uncertainties in the constraints. In addition to the above example, the WCLO with uncertainties also appears as a subproblem in stochastic optimization that has been widely used in a broad range of disciplines [13, 15, 16, 19]. The single-ellipsoid uncertainty set is a popular choice in modeling uncertainty and approximating general convex uncertainty set [12, 13].

Unfortunately, solving the WCLO is rather challenging as shown by the following proposition.

**Proposition 3.1.1.** *The WCLO problem is NP-hard.*

For self-completeness, we include its proof in Appendix B. One way to tackle the WCLO model is two-stage robust optimization [10], which can provide upper bounds to WCLO via reformulating it as an optimization problem over functional and restricting the feasible functional solution to be affine. The affine decision rules can be used for a large class of multistage robust optimization and the resulting problem is usually tractable conic program [10]. In [15, 16], Bertsimas and Goyal estimated the approximation rate of the solution obtained from the robust optimization model, and showed that good approximation bound can be obtained for several well-structured constraint/uncertainty sets. They also gave examples showing that the loss in optimality by using affine decision rule could be significant [15].

Different from the above-mentioned robust optimization approach, in this chapter we attempt to approximate the WCLO by means of semidefinite optimization (SDO). Our work is inspired by the following two observations. First, to some extent, the robust optimization approach to the WCLO can be interpreted as a tractable convex optimization approximation to the original hard problem. On the other hand, it is well-known that SDO is very successful in providing tight relaxations [66, 81, 83] and approximate solutions [51, 77, 100] for many NP-hard optimization problems.

To construct an SDO relaxations for the WCLO model, we first cast it as an equivalent optimization problem with linear and quadratic constraints via using the duality theory for LO. Then we relax the quadratic constraint to obtain the so-called coarse SDR for WCLO. By solving the coarse SDR, we can obtain both upper and
lower bounds for the original WCLO.

Note that the bounds from the coarse SDR might not be very tight. To obtain tighter bounds for WCLO, we consider the issue of how to utilize the solution from the current SDR to derive a stronger relaxation model. For such a purpose, we introduce an iterative procedure that can sequentially enhance the relaxation model and narrow the gap between the upper and lower bounds via changing slightly some parameters in the coarse SDR. We show that such an iterative procedure will lead to a series of SDRs that converges to a nonlinear SDO, which can provide much stronger upper and lower bounds to the original WCLO. Then we analyze the properties of the resulting nonlinear SDO and propose a bi-section search algorithm for it. Our preliminary experimental results illustrate that the solution from the nonlinear SDO can provide very tight bounds to the original WCLO and able to locate the globally optimal solutions for most tested instances.

The Chapter is organized as follows. In Section 3.2, we describe the coarse SDR for WCLO, and discuss the lower and upper bounds obtained from the coarse SDR. In Section 3.3, we first introduce an iterative procedure to enhance the coarse SDR and show that the resulting series of SDRs will converge to a nonlinear SDO. Then, we propose a bi-section search method for solving the nonlinear SDO. In Section 3.4, we give two application examples of the WCLO model: the worst case estimation of the systemic risk [41] and two-stage adaptive optimization [15]. Numerical results are reported in Section 3.5. We conclude the chapter with some remarks in Section 3.6.

3.2 The Coarse SDO Relaxation of WCLO

In this section, we first describe a coarse SDR for WCLO.

**Theorem 3.2.1.** For a given WCLO, let \((Y^*, y^*, y^*_i, t^*, s^*)\) be an optimal solution of
the following problem

\[
\begin{align*}
\max_{t, Y, y, t, s} & \quad t \\
\text{s.t.} & \quad \text{Tr} \left( \begin{bmatrix} Q Q^T - b_0 b_0^T & b_0 \\ b_0^T & -1 \end{bmatrix} \begin{bmatrix} Y & y_t \\ y_t^T & s \end{bmatrix} \right) \geq 0, \\
& \quad \begin{bmatrix} Y & y_t \\ y_t^T & s \end{bmatrix} - \begin{bmatrix} y_t \\ t \end{bmatrix} = 0, \\
& \quad A^T Y = c y^T, \quad Y \geq 0, \quad A^T y = c, \quad y \leq 0.
\end{align*}
\] (3.1)

Let \( u_1 = t^* \), \( l_1 = \|Q^T y^*\|_2 + b_0^T y^* \). Then we have

\[ l_1 \leq t^*_{\text{wclo}} \leq u_1. \]

Here \( t^*_{\text{wclo}} \) denotes the optimal value of WCLO.

**Proof.** Using the strong duality for LO, we have

\[
\begin{align*}
\max_{\|u\|_2 \leq 1} & \quad \min_{A x \leq Q u + b_0} c^T x \\
= & \quad \max_{\|u\|_2 \leq 1} \max_{A^T y = c, \ y \leq 0} u^T Q^T y + b_0^T y \\
= & \quad \max_{\|u\|_2 \leq 1} \max_{A^T y = c, \ y \leq 0} \|u\|_2 \|Q^T y\|_2 + b_0^T y \\
= & \quad \max_{A^T y = c, \ y \leq 0} \|Q^T y\|_2 + b_0^T y \\
\max & \quad t \\
\text{s.t.} & \quad \|Q^T y\| \geq t - b_0^T y \\
& \quad A^T y = c, \ y \leq 0
\end{align*}
\] (3.2)

\[
\begin{align*}
\max & \quad t \\
\text{s.t.} & \quad \|Q^T y\|^2 \geq (t - b_0^T y)^2 \\
& \quad A^T y = c, \ y \leq 0
\end{align*}
\] (3.3)
The last equality holds because the first constraint in (3.3) is equivalent to

$$-\|Q^T y\| \leq t - b_0^T y \leq \|Q^T y\|$$

and the first inequality in the above relation will never be active at the optimal solution.

By rewriting (3.3), we can reformulate WCLO as the following optimization problem with quadratic constraint

$$\max \quad t$$

$$\text{s.t.} \quad y^T (QQ^T - b_0 b_0^T) y + 2tb_0^T y - t^2 \geq 0; \quad (3.5)$$

$$A^T y = c, \quad y \leq 0. \quad (3.6)$$

Let us define

$$Y = yy^T, \quad y_t = ty, \quad s = t^2.$$  

For any feasible solution to the above problem, we have

$$\begin{bmatrix} Y & y_t \\ y_t^T & s \end{bmatrix} - \begin{bmatrix} y \\ t \end{bmatrix} \begin{bmatrix} y \\ t \end{bmatrix}^T \succeq 0, \quad A^T Y = c y^T, \quad Y \succeq 0.$$  

which further leads to the SDO relaxation (3.1). It follows immediately

$$t_{w clo}^* \leq u_1.$$  

Since \((Y^*, y^*, y_t^*, t^*, s^*)\) is an optimal solution to problem (3.1), \(y^*\) must be a feasible solution to problem (3.2). Therefore, we have

$$t_{w clo}^* \geq l_1 = \|Q^T y^*\|_2 + b_0^T y^*.$$  

This finishes the proof of the theorem.

We next estimate the gap between the lower and upper bounds derived from problem (3.1).

**Theorem 3.2.2.** Let \((Y^*, y^*, y_t^*, t^*, s^*)\) be the optimal solution to problem (3.1). We
have

\[ u_1 - l_1 \leq \sqrt{\text{Tr}(QQ^T(Y^* - y^*y^*T))}. \]  \hspace{1cm} (3.7)

Proof. From the first constraint in problem (3.1), we have

\[
0 \leq \text{Tr}
\left(
\begin{bmatrix}
QQ^T - b_0b_0^T & b_0 \\
-b_0^T & -1
\end{bmatrix}
\begin{bmatrix}
Y^* & y_t^* \\
y_t^T & s^*
\end{bmatrix}
\right)
= \text{Tr}(QQ^TY^*) - \text{Tr}
\left(
\begin{bmatrix}
b_0b_0^T & b_0 \\
-b_0^T & +1
\end{bmatrix}
\begin{bmatrix}
Y^* & y_t^* \\
y_t^T & s^*
\end{bmatrix}
\right)
\leq \text{Tr}(QQ^TY^*) - \text{Tr}
\left(
\begin{bmatrix}
b_0 & b_0 \\
-1 & -1
\end{bmatrix}
\begin{bmatrix}
y^* & y^* \\
t^* & t^*
\end{bmatrix}
\right)
= \text{Tr}(QQ^TY^*) - (t^* - b_0^Ty^*)^2.
\]

Thus, we have,

\[ u_1 = t^* \leq b_0^Ty^* + \sqrt{\text{Tr}(QQ^TY^*)}. \]

On the other hand, recall that \( l_1 = b_0^Ty^* + \|Q^Ty^*\|_2 \). It follows

\[
u_1 - l_1 \leq \sqrt{\text{Tr}(QQ^TY^*)} - \sqrt{\text{Tr}(QQ^Ty^*y^*T)}
\]
\[ \leq \sqrt{\text{Tr}(QQ^T(Y^* - y^*y^*T))}. \]

\[ \square \]

3.3 A Nonlinear SDO Relaxation for WCLO

In this section, we consider how to further improve the coarse SDR for WCLO introduced in the previous section. The section consists of two parts. In the first subsection, we introduce an iterative procedure to sequentially enhance the coarse
SDR, and establish the convergence of the SDR series in the iterative procedure. In the second subsection, we explore the theoretical properties of the nonlinear SDR resulted from the iterative procedure, and propose a bi-section search algorithm for it.

3.3.1 An iterative procedure to tighten the SDO relaxation

In this subsection, we propose an iterative procedure that can sequentially tighten the SDR for WCLO. Such a procedure is building upon the idea of constructing additional cuts for the SDR based on the solution from the current SDR. To see how this works, let us assume that we have an upper bound $\Delta$ for the WCLO obtained by solving the current SDR. For simplicity of discussion, we first consider the case where both $t_{wclo}^*$ and $\Delta$ are non-negative. In such a case, we have

$$0 \leq t_{wclo}^* \leq \Delta.$$ 

Therefore

$$s \leq \Delta^2$$

is a valid constraint for the relaxation of WCLO. We thus derive the following enhanced SDR for WCLO:

$$\max t$$

s.t. $\text{Tr} \left( \begin{bmatrix} QQ^T - b_0b_0^T & b_0 \\ b_0^T & -1 \end{bmatrix} \begin{bmatrix} Y & y_t \\ y_t^T & s \end{bmatrix} \right) \geq 0$

$$\begin{bmatrix} Y & y_t \\ y_t^T & s \end{bmatrix} - \begin{bmatrix} y \\ t \end{bmatrix} \begin{bmatrix} y \\ t \end{bmatrix}^T \geq 0$$

$$A^TY = cy^T, \ Y \geq 0, \ A^T y = c, \ y \leq 0$$

$$s \leq \Delta^2.$$
Let $t^*_\Delta$ denote the objective value at the optimal solution of the above problem. Then $t^*_\Delta$ will be a new upper bound for WCLO and thus we can resolve problem (3.9) with the updated new upper bound. This leads to Algorithm 1 that will sequentially tighten the SDO relaxations for WCLO.

**Algorithm 1**

A Iterative Procedure For tightening SDO relaxation

Let $\Delta_0 = u_1$, $\Delta_1 = t^*_{\Delta_0}$, $i = 1$

while $\Delta_i < \Delta_{i-1}$ do

- Solve problem (3.9) with parameter $\Delta = \Delta_i$
- Update $\Delta_{i+1} = t^*_{\Delta_i}$ and set $i = i + 1$

end while

Our next theorem establishes the convergence of the above procedure.

**Theorem 3.3.1.** Suppose that $t^*_{wclo} \geq 0$. Then the sequence $\{\Delta_i\}$ generated by Algorithm 1 converges.

**Proof.** For $i = 1$, since $\Delta_0 \geq t^*_{wclo}$, problem (3.9) is a relaxation of WCLO. Let $(t^* = t^*_{\Delta_0}, s^*)$ denote the optimal solution of problem (3.9) with $\Delta = \Delta_0$. Since

$$\Delta_1^2 = (t^*_{\Delta_0})^2 \leq s^* \leq \Delta_0^2$$

we have

$$t^*_{wclo} \leq \Delta_1 \leq \Delta_0.$$ 

Similarly for $k \geq 2$, we have

$$t^*_{wclo} \leq \Delta_k \leq \Delta_{k-1}.$$ 

The above discussion indicates that the sequence $\{\Delta_i\}$ is non-increasing and bounded from below by $t^*_{wclo}$. Thus $\{\Delta_i\}$ converges. \qed

We remark that though in Algorithm 1, we consider only the case $t^*_{wclo} \geq 0$, a similar procedure can be applied to the case $t^*_{wclo} < 0$. To see this, let us recall Theorem 3.2.1, we can obtain both a lower bound $l_1$ and an upper bound $\Delta_1$ for $t^*_{wclo}$
by solving the coarse SDR. Correspondingly, we have

\[ 0 \leq t^*_{\text{welo}} - l_1 \leq \Delta_1 - l_1, \]

which implies

\[ 0 \leq (t^*_{\text{welo}} - l_1)^2 \leq (\Delta_1 - l_1)^2. \]

Therefore, the following relation

\[ s - 2l_1 t^*_{\text{welo}} \leq \Delta_1^2 - 2l_1 \Delta_1, \]

will be a valid constraint for the SDR. By solving the refined SDR with the new constraint, we will obtain a new upper bound \( \Delta_2 \), which can be further used to construct a new constraint to enhance the relaxation model. Similarly, we can also prove the convergence of the sequence \( \{\Delta_i\} \).

We next characterize the limit point of the convergent sequence \( \{\Delta_i\} \).

**Proposition 3.3.2.** The accumulation point \( (\Delta^*) \) of the sequence \( \{\Delta_i\} \) is equivalent to the objective value at the optimal solution of the following nonlinear SDO

\[
\begin{align*}
\max & \quad b_0^T y + \sqrt{(b_0^T y)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y)} \\
\text{s.t.} & \quad A^T Y = c y^T, \quad Y \geq 0, \quad A^T y = c, \quad y \leq 0 \\
& \quad Y - yy^T \succeq 0.
\end{align*}
\]

\[ (3.10) \]

**Proof.** Since \( \Delta^* \) is the accumulation point of the sequence \( \{\Delta_i\} \), it follow from problem (3.9) that at its optimal solution, we have

\[ (\Delta^*)^2 = t^2 = s, \quad y_1 = y_1. \]

\[ (3.11) \]

Applying the above relations to the first constraint in problem (3.9) with \( \Delta = \Delta^* \),
we obtain
\[
\text{Tr}((QQ^T - b_0b_0^T)Y) + 2b_0^Tyt - t^2 \geq 0
\]
\[\Leftrightarrow (t - b_0^Ty)^2 \leq (b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y)
\]
\[\Leftrightarrow t \leq b_0^Ty + \sqrt{(b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y)}.
\]
This establishes the equivalence between the two problems (3.10) and (3.9) with \(\Delta = \Delta^*\). \(\square\)

We remark that another way to explore the relation between the coarse SDR (3.1) and the nonlinear SDR (3.10) is as follows. First we note that by using (3.2), we can rewrite problem (3.1) as the following problem

\[
\begin{align*}
\max & \quad b_0^Ty + t \\
\text{s.t.} & \quad t^2 \leq \text{Tr}(QQ^TY); \\
& \quad A^TY = cy^T, \quad Y \geq 0, \quad A^Ty = c, \quad y \leq 0, \\
& \quad Y - yy^T \succeq 0,
\end{align*}
\]

or equivalently the following nonlinear SDO

\[
\begin{align*}
\max & \quad b_0^Ty + \sqrt{\text{Tr}(QQ^TY)} \\
\text{s.t.} & \quad A^TY = cy^T, \quad Y \geq 0, \quad A^Ty = c, \quad y \leq 0, \\
& \quad Y - yy^T \succeq 0.
\end{align*}
\]

Now using the fact that
\[
\sqrt{(b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y)} \leq \sqrt{\text{Tr}(QQ^TY)},
\]
one can see that the bound provided by the nonlinear SDR (3.10) is usually tighter than that by the coarse SDR (3.1). However, the iterative procedure describes how to improve the coarse SDR step by step.

Our next result estimates the gap between the lower and upper bounds derived
from the nonlinear SDR (3.10).

**Theorem 3.3.3.** Let \( l_2, u_2 \) denote the lower and upper bounds for WCLO obtained by solving problem (3.10) and \((Y, y)\) be the optimal solution to problem (3.10). We have

\[
    u_2 - l_2 \leq \sqrt{\text{Tr}((QQ^T - b_0 b_0^T)(Y - yy^T))}. \tag{3.14}
\]

**Proof.** By Proposition 3.3.2,

\[
    u_2 - l_2 \leq \sqrt{(b_0^T y)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y)} - \sqrt{y^T QQ^T y}
    = \sqrt{\text{Tr}((QQ^T - b_0 b_0^T)(Y - yy^T)) + y^T QQ^T y} - \sqrt{y^T QQ^T y}
    \leq \sqrt{\text{Tr}((QQ^T - b_0 b_0^T)(Y - yy^T))}. \quad \square
\]

Comparing the estimates in Theorems 3.2.2 and 3.3.3, we can see that the bound in Theorem 3.3.3 is sharper.

### 3.3.2 A bi-section search algorithm

Though the iterative Procedure 1 can find the optimal solution to problem (3.10), it might converge very slowly in some cases. In this subsection, we propose a bi-section search procedure for solving problem (3.10).

For convenience, we first consider the following LO

\[
    \max \quad b_0^T y \tag{3.15}
    
    \text{s.t.} \quad A^T y = c, y \leq 0.
\]

Let \( \mu_{lo} \) denote the objective value at the optimal solution of the above problem. Now we are ready to state the main result in this subsection.

**Theorem 3.3.4.** Assume \( \Delta \geq \mu_{lo} \). Let \( \Delta^* \) denote the objective value at the optimal solution of problem (3.10) and \( \mu_{\Delta} \) denote the objective value at the optimal solution
of the following SDO

\[
\begin{align*}
\max \quad & \text{Tr}((QQ^T - b_0b_0^T)Y) + 2b_0^Ty \cdot \Delta - \Delta^2 \\
\text{s.t.} \quad & A^TY = cy^T, \quad Y \geq 0, \quad A^Ty = c, \quad y \leq 0; \\
& Y - yy^T \succeq 0.
\end{align*}
\] (3.16)

Then the following conclusions hold:

(i) \( \mu_\Delta > 0 \) if only if \( \Delta < \Delta^* \);

(ii) \( \mu_\Delta < 0 \) if only if \( \Delta > \Delta^* \);

(iii) \( \mu_\Delta = 0 \) if only if \( \Delta = \Delta^* \).

Proof. We first prove statement (i). Suppose that \((Y^*, y^*)\) be an optimal solution to problem (3.16). If

\[
\mu_\Delta = \text{Tr}((QQ^T - b_0b_0^T)Y^*) + 2b_0^Ty^* \cdot \Delta - \Delta^2 > 0,
\]

then it must hold

\[
\Delta < b_0^Ty^* + \sqrt{(b_0^Ty^*)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y^*)}.
\]

Because \((Y^*, y^*)\) is also feasible for problem (3.10), we thus have

\[
\Delta < \Delta^*.
\]

On the other hand, if \((Y^*, y^*)\) is the optimal solution to problem (3.10), then it holds

\[
\Delta < \Delta^* = b_0^Ty^* + \sqrt{(b_0^Ty^*)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y^*)}.
\] (3.17)

By the assumption in the theorem, we have \( \Delta - b_0^Ty^* \geq \Delta - \mu_{lo} \geq 0 \). It follows from inequality (3.17) that

\[
\mu_\Delta \geq \text{Tr}((QQ^T - b_0b_0^T)Y^*) + 2b_0^Ty^* \cdot \Delta - \Delta^2 > 0.
\]
This finishes the proof of statement (i).

We now proceed to prove statement (ii). Suppose that \( \Delta > \Delta^* \). Let \( (Y^*, y^*) \) be the optimal solution to problem (3.10), we have

\[
\Delta > \Delta^* = b_0^T y^* + \sqrt{(b_0^T y^*)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y^*)}. \tag{3.18}
\]

It follows immediately

\[
\mu_\Delta = \text{Tr}((QQ^T - b_0 b_0^T)Y^*) + 2b_0^T y^* \cdot \Delta - \Delta^2 < 0.
\]

On the other hand, if \( (Y^*, y^*) \) is the optimal solution to problem (3.10) and \( \mu_\Delta < 0 \), then we have

\[
(\Delta - b_0^T y^*)^2 > (b_0^T y^*)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y^*). \tag{3.19}
\]

Using the assumption in the theorem again, we have

\[
\Delta - b_0^T y^* \geq 0.
\]

Combining the above two relations, we obtain

\[
\Delta > b_0^T y^* + \sqrt{(b_0^T y^*)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y^*)} = \Delta^*.
\]

This finishes the proof of the second statement.

Statement (iii) follows directly from statements (i) and (ii). The proof of the theorem is completed. \(\square\)

Theorem 3.3.4 forms the basis of the bi-section search algorithm for problem (3.10) described as follows.

Note that according to S.3, the algorithm stops whenever \( u - l \leq \epsilon \), which further implies \( |\Delta_i - \Delta^*| \leq \epsilon \). In other words, the algorithm will stop if an approximate solution to problem (3.10) within tolerance is located. Since we use a bisection search procedure to reduce the interval \([l, u]\), one can easily prove the following result.
Algorithm 2 A Bisection Search Algorithm For Solving NLSDR

**Input:** $A, Q, b_0, c, N_{max}$ and the stop criteria $\epsilon$

**Output:** The lower and upper bound $(l, u)$

Solve problem (3.1) to get $l_1$ and $u_1$, and solve problem (3.15) to get $\mu_{lo}$

Let $l = \max(l_1, \mu_{lo}), u = u_1, \Delta = \frac{l_u + u_1}{2}, i = 0$

Evaluate $\mu_\Delta$ by solving (3.16); $i = i + 1$

while $|\mu_\Delta| \geq \epsilon \& i \leq N_{max}$ do
  if $\mu_\Delta > 0$ then
    $l = \Delta$
  else
    $u = \Delta$
  end if
  if The solution to (3.16) $Y^*$ has rank one and $\mu_\Delta < 0.1\Delta$ then
    $\Delta = b_0^T y^* + \|Q^T y^*\|$ where $y^*$ is the solution to (3.16)
  else
    $\Delta = \frac{l_u + u_1}{2}$
  end if
  Evaluate $\mu_\Delta$ by solving (3.16); $i = i + 1$
end while

return $(l, u)$
Theorem 3.3.5. The bisection search algorithm can find an approximation solution to problem (3.10) satisfying $|\Delta - \Delta^*| \leq \epsilon$ in at most $O(\log \frac{u_1 - l_1}{\epsilon})$ iterations.

3.4 Applications

In this section, we give two applications of solving WCLO.

3.4.1 Estimating worst-case systemic risk

Our first example is estimating the worst-case systemic risk. Given an interconnected system, systemic risk refers to the potential loss of the whole system as a result of the actions taken by its individual components. Generally speaking, systemic risk can be used to analyze the resilience of any complex system composed of interlinked components subject to external perturbations. For example, Eisenberg and Noe studied the systemic risk in financial system [41]; Crowther and Haimes [33] investigated the systemic risk of infrastructure system; Kambhu, Weidman and Krishnan [61] reviewed the systemic risk studied in ecology system, engineering system and epidemiology.

For convenience, we use the terminology in [41] in the remaining of this section. Consider an interbank market consisting of $n$ banks. The liability relationship between any two banks in the interbank market is described by a $n \times n$ liability matrix $L$ whose element $L_{ij}$ denotes the liability of bank $i$ to bank $j$ (e.g., the amount of money that bank $i$ borrowed from bank $j$). Let $c_i$ denote the exogenous operating cash flow received by bank $i$ which can be used to compensate the potential shortfall on incoming cash flow. Given $c$, a systemic loss of the financial system measuring the volume of failed liabilities (denoted by $l(c)$) is characterized by the following linear
In practice, the operating cash flow of a bank is usually constantly changing, thus uncertain. The systemic risk is the result of uncertain operating cash flow. A popular method to estimate the systemic risk is simulation (see e.g., [42, 43]). For example, 100,000 scenarios of the realized cash flows were drawn from empirical distribution to estimate the systemic risk of Australian banking system [42]. The worst-case systemic risk (denoted by $\rho_{WCSR}$) is defined as the objective value at the optimal solution of the following problem

$$\max_{u \in U} \min_x \sum_{i=1}^n (1 - x_i)$$

subject to

$$\left( \sum_j L_{ij} \right) x_i - \sum_j L_{ji} x_j \leq c_i \forall i = 1, \ldots, n$$

$$x_i \leq 1 \forall i = 1, \ldots, n.$$  

The above model has several advantages over the traditional simulation-based method: (i) Worst-case systemic risk is a robust measure that is independent of sampling process comparing with the simulation-based method; (ii) The worst scenario can be identified by solving problem (3.20) so that policies can be made to prevent those scenarios from happening; (iii) It has been empirically observed that contagion is rare in practice [42], but it is dangerous to conclude small contagion based on empirical studies. Worst-case systemic risk on the other hand essentially can test all possible scenarios and guarantees small contagion if worst-case systemic risk is relatively small. It is easy to see that problem (3.20) is a special case of WCLO, thus
the methods developed in previous sections can be applied. We shall report our experimental results for problem (3.20) in Section 3.5.2.

3.4.2 Adjustable robust optimization

The second example arise from adjustable robust optimization which deals with the situation when decision maker has the option to adjust its decisions after the uncertainty is revealed. It has recently become a popular topic in the optimization community because of its difficulty and important applications (see e.g., [10, 15, 16]). In this section, we will show that WCLO can be used to evaluate the objective of adjustable robust optimization given the non-adjustable solution (or first-stage solution) and even solve the adjustable robust optimization.

Consider the following optimization problem,

\[
\min_x c^T x + \max_{b \in U} \min_y d^T y \\
\text{s.t. } Ax + By \leq b.
\]  

(3.21)

Given the non-adjustable decision \( x \), evaluating the objective is equivalent to solve the following problem,

\[
\mu_{ARC}(x) = c^T x + \max_{b \in U} \min_y d^T y \\
\text{s.t. } By \leq b - Ax,
\]  

(3.22)

which is a special case of WCLO. Consequently, problem (3.22) can be reformulated equivalently as the following problem

\[
\min_x \mu_{ARC}(x).
\]  

(3.23)

One can further show that the function \( \mu_{ARC}(x) \) is convex in \( x \). Therefore, the WCLO model and algorithms for it will play a critical role in some algorithm that solves problem (3.23) directly.
3.5 Numerical results

In this section, we report our experimental results comparing the bounds obtained by the following three relaxation models:

(\textbf{SDR}): The coarse SDO relaxation (3.1);

(\textbf{NLSDR}): The nonlinear SDO relaxation (3.10);

(\textbf{AFFINE}): Affine-rule approximation (see Appendix B).

In our numerical experiments, we noted that the SDP solver occasionally fail to solve problem (3.1) even when it is indeed feasible and bounded. We also note that if we know the optimal value of WCLO is non-negative (or negative) in advance, we can further add the following constraints to problem (3.1),

\begin{equation}
    y_t \leq 0 \text{(or } y_t < 0)\,, \quad A^T y_t = tc. \tag{3.24}
\end{equation}

We find that these extra constraints can help to resolve the above-mentioned numerical issue.

In our experiments, all the problems were solved in Matlab R2009b on a 3.33GHz Intel Core 2 Duo with 8GB memory. The SDP subproblems are automatically generated by CVX 1.2 [53] and solved by the SDP solver SDPT3 [97].

3.5.1 Randomly generated examples

We first test random WCLO instances of the following form,

\begin{equation*}
    \max_{\|u\|_2 \leq 1} \quad \min_x \quad c^T x \\
    \text{s.t.} \quad Ax \leq Qu + b_0, \quad x \geq 0
\end{equation*}

where \((A, Q, b_0, c) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times k} \times \mathbb{R}^m \times \mathbb{R}^n\). Specifically, elements of \(A, b_0\) are drawn from \(U(-5, 5)\) (i.e., uniformly distributed within interval \([-5, 5]\)), elements of \(Q\) are drawn from \(U(-2, 2)\) and elements of \(c\) are drawn from \(U(0, 1)\). 300 WCLO
instances with $m = 8, n = 20, l = 5$ and 300 with $m = 10, n = 30, k = 10$ are generated. The non-negativity constraint on $x$ in the above problem and the choice of non-negative vector $c$ make the randomly generated problem feasible and bounded in most cases.

Three sets of instances: (a) 300 WCLO instances with $m = 8, n = 20, k = 5$; (b) 300 WCLO instances with integral data and $m = 8, n = 20, k = 5$; (c) 300 with $m = 10, n = 30, k = 10$, are generated and all the three models (SDR, NLSDR & AFFINE) are applied to obtain bounds. The lower and upper bounds obtained from these three models are used to compute the gap defined by

$$\text{gap} := \frac{\text{upperbound} - \text{lowerbound}}{\text{lowerbound}} \cdot 100\%.$$ 

The distribution of the gap $P(x)$ is given by

$$P(x) := \frac{\# \text{ of instances with gap less than } x}{300} \cdot 100\%,$$

as plotted in all the figures in this section.

Figure 3.1 shows that NLSDR can solve most of the WCLO instances to global optimality and solve all the instances within a very small gap. As one can see from the figure, there is no significant difference in performance between fractional data and integral data. Another interesting observation is that the gap of affine-rule approximation tends to increase as the size of the problem increases, which is consistent with the theoretical work discussed in [15, 16].

To further illustrate the efficacy of the NLSDR model, in Table 3.1 we also report the minimum, average and maximum number $^1$ of iterations that the bisection search algorithm used to solve problem (3.10). The statistics in Table 3.1 has excluded the WCLO instances that SDR relaxation is exact since in such a case, the NLSDR model will not be solved at all. Table 3.1 shows the bisection search algorithm converges in a few iterations on average.

$^1$In our experiments, the maximum number of iterations is set to be 30.
### Table 3.1: Number of iterations to solve NLSDR model

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>Mean</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) WCLO with n = 20</td>
<td>2</td>
<td>3.7</td>
<td>19</td>
</tr>
<tr>
<td>(b) WCLO with n = 20 (integral)</td>
<td>2</td>
<td>5.0</td>
<td>30</td>
</tr>
<tr>
<td>(c) WCLO with n = 30</td>
<td>2</td>
<td>8.0</td>
<td>30</td>
</tr>
</tbody>
</table>

#### 3.5.2 Estimating the worst-case systemic risk

We next test all the three models on problem (3.20) with randomly generated data \((L, Q, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^n\). Specifically, the off-diagonal elements of \(L\) are drawn from \(LN(0.4, 1)\) (i.e., lognormal distribution with mean 0.4 and standard deviation 1), elements of \(Q\) are drawn from \(U(-2, 2)\) and elements of \(c\) are drawn from \(U(0.5, 5)\). 300 instances of problem (3.20) with \(m = 5, n = 20\) and 300 with \(m = 10, n = 30\) are generated and solved. The distributions of gaps are plotted in Figure 3.2.

Two sets of instances: (a) 300 instances of problem (3.20) with \(m = 5, n = 20\) and (b) 300 with \(m = 10, n = 30\), are generated and all the three models are applied to obtain bounds. The distributions of gaps are plotted in Figure 3.2.

As one can see from Figure 3.2, the NLSDR can solve all the instances to global optimality, while AFFINE has slightly better performance than SDR. Table 3.2 illustrates that on average, the bisection search algorithm converges in just a few iterations.

### Table 3.2: Number of iterations to solve NLSDR model

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>Mean</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) n = 20</td>
<td>2</td>
<td>2.8</td>
<td>30</td>
</tr>
<tr>
<td>(b) n = 30</td>
<td>2</td>
<td>3.3</td>
<td>30</td>
</tr>
</tbody>
</table>
3.6 Conclusions

In this Chapter, we considered the WCLO problem arising from numerous important applications and showed that it is NP-hard. Then we presented a coarse SDO relaxation to WCLO and observed that the coarse model can be further tighten by using the solution from the current SDR. An iterative procedure was proposed to sequentially enhance the relaxation model and it was shown that the sequence of the SDRs converges to a nonlinear SDO, which can provide stronger bounds than the coarse SDR. By exploring the properties of the nonlinear SDO, we developed a bisection search algorithm and established the global convergence of the algorithm. The complexity of the algorithm was estimated as well. Preliminary experimental results illustrated that the new nonlinear SDO and the proposed algorithm can help to find very tight bounds for the original WCLO and able to locate the global optimal solutions for most tested instances.

There are several issues left open. First of all, is it possible (and how) to estimate the quality of the bounds provided from the nonlinear SDO relaxation? A deeper investigation in this direction will help to build up the foundation of the nonlinear SDO relaxation. Secondly and more importantly, can we develop similar nonlinear SDO relaxation for other hard optimization problems that is stronger than the standard SDO relaxation and how to solve the resulting nonlinear SDO effectively? More study is needed to address these issues.
(a) Random WCLO with $m = 8$, $n = 20$, $k = 5$

(b) Random WCLO with integral data and $m = 8$, $n = 20$, $k = 5$

(c) Random WCLO with $m = 10$, $n = 30$, $k = 10$

Figure 3.1: Distribution of gaps obtained by applying SDR, NLSDR and AFFINE to randomly generated WCLO problems
Figure 3.2: Distribution of gaps obtained by applying SDR, NLSDR and AFFINE to randomly generated problems of (3.20)
CHAPTER 4
CONDITIONALLY QUASI-CONVEX SEMIDEFINITE RELAXATIONS

4.1 Introduction

Convex relaxation has become one of the main techniques to estimate bounds and/or obtain near-optimal solutions for NP-hard problems [30, 4, 82, 13, 51, 77]. Getting a strong convex relaxation is of both theoretical and practical interests [5, 3]. One way to strengthen convex relaxations is to add tractable constraints either by exploiting the structure of the original problems [74, 82] or by some rather-general scheme such as RLT [91] and triangle inequalities [24, 99]. Another approach is to gradually lift the dimension of decision variables to construct a sequence of relaxations that converges to global optimal of the original problem as the times of lifting goes to infinity [66]. In the second approach, the number of variables grows exponentially. In parallel of these method, we showed in Chapter 3 that by augmenting the objective with a nonconvex penalty term one can also strengthen the convex relaxation. The promising numerical results of the nonlinear relaxation model for WCLO problem motivate us to extend the result to more general scenario.

In this chapter, a new class of relaxation models called conditionally quasi-convex relaxation (CQCR) model is introduced. The CQCR model involves a parameter $\alpha$ (denoted by $\text{CQCR}(\alpha)$) that can be freely chosen from $[0, +\infty)$. The $\text{CQCR}(\alpha)$ model with sufficiently large $\alpha$ is tractable by a bisection procedure similar to Algorithm 2. However, we also show that the bound obtained from $\text{CQCR}(\alpha)$ is non-decreasing as $\alpha$ decreases. Therefore, $\text{CQCR}(0)$ is a strongest among all $\text{CQCR}(\alpha)$ models. Applying the bisection procedure to $\text{CQCR}(0)$ only allow us to obtain a
trivial lower bound of CQCR(0) which is equal to the bound obtained from linear semidefinite relaxation model. Alternatively, we propose an iterative procedure to approximately solve CQCR(0) and another bisection procedure to solve CQCR(0) under some assumption.

To facilitate the discussion, we first focus on the Quadratically Constrained Linear Program (QCLP) of the following form,

\[
\begin{align*}
\min_{x} & \quad c^T x \\
\text{s.t.} & \quad x^T P_i x + q_i^T x \leq r_i, \quad i = 1, \ldots, m \\
& \quad Ax \leq b, \quad x \geq 0
\end{align*}
\]

where \(P_i\) is indefinite in general. QCLP itself has broad applications in production planning [57], economics [89, 73] and design optimization [6, 94]. Besides, we will discuss applying conditionally quasi-convex relaxation to other NP-hard problems including Integer Linear Optimization, Quadratic Knapsack Problem and Box-constrained Nonconvex Quadratic Programming later in this Chapter.

In the remaining of this Chapter, we first give the definition of conditional quasi-convexity in Section 4.2. In Section 4.3, we introduce the conditionally quasi-convex relaxation (QCQR) model for QCLP and algorithms to solve it. Preliminary numerical results on three NP-hard problems including Integer Linear Optimization (ILP), Quadratic Knapsack Problem (QKP), Box-constrained Nonconvex Quadratic Programming (BoxQP) are presented in Section 4.5.

### 4.2 Conditionally Quasi-convex Function

Quasi-convex function, as the name indicated, possesses certain convexity property (i.e., sublevel sets are convex) so that optimizing quasi-convex function can be polynomial-time solvable [45, 44]. We first recall the definition of quasi-convexity as follows,

**Definition 4.2.1.** [21] A function \(f(x)\) is called quasi-convex (or quasi-concave) if
its domain $\mathcal{X}$ and its sublevel sets

\[ S_\beta = \{ x \in \mathcal{X} | f(x) \leq \beta \} \quad (\text{or } S_\beta = \{ x \in \mathcal{X} | f(x) \geq \beta \}) \]

are convex for all $\beta \in \mathbb{R}$.

However, the quasi-convexity requires all sublevel sets to be convex which might be too restrictive to satisfy in reality. For example, the objective function in NLSDR model introduced in Chapter 3.3.2 does not satisfy the quasi-convexity condition, but we utilized a property similar to quasi-convexity to solve NLSDR in the bisection algorithm. Next we introduce a new notion of conditional quasi-convexity, which is a relaxed version of quasi-convexity.

**Definition 4.2.2.** A function $f(x)$ is called conditionally quasi-convex (or conditionally quasi-concave) for some $\beta_0$ if its domain $\mathcal{X}$ and its sublevel sets

\[ S_\beta = \{ x \in \mathcal{X} | f(x) \leq \beta \} \quad (\text{or } S_\beta = \{ x \in \mathcal{X} | f(x) \geq \beta \}) \]

are convex for all $\beta \leq \beta_0$ (or $\beta \geq \beta_0$).

Clearly, any quasi-convex function is conditionally quasi-convex for any $\beta_0 \in \mathbb{R}$. Any conditionally quasi-convex function with $\beta_0 = +\infty$ is quasi-convex. As an example, we first show that the objective function of the NLSDR model introduced in Chapter 3 is conditionally quasi-convex.

**Proposition 4.2.3.** Given $b_0, Q$ and convex set $\mathcal{X}$, the function

\[ f(y, Y) = b_0^T y + \sqrt{(b_0^T y)^2 + \text{Tr}((QQ^T - b_0 b_0^T)Y)}, \quad (y, Y) \in \mathcal{X} \]

is conditionally quasi-concave with $\beta_0 = \max_{(y,Y) \in \mathcal{X}} b_0^T y$. 
Proof. For any $\beta \geq \beta_0$ and $(y,Y) \in X$, we have

\[
f(y,Y) \geq \beta \iff b_0^Ty + \sqrt{(b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y)} \geq \beta
\]

\[
\iff \sqrt{(b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y)} \geq \beta - b_0^Ty \geq 0
\]

\[
\iff (b_0^Ty)^2 + \text{Tr}((QQ^T - b_0b_0^T)Y) \geq (b_0^Ty - \beta)^2
\]

\[
\iff \text{Tr}((QQ^T - b_0b_0^T)Y) - 2b_0^Ty + \beta_0^2 \geq 0
\]

is convex in $(y,Y)$. The second equivalence is because

\[
\beta \geq \beta_0 \geq b_0^Ty, \forall (y,Y) \in X.
\]

This proves the proposition. \qed

4.3 Conditionally Quasi-convex Semidefinite Relaxation of QCLP

4.3.1 Conditionally Quasi-convex Semidefinite Relaxation of QCLP

First, we recall the semidefinite relaxation of QCLP,

\[
\min_{x,X} c^Tx \\
\text{s.t.} \quad (x,X) \in \mathbb{X}
\]

(4.2)

where

\[
\mathbb{X} = \left\{ (x,X) \mid \begin{array}{l}
X - xx^T \succeq 0 \\
\text{Tr}(P_iX) + q_i^Tx \leq r_i, \quad i = 1, \ldots, m \\
Ax \leq b, \quad AX \leq bx^T, \quad x \geq 0, \quad X \geq 0
\end{array} \right\}
\]

67
and consider a stronger relaxation of QCLP of the following form,

$$
(CQCR(\alpha)) \min_{x, X} \quad c^T x + \alpha - \sqrt{\alpha^2 + c^T (xx^T - X)c}
$$

s.t. $$(x, X) \in X$$

for some $\alpha \geq 0$. Let $u_0$ and $u_\alpha$ denote the optimal objective of problems (4.2) and CQCR($\alpha$). Note that

$$
\alpha - \sqrt{\alpha^2 + c^T (xx^T - X)c} \geq 0, \quad \forall (x, X) \in X
$$

we have $u_0 \leq u_\alpha$. Moreover, the equality holds if and only if

$$
c^T (xx^T - X)c = 0. \quad (4.3)
$$

Equation (4.3) does not hold for the solution of (4.2) in general, thus problem CQCR($\alpha$) can be considered as the linear semidefinite relaxation with objective augmented by a penalty function. We next show that the objective function of CQCR($\alpha$) is conditionally quasi-convex and can be solved using a bisection algorithm.

**Proposition 4.3.1.** Given $(c, \alpha)$ and convex set $X$, function

$$
f(x, X) = c^T x + \alpha - \sqrt{\alpha^2 + c^T (xx^T - X)c}, \quad (x, X) \in X
$$

is conditionally quasi-convex for $\beta_0 = \min_{(x, X) \in X} c^T x + \alpha$.

**Theorem 4.3.2.** Let $\beta \leq \beta_0 = \min_{(x, X) \in X} c^T x + \alpha$ and $u_\alpha$ denotes the optimal value of CQCR($\alpha$). Let $\mu_\beta$ denote the objective value at the optimal solution of the following SDP

$$
\begin{align*}
\min \quad & c^T Xc + 2(\alpha - \beta) \cdot c^T x + \beta^2 - 2\alpha \beta \\
\text{s.t.} \quad & (x, X) \in X
\end{align*}
$$

(4.4)

Then the following conclusions hold:
(i) \( \mu_\beta < 0 \) if only if \( \beta > u_\alpha \);

(ii) \( \mu_\beta > 0 \) if only if \( \beta < u_\alpha \);

(iii) \( \mu_\beta = 0 \) if only if \( \beta = u_\alpha \).

Proof. We first prove statement (i). If \( \mu_\beta < 0 \), let \((x^*, X^*)\) denote the solution to problem (4.4). We have,

\[
c^T X^* c + 2(\alpha - \beta) \cdot c^T x^* + \beta^2 - 2\alpha\beta < 0
\Rightarrow (c^T x^* + \alpha - \beta)^2 < \alpha^2 + (c^T x^*)^2 - c^T X^* c.
\]

(4.5)

On the other hand, \( \beta \leq \min_{x \in X} c^T x + \alpha \), we know

\[
c^T x + \alpha - \beta \geq 0 \quad \forall x \in X
\Rightarrow c^T x^* + \alpha - \beta \geq 0.
\]

(4.6)

Thus, (4.5) and (4.6) imply

\[
c^T x^* + \alpha - \beta < \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c}
\Rightarrow \beta > c^T x^* + \alpha - \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c} \geq u_\alpha.
\]

If \( \beta > u_\alpha \), then there exists \((x^*, X^*) \in X\) such that

\[
\beta > c^T x^* + \alpha - \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c},
\Rightarrow \mu_\beta \leq c^T X^* c + 2(\alpha - \beta) \cdot c^T x^* + \beta^2 - 2\alpha\beta < 0.
\]

This proves statement (i).

We now proceed to prove statement (ii). If \( \mu_\beta > 0 \). Suppose \( \beta \geq u_\alpha \), then there exists \((x^*, X^*) \in X\) such that

\[
\beta \geq c^T x^* + \alpha - \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c},
\Rightarrow \mu_\beta \leq c^T X^* c + 2(\alpha - \beta) \cdot c^T x^* + \beta^2 - 2\alpha\beta \leq 0.
\]

69
Contradiction. Thus, $\mu_\beta > 0$ implies $\beta < u_\alpha$.

If $\beta < u_\alpha$. Suppose $\mu_\beta \leq 0$ and let $(x^*, X^*)$ denote the solution to problem (4.4). We have,

$$c^T X^* c + 2(\alpha - \beta) \cdot c^T x^* + \beta^2 - 2\alpha\beta \leq 0$$
$$\Rightarrow (c^T x^* + \alpha - \beta)^2 \leq \alpha^2 + (c^T x^*)^2 - c^T X^* c. \quad (4.7)$$

On the other hand, $\beta \leq \min_{x \in X} c^T x + \alpha$, we know

$$c^T x + \alpha - \beta \geq 0 \ \forall x \in X$$
$$\Rightarrow c^T x^* + \alpha - \beta \geq 0. \quad (4.8)$$

Thus, (4.7) and (4.8) imply

$$c^T x^* + \alpha - \beta \leq \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c}$$
$$\Rightarrow \beta \geq c^T x^* + \alpha - \sqrt{\alpha^2 + (c^T x^*)^2 - c^T X^* c} \geq u_\alpha.$$

Contradiction. Thus, $\beta < u_\alpha$ implies $\mu_\beta > 0$. This proves statement (ii).

(iii) is the direct result of (i) and (ii). This completes the proof. \qed

Theorem 4.3.2 forms the basis of the following Bisection algorithm for solving CQCR($\alpha$) given some nonnegative $\alpha$.

4.3.2 Choosing $\alpha$

So far, we haven’t discussed how to choose the parameter $\alpha$ in the conditionally quasi-convex model CQCR($\alpha$) yet. In this section, we show that small $\alpha$ is preferred for the sake of obtaining strong relaxation.

**Proposition 4.3.3.** Let $u(\alpha)$ ($\alpha \geq 0$) denote the optimal value of CQCR($\alpha$), then $u(\alpha)$ is a non-decreasing as $\alpha$ decreases.

**Proof.** For $\alpha_2 \geq \alpha_1 \geq 0$, let $(x^*, X^*)$ denote the optimal solution corresponding to
Algorithm 3 A Bisection Search Algorithm For Solving CQCR(\(\alpha\))

**Input:** QCLP with data \((c, P_i, q_i, r_i, A, b), i = 1, 2, ..., m, N_{max}, \epsilon\) and \(\alpha\)

**Output:** The lower bound \(l\)

Solve problem (4.2) and let \(l\) denote the optimal value

Let \(l = l, u = l + \alpha, \beta = \frac{l + u}{2}, i = 0\)

Evaluate \(\mu_{\beta}\) by solving (4.4); \(i = i + 1\)

while \(|\mu_{\beta}| \geq \epsilon & u - l \geq \epsilon & i \leq N_{max}\) do

if \(\mu_{\beta} > 0\) then

\(l = \beta\)

else

\(u = \beta\)

end if

\(\beta = \frac{l + u}{2}\)

Evaluate \(\mu_{\beta}\) by solving (4.4); \(i = i + 1\)

end while

return \(l\)

\[u(\alpha_1).\] Then \((x^*, X^*)\) is also feasible for CQCR with \(\alpha = \alpha_2\). It is easy to see

\[u(\alpha_1) \geq \alpha_2 - \sqrt{\alpha_2^2 + (c^T x^*)^2} - c^T X^* c \geq u(\alpha_2)\]

is true by using the following technical inequality

\[b - \sqrt{b^2 - c^2} \geq a - \sqrt{a^2 - c^2}, \forall a \geq b \geq c \geq 0.\]

Due to Proposition 4.3.3, ideally we would like to choose \(\alpha = 0\). Unfortunately, applying Algorithm 3 to CQCR(0) only allows us to find a trivial lower bound \(\min_{x \in \mathbb{R}} c^T x\). In the next two sections, we propose an iterative procedure and a bisection procedure to solve CQCR(0). The iterative procedure can solve CQCR(0) in general but the running time is unknown, while the bisection procedure can solve CQCR(0) under some assumption.

\[\square\]
4.3.3 An Iterative Procedure to Solve CQCR(0)

Given a lower bound of CQCR(0), \( L \), consider the following auxiliary problem,

\[
\mu(L) = \min_{(x,X) \in X} c^T X c - 2Lc^T x + L^2 \tag{4.9}
\]

s.t. \((x,X) \in X\)

**Theorem 4.3.4.** Assume the QCLP is feasible. Let \( L_0 = \min_{x \in X} c^T x \), \( L_{i+1} = L_i + \sqrt{\mu(L_i)} \), then

\[
L_i \to u(0)
\]

where \( u(0) \) denote the optimal value of CQCR(0).

**Proof.** We first show that the sequence \( \{L_i\} \) is well defined and upper bounded by \( u(0) \). We prove by induction.

For \( i = 1 \), problem (4.9) with \( L = L_0 \) is feasible and

\[
\mu(L_0) = \min_{(x,X) \in X} c^T X c - 2L_0c^T x + L_0^2 \\
= \min_{(x,X) \in X} c^T (X - xx^T)c + (c^T x - L_0)^2 \geq 0.
\]

Thus, \( L_1 = L_0 + \sqrt{\mu(L_0)} \) is well defined. Moreover, let \((x^*, X^*) \in X\) be the optimal solution to CQCR(0), we have

\[
\mu(L_0) \leq c^T (X^* - xx^T)c + (c^T x^* - L_0)^2 = (u(0) - L_0)^2 \tag{4.10}
\]

As \( \mu(L_0) \geq 0 \) and \( u(0) \geq L_0 \), (4.10) implies

\[
L_1 = L_0 + \sqrt{\mu(L_0)} \leq u(0).
\]

Suppose \( L_i \) is well defined and \( L_i \leq u(0) \). It’s easy to see \( \mu(L_i) \geq 0 \), thus \( L_{i+1} \) is well defined. Moreover, let \((x^*, X^*) \in X\) be the optimal solution to CQCR(0), we
have
\[ \mu(L_i) \leq c^T(X^* - x^*x^T)c + (c^Tx^* - L_i)^2 = (u(0) - L_i)^2 \] (4.11)

As \( \mu(L_i) \geq 0 \) and \( u(0) \geq L_i \), (4.11) implies
\[ L_{i+1} = L_i + \sqrt{\mu(L_i)} \leq u(0). \]

This proves the sequence \( \{L_i\} \) is well defined and upper bounded by \( u(0) \). Actually \( \{L_i\} \) is non-decreasing and upper bounded, thus \( \{L_i\} \) converges. Let \( L^* \) denote the limit of \( \{L_i\} \). \( L^* \) satisfies
\[
\begin{align*}
\mu(L^*) &= 0 \\
\Rightarrow & \exists (x, X) \in \mathbb{X}, \ c^T(X - xx^T)c + (c^T x - L^*)^2 = 0 \\
\Rightarrow & \exists (x, X) \in \mathbb{X}, \ L^* = c^T x - \sqrt{c^T(xx^T - X)c} \\
\Rightarrow & L^* \geq u(0)
\end{align*}
\]

but \( L^* \leq u(0) \). Thus
\[ L^* = u(0). \]

This completes the proof.

In summary, the iterative procedure
\[ L_0 = \min_{x \in \mathbb{X}} c^T x, \ L_{i+1} = L_i + \sqrt{\mu(L_i)} \] (4.12)
can generate a sequence that converges to the solution of CQCR(0).

4.3.4 A Bisection Procedure to Solve CQCR(0)

The iterative procedure (4.12) can solve CQCR(0) in general, but the running time is unknown. In this subsection, we give another bisection search procedure which can solve CQCR(0) satisfying some condition.
Theorem 4.3.5. Assume the following set

\[ \{(x, X) \mid c^T(xx^T - X)c = 0\} \cap X \]

is connected. Let \( \Delta^* \) denotes the optimal value of \( CQCR(0) \) and let \( \mu_\Delta \) denote the optimal value of the following SDP

\[
\mu_\Delta = \min \ c^T Xc - 2c^T x \cdot \Delta + \Delta^2 \tag{4.13}
\]

s.t. \((x, X) \in X\).

If \( \Delta \leq \bar{\Delta} \) where \( \bar{\Delta} = c^T x \) for some \( x \) feasible for the original QCLP, then the following conclusions hold:

(i) \( \mu_\Delta = 0 \) if only if \( \Delta \geq \Delta^* \);

(ii) \( \mu_\Delta > 0 \) if only if \( \Delta < \Delta^* \).

Proof. We first prove statement (i). If \( \mu_\Delta = 0 \), then let \((x^*, X^*) \in X\) denote the optimal solution of problem (4.13) we have

\[
c^T X^* c - 2c^T x^* \cdot \Delta + \Delta^2 = 0.
\]

Thus,

\[
0 \leq (c^T x^* - \Delta)^2 = c^T (x^* x^*^T - X^*)c \geq 0.
\]

The only way that the above inequality holds is when

\[
c^T x^* = \Delta, \ c^T (x^* x^*^T - X^*)c = 0.
\]

Thus, \((x^*, X^*)\) is also feasible for \( CQCR(0) \) and we have

\[
\Delta^* \leq c^T x^* - \sqrt{c^T (x^* x^*^T - X^*)c} = \Delta.
\]

If \( \Delta \geq \Delta^* \), then by the assumption and the intermediate value theorem there must...
exist \((x^*, X^*) \in X\) such that
\[
\Delta = c^T x^* - \sqrt{c^T (x^* x^{*T} - X^*) c}
\]
which implies
\[
c^T X^* c - 2c^T x^* \cdot \Delta + \Delta^2 = 0.
\]
Thus, \(X^*, x^*\) is an optimal solution to problem (3.16) since \(\mu_\Delta \geq 0\). Therefore, \(\mu_\Delta = 0\).

As \(\mu_\Delta \geq 0\), (ii) is a direct result of (i).

Theorem 4.3.5 forms the basis for the following bisection algorithm.

**Algorithm 4.** A Bisection Search Algorithm For Solving CQCR(0)

**Input:** QCLP with data \((c, P_i, q_i, r_i, A, b), i = 1, 2, \ldots, m, N_{max}\) and \(\epsilon\)

**Output:** The lower bound \(l\)

Solve any relaxation of QCLP and let \(l_0\) denote the lower bound

Find a feasible solution of QCLP and let \(u_0\) denote the objective value

Let \(l = l_0, u = u_0, \Delta = \frac{l + u}{2}, i = 0\)

Evaluate \(\mu_\Delta\) by solving (4.13); \(i = i + 1\)

while \(u - l > \epsilon \& i \leq N_{max}\) do

if \(\mu_\Delta > \epsilon\) then

\(l = \Delta\)

else (\(\mu_\Delta \leq \epsilon\))

\(u = \Delta\)

end if

\(\Delta = \frac{l + u}{2}\)

Evaluate \(\mu_\Delta\) by solving (4.4); \(i = i + 1\)

end while

return \(l\)

**Theorem 4.3.6.** Algorithm 4 can find an approximate solution to CQCR(0) with objective value \(\Delta\) which satisfies \(|\Delta - \Delta^*| \leq \epsilon\) in at most \(O(\log \frac{u_0-l_0}{\epsilon})\) iterations.
4.4 Conditionally Quasi-convex Semidefinite Relaxations of WCLP

In this section, we first show a relationship between the conditionally quasi-convex relaxation and the nonlinear relaxation model (3.10) introduced in Chapter 3. Recall that the WCLP can be reformulated as the following nonconvex problem,

$$\min_{A^T y = c, \ y \leq 0} \ -b_0^T y - \|Q^T y\|_2$$

whose semidefinite relaxation can be written as

$$\min_{(y,Y) \in \mathcal{Y}} \ -b_0^T y - \sqrt{\text{Tr}(QQ^T Y)}$$

where

$$\mathcal{Y} = \left\{ (y,Y) \mid \begin{array}{l} Y - yy^T \succeq 0 \\ A^T Y = cy^T, \ Y \succeq 0 \\ A^T y = c, \ y \leq 0 \end{array} \right\}.$$ 

By augmenting the objective with the 'penalty' $$\alpha - \sqrt{\alpha^2 + b_0^T (yy^T - Y)b_0}$$, we obtain a stronger relaxation

$$\min_{(y,Y) \in \mathcal{Y}} \ -b_0^T y - \sqrt{\text{Tr}(QQ^T Y)} + \alpha - \sqrt{\alpha^2 + b_0^T (yy^T - Y)b_0}.$$ (4.15)

Although small $$\alpha$$ is preferred as we discussed in Section 4.3.2, it turns out that if we choose

$$\alpha = \sqrt{\text{Tr}(QQ^T Y)}$$

problem (4.15) is tractable and equivalent to the nonlinear semidefinite relaxation model (3.10).

The particular choice of $$\alpha$$ motivates us to obtain a conditionally quasi-convex semidefinite relaxation model by choosing $$\alpha = 0$$. To do that, we first rewrite (4.14)
as follows,

\[
\begin{aligned}
& \min \quad -b_0^T y + t \\
& \text{s.t.} \quad A^T Y = cy, \quad Y \geq 0, \quad A^T y = c, \quad y \leq 0 \\
& \quad \text{Tr}(QQ^T Y) = T, \quad \left[ \begin{array}{c} T \\ y_t \end{array} \right] - \left[ \begin{array}{c} t \\ y \end{array} \right] \left[ \begin{array}{c} t \\ y \end{array} \right]^T \succeq 0.
\end{aligned}
\]

The conditionally quasi-convex relaxation is then

\[
\begin{aligned}
& \min \quad -b_0^T y + t - \left[ \begin{array}{c} 1 \\ -b_0 \end{array} \right]^T \left( \begin{array}{c} t \\ y \end{array} \right) - \left[ \begin{array}{c} T \\ y_t \end{array} \right] - \left[ \begin{array}{c} 1 \\ -b_0 \end{array} \right] \\
& \text{s.t.} \quad A^T Y = cy, \quad Y \geq 0, \quad A^T y = c, \quad y \leq 0 \\
& \quad \text{Tr}(QQ^T Y) = T, \quad \left[ \begin{array}{c} T \\ y_t \end{array} \right] - \left[ \begin{array}{c} t \\ y \end{array} \right] \left[ \begin{array}{c} t \\ y \end{array} \right]^T \succeq 0.
\end{aligned}
\]

The iterative procedure (4.12) and Algorithm 4 can be used to solve problem (4.17). We remark that problem (4.17) is not equivalent to (4.15), thus we don’t know if (4.17) is always better than (4.15) although numerical experiments (see Section 4.5) shows (4.17) is preferred.

### 4.5 Preliminary Numerical Results

In this section, we evaluate the numerical performance of the CQCR(0) model and effectiveness of iterative procedure (4.12) and Algorithm 4. We first apply the CQCR(0) model to a Nonlinear Knapsack Problem. We next apply the CQCR(0) model to 0-1 Integer Linear Programming (ILP) problems as the 0-1 ILP problems is essentially QCLP since we can write the integer constraint \( x \in \{0, 1\} \) as \( x^2 - x = 0 \). We also apply the CQCR(0) model to Box-constrained Nonconvex Quadratic Programming problems. The bounds obtained by CQCR(0) model is compared with
other classic relaxation models. The gap is computed using the following formula

\[ \text{gap} = \frac{\text{OPT} - \text{LowerBound}}{\text{OPT}} \]  

(4.18)

where \( \text{OPT} \) denotes the global optimal or a valid upper bound, \( \text{LowerBound} \) denotes the lower bound found by the relaxation models.

The distributions of the gap plotted in this section are plots of function \( P(x) \) given by

\[ P(x) := \frac{\# \text{ of instances with gap less than } x}{200} \cdot 100\%. \]

4.5.1 Numerical results for Nonlinear Knapsack Problem

In this section, we consider the Nonlinear Knapsack Problem [22] of the following particular form

\[ \min w^T x \]

\[ \text{s.t. } x^T P x \geq q, \ x \in \{0, 1\}^n \]

where \( P \in S^n, \ w \in \mathbb{R}^n \) and \( q \in \mathbb{R} \) are given parameters. Because of the quadratic constraint, problem (4.19) is called Quadratically Constrained Knapsack Problem (QCKP). The QCKP has applications in location selection of freight handling terminals [88], portfolio selection problem [88] and the max clique problem [47]. Moreover, the QCKP can also be used to solve the Quadratic Knapsack Problem [84] of the following form

\[ \text{(QKP)} \quad \max x^T P x \]

\[ \text{s.t. } w^T x \leq c, \ x \in \{0, 1\}^n \]

using a binary search with \( q \) in \([0, \sum_{ij} P_{ij}]\).

In the numerical experiments, we consider the following two relaxations of QCKP
problems.

\[
\text{(SDR-QCKP)} \quad \min \ w^T x \\
\text{s.t.} \quad \text{Tr}(PX) \geq q, \ \text{diag}(X) = x \\
x \in [0, 1]^n, \ X \in [0, 1]^n.
\]

\[
\text{(CQCR-QCKP)} \quad \min \ w^T x \\
\text{s.t.} \quad w^T (xx^T - X)w = 0 \\
\text{Tr}(PX) \geq q, \ \text{diag}(X) = x \\
x \in [0, 1]^n, \ X \in [0, 1]^n.
\]

We compare the following four lower bounds.

- SDR - bound obtained by directly solving (SDR-QCKP);
- ITER-1 - bound obtained by applying iterative procedure (4.12) to (CQCR-QCKP) with 1 iteration;
- ITER-5 - bound obtained by applying iterative procedure (4.12) to (CQCR-QCKP) with 5 iteration;
- CQCR - bound obtained by applying Algorithm 4 to (CQCR-QCKP) assuming the connectivity condition in Theorem 4.3.5 is satisfied.

200 QCKP instances are randomly generated. Specifically, the elements of \( w, P \) are drawn from uniform distribution \( U[0, 1] \) and \( q = 0.7r \cdot e^T Pe \) where \( r \) is drawn from \( U[0, 1] \). The OPT in (4.18) is an upper bound found by TOMLAB/MINLP [58, 48]. The distributions of gaps are plotted in Figure (4.1). From Theorem 4.3.4 and Algorithm 4, we know the iterative procedure (4.12) always output a lower bound of the actual optimal value of CQCR(0) while the bisection procedure 4 output a upper bound up to an \( \epsilon \) precision. Thus, the fact that curves of ITER-1, ITER-5 and CQCR are almost identical means (i) the iterative procedure is very effective and
can reach very close to the exact optimal of CQCR(0) after only a few iterations; (ii) the connectivity assumption may be a reasonable assumption in practice. Moreover, the margin improved by the CQCR(0) is significant.

Figure 4.1: Distributions of gaps obtained by solving the SDR-QCKP and CQCR-QCKP of 200 randomly generated QCKP
4.5.2 Numerical results for Integer Programming

Consider the following general 0-1 Integer Programming Problems (ILPs),

\[(ILP) \quad \min \quad c^T x \]
\[\text{s.t.} \quad Ax \leq b, \ x \in \{0,1\}^n\]

We consider the following two relaxation models,

\[(SDR-ILP) \quad \min \quad c^T x \]
\[\text{s.t.} \quad Ax \leq b, \ x \in [0,1]^n, \ X \in [0,1]^n, \ \text{diag}(X) = x, \ AX \leq bx^T, \ X - xx^T \succeq 0 \]
\[Axe^T - AX \leq be^T0 - bx^T \]
\[bb^T - Axb^T - bx^TA^T + AXA^T \succeq 0 \]
\[ex^T \succeq X, \ ee^T - xe^T - ex^T + X \equiv 0.\]

\[(CQCR-ILP) \quad \min \quad c^T x \]
\[\text{s.t.} \quad c^T(xx^T - X)c = 0 \]
\[Ax \leq b, \ x \in [0,1]^n, \ X \in [0,1]^n \]
\[\text{diag}(X) = x, \ AX \leq bx^T, \ X - xx^T \succeq 0 \]
\[Axe^T - AX \leq be^T0 - bx^T \]
\[bb^T - Axb^T - bx^TA^T + AXA^T \succeq 0 \]
\[ex^T \succeq X, \ ee^T - xe^T - ex^T + X \equiv 0.\]

We compare the following four lower bounds.

- SDR - bound obtained by directly solving (SDR-ILP);
- ITER-1 - bound obtained by applying iterative procedure (4.12) to (CQCR-ILP) with 1 iteration;
• ITER-5 - bound obtained by applying iterative procedure (4.12) to (CQCR-ILP) with 5 iteration;

• CQCR - bound obtained by applying Algorithm 4 to (CQCR-ILP) assuming the connectivity condition in Theorem 4.3.5 is satisfied.

200 ILP instances are randomly generated. Specifically, the elements of \( A, b \) are drawn from \( U[-5, 5] \)\(^1\) and the elements of \( c \) are drawn from \( U[0, 1] \). The \( OPT \) in (4.18) is the global optimal value of the ILP found by the bintprog() function in MATLAB. The distribution of gap for each of relaxation models is plotted in Figure (4.2). We can see that the margin improved by the CQCR is significant. Moreover, the ITER-5 matches CQCR very well which means (i) the iterative procedure is very effective and can reach very close to the exact optimal of CQCR(0) after only a few iterations; (ii) the connectivity assumption is a reasonable assumption in practice.

4.5.3 Numerical results for Box-constrained nonconvex QP

Consider the following Box-constrained Quadratic Programming (BoxQP)

\[
\text{(BoxQP)} \quad \min \quad x^T Q x + c^T x \\
\text{s.t.} \quad x \in [0, 1]^n
\]

We consider the following two relaxation models

\[
\text{(SDR2-BoxQP)} \quad \min \quad \text{Tr}(QX) + c^T x \\
\text{s.t.} \quad x \in [0, 1]^n, \quad X \in [0, 1]^n \\
\begin{bmatrix}
Y & Y_x \\
Y_x^T & X
\end{bmatrix} - \begin{bmatrix}
\text{vec}(X) \\
x
\end{bmatrix} \begin{bmatrix}
\text{vec}(X) \\
x
\end{bmatrix}^T \succeq 0 \\
\begin{bmatrix}
\text{mat}(Y_{x_i}) & X_i \\
X_i^T & x_i
\end{bmatrix} \succeq 0, \quad i = 1, \ldots, n \\
\begin{bmatrix}
X & x \\
x^T & 1
\end{bmatrix} - \begin{bmatrix}
\text{mat}(Y_{x_i}) & X_i \\
X_i^T & x_i
\end{bmatrix} \succeq 0, \quad i = 1, \ldots, n
\]

\(^1\)ILP problem randomly generated in this way may be infeasible, such problems are dropped.
(a) Integer Optimization problems with 20 variables and 10 constraints

(b) Integer Optimization problems with 30 variables and 10 constraints

Figure 4.2: Distributions of gaps obtained by solving the SDR-ILP and CQCR-ILP of 200 randomly generated ILP problems

\[ \text{(CQCR-BoxQP)} \quad \min \quad \text{Tr}(QX) + c^T x \]
\[ \text{s.t.} \quad \begin{bmatrix} \text{vec}(Q) \\ c \end{bmatrix} = 0 \]
\[ x \in [0,1]^n, \quad X \in [0,1]^n \]
\[ \begin{bmatrix} Y & Yz \\ Y^T & X \end{bmatrix} - \begin{bmatrix} \text{vec}(X) \\ x \end{bmatrix} \begin{bmatrix} \text{vec}(X) \\ x \end{bmatrix}^T \succeq 0 \]
\[ \begin{bmatrix} \text{mat}(Y_{zi}) \\ X_i \\ X_i^T \end{bmatrix} \begin{bmatrix} X_i \\ x_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, n \]
\[ \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} - \begin{bmatrix} \text{mat}(Y_{zi}) \\ X_i \\ X_i^T \end{bmatrix} \begin{bmatrix} X_i \\ x_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, n \]
We compare the following four lower bounds.

- **SDR** - bound obtained by directly solving (SDR2-BoxQP);
- **ITER-1** - bound obtained by applying iterative procedure (4.12) to (CQCR-BoxQP) with 1 iteration;
- **ITER-5** - bound obtained by applying iterative procedure (4.12) to (CQCR-BoxQP) with 5 iteration;
- **CQCR** - bound obtained by applying Algorithm 4 to (CQCR-BoxQP) assuming the connectivity condition in Theorem 4.3.5 is satisfied.

200 ILP problems are randomly generated. The $OPT$ in (4.18) is an upper bound found by KNITRO/TOMLAB [58, 28]. The distributions of gaps are plotted in Figure (4.3). We can see that we still have a significant improvement even though the SDR2 model is already very tight (i.e, the average gap is small). Moreover, the ITER-1 and ITER-5 match CQCR very well which means (i) the iterative procedure is very effective and can reach very close to the exact optimal of CQCR(0) after only a few iterations; (ii) the connectivity assumption is a reasonable assumption in practice.

### 4.6 Conclusions

In this Chapter, we introduced a new class of relaxation models called the conditionally quasi-convex relaxation model. The CQCR model can be seen as an linear semidefinite relaxation model augmented by a special kind of penalty function. The CQCR model has a parameter $\alpha$ to be determined. We showed that the CQCR model with $\alpha = 0$ (denoted by CQCR(0)) is the strongest CQCR model. However, the CQCR(0) is not readily tractable. We proposed an iterative procedure to approximately solve CQCR(0) and a bisection procedure to solve CQCR(0) under some assumption. The numerical experiments showed the iterative procedure is very effective and can approximately solve CQCR(0) in only a few steps. Moreover,
(a) Box-constrained (nonconvex) Quadratic Programming problems with 8 variables

(b) Box-constrained (nonconvex) Quadratic Programming problems with 10 variables

Figure 4.3: Distributions of gaps obtained by solving the SDR2-BoxQP and CQCR-BoxQP of 200 randomly generated BoxQP problems

the numerical experiments demonstrated the CQCR(0) model outperforms classic relaxation models.

There are several issues left open. Firstly, does there exist other forms of conditionally quasi-convex relaxation besides the one introduced in this chapter? Secondly, all our CQCR(0) models are based on QCLP, can we derive similar techniques for the more general Quadratically Constrained Quadratic Programming (QCQP) problems?
CHAPTER 5

CONCLUSIONS AND FUTURE REMARKS

This thesis introduced and studied several novel relaxation models for NP-hard problems. These relaxation models can be considered as the extensions of classic linear semidefinite relaxation model. We showed that the classic semidefinite relaxation models can be strengthened by (i) using the parameters from an optimal subset; (ii) constructing CQCR model by augmenting the objective with a special type of penalty function.

In particular, We first studied the semidefinite relaxation of Quadratic Assignment Problem (QAP) based on matrix splitting. We characterized an optimal subset of all valid matrix splittings and propose the minimal trace principle to find them by solving a tractable auxiliary problem. A new matrix splitting scheme called sum-matrix splitting is also proposed and its numerical performance is evaluated. The bounds obtained by using sum-matrix splitting have replaced several best-known bounds reported in QAPLIB [27].

We next considered the so-called Worst-case Linear Optimization (WCLO) problem which has applications in systemic risk estimation and stochastic optimization. We showed that WCLO is NP-hard and a coarse linear semidefinite relaxation is firstly presented. An iterative procedure is introduced to sequentially refine the the coarse relaxation model and we showed that the sequence of the gradually refined models actually converge to a nonlinear semidefinite relaxation (NLSDR) model. We then proposed a bisection algorithm to solve the NLSDR in polynomial time. Our preliminary numerical results showed that the NLSDR can provide very tight bounds for WCLO even the exact global solution.

Inspired by the NLSDR model of WCLO, we introduced a new class of relaxation
called conditionally quasi-convex relaxation (CQCR). General CQCR model has an undetermined nonnegative parameter $\alpha$ and the CQCR model with $\alpha = 0$ (denoted by CQCR(0)) is proved to be the strongest of all CQCR models. We next propose an iterative procedure to approximately solve CQCR(0) and a bisection procedure to solve CQCR(0) under some assumption. Preliminary numerical experiments illustrate the proposed algorithms are effective and the CQCR(0) model outperforms classic relaxation models.

5.1 Future Directions of Research

There are several possible directions for future research.

- For the SDP relaxation of QAP using matrix splitting, we are only able to characterize a optimal subset of parameters. It still remains an open question on how to find the strongest SDR based on matrix splitting given the specific QAP instance.

- The CQCR model of QCLP with the square-root penalty function is a specific instance of CQCR model. The conditional quasi-convexity definition is rather general. It will be interesting to find other useful CQCR models.

- The CQCR model introduced in Chapter 4 is mainly used for QCLP. Can we apply the CQCR model to QCQP in an efficient way without multiple lifting?
APPENDIX A

PROOF OF THEOREM 2.3.2

We first cite a well-known result for matrix product from [96] that will be used in the proof of Theorem 2.3.2.

**Lemma A.0.1.** Let $A, B \in S^n$ with the eigenvalues $\lambda_i(A)$ and $\lambda_i(B)$, $i = 1, \ldots, n$ listed in nonincreasing order. Then

$$\text{Tr}(AB) \leq \sum_{i=1}^{n} \lambda_i(A)\lambda_i(B)$$

where the equality holds if and only if there is an orthogonal matrix $P$ whose columns form a common set of eigenvectors for $A$ and $B$ and are ordered with respect to $\{\lambda_i(A)\}_{i=1}^{n}$ and $\{\lambda_i(B)\}_{i=1}^{n}$, such that $P^{-1}AP$ and $P^{-1}BP$ are diagonal.

Now we are ready to prove Theorem 2.3.2. Since $(\alpha, \beta)$ is an optimal solution of problem (2.25), there exists $U \in S^n$ such that

$$n - \text{Tr}(UE) = 0, \quad (A.1)$$
$$n - \text{Tr}(U) = 0, \quad (A.2)$$
$$\text{Tr}(U(\alpha E + \beta I - B)) = 0, \quad (A.3)$$
$$U \succeq 0, \quad \alpha E + \beta I - B \succeq 0. \quad (A.4)$$

From (A.1)-(A.3), we obtain directly

$$n(\alpha + \beta) = \text{Tr}(UB). \quad (A.5)$$
From (A.3) and (A.4), we follow that

\[ U(\alpha E + \beta I - B) = (\alpha E + \beta I - B)U = 0. \]  

(A.6)

This implies that \( U \) and \( \alpha E + \beta I - B \) can commute. By Theorem 1.3.12 in [59], \( U \) and \( \alpha E + \beta I - B \) are simultaneously diagonalizable. Since \( U \in S^n \), \( \alpha E + \beta I - B \in S^n \), there is an orthogonal matrix \( P \) such that \( P^{-1}UP \) and \( P^{-1}(\alpha E + \beta I - B)P \) are diagonal. So, we have

\[
\text{Tr}(U(\alpha E + \beta I - B)) = \sum_{i=1}^{n} \lambda_i(U)\lambda_i(\alpha E + \beta I - B) = 0
\]

which in turn by (A.4) implies that

\[
\lambda_i(U)\lambda_i(\alpha E + \beta I - B) = 0, \ i = 1, \ldots, n.
\]  

(A.7)

Due to the minimal trace principle, we have \( m = \text{Rank}(\alpha E + \beta I - B) < n \). Since \( \alpha E + \beta I - B \succeq 0 \), we assume that \( \lambda_i(\alpha E + \beta I - B) > 0 \), \( i = 1, \ldots, m \). The above equality (A.7) then yields \( \lambda_i(U) = 0 \), \( i = 1, \ldots, m \).

We now prove \( UE \neq EU \). Suppose to the contrary that \( UE = EU \). By Theorem 1.3.12 in [59], \( U \) and \( E \) are simultaneously diagonalizeable. Let

\[ \lambda_1(U) = \ldots = \lambda_s(U) = 0 < \lambda_{m+1}(U) \leq \ldots \leq \lambda_n(U). \]  

(A.8)

Note that the eigenvalues of \( E \) are \( 0, \ldots, 0, n \). Therefore, we have

\[ \text{Tr}(UE) = n\lambda_n(U), \]

which by (A.1) implies \( \lambda_n(U) = 1 \). Hence, we infer from (A.8) that

\[ \text{Tr}(U) = \sum_{i=1}^{n} \lambda_i(U) \leq n - m < n, \]
this contradicts (A.2).

Because $UE \neq EU$, from (A.6) we obtain $UB \neq BU$. Since $U \in S^n$ and $B \in S^n$, by Theorem 1.3.12 in [59], $U$ and $B$ are not simultaneously diagonalizable. Now using Lemma A.0.1, we have

$$\text{Tr}(UB) < \sum_{i=1}^{n} \lambda_i(U) \lambda_i(B),$$

which, together with (A.5), yields

$$n(\alpha + \beta) < \sum_{i=1}^{n} \lambda_i(U) \lambda_i(B). \quad (A.9)$$

Let $\lambda_{\text{max}}(B)$ be the largest eigenvalue of $B$. Note that $\sum_{i=1}^{n} \lambda_i(U) = \text{Tr}(U) = n$. Also, $\lambda_i(U) \geq 0$ for all $i$ since $U \succeq 0$. It then follows from (A.9) that

$$\alpha + \beta < \lambda_{\text{max}}(B). \quad (A.10)$$

On the other hand, from (A.4), we have

$$B - \alpha(E - I) - (\alpha + \beta)I \preceq 0.$$ 

This means that

$$\alpha + \beta \geq \lambda_{\text{max}}(B - \alpha(E - I)). \quad (A.11)$$

If $\alpha = 0$, then the combination of (A.10) and (A.11) leads to a contradiction.

If $\alpha < 0$. Let $\rho(B)$ be the spectral radius of $B$. Since $B \in S^n$ is non-negative, by Theorem 8.3.1 in [59], then $\rho(B)$ is an eigenvalue of $B$ and there exists nontrivial $\hat{x} \geq 0 \in \mathbb{R}^n$ such that $B\hat{x} = \rho(B)\hat{x}$. Without loss of generality, we can further assume that $\|\hat{x}\|_2 = 1$. Thus we have $\hat{x}^T B\hat{x} = \rho(B)$. Since $\hat{x} \succeq 0$, it holds $\hat{x}^T (E - I)\hat{x} \succeq 0$. 

90
It follows from (A.11) that

\[ \alpha + \beta \geq \lambda_{\text{max}}(B - \alpha(E - I)) \]

\[ = \max \{ x^T(B - \alpha(E - I))x : x^Tx = 1 \} \]

\[ \geq \hat{x}^T(B - \alpha(E - I))\hat{x} \]

\[ \geq \hat{x}^TB\hat{x} = \rho(B) \]

\[ \geq \lambda_{\text{max}}(B), \]

which contradicts to (A.10). Therefore, we can conclude \( \alpha > 0 \). This finishes the proof of the theorem. \( \square \)
APPENDIX B

PROOF OF PROPOSITION 3.1.1

Using the strong duality of linear program, we have

\[
\max_{\|u\|_2 \leq 1} \min_{Ax \leq Qu + b_0} c^T x \quad (B.1)
\]

\[
= \max_{\|u\|_2 \leq 1} \max_{A^T y = c, \ y \leq 0} u^T Q^T y + b_0^T y \quad (B.2)
\]

\[
= \max_{\|u\|_2 \leq 1} \max_{A^T y = c, \ y \leq 0} \|u\|_2 \|Q^T y\|_2 + b_0^T y \quad (B.3)
\]

\[
= \max_{A^T y = c, \ y \leq 0} \|Q^T y\|_2 + b_0^T y. \quad (B.4)
\]

With carefully chosen \((Q, b_0, A)\), the above optimization problem includes the following problem as a special case,

\[
\max_x \|x\|_2 \quad (B.5)
\]

\[
\text{s.t.} \quad Bx \geq d \quad (B.6)
\]

which has been shown to be NP-hard by a reduction from the NP-complete partition problem \([49, 72]\).
A common tool to approximately solve multilevel optimization including WCLO is using the affine-rule approximation (see e.g., [9, 10, 15, 14, 17, 31]). More specifically, for WCLO, we have

\[
\begin{align*}
\max_{\|u\|_2 \leq 1} \min_{Ax \leq Qu + b_0} c^T x \\
= \min_{Ax(u) \leq Qu + b_0, \forall \|u\|_2 \leq 1} \max_{\|u\|_2 \leq 1} c^T x(u)
\end{align*}
\]

(C.1)

since we can always choose

\[x(u) = \arg \min_{Ax \leq Qu + b_0} c^T x.\]

To make problem (C.1) tractable, we can artificially restrict the decision function \(x(\cdot)\) to be affine in uncertainties, i.e.,

\[x(u) = Pu + q.\]
Under such a circumstance, problem (C.1) reduces to

\[
\begin{align*}
\min_{P, q, t} \quad & c^T(Pu + q) \\
\text{s.t.} \quad & A(Pu + q) \leq Qu + b_0, \quad \forall \|u\|_2 \leq 1 \\
& c^T(Pu + q) \leq t, \quad \forall \|u\|_2 \leq 1
\end{align*}
\]

which is a tractable second-order conic optimization problem.
REFERENCES


[62] S. E. Karisch and F. Rendl. Lower bounds for the quadratic assignment prob-

[63] S. Kim and M. Kojima. Exact solutions of some nonconvex quadratic opti-
mization problems via sdp and socp relaxations. *Computational Optimization

[64] M. Kocvara, M. Stingl, and P. GbR. Pensdp users guide (version 2.2). *PENOP-
T GbR*, 2006.

[65] T. Koopmans and M. Beckmann. Assignment problems and the location of


cally constrained quadratic programs. *Mathematical programming*, 103(2):251–
282, 2005.

[69] E. Loiola, N. de Abreu, P. Boaventura-Netto, P. Hahn, and T. Querido. A

[70] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 opti-


[72] O. Mangasarian and T. Shiau. A variable-complexity norm maximization prob-


