Semidefinite Programming for Assignment and Partitioning Problems

by

Qing Zhao

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

Waterloo, Ontario, Canada, 1996

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Semidefinite programming, SDP, is an extension of linear programming, LP, where the nonnegativity constraints are replaced by positive semidefiniteness constraints on matrix variables. SDP has proven successful in obtaining tight relaxations for NP-hard combinatorial optimization problems of simple structure such as the max-cut and graph bisection problems. In this work, we try to solve more complicated combinatorial problems such as the quadratic assignment, general graph partitioning and set partitioning problems.

A tight SDP relaxation can be obtained by exploiting the geometrical structure of the convex hull of the feasible points of the original combinatorial problem. The analysis of the structure enables us to find the so-called "minimal face" and "gangster operator" of the SDP. This plays a significant role in simplifying the problem and enables us to derive a unified SDP relaxation for the three different problems. We develop an efficient "partial infeasible" primal-dual interior-point algorithm by using a conjugate gradient method and by taking advantage of the special data structure of our relaxation. Numerical tests show that the approximations given by our approach are of high quality.

Future work for solving a large sparse problem with our approach is also discussed for each of the applications. In particular, for a large sparse set partitioning problem, we propose an approach combining a mixed LP-SDP relaxation with matrix decomposition techniques.

Acknowledgements

I would like to thank my supervisor, Professor Henry Wolkowicz, for his excellent guidance and support throughout my Ph.D. program. I am also grateful to my co-supervisor, Professor Anthony Vannelli, for his support and advice during my Ph.D studies. My thanks go to the members of my examining committee for their helpful comments and suggestions regarding my thesis work.

I would like to acknowledge the financial support during my Ph.D studies provided by the Department of Combinatorics and Optimization at University of Waterloo.

I would like to thank my college and office-mate, Serge Kruk, for his valuable assistance.

For my memorable visit to the Technical University of Graz, I am grateful to Professor Franz Rendl and Dr. Stefan Karisch for inviting me and making my visit very productive and enjoyable.

Finally, I would like to thank my wife, Jiali Tao, for her encouragement and love throughout my graduate studies.

In memory of my mother Qiu, Liying

Contents

1 Introduction and Notation **1**

1.1 Introduction 1

1.2 Notation 4

2 Semidefinite Programming **7**

2.1 Introduction 7

2.2 Duality Theory and Interior Point Methods 8

2.3 Geometrical Structure of SDP 15

2.4 SDP Relaxation for Quadratically
Constrained Quadratic Programming 18

3 Quadratic Assignment Problem **22**

3.1 Introduction 22

3.2 An SDP Relaxation 27

3.3 Geometry of the Feasible Set 33

3.4 Gangster Operator and
Final SDP Relaxation 40

3.5 A Truncated Primal-infeasible Dual-feasible Interior-Point Method . 49

3.5.1 Why this Method? 50

3.5.2 The Preconditioned (Truncated) Conjugate Gradient Method 53

3.5.3 Implementation 55

3.6 Numerical Tests and Comments 56

3.6.1 Goal of the Numerical Tests 57

3.6.2 Measures of Performance 57

3.7 Future Work 60

4 Graph Partitioning Problem 61

4.1 Introduction 61

4.2 An SDP Relaxation 63

4.3 Geometry of the Feasible Set 65

4.4 Final SDP Relaxation 70

4.5 Numerical Tests 77

4.6 Conclusion 78

5 Set Partitioning Problems 83

5.1 Introduction 83

5.2 An SDP Relaxation 85

5.3 Numerical Testing for Small Problems 92

5.4 SDP Relaxation for Large Sparse Problems and Future Work 94

5.4.1 An SDP Relaxation with Block Structure 94

5.4.2 An Infeasible Primal-Dual Interior-Point Method 101

5.4.3 Preliminary Numerical Tests and Future Work 104

6 Summary and Discussion **107**

Bibliography **109**

Chapter 1

Introduction and Notation

1.1 Introduction

Semidefinite programming, SDP, is an extension of linear programming, LP, where the nonnegativity constraints are replaced by positive semidefiniteness constraints on matrix variables. It possesses almost as simple and almost as nice a structure as LP does. SDP not only can approximate more problems than LP does but also can provide better approximations. Moreover, through SDP, a lot of powerful tools developed in continuous optimization, such as interior-point methods, can be applied to tackle many hard discrete optimization problems.

SDP has recently been active in many mathematical and engineering research areas such as control, min-max eigenvalue problems and combinatorial optimization problems, see e.g. Alizadeh [ALI95, ALI92] and Vandenberghe and Boyd [VB96]. In those research activities, SDP has already shown its potential as a very powerful tool. In particular, SDP has proven successful in obtaining tight relaxations for NP-hard combinatorial optimization problems of simple structure such as the max-cut and graph bisection problems. These relaxations can be obtained from their corresponding quadratically constrained quadratic programming formulation through their Lagrangian dual, see e.g. [PRW95]. However, for a hard combinatorial problem with sophisticated structure, how to find a tight SDP relaxation is still an open question. In this work, we try to present a unified SDP approach for solving more complicated combinatorial problems such as the quadratic assignment problem, general graph partitioning problems and set partitioning problems.

The quadratic assignment problem, general graph partitioning problem and set partitioning problem, denoted QAP, GP and SP, respectively, have all been extensively investigated because of their special structures and their numerous practical applications. Since they are all well-known as NP-hard problems (see e.g. [GJ79]), the current popular approaches are focused on finding a "near optimal" solution by solving a relaxation problem for each of them. Therefore, to find a tight relaxation for each of these problems is essential for finding high quality "near optimal" solutions.

Although the QAP, GP and SP do not look alike, they do have some common structure. They all can be formulated so that their constraints look like the following.

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= 1 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n &= 1, \end{aligned}$$

where each a_{ij} is either 0 or 1 and $x = (x_1, \ldots, x_n)^t$ is a 0-1 vector. We call the above constraints the *assignment constraints*. This common structure turns out to be essential in developing an SDP framework to solve QAP, GP and SP.

Our work in this thesis serves two purposes:

- use semidefinite programming relaxations to obtain better lower bounds and high quality approximate solutions in order to solve QAP, GP and SP.
- try to develop a unified way of applying SDP to tackle hard combinatorial optimization problems having similar structure to QAP, GP and SP.

The structure of the thesis is as follows.

In the second part of Chapter 1, we introduce some notation which will be used throughout this thesis.

In Chapter 2 we introduce semidefinite programming. We will describe briefly the geometry of the feasible set of SDP and duality theory. We will demonstrate an SDP relaxation procedure for general quadratically constrained quadratic programming problems.

In Chapter 3, we develop an SDP relaxation for the quadratic assignment problem by exploiting the geometrical structure and using the *Gangster* operator. To solve the large SDP relaxation, a truncated conjugate gradient method is tried when implementing a primal-dual interior-point algorithm.

In Chapter 4, we develop an SDP relaxation for GP by following almost the same procedure as for QAP. As a byproduct, the SDP relaxation is almost the same as for the QAP. Therefore, it is solved by the same methods. Numerical tests are run for both unweighted and weighted graphs.

In Chapter 5, we develop an SDP relaxation for SP. Again we follow almost the same procedure as for QAP. Numerical tests are run by using the same algorithm for QAP. For large sparse SP with block structure, a mixed LP-SDP relaxation approach is proposed. Preliminary numerical tests are run by using an infeasible primal-dual interior-point algorithm.

In the last chapter, we will give our summary and discussion about future work.

1.2 Notation

In this section, we give some notations and terminologies which will be used throughout the thesis.

We work with the space of *real* $n \times n$ *symmetric matrices*, denoted $\mathcal{S}_n$, with the *trace inner product* $\langle A, B \rangle := \operatorname{trace} AB$. The dimension of the matrices is assumed to be n, unless otherwise noted.

Suppose $J \subset \{(i,j) : 1 \leq i \leq j \leq n\}$. The subspace of $n \times n$ symmetric matrices with nonzero index set J is denoted $\mathcal{S}_J$, i.e.,

$$\mathcal{S}_J := \{X \in \mathcal{S}_n : X_{ij} = 0 \text{ if both } (i,j) \text{ and } (j,i) \notin J\}.$$

The set of $n \times n$ *positive semidefinite matrices*, denoted $\mathcal{P}_n$, forms a closed convex cone, which is *self-polar*, i.e., the *polar cone.*

$$\mathcal{P}_n^+ := \{K = K^t : \operatorname{trace} KP \geq 0, \ \forall P \in \mathcal{P}_n\} = \mathcal{P}_n.$$

The space $\mathcal{S}_n$ is endowed with the Löwner partial order, i.e. $A \succ$ (resp. $\succeq$) B denotes $A - B$ is positive definite (resp. positive semidefinite). (Similarly for $\prec$ and $\preceq$)

For a matrix $Q \in \mathcal{S}$, $Q^\perp$ is defined as follows

$$Q^\perp := \{X \in \mathcal{S} : \operatorname{trace}(XQ) = 0\}.$$

For $v \in \Re^n$, $\operatorname{Diag}(v)$ denotes the diagonal matrix formed from the vector v. Conversely, for a matrix M, $\operatorname{diag}(M)$, with lower case d, denotes the column vector formed from the diagonal of M.

For an $n \times m$ matrix X, $\mathrm{vec}(X)$ denotes the $nm \times 1$ vector formed from the columns of the matrix X, while $\mathrm{Mat}(x)$ denotes the $n \times m$ matrix reshaped from an $nm \times 1$ vector x.

For a vector $v = (v_1, \ldots, v_n)^t \in \Re^n$, let $v_{p:q} \in \Re^{q-p+1}$ denote the $(q-p+1) \times 1$ vector $(v_p, \ldots, v_q)^t$. $v > 0$ denotes that $v_i > 0$ for $i = 1, \ldots, n$.

For a matrix M, $M_{:j}$ denotes its jth column.

The vector $e_n \in \Re^n$ denotes the *vector of ones*, while e is a vector of ones when there is no ambiguity. The vector u_i is the *i-th unit vector*; and $E_{ij} = u_i u_j^t + u_j u_i^t$. E_n is a $n \times n$ matrix with all its entries being equal to one. I_n is a $n \times n$ identical matrix. We use E and I when there is no ambiguity.

$\mathcal{R}(M)$, $\mathcal{N}(M)$ denote *range space* and *null space* of M, respectively. For a square matrix M, $\det(M)$ denotes the *determinant*. For two $m \times n$ matrices M, N, the *Hadamard product*, or entry-wise product, is denoted $M \circ N$. For two $m \times n$ matrices M, N, the *Kronecker product*, or tensor product, is denoted $M \otimes N$.

For a linear operator $\mathcal{A} : \Re^n \to \Re^m$, the adjoint operator of $\mathcal{A}$, denoted $\mathcal{A}^*$, is a linear operator mapping from $\Re^m$ to $\Re^n$ such that for any $x \in \Re^n$ and any $y \in \Re^m$,

$$\langle \mathcal{A}(x), y \rangle = \langle x, \mathcal{A}^*(y) \rangle.$$

The *arrow operator*, acting on an $n \times n$ matrix Y, is defined by

$$\mathrm{arrow}(Y) := (\mathrm{diag}(Y))_{1:n} - Y_{1:n,0},$$

where $Y_{1:n,0}$ is the $n \times 1$ vector formed from the last n components of the first, or 0th column of Y. The operator $\mathcal{G}_J : \mathcal{S}_n \to \mathcal{S}_n$ with its range $\mathcal{R}(\mathcal{G}_J) = \mathcal{S}_J$ is called the *Gangster* operator. It "shoots" holes or zeros in a given matrix, i.e., given a set

$J \subset \{(i,j) : i \leq j \in \{1,\ldots,n\}\}$ and a matrix $Y \in \mathcal{S}_n$ as

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nn} \end{bmatrix},$$

the operator $\mathcal{G}_J(Y)$ satisfies

$$(\mathcal{G}_J(Y))_{ij} := \begin{cases} y_{ij} & \text{if } (i,j) \text{ or } (j,i) \in J \\ 0 & \text{otherwise.} \end{cases}$$

The gangster operator is self-adjoint, i.e.,

$$\mathcal{G}_J = \mathcal{G}_J^*.$$

(See e.g [FLE95] for its application on large sparse quasi-Newton method. The name of the gangster operator was introduced in [TOI77].)

Chapter 2

Semidefinite Programming

2.1 Introduction

Semidefinite programming can be a very powerful tool for several different applications: e.g. min-max eigenvalue problem [RVW95]; trust region problems [RW95b]; control problems [VB93] and hard combinatorial optimization problems [ALI95].

In this chapter, we will first present some state of the art results on the theory and algorithms for SDP. We will present the duality theory and the primal-dual interior-point framework for SDP, which resembles that for linear programming.

In order to have a deeper insight into SDP, we will have a look at the geometrical structure such as faces and dimensions of feasible sets. Again we will see the similarity between the SDP feasible sets and polyhedra. Then we will discuss how to derive an SDP relaxation for the general quadratically constrained quadratic programming problem. We will see how an SDP relaxation, generated by Lagrangian dual relaxation and homogenization, yields a lower bound.

2.2 Duality Theory and Interior Point Methods

A semidefinite programming problem has the following form

$$
(P) \qquad \begin{array}{ll} \max & \operatorname{trace} CX \\ \text{s.t.} & \mathcal{A}(X) = a \\ & \mathcal{B}(X) \leq b \\ & X \succeq 0, \end{array}
$$

where both $\mathcal{A} : \mathcal{S}_n \to \Re^p$ and $\mathcal{B} : \mathcal{S}_n \to \Re^q$ are linear operators. The dual problem of (P) is

$$
(D) \qquad \begin{array}{ll} \min & a^t y + b^t t \\ \text{s.t.} & \mathcal{A}^*(y) + \mathcal{B}^*(t) \succeq C \\ & y \in \Re^p \quad t \in \Re^q_+, \end{array}
$$

where $\mathcal{A}^*$ and $\mathcal{B}^*$ are the adjoint operators of $\mathcal{A}$ and $\mathcal{B}$, respectively. The linear operators $\mathcal{A}$ and $\mathcal{B}$ acting on $X \in \mathcal{S}_n$ can be expressed explicitly by the following two vectors, respectively.

$$
\mathcal{A}(X) = \begin{pmatrix} \operatorname{trace}(A_1 X) \\ \vdots \\ \operatorname{trace}(A_p X) \end{pmatrix} \tag{2.2.0}
$$

and

$$
\mathcal{B}(X) = \begin{pmatrix} \operatorname{trace}(B_1 X) \\ \vdots \\ \operatorname{trace}(B_q X) \end{pmatrix}, \tag{2.2.1}
$$

where $A_i \in \mathcal{S}_n$ for $i = 1, \ldots, p$ can be constructed using $\mathcal{A}(E_{ij})$, while similarly for $B_i \in \mathcal{S}_n$ for for $i = 1, \ldots, q$.

Definition 2.1 *1. Problem (P) is called strictly feasible if there exists a feasible point $\hat{X}$ such that $\hat{X} \succ 0$ and $\mathcal{B}(\hat{X}) < 0$;*

2. *Problem* (D) *is called strictly feasible if there exists a feasible point* $\hat{y}$ *and* $\hat{t}$ *such that* $\mathcal{A}^*(\hat{y}) + \mathcal{B}^*(\hat{t}) \succ C$ *and* $\hat{t} > 0$.

The following theorem characterizes the duality of SDP. For a general theorem for cone-LP's and its proof, see e.g. [WOL81].

Theorem 2.1 *Let* (P) *or* (D) *be strictly feasible. Then:*

(a) Let X *and* (y,t) *be feasible solutions of* (P) *and* (D)*, respectively. Then* $\operatorname{trace} CX \leq a^t y + b^t t$.

(b) If one of the problems is infeasible, then the other is infeasible or unbounded.

(c) Let both (P) *and* (D) *be feasible, then their optimal values are equal. Furthermore, the dual (primal) optimal solution is attained if the primal problem* (P) *(the dual problem* (D)*) is strictly feasible.*

(d) Let X *and* (y,t) *be feasible solutions of* (P) *and* (D)*, respectively. Then* X *and* $(y^t, t^t)^t$ *are optimal if and only if*

$$\textit{duality gap} := t^t(b - \mathcal{B}(X)) + \operatorname{trace}\left((\mathcal{A}^*(y) + \mathcal{B}^*(t) - C)X\right) = 0,$$

or equivalently, if and only if

$$t_i(\mathcal{B}(X) - b)_i = 0, \forall i, \ \textit{and} \ \ (\mathcal{A}^*(y) + \mathcal{B}^*(t) - C)X = 0.$$

The following example shows that if the primal problem is not strictly feasible, then the dual may not attain its optimal solution.

Example 2.2.1 *Consider the SDP pair*

$$\begin{array}{ll} \min & 2X_{12} \\ s.t. & \operatorname{Diag}(X) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ & X \succeq 0, \end{array} \qquad \begin{array}{ll} \max & y_2 \\ s.t. & \begin{bmatrix} y_1 & 0 \\ 0 & y_2 \end{bmatrix} \preceq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{array}$$

The Slater condition holds for the dual but not for the primal. The optimal value for both is zero. The primal is attained, but the dual is not.

Based on the above optimality condition for SDP, we now outline an infeasible primal-dual interior-point approach for solving the above primal-dual pair (P) and (D). This approach is introduced in [HRVW96]. Because of examples like Example 2.2.1, we need to make the following assumption:
both the primal problem (P) and the dual problem (D) are strictly feasible.
We introduce a slack variable $Z \succeq 0$ for the dual constraint such that

$$\mathcal{A}^*(y) + \mathcal{B}^*(t) - C - Z = 0.$$

Then the log-barrier problem for the dual problem (D) can be described as follows:

$$\begin{array}{ll} \min & a^t y + b^t t - \mu(\log\det Z + \sum_{i=1}^q \log t_i) \\ s.t. & \mathcal{A}^*(y) + \mathcal{B}^*(t) - Z = C \\ & t \geq 0, \quad Z \succeq 0. \end{array}$$

Here μ is a positive real number called the *barrier parameter*. For each $\mu > 0$, there is a corresponding Lagrangian:

$$\begin{aligned} L_\mu(X, y, t, Z) = \; & a^t y + b^t t - \mu(\log\det Z + \textstyle\sum_{i=1}^q \log t_i) - \\ & \operatorname{trace}\left((\mathcal{A}^*(y) + \mathcal{B}^*(t) - Z - C)X\right). \end{aligned}$$

The stationary point of the Lagrangian yields the following optimality conditions for the log-barrier problem.

$$\begin{array}{lll} F_p & := \mathcal{A}(X) - a & = 0 \\ F_d & := \mathcal{A}^*(y) + \mathcal{B}^*(t) - C - Z & = 0 \\ F_{tB} & := t \circ (b - \mathcal{B}(X)) - \mu e & = 0 \\ F_{ZX} & := ZX - \mu I & = 0 \\ & X \succeq 0, \quad Z \succeq 0, \quad t > 0, & \end{array} \tag{2.2.2}$$

where the 4th equation is modified from $X - \mu Z^{-1} = 0$. The strict concavity of $\log\det Z$ and $\log t_i$ implies that there exists a unique solution to the optimality conditions for each $\mu > 0$. Denote the unique point corresponding to μ as $(X(\mu), y(\mu), t(\mu), Z(\mu))$. The set of such points for each $\mu > 0$

$$\{(X(\mu), y(\mu), t(\mu), Z(\mu)) : \mu > 0\}$$

defines a smooth curve which is called the *central path.* The central path plays a vital role in primal-dual interior-point methods. It has been found to be beneficial that the iterate points stay within a neighborhood of the central path. By doing so, robustness in convergence can be expected. For each point (X, y, t, Z) on the central path, it is easy to determine its associated μ value using the last two equations of the optimality conditions:

$$\mu = \frac{\operatorname{trace}(ZX)}{n} = \frac{t^t(b - \mathcal{B}(X))}{q} = \frac{\operatorname{trace}(ZX) + t^t(b - \mathcal{B}(X))}{n+q}. \tag{2.2.3}$$

(Note: $\operatorname{trace}(ZX) + t^t(b - \mathcal{B}(X))$ is just the duality gap.) We will use (2.2.3) to define the associated μ value for a point (X, y, t, Z) even when it is not on the central path. The interior-point algorithm is the following. We start with a point (X, y, t, Z) which satisfies $X \succ 0$, $Z \succ 0$, $t > 0$ and $b - \mathcal{B}(X) > 0$. We estimate its associated μ value and divide it by two:

$$\mu = \frac{\operatorname{trace}(ZX) + t^t(b - \mathcal{B}(X))}{2(n+q)}.$$

(Note: this simple heuristic performs very well in practice, even though it does not guarantee monotonic decrease in μ, see [VC93].) We attempt to find steps $(\delta X, \delta y, \delta t, \delta Z)$ such that the new point $(X+\delta X, y+\delta y, t+\delta t, Z+\delta Z)$ becomes close to the point $(X(\mu), y(\mu), t(\mu), Z(\mu))$ on the central path at this value of μ. We can find such a step with a variant of Newton's method in the following way. In order to apply operators $\mathcal{A}$ and $\mathcal{B}$ to nonsymmetric matrices, we extend their definition. For any nonsymmetric square matrix M, let

$$\mathcal{A}(M) := \frac{1}{2}\mathcal{A}(M+M^t)$$

and

$$\mathcal{B}(M) := \frac{1}{2}\mathcal{B}(M+M^t).$$

From the definitions of (2.2.0) and (2.2.1) and the fact that for $A \in \mathcal{S}_n$

$$\operatorname{trace}(AX) = \operatorname{trace}(AX^t) = \operatorname{trace}\frac{1}{2}\left(A(X+X^t)\right),$$

the above definitions are well defined. We linearize the system (2.2.2) by the following system of equations

$$\begin{array}{ll} \mathcal{A}(\delta X) & = -F_p \\ \mathcal{A}^*(\delta y) + \mathcal{B}^*(\delta t) - \delta Z & = -F_d \\ \delta t \circ (b - \mathcal{B}(X)) - t \circ \mathcal{B}(\delta X) & = -F_{tB} \\ Z(\delta X) + (\delta Z)X & = -F_{ZX}. \end{array} \tag{2.2.4}$$

This linear system, where δX, δZ are symmetric, may be inconsistent. By solving for δZ using the second equation, we have

$$\delta Z = F_d + \mathcal{A}^*(\delta y) + \mathcal{B}^*(\delta t), \tag{2.2.5}$$

which is symmetric. Then we solve for δX using the 4th equation in (2.2.4) and the definition of F_{ZX} from (2.2.2). We get

$$\delta X = \mu Z^{-1} - X - Z^{-1}F_d X - Z^{-1}(\mathcal{A}^*(\delta y) + \mathcal{B}^*(\delta t))X. \tag{2.2.6}$$

Evidently, δX is not symmetric in general. By substituting the above expression for δX into the first and the third equation of the linear system (2.2.4), we obtain the following linear system for $(\delta y, \delta t)$, called the *normal equations.*

$$\begin{aligned}
\mathcal{A}(Z^{-1}F_dX - Z^{-1}(\mathcal{A}^*(\delta y) + \mathcal{B}^*(\delta t))X) &= \mathcal{A}(\mu Z^{-1} - X - Z^{-1}F_dX) + F_p \\
t \circ \mathcal{B}(Z^{-1}F_dX - Z^{-1}(\mathcal{A}^*(\delta y) + \mathcal{B}^*(\delta t))X) & \\
+\delta t \circ (b - \mathcal{B}(X)) &= t \circ \mathcal{B}(\mu Z^{-1} - X - Z^{-1}F_dX) - F_{ZX}.
\end{aligned} \tag{2.2.7}$$

We solve the normal equations for $(\delta y, \delta t)$. Then, by substituting $(\delta y, \delta t)$ into (2.2.5) and (2.2.6), we have δZ and δX. Finally, we let

$$\delta X \leftarrow \frac{\delta X + \delta X^t}{2}$$

to symmetrize δX. As we have seen, we always symmetrize δX in order to update X. Hence, the symmetrized δX may no longer be a Newton direction. However, it is still a descent direction for the objective as shown in [HRVW96]. This is the reason we call this method *a variant of Newton's method.* This nonsymmetry issue has also been dealt with by other people, see e.g. Kojima et al. [KSH94], Monteiro [MON95], Zhang [ZHA95], Alizadeh et al. [AHO94] and Nesterov et al. [NT94, NT95].

The last part of the algorithm is the *line search.* To measure the progress of the algorithm, we use the convex merit function

$$\begin{aligned}
f_\mu(X, y, t, Z) \quad = \quad & \operatorname{trace}(ZX) - \mu \log\det(ZX) + \\
& t^t(b - \mathcal{B}(X)) - \mu e^t \log(t \circ (b - \mathcal{B}(X))) + \\
& \tfrac{1}{2}\|F_p\|^2 + \tfrac{1}{2}\|F_d\|^2.
\end{aligned} \tag{2.2.8}$$

To guarantee global convergence of the algorithm, the key conditions given in [HRVW96] state that the step size α should be such that the following Goldstein-

Armijo conditions are satisfied.

$$\delta_1 \alpha \left| \frac{\partial f_\mu}{\partial s} \delta s \right| \leq f_\mu(s) - f_\mu(s + \alpha \delta s) \leq \delta_2 \alpha \left| \frac{\partial f_\mu}{\partial s} \delta s \right|, \tag{2.2.9}$$

where $0 < \delta_1 < \delta_2 < 1$, $s := (X, y, t, Z)$ and $\delta s := (\delta X, \delta y, \delta t, \delta Z)$. Because of this result, a line search method was developed in [HRVW96]. This line search method has been used successfully in solving the SDP relaxation for the max-cut problem and the min-max eigenvalue problems, see [HRVW96]. We demonstrate this line search method by finding the primal step α_p in the following.

step 0: set $\alpha_p := 1$;

step 1: if $b - \mathcal{B}(X + \alpha_p \delta X) > 0$ and $X + \alpha_p \delta X \succ 0$, stop. Otherwise, go to step 2;

step 2: repeat $\alpha_p \leftarrow 0.8\alpha_p$ until both $b - \mathcal{B}(X + \alpha_p \delta X) > 0$ and $X + \alpha_p \delta X \succ 0$ are satisfied. go to step 3;

step 3: $\alpha_p \leftarrow 0.95\alpha_p$. (to make sure the next point is not too close to the boundary.)

Similarly, we can find the dual step size α_d.

After we find the constants α_p and α_d, we step to the new point

$$\begin{gathered} X + \alpha_p \delta X \\ y + \alpha_d \delta y \\ t + \alpha_d \delta t \\ Z + \alpha_d \delta Z. \end{gathered}$$

We repeat the same procedure as above until some stopping criterion is satisfied. A stopping criterion will be discussed in the next chapter.

2.3 Geometrical Structure of SDP

Note that in the SDP problem (P) the inequality constraints can always be changed to equality constraints by adding slack variables. Without loss generality, we rewrite the SDP problem (P) as the following SDP problem in matrix form.

$$\begin{array}{rl} \max & \operatorname{trace} CX \\ \text{s.t.} & \operatorname{trace} A_i X = a_i \ \text{ for } \ 1 \leq i \leq m \\ & X \succeq 0. \end{array} \tag{2.3.10}$$

Its dual is

$$\begin{array}{rl} \min & a^t y \\ \text{s.t.} & \sum_{i=1}^m y_i A_i \succeq C \\ & y \in \Re^m. \end{array} \tag{2.3.11}$$

The feasible set of the primal SDP (2.3.10) is defined as

$$\mathcal{F}_P := \{X \in S_n : \operatorname{trace}(A_i X) = a_i, \ \text{for } i = 1, \ldots, m, \ X \succeq 0\}. \tag{2.3.12}$$

The set $\mathcal{F}_P$ is called an *elliptope*, see e.g. Laurent and Poljak [LP95].

The feasible set of the dual SDP (2.3.11) is defined as

$$\mathcal{F}_D := \{y \in \Re^m : \sum_{i=1}^m y_i A_i \succeq C\}. \tag{2.3.13}$$

The set $\mathcal{F}_D$ is called *spectrahedra* in e.g. [RAM93].

In this section we will mainly discuss the geometrical structure of elliptopes $\mathcal{F}_P$, since the geometrical structure of spectrahedra $\mathcal{F}_D$ follows from similar arguments, see e.g. Ramana [RAM93]. The study of the facial structure of elliptopes is relatively new in the optimization literature. The facial structure of general convex sets was used by Borwein and Wolkowicz [BW81]. Pataki (in [PAT94a] and [PAT94b]) discussed the facial structure of cone-LP's and SDP's. Similar work can also be found in Ramana [RAM93] and Laurent and Poljak [LP95]. An introduction to the general structure of convex sets can be found in Rockafellar [ROC70].

Definition 2.2 *Given a convex set G, F is a face of G if*

$$x, y \in G \text{ and } \frac{1}{2}(x+y) \in F$$

imply that $x, y \in F$.

Now we characterize the faces of the closed convex cone of $n \times n$ positive semidefinite matrices. Recall that the closed convex cone is denoted $\mathcal{P}_n$.

Theorem 2.2 *The following statements are equivalent.*

(i) *F is a face of $\mathcal{P}_n$.*

(ii) *There exists an orthogonal projection matrix Q,*

$$Q = Q^t = Q^2,$$

such that $F = \mathcal{P}_n \cap Q^\perp$.

(iii) *There exists an orthogonal projection matrix Q,*

$$Q = Q^t = Q^2,$$

such that $F = (I - Q)\mathcal{P}_n(I - Q)$.

Moreover,

$$F = \{X \succeq 0 : \mathcal{N}(X) \supset \mathcal{R}(Q)\} \text{ for some } Q = Q^t,$$

and the relative interior of F satisfies

$$\operatorname{ri} F = \{X \succeq 0 : \mathcal{N}(X) = \mathcal{R}(Q)\} \text{ for some } Q = Q^t.$$

For proof and other details, see e.g. [BC75].

Now we consider the faces of elliptopes $\mathcal{F}_P$. The following two results are straightforward. See e.g. Pataki [PAT96].

Theorem 2.3 *The set F is a face of $\mathcal{F}_P$ if and only if there exists a face G of $\mathcal{P}_n$ such that*

$$F = G \cap \mathcal{F}_P.$$

Denote the dimension of a set F by $\dim F$.

Theorem 2.4 *Let G be a face of $\mathcal{P}_n$. Then, there exists a $n \times m$ matrix V with* $\operatorname{rank}(V) = m$ *such that:*
$Y \in G$ if and only if there exists a matrix $X \in \mathcal{P}_m$ such that $Y = VXV^t$; furthermore,

$$\dim G = \frac{m(m+1)}{2}$$

.

Now, we discuss the extreme points and extreme rays of the elliptope $\mathcal{F}_P$, see e.g. Rockafellar [ROC70].

Definition 2.3 *The ray Y is an extreme ray of $\mathcal{F}_P$ if:*
for any rays, $Y_1 and Y_2$ of $\mathcal{F}_P$,

$$Y = \frac{1}{2}Y_1 + \frac{1}{2}Y_2$$

implies $Y_1 = \lambda Y_2$, for some $\lambda \geq 0$.

The following theorem characterizes the structure of elliptopes $\mathcal{F}_P$ using its extreme points and extreme rays. For a more general result see e.g. Klee [KLE57].

Theorem 2.5 *An elliptope $\mathcal{F}_P$ is the Minkowski sum of convex hull of its set of extreme points and extreme rays.*

2.4 SDP Relaxation for Quadratically Constrained Quadratic Programming

Consider the following quadratically constrained quadratic programming problem

$$
\begin{array}{rll}
\mu^* = & \min & x^t A_0 x + 2b_0^t x + c_0 \\
& \text{subject to} & x^t A_i x + 2b_i^t x + c_i \leq 0, \;\; i = 1, \ldots, p \\
& & x^t A_i x + 2b_i^t x + c_i = 0, i = p+1, \ldots, m,
\end{array}
\tag{2.4.14}
$$

where $A_i \in \mathcal{S}_n$, $b_i \in \Re^n$ and c_i is a scalar. The matrices A_i can be indefinite, therefore, problem (2.4.14) is generally a very hard, non-convex optimization problem. (Note: if matrices A_i are all positive semidefinite, problem (2.4.14) can be solved efficiently by an interior point method. See e.g. [NN93]) A lot of hard combinatorial optimization problems can be written in the above form. For example, a 0-1 quadratic programming problem can be written in the form (2.4.14) by expressing its 0-1 variables as $x_i(x_i - 1) = 0$. The feasible set of (2.4.14) can be either a finite discrete set or a continuous dense set.

Now we describe the Lagrangian dual approach, by which an SDP relaxation for problem (2.4.14) can be derived.

Problem (2.4.14) can be written as

$$
\begin{array}{rll}
\mu^* := & \min & (1, x^t) \left[\begin{array}{c|c} c_0 & b_0^t \\ \hline b_0 & A_0 \end{array} \right] \begin{pmatrix} 1 \\ x \end{pmatrix} \\
& \text{subject to} & (1, x^t) \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array} \right] \begin{pmatrix} 1 \\ x \end{pmatrix} \leq 0 \quad i = 1, \ldots, p \\
& & (1, x^t) \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array} \right] \begin{pmatrix} 1 \\ x \end{pmatrix} = 0 \quad i = p+1, \ldots, m.
\end{array}
$$

We can homogenize the above problem by adding a new variable x_0 such that $x_0^2 = 1$. As a result, we get the following equivalent problem

$$
\begin{array}{rll}
\mu^* = & \min & (x_0, x^t) \left[\begin{array}{c|c} c_0 & b_0^t \\ \hline b_0 & A_0 \end{array}\right] \begin{pmatrix} x_0 \\ x \end{pmatrix} \\
& \text{subject to} & (x_0, x^t) \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] \begin{pmatrix} x_0 \\ x \end{pmatrix} \leq 0 \quad i = 1, \ldots, p \\
& & (x_0, x^t) \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] \begin{pmatrix} x_0 \\ x \end{pmatrix} = 0 \quad i = p+1, \ldots, m \\
& & (x_0, x^t) \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \end{array}\right] \begin{pmatrix} x_0 \\ x \end{pmatrix} - 1 = 0.
\end{array}
$$

From the Lagrangian of the above problem, we get the following lower bound.

$$
\mu^* \geq \mu_R := \max_{\substack{t,\, \tau_i \geq 0 \\ 1 \leq i \leq p}} \min_{x_0, x} (x_0, x^t) \left(\left[\begin{array}{c|c} c_0 - t & b_0^t \\ \hline b_0 & A_0 \end{array}\right] + \sum_{i=1}^{m} \tau_i \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] \right) \begin{pmatrix} x_0 \\ x \end{pmatrix} + t.
$$

To prevent the above quadratic form from going to negative infinity, the hidden constraint has to be satisfied.

$$
\left[\begin{array}{c|c} c_0 - t & b_0^t \\ \hline b_0 & A_0 \end{array}\right] + \sum_{i=1}^{m} \tau_i \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] \succeq 0.
$$

Therefore, we can get a lower bound for problem (2.4.14) by solving the Lagrangian dual relaxation problem

$$
\begin{array}{rll}
\mu_R = & \max & t \\
& \text{subject to} & \left[\begin{array}{c|c} c_0 - t & b_0^t \\ \hline b_0 & A_0 \end{array}\right] + \sum_{i=1}^{m} \tau_i \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] \succeq 0 \\
& & \tau_i \geq 0 \;\text{ for }\; i = 1, \ldots, p.
\end{array} \tag{2.4.15}
$$

The dual of the Lagrangian dual relaxation is then

$$\begin{array}{lll}
\min & \operatorname{trace} \left[\begin{array}{c|c} c_0 & b_0^t \\ \hline b_0 & A_0 \end{array}\right] Y & \\
\text{subject to} & Y_{00} = 1 & \\
& \operatorname{trace} \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] Y \leq 0 \quad i = 1, \ldots, p & \\
& \operatorname{trace} \left[\begin{array}{c|c} c_i & b_i^t \\ \hline b_i & A_i \end{array}\right] Y = 0 \quad i = p+1, \ldots, m & \\
& Y \succeq 0. &
\end{array} \tag{2.4.16}$$

To justify that the problem (2.4.16) is an SDP relaxation, we let

$$Y = \begin{pmatrix} x_0 \\ x \end{pmatrix} (x_0, x^t). \tag{2.4.17}$$

Then the above problem (2.4.16) becomes the original quadratic optimization problem. Therefore, the SDP problem (2.4.16) is really a relaxation for problem (2.4.14). In other words, the SDP relaxation can be derived by the following approach called the *direct approach.*

1. Find a representation for each of the original quadratic constraints for the rank-one matrix
$$\begin{pmatrix} x_0 \\ x \end{pmatrix} (x_0, x^t) \quad \text{with } x_0^2 = 1;$$
2. replace the rank-one matrix by an matrix $Y \succeq 0$.

Before we apply the Lagrangian dual approach to derive an SDP relaxation for each of the applications, we would like to point out the following:

- a redundant constraint for the original problem (2.4.14) may yield a non-redundant constraint for the SDP relaxation (2.4.16).

This means that finding the "right" constraints of the original problem is essential. As we will see when we move on, this can be achieved by exploiting the special structure for each of the applications. For a 0-1 quadratic programming problem, a recipe for deriving an SDP relaxation was given by Poljak, Rendl and Wolkowicz, see e.g. [PRW95]. Also see e.g. [FK95] for some characterization of an SDP relaxation for nonconvex quadratic programs.

Chapter 3

Quadratic Assignment Problem

3.1 Introduction

The quadratic assignment problem, QAP, can best be described by the following problem:

> We are given n facilities and n locations. There is a given amount of flow between every pair of facilities and a given cost rate per unit flow (distance) between every pair of locations; and, there is a setup cost for a facility in a given location. We want to assign each facility to a unique location in such a way that the total cost (sum of cost for every pair of facilities plus the sum of the setup cost) is minimized.

We use a_{ij} for the flow between facility i and facility j, b_{ij} for the cost rate per unit flow between location i and location j and f_{ij} for the cost for setting up facility i at location j, for all $i, j \in \{1, \ldots, n\}$. Let

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix},$$

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix}.$$

For convenience let $c_{ij} = -0.5 f_{ij}$ and define

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}.$$

The diagonal elements of A and B are all zero and both A and B are symmetric matrices. For a given assignment, let X be the permutation matrix defined by

$$X_{ij} := \begin{cases} 1 & \text{if } i \text{ assigned to } j \\ 0 & \text{otherwise.} \end{cases}$$

Thus the jth column $X_{:j}$ is the indicator set for the jth location. Such an X can represent the assignment. For each such assignment X,

$$\operatorname{trace}(AXBX^t - 2CX^t)$$

gives the total cost. Therefore, the minimal total cost is obtained by solving the quadratic assignment problem in the trace formulation

$$(QAP) \quad \mu^* := \min_{X \in \Pi} \operatorname{trace}(AXBX^t - 2CX^t),$$

where Π denotes the set of permutation matrices. As we can see, (QAP) is a 0-1 quadratic minimization problem. The quadratic term comes from the flow and distance matrices and the linear term arises from the setup cost.

The QAP is well known to be NP-hard and the traveling salesman problem (TSP), a well known NP-hard problem, can be formulated as QAP (see e.g. [SG76]). In practice, QAP problems larger than order $n = 16$ are still considered very hard. The Nugent test problem (see e.g. [CNR68] and [BKR91]) of dimension $n = 20$ has only recently been solved (See e.g.[LPR93]). The techniques used so far are based on branch and bound methods which use bounding techniques, such as Gilmore-Lawler bound [GIL62, LAW63], eigenvalue bounds [HRW92a, HRW92b] and bounds based on linear programming relaxation [AJ94] and [RRD94]. Many heuristic techniques, such as simulated annealing, also need a lower bound to see how good a solution is.

In both the Gilmore-Lawler bound technique and the eigenvalue bound technique, the quadratic term and the linear term are relaxed separated to form two different problems. The sum of the optimal values of the two relaxed problems gives the lower bound. This is the disadvantage of these two techniques due to the fact that the sum of the minimal values of two functions is less than or equal to the minimal value of the sum of the two functions. Therefore, a further improvement of the lower bound can be expected if the quadratic and linear terms are combined.

In this chapter, we describe and test a new approach based on a semidefinite programming relaxation. This relaxation prove to be numerically successful. In the SDP approach, the quadratic term and the linear term are treated together. The relaxation of the linear equality constraints, corresponding to the doubly stochastic property of permutation matrices, implies that the primal of our SDP relaxation does not satisfy the Slater constraint qualification (strict feasibility). Although there is no duality gap in theory, since the dual does satisfy the Slater constraint qualification, this leads to an unbounded dual optimal solution set. see Example 2.2.1. Numerical difficulties can arise when trying to implement interior-point methods. However, the minimal face of the semidefinite cone can be found by exploiting the structure of the barycenter of the convex hull of the permutation matrices. Then, the primal problem can be projected onto the minimal face. This, combined with the so called *Gangster operator*, yields a regularized SDP of smaller dimension, which can be solved in a numerically stable way.

Now we would like to present some special notations for this chapter.

We use the *Kronecker product*, or tensor product, of two matrices, $B \otimes A$, when discussing the quadratic assignment problem QAP. Note that the objective function

$$q(X) = \operatorname{trace}(AXBX^t - 2CX^t) = \operatorname{vec}(X)^t(B \otimes A)\operatorname{vec}(X) - 2\operatorname{vec}(C)^t\operatorname{vec}(X).$$

The Kronecker product gives rise to generalized notions of trace and diagonal.

For any $n \times n$ matrix X, we define the following $(n^2+1) \times (n^2+1)$ matrix

$$Y_X := \left[\begin{array}{c|c} 1 & \operatorname{vec}(X)^t \\ \hline \operatorname{vec}(X) & \operatorname{vec}(X)\operatorname{vec}(X)^t \end{array}\right]. \tag{3.1.18}$$

The *principal-block-diagonal-operator* denoted $\mathrm{b}^0\mathrm{diag}: \mathcal{S}_{n^2+1} \to \mathcal{S}_n$, is defined by

$$\mathrm{b}^0\mathrm{diag}(Y) := \sum_{i=1}^n Y_{ii},$$

where Y is written as

$$Y = \left[\begin{array}{c|cccc} Y_{00} & Y_{01} & \dots & Y_{0n} \\ \hline Y_{10} & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \dots & Y_{nn} \end{array}\right],$$

where Y_{00} is a scalar, Y_{i0} and Y_{0i}, for $i = 1, \dots, n$, are $n \times 1$ and $1 \times n$ vectors, respectively, and Y_{ij}, for $i, j = 1, \dots, n$, are $n \times n$ blocks of Y.

The adjoint operator of $\mathrm{b}^0\mathrm{diag}$ is denoted $\mathrm{B}^0\mathrm{Diag}$

$$\mathrm{B}^0\mathrm{Diag}\ : \mathcal{S}_n \to \mathcal{S}_{n^2+1}$$

and for a matrix $S \in \mathcal{S}_n$

$$\mathrm{B}^0\mathrm{Diag}\,(S) := \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes S \end{array}\right].$$

Thus the adjoint equation

$$\mathrm{trace}\,(W \mathrm{b}^0\mathrm{diag}\,(Y)) = \mathrm{trace}\,(\mathrm{B}^0\mathrm{Diag}\,(W)Y)$$

holds for all $W \in \mathcal{S}_n$ and $Y \in \mathcal{S}_{n^2+1}$.

The *off-block-diagonal-operator* denoted $\mathrm{o}^0\mathrm{diag}\,(Y) : \mathcal{S}_{n^2+1} \to \mathcal{S}_n$, is defined by

$$\mathrm{o}^0\mathrm{diag}\,(Y) := \begin{bmatrix} \mathrm{trace}\,(Y_{11}) & \dots & \mathrm{trace}\,(Y_{1n}) \\ \vdots & \ddots & \vdots \\ \mathrm{trace}\,(Y_{n1}) & \dots & \mathrm{trace}\,(Y_{nn}) \end{bmatrix},$$

where Y is written in the same block matrix form as for the $\mathrm{b}^0\mathrm{diag}$ operator.

The adjoint operator of $\mathrm{o}^0\mathrm{diag}$ is denoted $\mathrm{O}^0\mathrm{Diag}\,(S)$,

$$\mathrm{O}^0\mathrm{Diag}\ : \mathcal{S}_n \to \mathcal{S}_{n^2+1}$$

and for a matrix $S \in \mathcal{S}_n$

$$\mathrm{O}^0\mathrm{Diag}\,(S) := \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & S \otimes I \end{array}\right].$$

Thus the adjoint equation

$$\mathrm{trace}\,(W \mathrm{o}^0\mathrm{diag}\,(Y)) = \mathrm{trace}\,(\mathrm{O}^0\mathrm{Diag}\,(W)Y)$$

holds for all $W \in \mathcal{S}_n$ and $Y \in \mathcal{S}_{n^2+1}$.

The adjoint operator of arrow is defined by

$$\text{Arrow} : \quad \Re^{n^2} \to \mathcal{S}_{n^2+1},$$

and for a vector $w \in \Re^{n^2}$

$$\text{Arrow}\,(w) = \left[\begin{array}{c|c} 0 & -\frac{1}{2}w^t \\ \hline -\frac{1}{2}w & \text{Diag}\,(w) \end{array}\right].$$

(The name arrow comes from the pattern of nonzero elements.) Note that

$$\text{trace}\,(\text{Arrow}\,(w)Y) = w^t \text{arrow}\,(Y).$$

The set of matrices with row and column sums one is denoted

$$\mathcal{E} := \{X : Xe = X^t e = e\} = \{X : \|Xe - e\|^2 + \|X^t e - e\|^2 = 0\}.$$

The set of 0-1 matrices is denoted

$$\mathcal{Z} := \{X : X_{ij}^2 = X_{ij},\ i,j = 1,\dots,n\}.$$

The set of orthogonal matrices is denoted as $\mathcal{O}$, and the set of (entrywise) non-negative matrices is denoted as $\mathcal{N}$.

The set of matrices for which the Hadamard product of any pair of distinct rows (and any two distinct columns) is equal to the zero vector is denoted

$$\mathcal{H} := \{X : \forall p \neq q\ , X_{ip}X_{iq} = 0, \forall i, \ \text{ and } \ X_{pj}X_{qj} = 0\ \forall j\}.$$

3.2 An SDP Relaxation

It is well known that the set of permutation matrices Π can be characterized as the intersection of $\mathcal{Z}$ and $\mathcal{E}$ and also as the intersection of $\mathcal{O}$ and $\mathcal{N}$ (see e.g.

[HRW92a]) i.e.

$$\Pi = \mathcal{E} \cap \mathcal{Z} = \mathcal{O} \cap \mathcal{N}. \tag{3.2.19}$$

With the introduction of the matrix set $\mathcal{H}$, we have the following result.

Lemma 3.1 $\Pi = \mathcal{E} \cap \mathcal{H}$.

Proof. It is easy to see that

$$\Pi \subset \mathcal{E} \cap \mathcal{H}.$$

We want to show that

$$\mathcal{E} \cap \mathcal{H} \subset \Pi.$$

Let

$$X \in \mathcal{E} \cap \mathcal{H}.$$

From the definition of $\mathcal{E}$, we know that in each column or row there exists an entry $X_{ij} \neq 0$. Since $X \in \mathcal{H}$, we have $X_{ip} = 0$ for $p \neq j$ and $X_{qj} = 0$ for $q \neq j$. Therefore $X_{ij} = 1$ and thus is the only nonzero entry in row i and column j, i.e. $X \in \Pi$. $\square$

The set $\mathcal{H}$ will be used later. We rewrite QAP using (3.2.19).

$$(QAP_E) \qquad \begin{array}{rlll} \mu^* := & \min & \operatorname{trace}(AXBX^t - 2CX^t) & \\ & \text{s.t.} & XX^t = X^tX = I & (X \text{ is orthogonal}) \\ & & Xe = X^te = e & (X \text{ is doubly stochastic}) \\ & & X_{ij}^2 - X_{ij} = 0, \quad \forall i,j. & (X \text{ is 0-1}) \end{array}$$

We can see that there are a lot of redundant constraints in QAP_E. Surprisingly, however, the SDP relaxation of these constraints are not all redundant. This can help tighten the SDP relaxation. We will discuss the reason in detail for using the redundant constraints $XX^t = X^tX = I$ below. We will also see the advantage of using $Xe = X^te = e$.

The constraints can now be relaxed to get an SDP relaxation of QAP_E. This can be done either via Lagrangian duality or directly from the QAP. We shall outline how the Lagrangian relaxation yields an SDP relaxation of QAP. (See also [PRW95].) In the process, we also introduce several of the linear operators used in our relaxations. We change the row and column sum constraints into $\|Xe-e\|^2 + \|X^te-e\|^2 = 0$. Consider the following equivalent formulation of the QAP problem (QAP).

$$
(QAP_1) \qquad
\begin{array}{rll}
\mu_{\mathcal{O}} := & \min & \operatorname{trace}(AXBX^t - 2CX^t) \\
& \text{s.t.} & XX^t = I \\
& & X^tX = I \\
& & \|Xe-e\|^2 + \|X^te-e\|^2 = 0 \\
& & X_{ij}^2 - X_{ij} = 0, \quad \forall i,j.
\end{array}
$$

As we can see from the Lagrangian dual approach described in Chapter 2.4, we will homogenize the problem by increasing the dimension of the problem by one. We first add the (0,1)-constraints and equation $\|Xe-e\|^2 + \|X^te-e\|^2 = 0$ to the objective function using Lagrange multipliers W_{ij} and u_0 respectively. We get

$$
\begin{array}{rl}
\mu_{\mathcal{O}} = \min\limits_{XX^t=X^tX=I} \max\limits_{W,u_0} & \operatorname{trace}(AXBX^t - 2CX^t) + \sum_{ij} W_{ij}(X_{ij}^2 - X_{ij}) \\
& +u_0(\|Xe-e\|^2 + \|X^te-e\|^2).
\end{array}
\tag{3.2.20}
$$

Interchanging min-max yields

$$
\begin{array}{rl}
\mu_{\mathcal{O}} \geq \max\limits_{W,u_0} \min\limits_{XX^t=X^tX=I} & \operatorname{trace}(AXBX^t - 2CX^t) + \sum_{ij} W_{ij}(X_{ij}^2 - X_{ij}) \\
& +u_0(\|Xe-e\|^2 + \|X^te-e\|^2).
\end{array}
\tag{3.2.21}
$$

We now homogenize the objective function by multiplying the linear terms by a constrained scalar x_0.

$$\mu_{\mathcal{O}} \geq \mu_R := \max_{W} \min_{\substack{XX^t = I \\ X^tX = I \\ x_0^2 = 1}} \quad \text{trace}\, [AXBX^t + W(X \circ X)^t + u_0(\|Xe\|^2 + \|X^te\|^2) \\ -x_0(2C + W)X^t] - 2x_0u_0e^t(X + X^t)e + 2nu_0. \tag{3.2.22}$$

Introducing a Lagrange multiplier w_0 for the constraint on x_0 and Lagrange multipliers S_b for $XX^t = I$ and S_o for $X^tX = I$, we get

$$\begin{aligned} \mu_R = \max_{W,S_b,S_o,u_0,w_0} \min_{X,\ x_0} \quad & \text{trace}\, [AXBX^t + u_0(\|Xe\|^2 + \|X^te\|^2) \\ & +W(X \circ X)^t + w_0x_0^2 + S_bXX^t + S_oX^tX\,] \\ & -\text{trace}\, x_0(2C + W)X^t - 2x_0u_0e^t(X + X^t)e \\ & -w_0 - \text{trace}\, S_b - \text{trace}\, S_o + 2nu_0. \end{aligned} \tag{3.2.23}$$

We have grouped the quadratic, linear, and constant terms together. We now define $x := \text{vec}\,(X)$, $y^t := (x_0, x^t)$ and $w := \text{vec}\,(W)$ and get

$$\begin{aligned} \mu_R = \max_{W} \min_{y} \quad & y^t\, [L_Q + \text{Arrow}\,(w) w_0 E_{00} + \text{B}^0\text{Diag}\,(S_b) + \text{O}^0\text{Diag}\,(S_o) + u_0 D]\, y \\ & -w_0 - \text{trace}\, S_b - \text{trace}\, S_o, \end{aligned} \tag{3.2.24}$$

where we define the $(n^2 + 1) \times (n^2 + 1)$ matrix

$$L_Q := \left[\begin{array}{c|c} 0 & -\text{vec}\,(C)^t \\ \hline -\text{vec}\,(C) & B \otimes A \end{array}\right], \tag{3.2.25}$$

and the linear operators

$$\text{Arrow}\,(w) := \left[\begin{array}{c|c} 0 & -\frac{1}{2}w^t \\ \hline -\frac{1}{2}w & \text{Diag}\,(w) \end{array}\right], \tag{3.2.26}$$

$$\text{B}^0\text{Diag}\,(S) := \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes S_b \end{array}\right], \tag{3.2.27}$$

and

$$\mathrm{O^0Diag}\,(S) := \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & S_o \otimes I \end{array}\right], \tag{3.2.28}$$

and

$$D := \left[\begin{array}{c|c} n & -e^t \otimes e^t \\ \hline -e \otimes e & I \otimes E \end{array}\right] + \left[\begin{array}{c|c} n & -e^t \otimes e^t \\ \hline -e \otimes e & E \otimes I \end{array}\right].$$

Note that we will refer to the additional row and column generated by the homogenization of the problem as the 0-th row and column. By using the hidden semidefinite constraint, i.e., the pure quadratic function is bounded below only if the Hessian

$$L_Q + \mathrm{Arrow}\,(w) w_0 E_{00} + \mathrm{B^0Diag}\,(S_b) + \mathrm{O^0Diag}\,(S_o) + u_0 D$$

is positive semidefinite, we see that (3.2.24) is equivalent to

$$(D_1) \quad \begin{array}{ll} \max & -w_0 - \mathrm{trace}\, S_b - \mathrm{trace}\, S_o \\ \text{s.t.} & L_Q + \mathrm{Arrow}\,(w) w_0 E_{00} + \mathrm{B^0Diag}\,(S_b) + \mathrm{O^0Diag}\,(S_o) + u_0 D \succeq 0. \end{array}$$

We introduce the $(n^2+1)\times(n^2+1)$ dual matrix variable $Y \succeq 0$ and derive the dual of the SDP problem D_1. Then, we obtain our desired SDP relaxation of QAP_1 as follows.

$$(P_1) \quad \begin{array}{lll} \min & \mathrm{trace}\, L_Q Y & \\ \text{s.t.} & \mathrm{b^0diag}\,(Y) = I & \mathrm{o^0diag}\,(Y) = I \\ & \mathrm{arrow}\,(Y) = 0 & \mathrm{trace}\, DY = 0 \\ & Y_{00} = 1 & Y \succeq 0, \end{array}$$

where the *arrow* operator, $\mathrm{b^0diag}$ and $\mathrm{o^0diag}$ are the the adjoint operators to $\mathrm{Arrow}\,(\cdot)$, $\mathrm{B^0Diag}$ and $\mathrm{O^0Diag}$, respectively; (They are defined in Chapter 1.2 and Section 1 of this chapter) the arrow operator represents the 0-1 constraints by guaranteeing that the diagonal and 0th column are identical; the $\mathrm{b^0diag}$ and

$\mathrm{o^0diag}$ represent the orthogonality constraints; and, finally, the norm constraint is represented by the constraint $\operatorname{trace} DY = 0$. Now we can show that there is exactly one redundancy among the constraints given by the operator $\mathrm{b^0diag}$ and the operator $\mathrm{o^0diag}$.

Theorem 3.1 *Among the constraints given by the operator* $\mathrm{b^0diag}$ *and the operator* $\mathrm{o^0diag}$ *there is exactly one redundant constraint. More precisely, let*

$$\mathcal{B} := \begin{pmatrix} \mathrm{b^0diag} \\ \mathrm{o^0diag} \end{pmatrix}.$$

Then

$$\dim(\mathcal{N}(\mathcal{B}^*)) = 1,$$

i.e. the null space of the operator $\mathcal{B}^*$ *is of dimension one.*

Proof. Let $S, T \in \mathcal{S}_n$ be the dual variables corresponding to $\mathrm{b^0diag}$ and $\mathrm{o^0diag}$, respectively. We first choose $S = -T = I$. Then $\mathcal{B}^*(S,T) = 0$. Hence, the null space of $\mathcal{B}^*$ is not empty.

Now let $T_{11} = 0$. We need only prove the following.

$$\mathcal{B}^*(S,T) = \mathrm{B^0Diag}\,(S) + \mathrm{O^0Diag}\,(T) = 0 \text{ implies } S = 0, \quad T = 0.$$

Since $T_{11} = 0$ and

$$\mathrm{B^0Diag}\,(S) + \mathrm{O^0Diag}\,(T) = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes S \end{array}\right] + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & T \otimes I \end{array}\right] = 0,$$

we have

$$1 \otimes S = -T_{11} \otimes I = 0,$$

thus, $S = 0$. This implies that $T = 0$ as well. □

Remark: an alternative proof can be done based on the fact that if the assignment polytope $\{X : Xe = X^t e = e, X \geq 0\}$ is expressed as

$$\{X : AX = e, X \geq 0\},$$

then the linear operator A^* has the property $\dim(\mathcal{N}(\mathcal{B}^*)) = 1$.

3.3 Geometry of the Feasible Set

In this section we study the geometrical structure of the feasible set of the SDP relaxation (P_1). We have expressed the orthogonality constraints with both $XX^t = I$ and $X^t X = I$. It is interesting to note that this redundancy adds extra constraints into the relaxation which are not redundant. These constraints reduce the size of the feasible set of the relaxation and so tighten the resulting bounds. We denote the feasible set of the SDP relaxation (P_1) by F_1. Note that $D \neq 0$ is positive semidefinite, therefore, to satisfy $\operatorname{trace} DY = 0$, Y has to be singular, which means that the feasible set of the primal problem P_1 is not strictly feasible. From this we can see that the relaxation of the redundant constraints $Xe = X^t e = e$ can actually help us see the geometric structure of the feasible set. It is not difficult to find an interior point for the feasible set of the dual (D_1), which means that Slater constraint qualification (strict feasibility) holds for (D_1). Therefore (P_1) is attained and there is no duality gap in theory for this primal-dual pair. However, since Slater constraint qualification for the primal fails, this is not truly a proper dual pair. This is because we cannot stay exactly feasible, $\succeq 0$, in the absence of Slater condition. (See [RTW95].) Moreover, because the supremum of (D_1) may never be attained, numerical instability is likely to occur. In order to overcome this difficulty, we need to explore the geometrical structure of F_1.

It is clear that the matrices

$$Y_X := \begin{pmatrix} 1 \\ \mathrm{vec}\,(X) \end{pmatrix} (1 \quad \mathrm{vec}\,(X)^t) \quad \text{for} \quad X \in \Pi$$

are feasible points of F_1. Moreover, since these points are rank-one matrices, we see that they are contained in the set of extreme points of F_1, see e.g. Pataki [PAT94a]. We need only to consider faces of $\mathcal{P}$ which contain all of these extreme points Y_X for $X \in \Pi$. We want to find the *minimal face*, which is the intersection of all these faces. The following theorem characterizes the minimal face by finding a point in its relative interior, namely the *barycenter*. This point has a very simple and elegant structure.

Theorem 3.2 *Let* $x = \mathrm{vec}\,(X)$*. Define the barycenter point*

$$\hat{Y} := \frac{1}{n!} \sum_{X \in \Pi_n} \left[\begin{array}{c|c} 1 & x^t \\ \hline x & xx^t \end{array} \right]. \tag{3.3.29}$$

Then:

1. *$\hat{Y}$ has a 1 in the (1,1) position and n diagonal $n \times n$ blocks with diagonal elements $1/n$. The first row and column equal the diagonal. The rest of the matrix is made up of $n \times n$ blocks with all elements equal to $1/(n(n-1))$*

except for the diagonal elements which are zeros:

$$\hat{Y} = \left[\begin{array}{c|cccc} 1 & & \frac{1}{n}e^t & & \\ \hline & \mathrm{Diag}\,(\frac{1}{n}e) & \frac{1}{n(n-1)}(E-I) & \cdots & \frac{1}{n(n-1)}(E-I) \\ & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n}e & \cdots & \cdots & \cdots & \cdots \\ & \cdots & \cdots & \cdots & \cdots \\ & \frac{1}{n(n-1)}(E-I) & \cdots & \frac{1}{n(n-1)}(E-I) & \mathrm{Diag}\,(\frac{1}{n}e) \end{array}\right]$$

$$= \begin{pmatrix} 1 \\ \frac{1}{n}e \end{pmatrix} (1, \tfrac{1}{n}e^t) + \left[\begin{array}{c|c} 0 & 0 \\ \hline & \\ 0 & \frac{1}{n^2(n-1)}(nI_n - E) \otimes (nI_n - E) \end{array}\right];$$

2.

$$\mathrm{rank}\,(\hat{Y}) = (n-1)^2 + 1;$$

3. the n^2+1 *eigenvalues of* $\hat{Y}$ *are given in the vector*

$$\left(2, \frac{1}{n-1}e_{(n-1)^2}, 0, \ldots, 0\right)^t;$$

4.

$$\mathcal{N}(\hat{Y}) = \left\{u : u \in \mathcal{R}(T^t)\right\},$$

where the assignment constraint matrix T *is*

$$T := \left[\begin{array}{c|cccccc} -1 & e^t & 0 & \cdots & \cdots & 0 \\ -1 & 0 & e^t & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & \cdots & \cdots & 0 & e^t \\ -e & I & I & \cdots & \cdots & I \end{array}\right];$$

5. *the range of* $\hat{Y}$ *can be expressed by the columns of the* $(n^2+1)\times((n-1)^2+1)$ *matrix*

$$\hat{V} = \left[\begin{array}{c|c} 1 & 0 \\ \hline e/n & V \otimes V \end{array}\right], \tag{3.3.30}$$

where the matrix V

$$V := \begin{bmatrix} I_n \\ -e_n^t \end{bmatrix}.$$

Furthermore, $T\hat{V} = 0$.

Proof. Fix $X \in \Pi_n$ and let

$$Y = Y_X = \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} (1 \ \operatorname{vec}(X)^t).$$

Consider the entries of the 0th row of Y. Since $Y_{0,(i-1)n+j} = 1$ means i is assigned to j and there are $(n-1)!$ such permutations, the components of the 0th row of $\hat{Y}$ are given by

$$\hat{Y}_{0,(i-1)n+j} = \frac{1}{n!}(n-1)! = \frac{1}{n}.$$

Now consider the entries of Y in the other rows, $Y_{(p-1)n+q,(i-1)n+j}$.

i) if $p = i$ and $q = j$, then $Y_{(p-1)n+q,(i-1)n+j} = 1$ means that i is assigned to j and there are $(n-1)!$ such permutations, therefore the diagonal elements are

$$\hat{Y}_{(p-1)n+q,(i-1)n+j} = \frac{1}{n!}(n-1)! = \frac{1}{n}.$$

ii) Now suppose that $p \neq i$ and $q \neq j$, i.e., the element is an off-diagonal element in an off-diagonal block. then $Y_{(p-1)n+q,(i-1)n+j} = 1$ means that i is assigned to j and p is assigned to q and since there are $(n-2)!$ such permutations, therefore

$$\hat{Y}_{(p-1)n+q,(i-1)n+j} = \frac{1}{n!}(n-2)! = \frac{1}{n(n-1)}.$$

iii) Otherwise, suppose that $p = i$ or $q = j$ but not both, i.e., we consider the off-diagonal elements of the diagonal block and the diagonal elements of the off-diagonal blocks. By the property of permutation matrices, these elements are all 0.

This proves the representation of $\hat{Y}$ in 1.

Let us find the rank and eigenvalues of $\hat{Y}$. We partition

$$\hat{Y} = \left[\begin{array}{c|c} 1 & \frac{1}{n}e^t \\ \hline \frac{1}{n}e & W \end{array}\right],$$

thus defining the block W. We have

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline -\frac{1}{n}e & I \end{array}\right] \hat{Y} \left[\begin{array}{c|c} 1 & -\frac{1}{n}e^t \\ \hline 0 & I \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & S \end{array}\right], \tag{3.3.31}$$

where $S = X - \frac{1}{n^2}E$. As a result, we have

$$\operatorname{rank}(\hat{Y}) = 1 + \operatorname{rank}(S).$$

Direct verification shows that

$$S = \frac{1}{n^2(n-1)}(nI_n - E) \otimes (nI_n - E).$$

The eigenvalues of $nI_n - E$ are n, with multiplicity $n-1$, and 0. Note that the eigenvalues of a Kronecker product are given by the Kronecker product of eigenvalues. Therefore, we have that the eigenvalues of S are $1/(n-1)$, with multiplicity $(n-1)^2$, and 0, with multiplicity $2n-1$. Therefore, we have

$$\operatorname{rank}(\hat{Y}) = 1 + \operatorname{rank}(S) = (n-1)^2 + 1.$$

This proves 2.

By (3.3.31), we can easily see that $1/(n-1)$, with multiplicity $(n-1)^2$, are also eigenvalues of $\hat{Y}$. Also, 2 is an eigenvalue of $\hat{Y}$. Therefore, since $\operatorname{rank}(\hat{Y}) = (n-1)^2+1$, we have that the eigenvalues of $\hat{Y}$ are 2, $1/(n-1)$ with multiplicity $(n-1)^2$, and 0 with multiplicity $2n-1$. This proves 3.

Note that $\operatorname{rank}(T) = 2n-1$ and $T\hat{Y} = 0$. Therefore, we have

$$\mathcal{N}(\hat{Y}) = \left\{u : u \in \mathcal{R}(T^t)\right\}.$$

This proves 4.

Since $\operatorname{rank}(\hat{V}) = (n-1)^2+1$ and $T\hat{V} = 0$, the columns of $\hat{V}$ span the range space of $\hat{Y}$. $\square$

Remark: The structure of the assignment polytope has been well studied. An alternative proof for part 1 and part 2 can be done based on the well known fact that the dimension of the assignment polytope is $(n-1)^2$.

The above characterization of the barycenter enables us to find the minimal face of F_1 that contains the feasible set of the SDP relaxation. Note that the range space of the barycenter $\hat{Y}$ spanned by the columns of $\hat{V}$ is the null space of the assignment matrix T. However, we would like to point out that this property of QAP is not true for a general feasible set with an assignment structure. Here is a counter example.

Consider the problem

$$\begin{array}{lllll} x_1 & & & & = 1 \\ x_1 & +x_2 & +x_3 & +x_4 & = 1 \\ x_1, & x_2, & x_3, & x_4 & \geq 0. \end{array}$$

As we can see its only solution is $(1,0,0,0)^t$, hence, its barycenter is a rank-one matrix. However, the null space of the above system is of dimension 3.

This fact again tells us that a success in finding a barycenter is the key in exploiting the geometrical structure of a given problem with an assignment structure.

Finally, let $t(n) := \frac{n(n+1)}{2}$. We have the following corollary.

Corollary 3.1 *The dimension of the minimal face is* $t((n-1)^2+1)$. *Moreover, the minimal face can be expressed as* $\hat{V}\mathcal{S}_{(n-1)^2+1}\hat{V}^t$.

From Theorem 3.2 we conclude that $Y \succeq 0$ is in the minimal face if and only if $Y = \hat{V}R\hat{V}^t$ for some $R \succeq 0$. We can now replace the matrix Y by $\hat{V}R\hat{V}^t$ in the SDP relaxation (P_1). As a result we get the following projected SDP relaxation.

$$
(Relax1) \qquad
\begin{array}{rl}
\min & \operatorname{trace}(\hat{V}^t L_Q \hat{V})R \\
\text{s.t.} & \mathrm{b}^0\mathrm{diag}\,(\hat{V}R\hat{V}^t) = I \\
 & \mathrm{o}^0\mathrm{diag}\,(\hat{V}R\hat{V}^t) = I \\
 & \operatorname{arrow}(\hat{V}R\hat{V}^t) = 0 \\
 & (\hat{V}R\hat{V}^t)_{00} = 1 \\
 & R \succeq 0.
\end{array}
$$

By construction, this program satisfies the generalized Slater constraint qualification for both the primal and the dual. Therefore there will be no duality gap and the optimal solutions are attained for both primal and dual. The projected SDP relaxation $(Relax1)$ has been solved by a primal-dual interior-point method. (See e.g. [KAR95]).

3.4 Gangster Operator and Final SDP Relaxation

It is very interesting to study the structure of $\hat{Y}$. Because of the symmetry, we only consider the upper triangular part. We denote the zero entries of $\hat{Y}$ by the following set

$$J := \left\{ (i,j) : \begin{array}{ll} i = (p-1)n + q \;\; j = (p-1)n + r & \text{for } q < r \;\text{ or} \\ i = (p-1)n + q \;\; j = (r-1)n + q & \text{for } p < r \end{array} \right\}.$$

With the set J we define the gangster operator

$$(\mathcal{G}_J(Y))_{ij} := \begin{cases} Y_{ij} & \text{if } \;(i,j) \text{ or } (j,i) \in J \\ 0 & \text{otherwise.} \end{cases}$$

As a result, we have

$$\mathcal{G}_J(\hat{Y}) = 0. \tag{3.4.32}$$

For any permutation matrix $X \in \Pi$, Y_X has all its entries either 0 or 1, and $\hat{Y}$ is just a convex combination of all these matrices Y_X for $X \in \Pi$. Hence, from (3.4.32), we have

$$\mathcal{G}_J(Y_X) = 0 \;\; \forall X \in \Pi.$$

Therefore, we can even further tighten the feasible set of the projected SDP relaxation problem ($Relax1$) by adding the natural constraints $\mathcal{G}_J(Y) = 0$. Note that the gangster operator constraints $\mathcal{G}_J(Y) = 0$ can be directly derived from the expression of the QAP feasible set, $\mathcal{E} \cap \mathcal{H}$.

The following useful properties can be derived from the fact that $T\hat{V} = 0$.

Lemma 3.2 *Let R be arbitrary $(n-1)^2+1 \times (n-1)^2+1$ symmetric matrix with*

$$R = \left[\begin{array}{c|ccc} R_{00} & R_{01} & \dots & R_{0(n-1)} \\ \hline R_{10} & R_{11} & \dots & R_{1(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ R_{(n-1)0} & R_{(n-1)1} & \dots & R_{(n-1)(n-1)} \end{array}\right],$$

where R_{00} is a scalar, R_{i0}, for $i = 1, \dots, n-1$, are $(n-1) \times 1$ vectors and R_{ij}, for $i, j = 1, \dots, n-1$, are $(n-1) \times (n-1)$ blocks of R. Let $Y = \hat{V} R \hat{V}^t$ and partition Y as

$$Y = \left[\begin{array}{c|ccc} Y_{00} & Y_{01} & \dots & Y_{0n} \\ \hline Y_{10} & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \dots & Y_{nn} \end{array}\right],$$

where Y_{00} is a scalar, Y_{i0}, for $i = 1, \dots, n$, are $n \times 1$ vectors and Y_{ij}, for $i, j = 1, \dots, n$, are $n \times n$ blocks of Y. Then

a)

$$Y_{00} = R_{00},$$

$$Y_{0i} e = R_{00}, \quad \text{for} \ \ i = 1, \dots, n,$$

and

$$\sum_{i=1}^{n} Y_{0i} = R_{00} e^t.$$

b)

$$Y_{0j} = e^t Y_{ij}, \quad \text{for} \ \ i, j = 1, \dots, n.$$

c)

$$\sum_{i=1}^{n} Y_{ij} = e X_{0j}, \quad \text{for}, \quad j = 1, \dots, n.$$

In particular

$$\sum_{i=1}^{n} \operatorname{diag}(Y_{ij}) = R_{0j}, \quad \textit{for}, \quad j = 1, \ldots, n.$$

Proof. We can easily check that $Y_{00} = R_{00}$. Since $T\hat{V} = 0$, we have

$$TY = T\hat{V}R\hat{V}^t = 0.$$

The remaining results follows from direct verification. □

Now, we add the gangster operator to the projected SDP relaxation problem $(Relax1)$. From Lemma 3.2, we have

$$Y_{0j} = e^t Y_{jj} \text{ for } j = 1, \ldots, n.$$

Note that the off-diagonal entries for each Y_{jj} are zeros. Therefore it follows that the arrow operator is redundant. Furthermore, by part $a)$ of Lemma 3.2, we can see that the principal-block-diag operator is redundant. Similarly, the off-block-diag operator is redundant.

We now define a subset $\hat{J}$ of J,

$$\hat{J} := \left\{ (i,j) : \begin{array}{lllll} i = (p-1)n + q & j = (p-1)n + r & \text{for} & q < r & \text{or} \\ i = (p-1)n + q & j = (r-1)n + q & \text{for} & p < r & r \neq n \\ & & & (p,r) \neq (n-2, n-1) & \end{array} \right\}.$$

With the new index set $\hat{J}$ we have the following lemma.

Lemma 3.3 *For any matrix* $Y \in \mathcal{S}_{\hat{J}}$,

$$\hat{V}^t Y \hat{V} = 0 \implies Y = 0.$$

Proof. The matrix Y can be written as

$$Y = \left[\begin{array}{c|ccc} 0 & 0 & \dots & 0 \\ \hline 0 & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & Y_{n1} & \dots & Y_{nn} \end{array}\right].$$

We let

$$Z = (V \otimes V)^t \begin{bmatrix} Y_{11} & \dots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \dots & Y_{nn} \end{bmatrix} (V \otimes V).$$

Then from $\hat{V}^t Y \hat{V} = 0$, we have $Z = 0$. Note that

$$V \otimes V = \begin{bmatrix} V & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & V \\ -V & \dots & -V \end{bmatrix}.$$

Therefore if we write the above matrix Z as

$$\begin{bmatrix} Z_{11} & \dots & Z_{1n-1} \\ \vdots & \ddots & \vdots \\ Z_{n-11} & \dots & Z_{n-1n-1} \end{bmatrix},$$

then we have, for $i, j \in \{1, \dots, n-1\}$,

$$Z_{ij} = V^t(Y_{ij} - Y_{nj} - Y_{in} + Y_{nn})V = 0. \tag{3.4.33}$$

Note that $Y_{ni} = Y_{in} = 0$, for $i = 1, \dots n-1$, and
$Y_{n-2n-1} = Y_{n-1n-2} = 0$. We have $V^t Y_{nn} V = 0$ and hence

$$Z_{ij} = V^t(Y_{ij})V = 0,$$

for $i, j \in \{1, \ldots, n-1\}$.

Since Y_{ij} can be either a diagonal matrix or a matrix with diagonal equal to zeros, we have the following two cases.

Case 1: Y_{ij} is a diagonal matrix.

Let

$$Y_{ij} = \begin{bmatrix} a_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & a_n \end{bmatrix}.$$

Then

$$Z_{ij} = \begin{bmatrix} a_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & a_{n-1} \end{bmatrix} + a_n E = 0,$$

which implies that $Y_{ij} = 0$.

Case 2: Y_{ij} is a matrix with diagonal equal to zeros.

Let

$$Y_{ij} = \begin{bmatrix} A & b \\ b^t & 0 \end{bmatrix},$$

where A is a $n-1$ by $n-1$ matrix with diagonal equal to zeros. Thus, we have

$$Z_{ij} = A - eb^t - be^t = 0,$$

which implies that $b = 0$ and $A = 0$, i.e. $Y_{ij} = 0$. Therefore, we have $Y_{nn} = 0$ and

$$Y_{ij} = 0,$$

for $i, j \in \{1, \ldots, n-1\}$, i.e.,

$$Y = 0.$$

□

We can get rid of all of the redundant constraints from the gangster operator $\mathcal{G}_J$ based on the following theorem.

Theorem 3.3 *Let* $Y = \hat{V}R\hat{V}^t$ *with form*

$$Y = \left[\begin{array}{c|ccc} Y_{00} & Y_{01} & \dots & Y_{0n} \\ \hline Y_{10} & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \dots & Y_{nn} \end{array}\right].$$

Then

1. $\mathcal{G}_{\hat{J}}(Y) = 0$ *implies that* $\operatorname{diag}(Y_{1n}) = 0, \dots, \operatorname{diag}(Y_{1n-1}) = 0$ *and* $\operatorname{diag}(Y_{n-2n-1}) = 0$.
2. *Let* $\bar{J} = \hat{J} \cup (0,0)$, *then the mapping* $\mathcal{G}_{\bar{J}}(Y)$ *is onto.*

Proof. Let $\mathcal{G}_{\hat{J}}(Y) = 0$. Then from Lemma 3.2, we have, for each $i = 1, \dots, n$,

$$\sum_{j=1}^{n} \operatorname{diag}(Y_{ij}) = Y_{i0}$$

and

$$\operatorname{diag}(Y_{ii}) = Y_{i0},$$

which implies that $\operatorname{diag}(Y_{in}) = 0$, for $i = 1, \dots n-3$, and therefore

$$\begin{cases} \operatorname{diag}(Y_{n-2n-1}) + \operatorname{diag}(Y_{n-2n}) = 0 \\ \operatorname{diag}(Y_{n-2n-1}) + \operatorname{diag}(Y_{n-1n}) = 0 \\ \operatorname{diag}(Y_{n-2n}) + \operatorname{diag}(Y_{n-1n}) = 0, \end{cases}$$

which implies that

$$\begin{cases} \operatorname{diag}(Y_{n-2n-1}) = 0 \\ \operatorname{diag}(Y_{n-2n}) = 0 \\ \operatorname{diag}(Y_{n-1n}) = 0. \end{cases}$$

This completes the proof for 1.

Since $\hat{Y}_{00} = 1$ and $\mathcal{G}_{\hat{J}}(\hat{Y}) = 0$, we know that Y_{00} is linearly independent on $\mathcal{G}_{\hat{J}}$. The rest of the proof for 2 follows immediately from Lemma 3.3. □

Therefore, by eliminating the redundant constraints we now can get a very simple final SDP relaxation.

$$(Relax2) \qquad \begin{array}{ll} \min & \operatorname{trace}(\hat{V}^t L_Q \hat{V})X \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(\hat{V}X\hat{V}^t) = E_{00} \\ & X \succeq 0, \end{array}$$

where $X \in \mathcal{P}_{(n-1)^2-1}$. (Note: in the rest of the chapter, X is not for a permutation matrix but the primal matrix variable for the SDP relaxation.) Its dual problem is

$$(D_2) \qquad \begin{array}{ll} \max & -W_{00} \\ \text{s.t.} & \hat{V}^t(L_Q + W)\hat{V} \succeq 0 \\ & W \in \mathcal{S}_{\bar{J}}. \end{array}$$

Note that the gangster operator is self adjoint and $\mathcal{G}_{\bar{J}}(\mathcal{S}) = \mathcal{S}_{\bar{J}}$. The following theorem gives a very interesting property of a feasible solution of the final SDP relaxation.

Theorem 3.4 *Let X be a feasible solution of $(Relax2)$. Then, the $n \times n$ matrix* $\operatorname{Mat}((\operatorname{diag}(\hat{V}X\hat{V}^t))_{1:n^2})$ *is a doubly stochastic matrix, i.e.*

$$\operatorname{Mat}((\operatorname{diag}(\hat{V}X\hat{V}^t))_{1:n^2}) \in \mathcal{E}.$$

Proof. Let $Y = \hat{V}X\hat{V}^t$. Then from Lemma 3.2 and $\mathcal{G}_{\bar{J}}(Y) = E_{00}$, we have $Y_{:0} = \operatorname{diag}(Y)$ and $Y_{00} = 1$. The rest of the proof follows immediately from the part a) of Lemma 3.2. □

From the above theorem, we can see that the final SDP relaxation can not only give a lower bound for the QAP, but also yields a doubly stochastic matrix, which may be used to derive a good feasible solution for the QAP.

Before we solve the final SDP relaxation, we would like to give interior points for both the primal feasible set and the dual feasible set.

Theorem 3.5 *The* $((n-1)^2+1) \times ((n-1)^2+1)$ *matrix*

$$\hat{X} = \left[\begin{array}{c|c} 1 & 0 \\ \hline & \\ 0 & \frac{1}{n^2(n-1)}(nI_{n-1} - E_{n-1}) \otimes (nI_{n-1} - E_{n-1}) \end{array}\right]$$

is an strictly interior point of the feasible set for $(Relax2)$.

Proof. Note that $\hat{X}$ is positive definite since $nI_{n-1} - E_{n-1}$ is positive definite.

The rest of the proof follows from showing that

$$\hat{V}\hat{X}\hat{V}^t = \hat{Y},$$

where $\hat{Y}$ is the barycenter.

$$\begin{aligned}
\hat{V}\hat{X}\hat{V}^t &= \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}e_n \otimes e_n & V \otimes V \end{array}\right] \hat{X} \left[\begin{array}{c|c} 1 & \frac{1}{n}e_n^t \otimes e_n^t \\ \hline 0 & V^t \otimes V^t \end{array}\right] \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}e_n \otimes e_n \end{pmatrix} \left(1, \frac{1}{n}e_n^t \otimes e_n^t\right) + \\
&\quad \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}V(nI_{n-1} - E_{n-1})V^t \otimes V(nI_{n-1} - E_{n-1})V^t \end{array}\right] \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}e_n \otimes e_n \end{pmatrix} \left(1, \frac{1}{n}e_n^t \otimes e_n^t\right) + \\
&\quad \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}(nI_n - E_n) \otimes (nI_n - E_n) \end{array}\right] \\
&= \hat{Y}
\end{aligned}$$

□

Theorem 3.6 *Let*

$$\bar{W} = M \left[\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right].$$

Then for a sufficiently large scalar M, $\bar{W}$ is a strictly interior point of feasible set (D_2).

Proof. It is obvious that we only need to show the matrix

$$\hat{V}^t \left[\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right] \hat{V}$$

is positive definite. Notice that $e^tV = 0$, we have

$$\begin{aligned}
\hat{V}^t\bar{W}\hat{V} &= \left[\begin{array}{c|c} 1 & e^t/n \\ \hline 0 & V^t \otimes V^t \end{array}\right] \left[\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline e/n & V \otimes V \end{array}\right] \\
&= \left[\begin{array}{c|c} n + (e^tI_ne)(e^t(I_n - E_n)e)/n^2 & (e^tV) \otimes (e^t(I_n - E_n)V) \\ \hline (V^te) \otimes (V^t(I_n - E_n)e) & (V^tV) \otimes (V^t(I_n - E_n)V) \end{array}\right] \\
&= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & (V^tV) \otimes (V^t(I_n - \frac{1}{n}E_n)V) \end{array}\right] \\
&= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & V^tV \otimes V^tV \end{array}\right] \\
&= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & (I_{n-1} + E_{n-1}) \otimes (I_{n-1} + E_{n-1}) \end{array}\right].
\end{aligned}$$

Since the matrix $I_{n-1} + E_{n-1}$ is positive definite, we have that

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & (I_{n-1} + E_{n-1}) \otimes (I_{n-1} + E_{n-1}) \end{array}\right]$$

is positive definite. $\square$

3.5 A Truncated Primal-infeasible Dual-feasible Interior-Point Method

Helmberg et al. [HEL94, HRVW96] proposed a primal-dual interior-point method for solving general semidefinite programming problem. We described the method in Chapter 2. With this method they successfully solved max-cut problems. Based on this method, we develop a so called *truncated primal-infeasible dual-feasible interior-point method* to solve the final SDP relaxation ($Relax2$). This method aims to generate a lower bound efficiently for large scale QAP. We describe this method in the rest of the section.

3.5.1 Why this Method?

We rewrite the dual problem (D_2) by introducing a slack matrix Z.

$$\begin{array}{ll} \max & -W_{00} \\ \text{s.t.} & \hat{V}^t(L_Q+W)\hat{V} - Z = 0 \\ & Z \succeq 0 \\ & Y \in \mathcal{S}_{\bar{J}}. \end{array} \tag{3.5.34}$$

The Karush-Kuhn-Tucker conditions of the dual log-barrier problem are

$$\begin{array}{rcccl} F_P & := & \mathcal{G}_{\bar{J}}(\hat{V}X\hat{V}^t) - E_{00} & = & 0 \\ F_D & := & \hat{V}^t(L_Q+W)\hat{V} - Z & = & 0 \\ F_{ZX} & := & ZX - \mu I & = & 0, \end{array} \tag{3.5.35}$$

where $X \succ 0$, $Z \succ 0$ and $W \in \mathcal{S}_{\bar{J}}$. The first equation is primal feasibility conditions, while the second is the dual feasibility conditions and the third takes care of complementary slackness for X and Z. We solve this system of equations with the variant of Newton's method discussed in Chapter 2, i.e. we always symmetrize δX after we obtain a solution $(\delta X, \delta W, \delta Z)$ by solving the following system of equations

$$\begin{array}{rcl} \mathcal{G}_{\bar{J}}(\hat{V}\delta X\hat{V}^t) & = & -F_P \\ \hat{V}^t\delta W\hat{V} - \delta Z & = & -F_D \\ \delta ZX + Z\delta X & = & -F_{ZX}. \end{array} \tag{3.5.36}$$

From the second equation, we have

$$\delta Z = \hat{V}^t\delta W\hat{V} + F_D. \tag{3.5.37}$$

Substituting it into the third equation, we have

$$\delta X = -Z^{-1}\hat{V}^t\delta W\hat{V}X - Z^{-1}F_D X - Z^{-1}F_{ZX}. \tag{3.5.38}$$

Substituting this to the first equation, we obtain the following normal equation

$$\mathcal{G}_{\bar{J}}(\hat{V}Z^{-1}\hat{V}^t\delta W\hat{V}X\hat{V}^t) = F_P - \mathcal{G}_{\bar{J}}(\hat{V}(Z^{-1}F_D X + Z^{-1}F_{ZX})\hat{V}^t). \tag{3.5.39}$$

Since in our following algorithm, we can always maintain dual feasibility, we can let $F_D = 0$. We denote the linear operator $\mathcal{G}_{\bar{J}}(\hat{V}Z^{-1}\hat{V}^t\delta W\hat{V}X\hat{V}^t)$ by $\mathcal{A}$ and the right hand side by b.

$$\mathcal{A}(\cdot) := \mathcal{G}_{\bar{J}}(\hat{V}Z^{-1}\hat{V}^t(\cdot)\hat{V}X\hat{V}^t) \tag{3.5.40}$$

and

$$b := F_P - \mathcal{G}_{\bar{J}}(\hat{V}Z^{-1}F_{ZX})\hat{V}^t), \tag{3.5.41}$$

where we use the dot $\cdot$ to represent a variable for the operator. Then the normal equation becomes

$$\mathcal{A}(\delta W) = b. \tag{3.5.42}$$

The size of the above problem is $m = n^3 - 2n^2 + 1$. For $n = 20$ and $n = 30$, $m = 7201$ and $m = 25201$, respectively. Therefore, to solve such a huge (and most likely dense) system of equations, a direct solver such as the direct Cholesky factorization is not likely to be efficient or most probably can not even be implemented due to memory limitations. It is worthwhile to note that even if the above system of equations can be solved directly, it is very time consuming to create the explicit matrix form for the system. An alternative approach to solving the normal equations is to use an iterative solver such as the preconditioned conjugate gradient algorithm. The algorithm is shown below:

The Preconditioned Conjugate Gradient Algorithm

Given: initial solution y, a left-hand operator $\mathcal{A}$, right-hand-side vector b and preconditioner $\mathcal{Q}$.
Initialization: $r = b - \mathcal{A}(y)$. Repeat until stopping criteria are satisfied:

1. Solve $\mathcal{Q}v = r$.

2. $\beta = r^t v$.

3. If first iteration, $q = v$;
   else $q = v + (\beta/\bar{\beta})q$.

4. $t = \mathcal{A}(q)$.

5. $\omega = \beta / q^t t$.

6. $y \leftarrow y + \omega q$.

7. $r \leftarrow r + \omega t$.

8. $\bar{\beta} \leftarrow \beta$.

The conjugate gradient method has been proven to be a very powerful tool in various interior-point methods to solve the Newton equation for large problems. In particular, for large sparse problems, the conjugate gradient method performs very well due to the fact that preconditioners of high quality can be derived efficiently by exploiting the sparity structure (see e.g. [CHI95]). In the conjugate gradient algorithm, the most expensive part is the evaluation of the vector $\mathcal{A}(q)$, the complexity of which for the normal equation (3.5.42) is $O(n^5)$. Therefore, to make the interior point method efficient, for each interior point update, it is necessary to stop the conjugate gradient algorithm as soon as possible. One way to do so is to solve the normal equation approximately i.e., truncating or stopping the conjugate gradient algorithm early. As we expect, this may result in primal infeasibility. However, the strict feasibility of the dual can still be maintained. As a lower bound is given by the dual objective value, it is not necessary to solve the problem ($Relax2$) to optimality to generate a good lower bound. In other words, by not aiming to solve the problem ($Relax2$) to optimality, we can make this interior-point method more efficient. The essence of this method is the so called *inexact Newton method* (see e.g [DES82]). One successful application of a similar method can be found in Portugal, Resende et al. [PRVJ94]. See e.g. [VB95] for a similar approach for solving SDP problems derived from the control applications.

3.5.2 The Preconditioned (Truncated) Conjugate Gradient Method

Stopping criteria for the Conjugate Gradient Method

For the conjugate gradient algorithm, a limit on the number of iterations is set up depending on the compromise between accuracy and efficiency. In our numerical test, the limit is less than the square root of the size of the above system.

The angle θ between $\mathcal{A}(\delta W)$ and b can be computed by

$$\cos\theta = \frac{|b^t\mathcal{A}(\delta W)|}{\|b\|\cdot\|\mathcal{A}(\delta W)\|},$$

where r is the residual. We choose a small number σ (in our test $\sigma = 0.001$). The stopping criterion for the conjugate gradient method is the following:

$$1 - \cos\theta \leq \sigma, \quad \text{or the number of iteration reaches the limit.} \tag{3.5.43}$$

When (3.5.43) is satisfied, we terminate the conjugate gradient algorithm. See e.g. [PRVJ94] for reference for the stopping criterion.

Preconditioner

It is well known that a good and cheap preconditioner is the key factor to the success of a conjugate gradient method. A preconditioner is usually constructed from the information contained in the matrix or linear operator. Some popular preconditioners, constructed by various techniques such as minimum spanning tree and incomplete Cholesky, can be obtained very efficiently from an explicit matrix representation of the constraints. Unlike a linear system with explicit matrix form, it can be very expensive to construct such preconditioners for a general linear operator system. Fortunately, we will see from the following that the special structure of the gangster operator makes it cheap to construct a preconditioner.

Let $\mathcal{K}$ be the explicit $m \times m$ matrix form for the linear operator system (3.5.39) and let $\hat{X} = \hat{V} X \hat{V}^t$ and $\hat{Z} = \hat{V} Z^{-1} \hat{V}^t$. Then the linear operator system (3.5.39) becomes

$$\mathcal{G}_{\bar{J}}(\hat{Z} \delta W \hat{X}) = b.$$

For $1 \leq k, l \leq m$, let us try to calculate $\mathcal{K}_{kl}$, the (k,l) entry of $\mathcal{K}$. Note that we can always order the index set $\bar{J}$. Let (k_i, k_j) and (l_i, l_j) be the index pairs from $\bar{J}$ corresponding to k and l, respectively. The lth column of $\mathcal{K}$ is

$$\left(\ \mathcal{G}_{\bar{J}}(\hat{Z}(0.5 e_{l_i} e_{l_j}^t + 0.5 e_{l_j} e_{l_i}^t) \hat{X}) \ \right).$$

Therefore,

$$\mathcal{K}_{kl} = (\hat{Z}_{k_i l_j} \hat{X}_{l_i k_j} + \hat{Z}_{k_i l_i} \hat{X}_{l_j k_j} + \hat{Z}_{k_j l_i} \hat{X}_{l_j k_i} + \hat{Z}_{k_j l_j} \hat{X}_{l_i k_i})/2.$$

The above formula can be used to construct a preconditioner in an efficient way that exploits sparsity. In our numerical test, we took the diagonal of $\mathcal{K}$ as the preconditioner.

3.5.3 Implementation

In our truncated primal-infeasible dual-feasible interior-point algorithm for QAP, we use a stopping criterion that differ from the standard primal-dual interior-point method for general semidefinite programming problems. We describe the stopping criterion as follows.

Stopping Criterion for the Interior-Point Method

Because of the primal infeasibility, instead of using the duality gap, we use the increasing rate of the dual objective value, which is defined as

$$\delta W_{00}^k := \frac{W_{00}^{k+1} - W_{00}^k}{W_{00}^{k+1}},$$

where W_{00}^k is the dual objective value at the iteration k. At each iteration the dual objective value gives a lower bound and the lower bound increases as k increases. We choose a small number ϵ such that when

$$\delta W_{00}^k < \epsilon, \tag{3.5.44}$$

we terminate the algorithm. In other words, when the gain for increasing the lower bound is not worth the computation expense, we stop the algorithm. Since our goal is to find a lower bound, this stopping criterion is quite reasonable. In our numerical tests, we took $\epsilon = 0.001$.

Flow Chart of the Primal-infeasible Primal-dual Interior-Point Algorithm

step 0: initial primal and dual interior-points. They are given by Theorem 3.5 and Theorem 3.6.

$X := \hat{X}$, $W := \bar{W}$, $Z = \hat{V}^t(L_Q + \bar{W})\hat{V}$

$\mu := \mathrm{trace}\,(ZX)/(2n)$;

step 1. if stopping criterion (3.5.44) is not satisfied, then compute F_p, F_{ZX} (see (3.5.35)) and b (see (3.5.41)). Solve (3.5.42) by the preconditioned (truncated) conjugate gradient method with the stopping criteria (3.5.43); (see Subsection 3.5.1)

step 2. compute δX, δZ by (3.5.38) and (3.5.37) and by symmetrization. Use the line search algorithm described in Chapter 2.2 to find α_p and α_d such that

$$X + \alpha_p \delta X \succ 0, \quad Z + \alpha_d \delta Z \succ 0$$

and both α_p and α_d are as close to one as possible.

step 3 update X, W and Z by

$$X \leftarrow X + \alpha_p \delta X, \quad W \leftarrow W + \alpha_d \delta W, \quad Z \leftarrow Z + \alpha_d \delta Z$$

$\mu \leftarrow \mathrm{trace}\,(ZX)/(2n)$. Goto step 1.

3.6 Numerical Tests and Comments

We coded the truncated primal-infeasible and dual-feasible interior-point algorithm in both C and Matlab. We tested our code by using some test problems from

QAPLIB, see e.g. [BKR91]. We present the results of our numerical testing in this section.

3.6.1 Goal of the Numerical Tests

The numerical tests serve two purposes

- compare the lower bounds given by the SDP relaxation with the existing bounds,
- understand the performance of the truncated primal-infeasible dual-feasible interior-point approach to see how truncation affects both the lower bound and the CPU time.

3.6.2 Measures of Performance

The comparison of the SDP bound with the existing bounds is summarized in Table 3.6.0. The lower bounds given by the existing bounding techniques in the literature for the testing problems are included for comparison. The table reads as follows. The first column indicates the problem instance looked at and its size, *nug*xx refers to the Nugent example of size xx. For references of the problem instances we refer to QAPLIB. The second to the seventh columns give the optimal solution OPT, the Gilmore-Lawler bound GLB [GIL62, LAW63], the projection or elimination bound ELI of [HRW92a], the bound RRD obtained by [RRD94], EVB3 from [RW92], and bound GAN given by the SDP relaxation with gangster operator, respectively. The last column shows the semidefinite bound given by Karisch [KAR95]. An 'n.a.' means that the value of the bound is not available.

	OPT	GLB	ELI	RRD	EVB3	GAN	$\mathcal{B}_0$
nug05	50	50	47	50	50	50	49
nug06	86	84	69	86	70	86	74
nug07	148	137	125	148	130	144	132
nug08	214	186	167	204	174	204	179
nug12	578	493	472	523	498	534	487
nug15	1150	963	973	1041	1002	1074	1009
nug20	2570	2057	2196	2182	2286	2385	2281
nug30	6124	4539	5266	4805	5450	5648	5424
Had12	1652	1536	1573	n.a.	1589	1640	1198
Had14	2724	2492	2609	n.a.	2630	2709	2651
Had16	3720	3358	3560	n.a.	3594	3678	3612
Had18	5358	4776	5104	n.a.	5150	5286	5174
Had20	6922	6166	6625	n.a.	6678	6847	6713
car10ga	4954	3586	4079	n.a.	4541	4847	4436
car10gb	8082	6139	7211	n.a.	7617	7941	7603
car10gc	8649	7030	7837	n.a.	8233	8546	8208
car10gd	8843	6840	8006	n.a.	8364	8658	8319
car10ge	9571	7627	8672	n.a.	8987	9327	8912

Table 3.6.0: comparison of lower bounds

	OPT	FEAS	IFEAS	CPU(sec) FEAS	CPU(sec) IFEAS
nug05	50	50	50	1.817	0.416
nug06	86	86	86	11.08	1.733
nug07	148	144	144	41.02	8.3
nug08	214	204	204	120.5	15.23
nug12	578	534	534	4589	343
nug15	1150	1076	1074	29520	2776
car10ga	4954	4853	4847	920.4	99.75
car10gb	8082	7960	7941	1019	111.7
car10gc	8649	8561	8546	1061	106.4
car10gd	8843	8666	8658	959.1	93.37
car10ge	9571	9349	9327	1030	102.5

Table 3.6.1: Feasible solution vs. infeasible solution

The above numerical results show that both relaxation 2 and relaxation 3 give very good bounds, especially for problems with linear terms. Therefore, semidefinite relaxation approach for QAP is very promising. In Table 3.6.1, we study how truncation affects the quality of lower bound and CPU times. The column under OPT is for optimal objective value. The columns under FEAS and IFEAS are for the lower bounds obtained by both feasible interior-point method and infeasible interior-point method, respectively. The last two columns are for the CPU times for both feasible and infeasible interior-point methods, respectively.

From Table 3.6.1, we can see that by truncating the conjugate gradient iterations, the infeasible interior-point approach can still give almost as good lower bound as the feasible interior-point method but with much less CPU time. Also, we observed that the truncation happened mostly during the final iterations of the interior-point algorithm (where the increasing rate for the objective is already quite small), which indicates that truncation is necessary.

3.7 Future Work

Our future work will be to make our approach more efficient for large scale problems. In addition to optimizing our code and using some fast matrix computation packages such as LAPACK, we would like to apply Paulina Chin's approach (see e.g. [CHI95]) to solve the Newton's equation, i.e. instead of solving the small but dense normal equation, we will try to solve a larger but sparse system of equations. In this way, we can fully take advantage of the sparsity and, possibly, find a better preconditioner.

Chapter 4

Graph Partitioning Problem

4.1 Introduction

The graph partitioning problem, GP, can be described as follows:

> Given: an undirected graph $G = (\mathcal{V}, \mathcal{E})$ having node $\mathcal{V}$ and edge $\mathcal{E}$ and a weight, $a_{ij} \geq 0$ for the edge between node i and node j. We consider the problem of partitioning $\mathcal{V}$ into k disjoint subsets $\mathcal{V}_1, \ldots, \mathcal{V}_k$ of given sizes $m_1, \ldots, m_k$ in such a way that the sum of weights of edges that connect nodes in different subsets is minimal.

We let $a_{ij} = 0$ if there is no edge between node i and node j. Then the symmetric matrix $A = (a_{ij})$ is the *weighted adjacency matrix* of the graph G. The matrix A can be written in the following.

$$A := \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix}.$$

We assume that the graph has no loops, hence the diagonal elements of A are all zeros. The graph G is an *unweighted* graph if a_{ij} for each edge is either 0 or 1. Otherwise, the graph G is *weighted* graph. For a given partition of the graph into k subsets, let $X = (x_{ij})$ be the $n \times k$ matrix ($n = \sum_i m_i$ is the cardinality of $\mathcal{V}$) defined by

$$x_{ij} := \begin{cases} 1 & \text{if node } i \text{ is in the } j\text{th subset} \\ 0 & \text{if node } i \text{ is not in the } j\text{th subset.} \end{cases}$$

Thus the jth column $X_{:j}$ is the indicator set for the jth subset. Such a matrix X can represent the partition. Let Π be the set of such matrices. An edge between nodes i and j is called an *uncut edge* if both i and j are in the same subset. Then for each partition $X \in \Pi$,

$$\frac{1}{2}\operatorname{trace} X^t A X = \frac{1}{2}\operatorname{trace} A X X^t = \frac{1}{2}\sum_{i,j=1}^{n}\sum_{l=1}^{k} a_{ij} x_{il} x_{jl}$$

gives the total weight for the uncut edges. As a result, the total weight for the cut edges is

$$w(E_{cut}) := \frac{1}{2}(e^t A e - \operatorname{trace}(X^t A X)).$$

Note that for any partition matrix $X \in \Pi$, we have

$$\operatorname{trace} X^t \operatorname{Diag}(Ae) X = e^t A e.$$

Therefore, the minimal weights $w^*(E_{cut})$ is obtained by solving the graph partitioning problem in the trace formulation

$$(GPI) \qquad \begin{array}{cc} w^*(E_{cut}) = \min & \frac{1}{2}\operatorname{trace} X^t L X \\ \text{subject to} & X \in \Pi, \end{array}$$

where the matrix

$$L := \operatorname{Diag}(Ae) - A$$

is called the *Laplacian matrix of a graph.*

The graph partitioning problem is well known to be NP-hard and therefore finding an optimal solution is likely very difficult. Yet this problem has many applications in various areas. One important application is VLSI design; see e.g. [LEN90] for a survey of Integrated Circuit Layout. See also [HMV92] for its application to netlist partition.

One popular and very successful heuristic for finding "good" partitions was proposed by Kernighan and Lin [KL70] in 1970. In the early 70's Donath and Hoffman [DH73] provided an eigenvalue-based bound. Several new eigenvalue-based bound techniques were presented by Rendl and Wolkowicz in [RW95a] and a computational study showed these bounds are very good, see e.g [FRW94]. An SDP relaxation technique for equal-partitioning problem, i.e., the sizes of the subsets are all equal, has been successfully developed in [KR94]. In this chapter, we are going to develop an SDP relaxation for the general graph partitioning problem as described above.

4.2 An SDP Relaxation

In order to have an SDP relaxation for (GPI), we will reformulate (GPI) as a quadratically constrained quadratic programming problem. Since the matrix X is restricted to 0-1 components, we have $X_{ij} = X_{ij}^2$, i.e.,

$$X = X \circ X.$$

Also, since $Xe_k = e_n$, we have

$$X_{:i} \circ X_{:j} = 0,$$

for any $i \neq j \in \{1, \ldots, k\}$. Therefore we can reformulate (GPI) as the following problem

$$\begin{array}{rl} \min & \frac{1}{2}\mathrm{trace}\, X^t L X \\ \text{subject to} & X \circ X = X \\ & X e_k = e_n \\ & X^t e_n = \bar{m} \\ & X_{:i} \circ X_{:j} = 0 \ \forall i \neq j, \end{array}$$

where we let $\bar{m} = (m_1, \ldots, m_k)^t$. Several of these constraints are clearly redundant. Redundant constraints can still be nonredundant in the SDP relaxation as we have seen in Chapter 3. An equivalent quadratic constrained quadratic programming problem is

$$\begin{array}{rl} \min & \frac{1}{2}\mathrm{trace}\, X^t L X \\ \text{subject to} & X \circ X = X \\ & \|X e_k - e_n\|^2 = 0 \\ & \|X^t e_n - \bar{m}\|^2 = 0 \\ & X_{:i} \circ X_{:j} = 0 \ \forall i \neq j. \end{array}$$

By following the same procedure as for the quadratic assignment problem, we have the following SDP relaxation for GP.

$$(RGP) \qquad \begin{array}{rl} \min & \mathrm{trace}\, L_A Y \\ \text{s.t.} & \mathrm{arrow}\,(Y) = 0 \\ & \mathrm{trace}\, D_1 Y = 0 \\ & \mathrm{trace}\, D_2 Y = 0 \\ & \mathcal{G}_J(Y) = 0 \\ & Y_{00} = 1 \\ & Y \succeq 0, \end{array}$$

where:

$$L_A := \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{2} I \otimes L \end{array}\right]; \tag{4.2.45}$$

$$J := \left\{ (i,j) : i = (p-1)n + q \quad j = (r-1)n + q \quad \text{for} \quad \begin{array}{l} p < r,\ p, r \in \{1, \dots, k\} \\ q \in \{1, \dots, n\} \end{array} \right\};$$

the gangster operator constraint represents the (Hadamard) orthogonality of the columns, $X_{:i} \circ X_{:j} = 0, \forall i \neq j$; and, finally, the norm constraints are represented by the constraints with The $(kn+1) \times (kn+1)$ matrices

$$D_1 := \left[\begin{array}{c|c} n & -e_k^t \otimes e_n^t \\ \hline -e_k \otimes e_n & (e_k e_k^t) \otimes I_n \end{array}\right]$$

and

$$D_2 := \left[\begin{array}{c|c} \bar{m}^t \bar{m} & -\bar{m}^t \otimes e_n^t \\ \hline -\bar{m} \otimes e_n & I_k \otimes (e_n e_n^t) \end{array}\right].$$

Since both D_1 and D_2 are positive semidefinite, the feasible set of the problem (RGP) is not strictly feasible. Hence we can not apply an interior-point method right away. However, one find a very simple structured matrix in the relative interior of the feasible set, which we do in the next section.

4.3 Geometry of the Feasible Set

In this section we study the geometrical structure of the feasible set, denoted $\mathcal{F}$, and of the convex cone $\mathcal{P}$ of the SDP relaxation (RGP). As $\mathcal{F}$ is not strictly feasible, we need to find the minimal face of the feasible set $\mathcal{F}$ as in the case of quadratic assignment problem. It is clear that the matrices

$$Y_X := \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} (1 \quad \operatorname{vec}(X)^t), \quad \text{for} \quad X \in \Pi,$$

are in $\mathcal{F}$. Moreover, since these points are rank-one matrices, we see that they are contained in the set of extreme points of $\mathcal{F}$. We need only to consider the *minimal face*, the intersection of faces of $\mathcal{P}$ which contain all of these extreme points Y_X, for $X \in \Pi$. The following theorem characterizes the minimal face by finding a point in its relative interior, namely the *barycenter*. This point has a very simple and elegant structure.

Theorem 4.1 *Let* $x = \mathrm{vec}\,(X)$*. Define the barycenter point*

$$\hat{Y} := \frac{m_1! \dots m_k!}{n!} \sum_{X \in \Pi} \left[\begin{array}{c|c} 1 & x^t \\ \hline x & xx^t \end{array}\right]. \tag{4.3.45}$$

Then:

1.

$$\begin{aligned}
\hat{Y} &= \left[\begin{array}{c|cccc} 1 & \frac{m_1}{n} e_n^t & \dots & \frac{m_k}{n} e_n^t \\ \hline \frac{m_1}{n} e_n & \left(\frac{m_1}{n} I + \frac{m_1(m_1-1)}{n(n-1)}(E-I)\right) & \dots & \left(\frac{m_1 m_k}{n(n-1)}\right)(E-I) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{m_k}{n} e_n & \left(\frac{m_1 m_k}{n(n-1)}\right)(E-I) & \dots & \left(\frac{m_k}{n} I + \frac{m_k(m_k-1)}{n(n-1)}(E-I)\right) \end{array}\right] \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}\bar{m} \otimes e_n \end{pmatrix} \left(1, \frac{1}{n}\bar{m}^t \otimes e_n^t\right) \\
&\quad + \left[\begin{array}{c|c} 0 & 0 \\ \hline & \\ 0 & \frac{1}{n^2(n-1)}(n\mathrm{Diag}\,(\bar{m}) - \bar{m}\bar{m}^t) \otimes (nI_n - E_n) \end{array}\right];
\end{aligned}$$

2. the rank of the barycenter

$$\mathrm{rank}\,(\hat{Y}) = (k-1)(n-1) + 1;$$

3. the rows of the matrix

$$T := \left[\begin{array}{c|ccccc} -m_1 & e_n^t & 0 & \cdots & \cdots & 0 \\ -m_2 & 0 & e_n^t & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -m_k & 0 & \cdots & \cdots & 0 & e_n^t \\ -e_n & I_n & I_n & \cdots & \cdots & I_n \end{array}\right]$$

form a basis for the null space of $\hat{Y}$*;*

4. the columns of the matrix

$$\hat{V} := \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}\bar{m} \otimes e_n & V_k \otimes V_n \end{array}\right]$$

form a basis for the range space of $\hat{Y}$*, where*

$$V_s := \begin{bmatrix} I_{s-1} \\ -e_{s-1}^t \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ -1 & \dots & \dots & 0 & -1 \end{bmatrix}_{s\times(s-1)} .$$

Proof. There are $n!$ ways to permute the nodes and there are $m_j!$ ways to permute the members of each set. Therefore, there are $\frac{n!}{m_1!\dots m_k!}$ possible partition matrices.

Consider the $(n(j-1)+i)$th column of Y_X is

$$\begin{pmatrix} 1 \\ x \end{pmatrix} x_{n(j-1)+i}.$$

The column is zero unless $x_{n(j-1)+i} = 1$. The element $x_{n(j-1)+i}$ corresponds to the i, j element of the partition matrix X, i.e., this element is 1 if node i is in set j. There are $\frac{(n-1)!m_j}{(m_1!\dots m_k!)}$ partition matrices, X, to (GPI) with $x_{n(j-1)+i} = 1$. Therefore the components of the 0th row of $\hat{Y}$ are given by

$$\hat{Y}_{0,n(j-1)+i)} = \frac{(m_1!\dots m_k!)}{n!} \sum_{x_{n(j-1)+i}=1} 1 = \frac{(m_1!\dots m_k!)}{n!} \frac{(n-1)!m_j}{(m_1!\dots m_k!)} = \frac{m_j}{n}.$$

Now look at the $(q-1)n+p$ element of $\begin{pmatrix} 1 \\ x \end{pmatrix} x_{n(i-1)+j}$. We distinguish four cases:

1. Assume that $j = q$ and $i = p$. There are again $\frac{(n-1)!m_j}{(m_1!\dots m_k!)}$ partitions to (GPI) with $x_{(p-1)n+q} = 1$, i.e.r, this confirms the fact that the diagonal elements are equal to the elements of the 0th row.

2. Assume that the node indices $i = p$ while the set indices $j \neq q$. Since the same node cannot be in two different sets, this implies that the diagonal elements of the off-diagonal blocks of the matrices Y_X are all 0.

3. Assume that the node indices $i \neq p$ while the set indices $j = q$. these are the off-diagonal elements of the diagonal blocks. Then there are $\frac{(n-2)!m_j(m_j-1)}{(m_1!\dots m_k!)}$ possible partitions.

4. If both the node indices $i \neq p$ and the set indices $j \neq q$, then these are the off-diagonal elements of the off-diagonal blocks. There are $\frac{(n-2)!m_j m_q}{m_1!\dots m_k!}$ possible partitions.

Dividing these expressions in the four cases by $\frac{n!}{m_1!\dots m_k!}$ we get the representation of $\hat{Y}$ in 1.

Now let us find a basis for the range space of $\hat{Y}$. We partition

$$\hat{Y} = \left[\begin{array}{c|c} 1 & z^t \\ \hline z & W \end{array}\right],$$

where $z = \frac{1}{n}\bar{m} \otimes e_n$, thus defining the block W. Then

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline -\frac{1}{n}\bar{m} \otimes e_n & I \end{array}\right] \hat{Y} \left[\begin{array}{c|c} 1 & -\frac{1}{n}\bar{m}^t \otimes e_n^t \\ \hline 0 & I \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & S \end{array}\right], \tag{4.3.45}$$

where $S = W - \frac{1}{n^2}\bar{m}\bar{m}^t \otimes E_n$. As a result, we have

$$\operatorname{rank}(\hat{Y}) = 1 + \operatorname{rank}(S).$$

Direct verification shows that

$$S = \frac{1}{n^2(n-1)}(n\mathrm{Diag}\,(\bar{m}) - \bar{m}\bar{m}^t) \otimes (nI_n - E_n).$$

The null space of the matrix $(n\mathrm{Diag}\,(\bar{m}) - \bar{m}\bar{m}^t)$ and the null space of the matrix $(nI_n - E_n)$ are spanned by e_k and e_n, respectively. Therefore, their range spaces are spanned by the columns of V_k and V_n, respectively. Hence, the range space of S is spanned by the columns of $V_k \otimes V_n$. This implies that $\operatorname{rank}(S) = (k-1)(n-1)$ This proves 2. Moreover, we have that the null space of $\hat{Y}$ is of dimension $k+n-1$.

Since

$$\operatorname{rank}(T) = k + n - 1,$$

and

$$T\hat{Y} = 0, \quad T\hat{V} = 0.$$

This implies that the rows of T span the null space of $\hat{Y}$ and the columns of $\hat{V}$ span the range space of $\hat{Y}$. □

Remark: The structure of the polytope of partitions has been well studied. The feasible set $\mathcal{F}$ is a relaxation of the polytope obtained by lifting the partition matrices into the higher dimensional matrix space. Therefore the dimension of the minimal face and the structure of the null space can be studied from the known results of the polytope of partitions.

4.4 Final SDP Relaxation

From Theorem 3.2 we conclude that $Y \succeq 0$ is in the minimal face if and only $Y = \hat{V}R\hat{V}^t$, for some $R \succeq 0$. We can now substitute $\hat{V}R\hat{V}^t$ for Y in the SDP relaxation (RGP). we get the following projected SDP relaxation.

$$\begin{array}{ll} \min & \operatorname{trace} \hat{V}^t L_A \hat{V} R \\ \text{s.t.} & \operatorname{arrow}(\hat{V}R\hat{V}^t) = 0 \\ & \mathcal{G}_J(\hat{V}R\hat{V}^t) = 0 \\ & (\hat{V}R\hat{V}^t)_{00} = 1 \\ & R \succeq 0. \end{array}$$

The following useful properties can be derived from the fact that $T\hat{V} = 0$.

Lemma 4.1 *Let R be an arbitrary $(n-1)(k-1)+1 \times (n-1)(k-1)+1$ symmetric matrix with*

$$R = \left[\begin{array}{c|ccc} R_{00} & R_{01} & \dots & R_{0(k-1)} \\ \hline R_{10} & R_{11} & \dots & R_{1(k-1)} \\ \vdots & \vdots & \ddots & \vdots \\ R_{(k-1)0} & R_{(k-1)1} & \dots & R_{(k-1)(k-1)} \end{array}\right],$$

where R_{00} is a scalar, R_{i0}, for $i = 1, \dots, k-1$, are $(n-1) \times 1$ vectors and R_{ij}, for $i, j = 1, \dots, n-1$, are $(n-1) \times (n-1)$ blocks of R. Let $Y = \hat{V}R\hat{V}^t$ and *partition Y as*

$$Y = \left[\begin{array}{c|ccc} Y_{00} & Y_{01} & \dots & Y_{0k} \\ \hline Y_{10} & Y_{11} & \dots & Y_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k0} & Y_{k1} & \dots & Y_{kk} \end{array}\right],$$

where Y_{00} is a scalar, Y_{i0}, for $i = 1, \dots, k$, are $n \times 1$ vectors and Y_{ij}, for $i, j = 1, \dots, k$, are $n \times n$ blocks of Y. Then

a)

$$Y_{00} = R_{00},$$

$$Y_{0i} e_n = m_i R_{00}, \quad \textit{for} \ \ i = 1, \dots, k$$

and

$$\sum_{i=1}^{k} Y_{0i} = R_{00} e_n^t.$$

b)

$$m_i Y_{0j} = e_n^t Y_{ij}, \quad \textit{for} \ \ i, j = 1, \dots, k.$$

c)

$$\sum_{i=1}^{k} Y_{ij} = e_n R_{0j}, \quad \textit{for} \ \ j = 1, \dots, k.$$

In particular

$$\sum_{i=1}^{k} \operatorname{diag}(Y_{ij}) = R_{0j}, \quad \textit{for} \ \ j = 1, \dots, k.$$

Proof. From the equation between Y and R, we see that $Y_{00} = R_{00}$. In addition, since $T\hat{V} = 0$, we have

$$TY = T\hat{V}R\hat{V}^t = 0.$$

The remaining results follow from direct verification. $\square$

From Lemma 4.1, we conclude that the arrow operator is redundant given the gangster constraint hold and $(\hat{V}R\hat{V}^t)_{00} = 1$. Now we will show that there are no redundant constraints left. We do this by showing that the null space of the adjoint operator is 0.

Theorem 4.2 *Suppose that $W \in \mathcal{S}_J$. Then*

$$\hat{V}^t \mathcal{G}_J(W)\hat{V} = 0 \implies \mathcal{G}_J(W) = 0.$$

Proof. Let $Y = \mathcal{G}_J(W)$. Y can be written as

$$Y = \left[\begin{array}{c|ccc} 0 & 0 & \ldots & 0 \\ \hline 0 & Y_{11} & \ldots & Y_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & Y_{k1} & \ldots & Y_{kk} \end{array}\right].$$

where Y_{ij}, for $i, j \in \{1, \ldots, k\}$, are $n \times n$ matrices. We let

$$Z = (V_k \otimes V_n)^t \begin{bmatrix} Y_{11} & \ldots & Y_{1k} \\ \vdots & \ddots & \vdots \\ Y_{k1} & \ldots & Y_{kk} \end{bmatrix} (V_k \otimes V_n).$$

Then $Z = 0$. Note that

$$V_k \otimes V_n = \begin{bmatrix} V_n & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & V_n \\ -V_n & \ldots & -V_n \end{bmatrix}.$$

Therefore if we write the above matrix Z as

$$\begin{bmatrix} Z_{11} & \ldots & Z_{1k-1} \\ \vdots & \ddots & \vdots \\ Z_{k-11} & \ldots & Z_{k-1k-1} \end{bmatrix},$$

then we have, for $i, j \in \{1, \ldots, n-1\}$,

$$Z_{ij} = V_n^t(Y_{ij} - Y_{kj} - Y_{ik} + Y_{kk})V_n = 0. \tag{4.4.46}$$

Note that $Y_{kk} = Y_{ii} = 0$ for $i = 1, \ldots k-1$. We have $V_n^t Y_{ik} V_n = 0$ for $i = 1, \ldots k-1$. Therefore,

$$Z_{ij} = V^t(Y_{ij})V = 0,$$

for $i, j \in \{1, \ldots, k-1\}$. Since Y_{ij} can be either a diagonal matrix or a zero matrix, we can write

$$Y_{ij} = \begin{bmatrix} a_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & a_n \end{bmatrix}.$$

Then

$$Z_{ij} = \begin{bmatrix} a_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & a_{n-1} \end{bmatrix} + a_n E_{n-1} = 0.$$

Thus we have $Y_{ij} = 0$ for $i, j \in \{1, \ldots, k-1\}$. Therefore,

$$Y = 0.$$

$\square$

Therefore, by eliminating the redundant constraints we can get a very simple projected SDP relaxation. We let $\bar{J} = J \cup (0, 0)$.

$$(RELAXP) \qquad \begin{array}{ll} \min & \operatorname{trace}(\hat{V}^t L_A \hat{V})R \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(\hat{V} R \hat{V}^t) = E_{00} \\ & R \succeq 0, \end{array}$$

where $R \in \mathcal{P}_{(k-1)(n-1)-1}$.

Its dual problem is

$$(RELAXD) \qquad \begin{array}{ll} \max & -W_{00} \\ \text{s.t.} & \hat{V}^t(L_A + W)\hat{V} \succeq 0 \\ & W \in \mathcal{S}_{\bar{J}}. \end{array}$$

Note that the gangster operator is self adjoint and $\mathcal{G}_{\bar{J}}(\mathcal{S}) = \mathcal{S}_{\bar{J}}$. The following theorem gives a very interesting property of a feasible solution of the projected SDP relaxation.

Theorem 4.3 *Let R be a feasible solution of $(RELAXP)$. Then the $n \times k$ matrix* $\mathrm{Mat}\,((\mathrm{diag}\,(\hat{V}R\hat{V}^t))_{1:nk})$ *satisfies*

$$\mathrm{Mat}\,((\mathrm{diag}\,(\hat{V}R\hat{V}^t))_{1:nk})e_k = e_n$$

and

$$\mathrm{Mat}\,((\mathrm{diag}\,(\hat{V}R\hat{V}^t))_{1:nk})^t e_n = \bar{m}.$$

Proof. Let $Y = \hat{V}R\hat{V}^t$. Then from Lemma 4.1 and $\mathcal{G}_{\bar{J}}(Y) = E_{00}$, we have $Y_{:0} = \mathrm{diag}\,(Y)$ and $Y_{00} = 1$. The rest of the proof follows immediately from part a) of Lemma 4.1. □

From the above theorem, we can see that the final SDP relaxation can not only give a lower bound for the GP, but it also yields a $n \times k$ matrix which may be used to derive a good feasible solution for the GP.

Before we solve the final SDP relaxation, we would like to give interior points for both the primal feasible set and the dual feasible set.

Theorem 4.4 *The* $((k-1)(n-1)+1) \times ((k-1)(n-1)+1)$ *matrix*

$$\hat{R} = \left[\begin{array}{c|c} 1 & 0 \\ \hline & \\ 0 & \frac{1}{n^2(n-1)}(n\mathrm{Diag}\,(\bar{m}_{k-1}) - \bar{m}_{k-1}\bar{m}_{k-1}^t) \otimes (nI_{n-1} - E_{n-1}) \end{array}\right]$$

is a strictly feasible point of the feasible set for $(RELAXP)$*, where*

$$\bar{m}_{k-1}^t = (m_1, \ldots, m_{k-1}).$$

Proof. Note that $\hat{R}$ is positive definite since both $n\mathrm{Diag}\,(\bar{m}_{k-1}) - \bar{m}_{k-1}\bar{m}_{k-1}^t$ and $nI_{n-1} - E_{n-1}$ are positive definite.

The rest of the proof follows from showing that

$$\hat{V}\hat{R}\hat{V}^t = \hat{Y},$$

where $\hat{Y}$ is the barycenter. We see that

$$\begin{aligned}
\hat{V}\hat{R}\hat{V}^t &= \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}\bar{m}\otimes e_n & V_k \otimes V_n \end{array}\right] \hat{R} \left[\begin{array}{c|c} 1 & \frac{1}{n}\bar{m}^t \otimes e_n^t \\ \hline 0 & V_k^t \otimes V_n^t \end{array}\right] \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}\bar{m}\otimes e_n \end{pmatrix} \left(1, \tfrac{1}{n}\bar{m}^t \otimes e_n^t\right) + \\
&\quad \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}(V_k(n\mathrm{Diag}\,(\bar{m}_{k-1}) - \bar{m}_{k-1}\bar{m}_{k-1}^t)V_k^t) \otimes (V_n(nI_{n-1} - E_{n-1})V_n^t) \end{array}\right] \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}\bar{m}\otimes e_n \end{pmatrix} \left(1, \tfrac{1}{n}\bar{m}^t \otimes e_n^t\right) + \\
&\quad \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}(n\mathrm{Diag}\,(\bar{m}) - \bar{m}\bar{m}^t) \otimes (nI_n - E_n) \end{array}\right] \\
&= \hat{Y},
\end{aligned}$$

where it is straightforward to check that

$$V_k(n\mathrm{Diag}\,(\bar{m}_{k-1}) - \bar{m}_{k-1}\bar{m}_{k-1}^t)V_k^t = n\mathrm{Diag}\,(\bar{m}) - \bar{m}\bar{m}^t,$$

and in particular

$$V_n(nI_{n-1} - E_{n-1})V_n^t = nI_n - E_n.$$

$\square$

Theorem 4.5 *The matrix*

$$\bar{W} = \left[\begin{array}{c|c} M & 0 \\ \hline 0 & (I_k - E_k) \otimes I_n \end{array}\right]$$

is a strictly feasible point for the dual problem (RELAXD), if M is a sufficiently large scalar.

Proof. We can write $\hat{V}^t(L_A + \bar{W})\hat{V}$ as the following summation of two parts.

$$\hat{V}^t \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes L \end{array}\right] \hat{V} + \hat{V}^t \left[\begin{array}{c|c} M & 0 \\ \hline 0 & (I_k - E_k) \otimes I_n \end{array}\right] \hat{V}.$$

Note that

$$Le = (\mathrm{Diag}\,(Ae) - A)e = Ae - Ae = 0.$$

We have for the first part

$$\begin{aligned} \hat{V}^t \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes L \end{array}\right] \hat{V} &= \left[\begin{array}{c|c} 1 & \bar{m}^t \otimes e^t/n \\ \hline 0 & V_k^t \otimes V_n^t \end{array}\right] \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I \otimes L \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline \bar{m} \otimes e/n & V_k \otimes V_n \end{array}\right] \\ &= \left[\begin{array}{c|c} 0 + (\bar{m}^t\bar{m}) \otimes (e^tLe)/n^2 & (\bar{m}^tV_k) \otimes (e^tLV_n)/n \\ \hline (V_k^t\bar{m}) \otimes (V_n^tLe)/n & (V_k^tV_k) \otimes (V_n^tLV_n) \end{array}\right] \\ &= \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & (I_{k-1} + E_{k-1}) \otimes (V_n^tLV_n) \end{array}\right]. \end{aligned}$$

Since the matrix $I_{k-1} + E_{k-1}$ is positive definite and matrix $V_n^tLV_n$ is positive semidefinite, their Kronecker product $(I_{k-1} + E_{k-1}) \otimes (V_n^tLV_n)$ is positive semidefinite, i.e., matrix $\hat{V}^tL_A\hat{V}$ is positive semidefinite. Now for the second part, note that $e^tV = 0$ and we have

$$\begin{aligned}
\hat{V}^t\bar{W}\hat{V} &= \left[\begin{array}{c|c} 1 & \bar{m}^t \otimes e^t/n \\ \hline 0 & V_k^t \otimes V_n^t \end{array}\right] \left[\begin{array}{c|c} M & 0 \\ \hline 0 & ((I_k - E_k) \otimes I_n \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline \bar{m} \otimes e/n & V_k \otimes V_n \end{array}\right] \\
&= \left[\begin{array}{c|c} M + \bar{m}^t(I_k - E_k)\bar{m}/n & (\bar{m}^t(I_k - E_k)V_k) \otimes (e^tV_n)/n \\ \hline (V_k^t(I_k - E_k)\bar{m}) \otimes (V_n^te)/n & (V_k^t(I_k - E_k)V_k) \otimes (V_n^tV_n) \end{array}\right] \\
&= \left[\begin{array}{c|c} M + \bar{m}^t(I_k - E_k)\bar{m}/n & 0 \\ \hline 0 & (I_{k-1} + E_{k-1}) \otimes (I_{n-1} + E_{n-1}) \end{array}\right].
\end{aligned}$$

Since both matrix $I_{k-1} + E_{k-1}$ and matrix $I_{n-1} + E_{n-1}$ are positive definite, we can see that when M is large enough matrix $\hat{V}^t\bar{W}\hat{V}$ is positive definite. This completes the proof. $\square$

4.5 Numerical Tests

Since the final SDP relaxation is similar to the QAP, we use the same techniques as in the chapter for QAP to solve it. After solving the final SDP relaxation, we obtain not only a lower bound for the graph partitioning problem but a solution R for the SDP relaxation. By reshaping the diagonal of $\hat{V}R\hat{V}^t$, we can get a $n \times k$ matrix X which satisfies all the feasible constraints except the 0-1 constraint for the original graph partitioning problem. By solving a network subproblem with X as its adjacent matrix, we can find a feasible solution for the graph partitioning problem, which gives an upper bound. With this feasible solution as an initial solution, we use Adaptive Simulated Annealing technique (or VFSR) (See e.g [ING89]) to generate a better upper bound. To measure how close our upper bound is to the optimal objective value, we use the measure

$$\text{relative gap} := \frac{\text{upperbound} - \text{lowerbound}}{\text{lowerbound}}.$$

Our numerical results are based on random unweighted and weighted graphs. We include two instances for each case. First, eight unweighted graphs were randomly generated. Each edge was generated independent of other edges with probability 0.5. These graphs have vertices of $36, 60, 84$ and 108, respectively. The number of subset K are $2, 3$ and 4. The size for each partition is randomly generated. Next, another eight weighted graphs were randomly generated. Each edge was generated independent of the other edges. The weights are integer numbers between 0 and 10. Again these graphs have vertices of $36, 60, 84$ and 108, respectively. The number of subsets k are $2, 3$ and 4. The size for each partition is randomly generated. In the tables, the column under LB is the lower bound, the column under INIT is the initial upper bound and the column under BEST is the upper bound generated by the VFSR. The last column under GAP is for the gap.

From the table for weighted graphs, we observe that the gaps are less than 0.05. However, for unweighted graph the gaps are mostly between 0.05 and 0.10. The initial upper bounds derived from the SDP solution are very good as we can see that the upper bound can hardly be improved by VFSR. The results significantly improve those in [FRW94] and are comparable to the results in [KAR95, KR94] which are restricted to the equipartition case.

4.6 Conclusion

In this chapter, using the same approach as in the QAP chapter, we derive an SDP relaxation for the general graph partitioning problem. This relaxation is almost the same as the one for QAP. Numerical tests show that this relaxation can give a good lower bound, in particular for weighted graphs. Therefore, this SDP relaxation approach for the general graph partition problem is very promising. As we can see,

	BEST	INIT	LB	GAP
a36	114	116	106	0.076
b36	71	72	66	0.076
a60	217	229	203	0.069
b60	352	370	336	0.048
a84	423	427	406	0.042
b84	420	428	401	0.047
a108	747	767	708	0.055
b108	753	769	713	0.056

Table 4.5.2: Bisection for Unweighted Graphs

	BEST	INIT	LB	GAP
a36	122	122	111	0.099
b36	103	108	97	0.062
a60	321	332	297	0.081
b60	475	499	431	0.102
a84	647	654	609	0.062
b84	646	646	606	0.066
a108	1120	1120	1030	0.087
b108	1113	1113	1038	0.072

Table 4.5.3: 3-partition for Unweighted Graphs

	BEST	INIT	LB	GAP
a36	176	192	162	0.086
b36	157	162	143	0.098
a60	492	522	451	0.091
b60	480	517	432	0.111
a84	1017	1032	912	0.115
b84	1051	1051	916	0.147
a108	1703	1703	1537	0.108
b108	1680	1680	1548	0.085

Table 4.5.4: 4-partition for Unweighted Graphs

	BEST	INIT	LB	GAP
wa36	919	938	897	0.025
wb36	815	815	785	0.038
wa60	4095	4095	4027	0.017
wb60	2250	2254	2196	0.025
wa84	4755	4773	4642	0.024
wb84	1604	1619	1573	0.020
wa108	8259	8329	8125	0.017
wb108	7430	7448	7264	0.023

Table 4.5.5: bi-partition for weighted Graphs

	BEST	INIT	LB	GAP
wa36	1336	1336	1302	0.026
wb36	521	521	506	0.030
wa60	4243	4246	4178	0.016
wb60	4366	4391	4293	0.017
wa84	11012	11012	10561	0.043
wb84	6445	6445	6261	0.029
wa108	12013	12013	11755	0.022
wb108	10786	10786	10511	0.026

Table 4.5.6: 3-partition for weighted Graphs

	BEST	INIT	LB	GAP
wa36	1912	1931	1853	0.032
wb36	1708	1750	1650	0.035
wa60	5423	5427	5200	0.043
wb60	4922	4945	4751	0.036
wa84	10643	10643	10195	0.044
wb84	9632	9632	9246	0.042
wa108	17820	17820	17299	0.030
wb108	15946	15946	15461	0.031

Table 4.5.7: 4-partition for weighted Graphs

the dual problem of our SDP relaxation is very sparse. Our future work will be focused on solving the large sparse problems. We have successfully extended to GP the theoretical results and algorithm for the QAP as the SDP relaxations for both QAP and GP are similar. We expect any large scale implementation of our algorithm to apply equally well to both problems.

Chapter 5

Set Partitioning Problems

5.1 Introduction

The set partitioning problem, SP, can be described as follows.

Suppose we are given a set M with m elements; and let

$$\mathcal{M} = \{M_j : j \in N := \{1 \ldots, n\}\}$$

be a given collection of subsets of M such that the union contains M i.e., $\cup_{j \in N} M_j = M$. For each M_j, there is an associated cost c_j. We want to find a subset F of the index set N such that:

1. the union still contains M, $\cup_{j \in F} M_j = M$;
2. the sets are pairwise disjoint, $M_k \cap M_j = \phi$, for $k \neq j \in F$;
3. and the sum of the costs $\sum_{i \in F} c_j$ is minimized.

Let $A = (a_{ij})$ be the $m \times n$ matrix with

$$a_{ij} = \begin{cases} 1 & \text{if element } \ i \in M_j \\ 0 & \text{otherwise.} \end{cases}$$

The matrix A is called the *incidence matrix* of the collection $\mathcal{M}$; each column of A is the indicator vector for the set M_j. Each subset $F \subset N$, for which the collection of sets $\{M_j, j \in F\}$ satisfies 1 and 2, is called a *set partition* of the set M. For a given set partition, we let $x \in \{0,1\}^n$ defined by

$$x_j = \begin{cases} 1 & \text{if } \ j \in F \\ 0 & \text{otherwise.} \end{cases}$$

Such an x can represent the set partition.

The set partitioning problem can now be formulated as the following 0-1 integer programming problem

$$\begin{array}{llll} & \mu^* := & \min & c^t x \\ (SPT) & & \text{subject to} & Ax = e \\ & & & x \in \{0,1\}^n. \end{array}$$

Without loss generality, we assume that A has full row rank. For each $i \in \{1, 2, \ldots, m\}$, we let

$$a_i := (a_{i1}, a_{i2}, \ldots, a_{in}).$$

The ith row of the constraints, $a_i x = 1$, guarantees that the ith element is in exactly one set.

The set partitioning problem has been extensively investigated because of its special structure and its numerous practical applications. The best known application is airline crew scheduling, see e.g. the recent reference [HP93]. Other applications include: truck scheduling; bus scheduling; facility location; circuit design and capital investment. (See e.g Garfinkel and Nemhauser [GN69], Marsten [MAR74], Balas and Padberg [BP76], Balas [BAL77], Nemhauser and Weber [NW79], Fisher and Kedia [FK90] Chan and Yano [CY92] and Hoffman and Padberg [HP92].)

Since the set partitioning problem is well-known to be NP-hard, many current approaches focus on finding a "near optimal" solution using various heuristic techniques. A natural candidate for generating a lower bound is the linear programming relaxation. The linear programming relaxation is as follows

$$(SPLP) \qquad \begin{array}{rcl} \mu^*_{LP} := & \min & c^t x \\ & \text{subject to} & Ax = e \\ & & x \geq 0. \end{array}$$

To improve the approximate solution for (SPT), one can use cutting planes and/or branch-and-bound techniques in conjunction with various bound improvement techniques. (See chu and Beasley [CB95] for a literature survey on exact and heuristic algorithms for SP.) We include the following related papers in the bibliography [AFST69, BF81, BH90, GER89, HT94, RF88].

In this chapter, we develop an SDP relaxation for the set partitioning problem. In our approach, in addition to taking care of all the linear programming relaxation constraints, we employ the "gangster operator" to efficiently model the special 0-1 structure of (SPT). By this SDP approach we can generate a better lower bound for the set partitioning problem. In addition, we combine the SDP relaxation with the standard LP relaxation and take advantage of block structures in the data.

5.2 An SDP Relaxation

To derive an SDP relaxation for SP, we reformulate the 0-1 integer programming model (SPT) as a quadratically constrained quadratic programming problem.

Since the variables x_i are restricted to 0-1, we have $x_i = x_i^2$, i.e.

$$x = x \circ x.$$

In addition, since $a_i x = 1$ for each $i \in \{1, \ldots, m\}$, we have

$$\{k \neq j, \quad a_{ik} = 1, \quad a_{ij} = 1\} \Rightarrow x_k x_j = 0. \tag{5.2.47}$$

Therefore (SPT) is equivalent to the following.

$$(SPQP) \quad \begin{array}{rrl} \mu^* = & \min & c^t(x \circ x) \\ & \text{subject to} & A(x \circ x) = e \\ & & (a_i x - 1)^2 = 0, \quad \text{for} \;\; i \in \{1, 2, \ldots, m\} \\ & & (x \circ x) - x = 0 \\ & & x_k x_j = 0, \quad \text{if } k \neq j, \quad a_{ik} = 1, \quad a_{ij} = 1 \;\; \text{for some } i. \end{array}$$

By adding a scalar x_0, we can eliminate the linear terms (homogenize) in the existing constraints of the above problem.

$$(SPQPH) \quad \begin{array}{rl} \mu^* = \min & c^t(x \circ x) \\ \text{subject to} & A(x \circ x) = e \\ & (-1, a_i)(x_0, x^t)^t(x_0, x^t)(-1, a_i)^t = 0 \\ & \text{for} \;\; i \in \{1, 2, \ldots, m\} \\ & (x \circ x) - x_0 x = 0 \\ & x_k x_j = 0, \quad \text{if } k \neq j, \quad a_{ik} = 1, \quad a_{ij} = 1 \;\; \text{for some } i \\ & x_0^2 = 1. \end{array}$$

We now replace the quadratic terms with a matrix, i.e., we replace the rank one matrix $(x_0, x^t)^t(x_0, x^t)$ by the positive semidefinite matrix $Y \succeq 0$ with $Y \in \mathcal{S}_{n+1}$.

We get the following SDP relaxation.

$$
(PSDP) \qquad
\begin{array}{ll}
\min & \operatorname{trace} CY \\
\text{subject to} & \operatorname{trace}(\operatorname{Diag}(0, a_i)Y) = 1, \quad i = 1, \dots, m \\
& (-1, a_i)Y(-1, a_i)^t = 0, \qquad i = 1, \dots, m \\
& \operatorname{arrow}(Y) = 0 \\
& \mathcal{G}_J(Y) = 0 \\
& Y_{00} = 1 \\
& Y \succeq 0,
\end{array}
$$

where $C = \operatorname{Diag}(0, c^t)$ and the operator $\mathcal{G}_J$ is a gangster operator with

$$J := \{(k, j) : \text{if } a_{ik} = a_{ij} = 1 \;\; k < j \text{ for some } i\};$$

the arrow constraint represents the 0-1 constraints by guaranteeing that the diagonal and 0th column (or row) are identical; the gangster operator constraint represents constraints in (5.2.47); and, finally, the assignment constraints $Ax = e$ are represented by the first two set of constraints in (PSDP).

Define the $m \times (n+1)$ *assignment constraint matrix*

$$T := [-e, A].$$

Each feasible Y satisfies $Y \succeq 0$ and

$$(-1, a_i)Y(-1, a_i)^t = 0, \quad i = 1, \dots, m.$$

Therefore the range space and null space satisfy

$$\mathcal{R}(T^t) \subset \mathcal{N}(Y) \text{ or alternatively } \mathcal{R}(Y) \subset \mathcal{N}(T).$$

Now let the null space of T be spanned by the columns of a $(n+1) \times (n-m+1)$ matrix V, i.e., let

$$\mathcal{N}(T) = \mathcal{R}(V).$$

This implies that $Y = VXV^t$ for some $X = X^t \succeq 0$, i.e., we are able to express each feasible Y as VXV^t. In order to solve large scale problems, a sparse representation of the null space of T is useful. We use a simple technique, called *Wolfe's variable-reduction* technique [WOL62]. (For a "sparsest" representation, see e.g. [CP86].)

Without loss generality, we assume that

$$T = [T_B, T_N],$$

where T_B is a $m \times m$ matrix with full rank and T_N is a $m \times (n-m+1)$ matrix. Then, the matrix

$$V = \begin{bmatrix} -T_B^{-1}T_N \\ I_{n-m+1} \end{bmatrix}$$

satisfies $\mathcal{N}(T) = \mathcal{R}(V)$.

We now take a look at the following interesting properties of the matrix VXV^t.

Lemma 5.1 *For any arbitrary $(n-m+1) \times (n-m+1)$ symmetric matrix*

$$X = \left[\begin{array}{c|ccc} X_{00} & X_{01} & \dots & X_{0(n-m)} \\ \hline X_{10} & X_{11} & \dots & X_{1(n-m)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{(n-m)0} & X_{(n-m)1} & \dots & X_{(n-m)(n-m)} \end{array}\right],$$

let $Y = VXV^t$ and write Y as

$$Y = \left[\begin{array}{c|ccc} Y_{00} & Y_{01} & \dots & Y_{0n} \\ \hline Y_{10} & Y_{11} & \dots & Y_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n0} & Y_{n1} & \dots & Y_{nn} \end{array}\right].$$

Then

a)

$$a_i Y_{1:n,0} = Y_{00}, \quad for \quad i = 1, \ldots, m;$$

b)

$$Y_{0j} = a_i Y_{1:n,j}, \ for \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

Proof. Since

$$TY = TVXV^t = 0,$$

we have, $(-1, a_i)Y = 0$, for each $1 \leq i \leq m$. □

This shows that the first two sets of constraints in (PSDP) are redundant. Before we write our final SDP relaxation, we present another lemma which helps get rid of more redundant constraints.

Lemma 5.2 *Let* $Y = VXV^t$. *Then*

$$\mathcal{G}_J(Y) = 0 \quad \Longrightarrow \text{arrow}\,(Y) = 0.$$

Proof. Suppose $Y = VXV^t$ and $\mathcal{G}_J(Y) = 0$. Let $j \in \{1, 2, \ldots, n\}$, then there exists $i \in \{1, 2, \ldots, m\}$ such that

$$a_{ij} = 1.$$

By Lemma 5.1, we have

$$(a_{i1}, \ldots, a_{in})Y_{1:n,j} = Y_{0j}.$$

This implies that

$$Y_{jj} + \sum_{\substack{k \\ k \neq j,\ a_{ik}=1}} Y_{kj} = Y_{0j}.$$

From the definition of the gangster operator, we have

$$\sum_{k\neq j,\ a_{ik}=1} Y_{kj} = 0.$$

Therefore

$$Y_{jj} = Y_{0j}.$$

□

Now replacing Y by VXV^t in (PSDP) and getting rid of the redundant constraints, we have the following final SDP relaxation for SP. Let $\bar{J} = J \cup (0,0)$.

$$\begin{array}{llll} & \mu^*_{SDP} := & \min & \operatorname{trace} V^tCVX \\ (PSDPF) & & \text{subject to} & \mathcal{G}_{\bar{J}}(VXV^t) = E_{00} \\ & & & X \succeq 0, \end{array}$$

where $X \in \mathcal{P}_{n-m+1}$, $C = \operatorname{Diag}(0, c^t)$. The dual is

$$\begin{array}{lll} & \max & W_{00} \\ (DSDPF) & \text{subject to} & V^tWV \preceq V^tCV \\ & & W \in \mathcal{S}_{\bar{J}}. \end{array}$$

Note that the gangster operator is self adjoint and $\mathcal{G}_{\bar{J}}(\mathcal{S}) = \mathcal{S}_{\bar{J}}$.

From Lemma 5.1 and Lemma 5.2, we can immediately derive the following.

Theorem 5.1 *Let X be any feasible solution of (PSDPF). Then* $(\operatorname{diag}(VXV^t))_{1:n}$ *(the last n diagonal element of the matrix VXV^t), is a feasible solution of the linear programming relaxation (SPLP).*

Proof. Let X be a feasible solution of (PSDPF) and $Y = VXV^t$. Then $\mathcal{G}_{\bar{J}}(Y) = E_{00}$ and

$$Y_{jj} \geq 0, \quad \text{for} \quad i \in \{1, \ldots, n\}.$$

From Lemma 5.1 and Lemma 5.2, we have $Y_{:0} = \operatorname{diag}(Y)$ and $Y_{00} = 1$, and thus for each $i \in \{1, \ldots, m\}$,

$$a_i(Y_{11}, \ldots, Y_{nn})^t = a_i Y_{0j}^t = Y_{00} = 1.$$

$\square$

Based on the theorem above and the fact that the objective value of the SDP relaxation is $(0, c^t)\operatorname{diag}(VXV^t)$, the following corollary follows.

Corollary 5.1 *The lower bound given by the SDP relaxation $(PSDPF)$ is greater than or equal to the one given by the LP relaxation, i.e.* $\mu^*_{SDP} \geq \mu^*_{LP}$.

In addition, we now see that there is no duality gap between (PSDPF) and (DSDPF).

Theorem 5.2 *Problem (DSDPF) is strictly feasible.*

Proof. From Lemma 5.2, we have, for any X,

$$\mathcal{G}_J(VXV^t) = 0 \quad \Longrightarrow \operatorname{arrow}(VXV^t) = 0.$$

Therefore

$$\mathcal{N}(\mathcal{G}_J(V \cdot V^t)) \subset \mathcal{N}(\operatorname{arrow}(V \cdot V^t)).$$

(where the dot $\cdot$ represents the variables for the operators.) In other word, their adjoint operators satisfy

$$\mathcal{R}(V^t \operatorname{Arrow}(\cdot)V)) \subset \mathcal{R}(V^t \mathcal{G}_J(\cdot)V) = \mathcal{R}(V^t \cdot V).$$

Therefore, for $y = -e \in \Re^n$, there exists $W \in S_J$ such that

$$V^t \operatorname{Arrow}(y)V = V^t W V$$

and, by using Schur complements, we see that

$$V^t(-ME_{00} + W)V = V^t(-ME_{00} - \mathrm{Arrow}\,(e_n))V \prec 0,$$

for M big enough. Therefore $\beta(-ME_{00} + W)$ is strictly feasible for large enough β. $\square$

From Theorem 5.2, we know that the dual problem satisfies the Slater condition. Therefore, there is no duality gap between the primal problem (PSDPF) and the dual problem (DSDPF) and, moreover, the primal optimal value is attained. However, the primal problem (PSDPF) may not be strictly feasible. Consider the same example problem as in Chapter 3.3

$$\begin{array}{llll} x_1 & & & = 1 \\ x_1 & +x_2 & +x_3 & +x_4 = 1 \\ x_1, & x_2, & x_3, & x_4 \geq 0. \end{array}$$

Observe that the feasible set is a singleton $(1,0,0,0)^t$. Note that for this problem $n = 4$ and $m = 2$, so V is a 5×3 matrix. Thus, for any feasible solution of its final SDP relaxation $X \in \mathcal{P}_3$, the diagonal of VXV^t is $(1,1,0,0,0)^t$. This means that $\mathrm{rank}\,(VXV^t) \leq 2$, which implies that $\mathrm{rank}\,(X) \leq 2$. Therefore, the final SDP relaxation is not strictly feasible.

5.3 Numerical Testing for Small Problems

Since the final SDP relaxation is exactly the same as the one for QAP, we use the same technique as in the chapter for QAP to solve it, i.e., we use the same infeasible primal-dual interior-point algorithm as for QAP. As we have seen from the geometrical discussion above, the algorithm may have to deal with those problems whose primal SDP relaxation are not strictly feasible and whose dual SDP relaxation can not attain their optimal value. As the main purpose of our algorithm is to find a lower bound, we expect that our infeasible primal-dual interior-point algorithm can handle those problems due to the following reason.

- Because the dual problem is strictly feasible and only has inequality constraints, the line search can easily maintain dual feasibility. Therefore, a lower bound can always be obtained from the dual objective value.

The purpose of our numerical tests is to illustrate that the lower bound given by our algorithm for the SDP relaxation is better than the one given by LP relaxation. In addition, after solving the relaxation, The diagonal of VXV^t satisfies the constraints of the linear programming relaxation.

Our numerical tests for small problems are based on real data for bus scheduling problems. The results are summarized in Table 5.3.8. The columns under nrow, ncol and nzero are for the number of rows, columns and nonzero elements, respectively. The last two columns show the lower bounds by LP and SDP relaxations, respectively. A lower bounds marked with a star means that the lower bound is equal to the optimal objective value. As we can see, numerical results show that our SDP approach is very promising.

	nrow	ncol	nzero	LP	SDP
small01	14	34	108	1864	1864*
small02	16	46	139	2259	2259*
small03	27	97	234	17327	18324
small04	33	192	584	4503	4503*
small05	44	277	770	21706	21706*
tiny04	6	27	72	1035	1091
tiny01	3	6	9	17.5	25.00
tiny05	7	35	70	1215	1257

Table 5.3.8: Numerical Results

5.4 SDP Relaxation for Large Sparse Problems and Future Work

5.4.1 An SDP Relaxation with Block Structure

As we see from the introduction, the set partitioning problems are usually derived from real world problems such as scheduling problems. These problems can be of very large size ($> 10,000$) and very sparse.

Currently, an approximate solution for a large size set partitioning problem can be obtained by solving a corresponding large sparse linear programming relaxation and the information from the primal and dual optimal solutions are used to decide which columns, or sets M_j, should be chosen for the partition. Since the diagonal of an SDP solution is a feasible solution of the LP relaxation, we expect that this solution can help in making the choices. On the other hand, it is hard to solve an SDP problem of size e.g. over 10,000. In order to make SDP relaxation more competitive with LP to solve the large sparse problem, we have to find a way to exploit the sparsity of the set partitioning problem. In this section, we relax part of the variables of the set partitioning problem by SDP while we treat the others with an LP relaxation.

Consider a large sparse set partitioning problem

$$
\begin{array}{llll}
 & \mu^* = & \min & c^t x \\
(LSP) & & \text{subject to} & Ax = e \\
 & & & x \in \{0,1\}^n.
\end{array}
$$

By permuting the rows and columns of A, we can rewrite A as in the following form

$$
A = \left[\begin{array}{cccc|c}
F_1 & 0 & \dots & 0 & 0 \\
0 & F_2 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & F_k & 0 \\
\hline
G_1 & G_2 & \dots & G_k & H
\end{array}\right], \tag{5.4.48}
$$

where for each $i \in \{1, \dots, k\}$, F_i is a $m_i \times n_i$ matrix and G_i is a $m_G \times n_i$ matrix and H is a $m_G \times n_H$ matrix, and

$$
m_1 + \dots + m_k + m_G = m;\ n_1 + \dots + n_k + n_H = n.
$$

The sparsity pattern of the matrix A is illustrated in Figure 5.1.

Corresponding to each submatrix F_i, for $i \in \{1, \dots, k\}$, we define

$$
x_{B_i} = (x_{B_i}^1, \dots, x_{B_i}^{n_i})^t
$$

and

$$
x_N = (x_N^1, \dots, x_N^{n_H})^t
$$

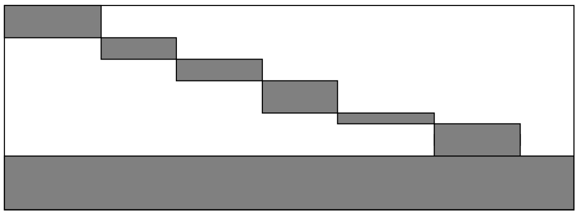

Figure 5.1: Sparsity Pattern of Matrix A

such that

$$x = \begin{pmatrix} x_{B_1} \\ \vdots \\ x_{B_k} \\ x_N \end{pmatrix}.$$

Similarly, we define

$$c_{B_i} = (c_{B_i}^1, \ldots, c_{B_i}^{n_i})^t$$

and

$$c_N = (c_N^1, \ldots, c_N^{n_H})^t$$

such that

$$c = \begin{pmatrix} c_{B_1} \\ \vdots \\ c_{B_k} \\ c_N \end{pmatrix}.$$

For each $i \in \{1, \ldots, k\}$, we write

$$F_i = \begin{bmatrix} F_i^1 \\ \vdots \\ F_i^{m_i} \end{bmatrix} = \begin{bmatrix} F_i^{11} & \ldots & F_i^{1n_i} \\ \vdots & \ddots & \vdots \\ F_i^{m_i1} & \ldots & F_i^{m_in_i} \end{bmatrix}.$$

Similarly, for each $i \in \{1, \dots, k\}$, we can write

$$G_i = \begin{bmatrix} G_i^1 \\ \vdots \\ G_i^{m_G} \end{bmatrix} = \begin{bmatrix} G_i^{11} & \dots & G_i^{1n_i} \\ \vdots & \ddots & \vdots \\ G_i^{m_G 1} & \dots & G_i^{m_G n_i} \end{bmatrix},$$

$$H = \begin{bmatrix} H^1 \\ \vdots \\ H^{m_G} \end{bmatrix} = \begin{bmatrix} H^{11} & \dots & H^{1n_H} \\ \vdots & \ddots & \vdots \\ H^{m_G 1} & \dots & H^{m_G n_H} \end{bmatrix}.$$

Now for each $i \in \{1, \dots, k\}$, we define an index set for a gangster operator.

$$J_i := \left\{ (p,q): \; p < q \text{ for some } j \; \begin{array}{c} F_i^{jp} = F_i^{jq} = 1 \text{ or} \\ G_i^{jp} = G_i^{jq} = 1 \end{array} \right\}.$$

We rewrite (LSP) as

$$\begin{array}{rcl} \mu^* = & \min & \sum_{i=1}^k c_{B_i}^t x_{B_i} + c_N^t x_N \\ & \text{subject to} & F_i x_{B_i} = e_{m_i}, \quad i \in \{1, \dots, k\} \\ & & G_1 x_{B_1} + \dots + G_k x_{B_k} + H x_N = e_{m_G} \\ & & x_{B_1}, \dots, x_{B_k}, x_N \in \{0,1\}^n. \end{array}$$

An equivalent quadratically constrained quadratic programming formulation can then be expressed as follows

$$\begin{array}{rcl} \mu^* = & \min & \sum_{i=1}^k c_{B_i}^t x_{B_i} \circ x_{B_i} + c_N^t x_N \circ x_N \\ & \text{subject to} & F_i x_{B_i} \circ x_{B_i} = e_{m_i}, \\ & & (F_i^j x_{B_i} - 1)^2 = 0, \quad \text{for } \; j \in \{1, 2, \dots, m_i\}, \\ & & x_{B_i} \circ x_{B_i} - x_{B_i} = 0, \\ & & x_{B_i}^p x_{B_i}^q = 0, \quad \text{for any pair } (p,q) \in J_i, \\ & & \text{for } \; i \in \{1, \dots, k\} \\ & & G_1 x_{B_1} \circ x_{B_1} + \dots + G_k x_{B_k} \circ x_{B_k} + H x_N \circ x_N = e. \end{array}$$

By adding, for each $i \in \{1, \ldots, k\}$, a scalar $x^0_{B_i}$, we homogenize the above problem as follows

$$
\begin{array}{rcl}
\mu^* = & \min & \sum_{i=1}^k c^t_{B_i} x_{B_i} \circ x_{B_i} + c^t_N x_N \circ x_N \\
& \text{subject to} & F_i x_{B_i} \circ x_{B_i} = e_{m_i}, \\
& & (-1, F^j_i)(x^0_{B_i}, x^t_{B_i})^t (x^0_{B_i}, x^t_{B_i})(-1, F^j_i)^t = 0, \\
& & \text{for} \ \ j \in \{1, 2, \ldots, m_i\}, \\
& & x_{B_i} \circ x_{B_i} - x^0_{B_i} x_{B_i} = 0, \\
& & (x^0_{B_i})^2 = 1, \\
& & x^p_{B_i} x^q_{B_i} = 0, \quad \text{for any pair } (p, q) \in J_i, \\
& & \text{for} \ \ i \in \{1, \ldots, k\} \\
& & G_1 x_{B_1} \circ x_{B_1} + \ldots + G_k x_{B_k} \circ x_{B_k} + H x_N \circ x_N = e.
\end{array}
$$

In the above quadratically constrained quadratic programming, we replace the rank-one matrix $(x^0_{B_i}, x^t_{B_i})^t (x^0_{B_i}, x^t_{B_i})$ by the matrix Y_i for each $i \in \{1, \ldots, k\}$, and also $X_N X^t_N$ by Y_N. Then we obtain an SDP relaxation as follows

$$
\begin{array}{rcl}
\mu^*_{LSDP} := & \min & \sum_{i=1}^k c^t_{B_i} (\operatorname{diag}(Y_i))_{1:n_i} + c^t_N \operatorname{diag}(Y_N) \\
& \text{subject to} & F_i (\operatorname{diag}(Y_i))_{1:n_i} = e_{m_i}, \\
& & (-1, F^j_i) Y_i (-1, F^j_i)^t = 0, \quad \text{for} \ \ j \in \{1, 2, \ldots, m_i\}, \\
& & \operatorname{arrow}(Y_i) = 0, \\
& & (Y_i)_{00} = 1, \\
& & \mathcal{G}_{J_i}(Y_i) = 0, \\
& & \text{for} \ \ i \in \{1, \ldots, k\} \\
& & \sum_{i=1}^k G_i (\operatorname{diag}(Y_i))_{1:n_i} + H \operatorname{diag}(Y_N) = e, \\
& & Y_1 \succeq 0, \ldots, Y_k \succeq 0, Y_N \succeq 0,
\end{array}
$$

where $Y_i \in \mathcal{P}_{n_i+1}$ for $i \in \{1, \ldots, k\}$ and $Y_N \in \mathcal{P}_{n_H}$. Since the coefficient matrices for Y_N are all diagonal, we can always write $Y_N = \operatorname{Diag}(x)$, where $x \in \Re^{n_H}, x \geq 0$.

For each $i \in \{1,\ldots,k\}$, we define an operator $\mathcal{A}_i : \mathcal{P}_{n_i+1} \to \Re^{m_G}$ such that

$$\mathcal{A}_i(Y_i) := G_i(\mathrm{diag}\,(Y_i))_{1:n_i}.$$

Then we have the following equivalent problem.

$$\begin{array}{rcl}
\mu^*_{LSDP} = & \min & \sum_{i=1}^k c_{B_i}^t(\mathrm{diag}\,(Y_i))_{1:n_i} + c_N^t x \\
& \text{subject to} & F_i(\mathrm{diag}\,(Y_i))_{1:n_i} = e_{m_i}, \\
& & (-1, F_i^j)Y_i(-1, F_i^j)^t = 0, \quad \text{for} \quad j \in \{1, 2, \ldots, m_i\}, \\
& & \mathrm{arrow}\,(Y_i) = 0, \\
& & (Y_i)_{00} = 1, \\
& & \mathcal{G}_{J_i}(Y_i) = 0, \\
& & \text{for} \quad i \in \{1,\ldots,k\} \\
& & \sum_{i=1}^k \mathcal{A}_i(Y_i) + Hx = e, \\
& & Y_{B_1} \succeq 0, \ldots, Y_{B_k} \succeq 0, x \geq 0.
\end{array}$$

For each $i \in \{1,\ldots,k\}$, we construct a $(n_i+1)\times(n_i - m_i + 1)$ matrix V_i such that the null space of $[-e_{m_i}, F_i]$ is spanned by the columns of V_i. We follow the same procedure as that in the above section, i.e., for $i \in \{1,\ldots,k\}$, we replace Y_i by $V_i X_i V_i^t$ and get rid of the redundant constraints. We denote $C_i := \mathrm{Diag}\,(0, c_{B_i}^t)$. Note that $c_{B_i}^t(\mathrm{diag}\,(Y_i))_{1:n_i} = \mathrm{trace}\,(\mathrm{Diag}\,(0, c_B^t)Y_i)$. Then we have the following final SDP relaxation.

$$(LPSDPF) \qquad \begin{array}{rcl}
\mu^*_{LSDP} = & \min & \sum_{i=1}^k \mathrm{trace}\, V_i^t C_i V_i X_i + c_N^t x \\
& \text{subject to} & \sum_{i=1}^k \mathcal{A}_i(V_i X_i V_i^t) + Hx = e_{m_G} \\
& & \mathcal{G}_{\bar{J}_i}(X_i) = E_{00}^i, \quad \text{for} \quad i \in \{1,\ldots,k\} \\
& & X_1 \succeq 0, \ldots, X_k \succeq 0, \quad x \geq 0,
\end{array}$$

where, for $i \in \{1,\ldots,k\}$, $X_i \in \mathcal{P}_{n_i - m_i + 1}$ and the operator $\mathcal{G}_{\bar{J}_i}$ is a gangster operator with

$$\bar{J}_i := \left\{ (p,q) : p < q \text{ for some } j \begin{array}{c} F_i^{jp} = F_i^{jq} = 1 \text{ or} \\ G_i^{jp} = G_i^{jq} = 1 \end{array} \right\} \cup (0,0).$$

Observe that in the final SDP relaxation $(LPSDPF)$ there are semidefinite matrix variables and nonnegative vector variables as well. Thus, we call the final SDP relaxation a *mixed LP-SDP* relaxation.

Its dual is

$$(LDSDPF) \qquad \begin{array}{rl} \max & \sum_{i=1}^{m_G} \lambda_i + \sum_{i=1}^{k} (W_i)_{00} \\ \text{subject to} & V_i^t(\operatorname{Diag}(0, \lambda^t G_i) + W_i)V_i \preceq V_i^t C_i V_i, \\ & W_i \in \mathcal{S}_{\bar{J}_i}, \\ & \text{for } \; i \in \{1, \ldots, k\} \\ & H^t \lambda \leq c, \end{array}$$

where for $i \in \{1, \ldots, k\}$, W_i and λ_i are dual variables.

For each feasible solution $(X_1, \ldots, X_k, x)$ of $(LPSDPF)$, we construct an $n \times 1$ vector

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_k \\ x \end{pmatrix}, \tag{5.4.49}$$

where $y_i = (\operatorname{diag}(V_i X_i V_i^t))_{1:n_i}$, for $i = 1, \ldots, k$. Applying the Theorem 5.1 to each block, we have $F_i y_i = e_{m_i}$ for $i = 1, \ldots, k$. Also note that $\sum_{i=1}^{k} G_i y_i + Hx = e_{m_G}$. Therefore, we have the following results.

Theorem 5.3 *Let* $(X_1, \ldots, X_k, x)$ *be any feasible solution of* $(LPSDPF)$*. Then* *the vector*

$$\begin{pmatrix} (\mathrm{diag}\,(V_1 X_1 V_1^t))_{1:n_1} \\ \vdots \\ (\mathrm{diag}\,(V_k X_k V_k^t))_{1:n_k} \\ x \end{pmatrix}$$

is a feasible solution of the linear programming relaxation $(SPLP)$.

Based on the above theorem and the fact that C_i, for $i = 1, \ldots, k$, are all diagonal matrices, the following corollary follows.

Corollary 5.2 *The lower bound given by the SDP relaxation* $(LPSDPF)$ *is great that or equal to the one given by the LP relaxation* $(SPLP)$, *i.e.,* $\mu^*_{LSDP} \geq \mu^*_{LP}$.

5.4.2 An Infeasible Primal-Dual Interior-Point Method

We rewrite the dual $(LDSDPF)$ by introducing a slack matrix Z_i for each $i \in \{1, \ldots, k\}$ and a slack vector z.

$$(LDSDPZ) \qquad \begin{array}{rl} \max & \sum_{i=1}^{m_G} \lambda_i + \sum_{i=1}^{k} (W_i)_{00} \\ \text{subject to} & V_i^t(\mathrm{Diag}\,(0, \lambda^t G_i) + W_i)V_i + Z_i = V_i^t C_i V_i, \\ & W_i \in \mathcal{S}_{\bar{J}_i}, \\ & \text{for } \; i \in \{1, \ldots, k\} \\ & H^t \lambda + z = c \\ & Z_1 \succeq 0, \ldots, Z_k \succeq 0, z \geq 0. \end{array}$$

The Karush-Kuhn-Tucker conditions of the dual log-barrier problem are

$$\begin{array}{rcccl}
\sum_{i=1}^k \mathcal{A}_i(V_i X_i V_i^t) + Hx - e_{m_G} & = & F_P^0 & = & 0 \\
\mathcal{G}_{\bar{J}_i}(V_i X_i V_i^t) - E_{00}^i & = & F_{P1}^i & = & 0, \\
\text{for } \; i \in \{1, \ldots, k\} & & & & \\
H^t\lambda + z - c & = & F_D^0 & = & 0 \\
V_i^t(\mathrm{Diag}\,(0, \lambda^t G_i) + W_i - C_i)V_i + Z_i & = & F_D^i & = & 0, \\
\text{for } \; i \in \{1, \ldots, k\} & & & & \\
z \circ x - \mu u & = & F_{ZX}^0 & = & 0 \\
Z_i X_i - \mu I & = & F_{ZX}^i & = & 0, \\
\text{for } \; i \in \{1, \ldots, k\}. & & & &
\end{array}$$

The first two equations are primal feasibility conditions, while the third and fourth are the dual feasibility conditions and the last two takes cares of complimentary slackness for X_i and Z_i and x and z, respectively. We solve this system of equations with a variant of Newton's method. We apply operators $\mathcal{A}_i$ and $\mathcal{G}_{\bar{J}_i}$ to nonsymmetric matrices and then we linearize the above system as follows.

$$\begin{array}{rcl}
\sum_{i=1}^k \mathcal{A}_i(V_i \delta X_i V_i^t) + H\delta x & = & -F_P^0 \\
\mathcal{G}_{\bar{J}_i}(V_i \delta X_i V_i^t) & = & -F_{P1}^i \\
\text{for } \; i \in \{1, \ldots, k\} & & \\
H^t \delta\lambda + \delta z & = & -F_D^0 \\
V_i^t(\mathrm{Diag}\,(0, \delta\lambda^t G_i) + \delta W_i))V_i + \delta Z_i & = & -F_D^i \\
\text{for } \; i \in \{1, \ldots, k\} & & \\
\delta z \circ x + z \circ \delta x & = & -F_{ZX}^0 \\
\delta Z_i X_i + Z_i \delta X_i & = & -F_{ZX}^i \\
\text{for } \; i \in \{1, \ldots, k\}. & &
\end{array} \tag{5.4.50}$$

From the third and fourth equations, we have, for $i \in \{1, \ldots, k\}$,

$$\delta Z_i = -F_D^i - V_i^t(\mathrm{Diag}\,(0, \delta\lambda^t G_i) + \delta W_i))V_i \tag{5.4.51}$$

and

$$\delta z = -F_D^0 - H^t \delta\lambda. \tag{5.4.52}$$

Substituting (5.4.51) and (5.4.52) into the last two equations, respectively, we have

$$\delta X_i = -Z_i^{-1} F_{ZX}^i + Z_i^{-1} F_D^i X_i + Z_i^{-1} V_i^t (\mathrm{Diag}\,(0, \delta\lambda^t G_i) + \delta W_i)) V_i X_i \tag{5.4.53}$$

and

$$\delta x = -z^{-1} \circ F_{ZX}^0 + z^{-1} \circ F_D^0 \circ x + z^{-1} \circ H^t \delta\lambda \circ x. \tag{5.4.54}$$

Substituting (5.4.53) and (5.4.54) into the first two equations, we have the following final normal equation.

$$\begin{array}{lcl} \sum_{i=1}^k \mathcal{A}_i(V_i Z_i^{-1} V_i^t (\mathrm{Diag}\,(0, \delta\lambda^t G_i) + \delta W_i) V_i X_i V_i^t) & & \\ \qquad\qquad +Hz^{-1} \circ H^t \delta\lambda \circ x & = & -F_P^0 + b_0 \\ \mathcal{G}_{\bar{J}_i}(V_i Z_i^{-1} V_i^t (\mathrm{Diag}\,(0, \delta\lambda^t G_i) + \delta W_i) V_i X_i V_i^t) & = & -F_{P1}^i + b_i \\ \qquad\qquad \text{for} \quad i \in \{1, \ldots, k\}, & & \end{array} \tag{5.4.55}$$

where

$$\begin{array}{ll} b_0 & = \sum_{i=1}^k \mathcal{A}_i(V_i(Z_i^{-1} F_{ZX}^i - Z_i^{-1} F_D^i X_i) V_i^t) + H(z^{-1} \circ F_{ZX}^0 - z^{-1} \circ F_D^0 \circ x), \\ b_i & = \mathcal{G}_{\bar{J}_i}(V_i(Z_i^{-1} F_{ZX}^i - Z_i^{-1} F_D^1 X_i) V_i^t), \\ & \text{for} \quad i \in \{1, \ldots, k\}. \end{array}$$

Denote the matrix representation of the left hand side of the normal equation by $\mathcal{K}$. The matrix $\mathcal{K}$ has a very nice sparsity structure shown in Figure 5.2, where the width of the long narrow bar is m_G which is much less than the size of the matrix.

We solve the normal equation by a preconditioned conjugate gradient method. Let $(\delta W_1^*, \ldots, \delta W_k^*, \delta\lambda^*)$ be the solution for the normal equation. By equations (5.4.51), (5.4.52), (5.4.53) and (5.4.54), we can obtain, for each $i \in \{1, \ldots, k\}$, δZ_i^*, δz_i^*, δX_i^* and δx_i^*, respectively. Finally, by symmetrizing δX_i^*, i.e.,

$$\delta X_i^* \leftarrow \frac{\delta X_i^* + (\delta X_i^*)^t}{2},$$

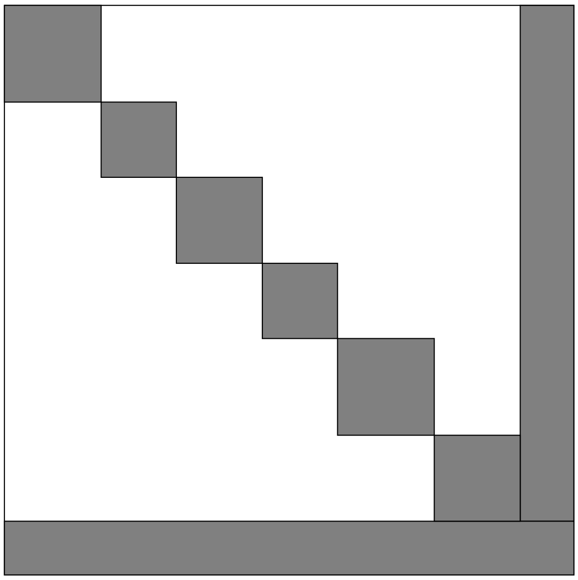

Figure 5.2: Sparsity Pattern

we obtain a search direction. We then do a line search and update the current point. Based on the duality gap, we update μ by using the following formula

$$\mu := \frac{\sum_{i=1}^{k} \operatorname{trace}(Z_i X_i) + z^t x}{2(n - m + m_G + k)}.$$

5.4.3 Preliminary Numerical Tests and Future Work

In the previous subsections, we have developed an approach for solving problems with matrix structure (5.4.48). We did some preliminary numerical tests just to see how this SDP relaxation works for small problems. In our testing, we use the diagonal of the matrix representation $\mathcal{K}$ as the preconditioner. The infeasible primal-dual interior-point algorithm for the mixed LP-SDP relaxation is coded in C and Matlab. The results are summarized in Table 5.4.9. In Table 5.4.9, the columns under nrow, ncol and nzero are for the number of rows, columns and

	nrow	ncol	nzero	LP	SDP	LP-SDP
small03	27	97	234	17327	18324	18320
tiny04	6	27	72	1037	1091	1066
tiny01	3	6	9	17.5	25	25
tiny05	7	35	70	1215	1257	1248

Table 5.4.9: Numerical Results

nonzero elements, respectively. The columns under LP and SDP show the lower bounds given by LP relaxation and SDP relaxation for a general dense problem, respectively, while the last column under LP-SDP shows the lower bounds given by our mixed LP-SDP relaxation.

For our future work, we would like to use the mixed LP-SDP relaxation to derive an approach to solve general large sparse set partitioning problems. To achieve this, we propose the following:

- to have the same matrix sparsity pattern as described for the mixed LP-SDP relaxation, the matrix for the general problem need to be transformed into form like (5.4.48). This can be done by treating the 0-1 matrix A as an incidence matrix of a graph or netlist and applying graph partitioning and netlist partitioning techniques;
- because of the nice sparsity structure as shown in Figure 5.1, more sophisticated incomplete factorization preconditioners can be used to improve the performance of primal-dual interior-point solvers, see e.g. [CHI95].

We would like to point out another future work. For a more general block structure

$$A = \left[\begin{array}{cccc|c} F_1 & 0 & \dots & 0 & H_1 \\ 0 & F_2 & \dots & 0 & H_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & F_k & H_k \\ \hline G_1 & G_2 & \dots & G_k & H \end{array}\right],$$

we should be able to develop an SDP relaxation as well. But this might involve a totally different approach since projection may not be easily applied.

Chapter 6

Summary and Discussion

In this thesis, we have developed a unified semidefinite programming relaxation approach to solve three different applications: quadratic assignment problem, graph partitioning problem and set partitioning problem. Numerical tests have shown that the bounds given by our SDP relaxations are of high quality for these three applications. This again demonstrates that semidefinite programming is really a very powerful tool for solving hard combinatorial optimization problems.

We feel that our contributions are not only in deriving better bounds for the applications using SDP but more interestingly, the SDP approach itself. Through the three different applications, we have illustrated our SDP approach for a general problem with a structure of assignment constraints. We summarize the SDP approach for a general problem with the special assignment structure in the following.

- derive a gangster operator based on the assignment constraints;
- derive some other operators, such as the arrow operator, based on the other special structure of the problem;

- generate a relative interior point for the minimal face containing all rank one feasible matrix solutions and then derive a projection matrix from the range space of this relative interior point;
- derive the final SDP relaxation by applying the projection matrix and by getting rid of the redundant constraints.

Finally we would like to point out that with the gangster operator, we can represent a combinatorial structure by a matrix sparsity structure and therefore be able to apply a lot of sparse matrix techniques to solve a large problem.

Bibliography

[AFST69] J. ARABEYRE, J. FEARNLEY, F. STEIGER, and W. TEATHER. The airline crew scheduling problem: a survey. *Transportation Science*, 2:140–163, 1969.

[AHO94] F. ALIZADEH, J-P.A. HAEBERLY, and M.L. OVERTON. A new primal-dual interior-point method for semidefinite programming. Technical report, Courant Institute of Mathematical Sciences, 1994. To appear in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, Utah, June, 1994.

[AJ94] W.P. ADAMS and T.A. JOHNSON. Improved linear programming-based lower bounds for the quadratic assignment problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16:43–75, 1994.

[ALI92] F. ALIZADEH. Combinatorial optimization with semidefinite matrices. In *Proceedings of the Second Annual Integer Programming and Combinatorial Optimization Conference*, Carnegie-Mellon University, 1992.

[ALI95] F. ALIZADEH. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[BAL77] E. BALAS. Some valid inequalities for the set partitioning problems. *Annals of Discrete Mathematics*, 1:13–47, 1977.

[BC75] G.P. BARKER and D. CARLSON. Cones of diagonally dominant matrices. *Pacific J. of Math.*, 57:15–32, 1975.

[BF81] E. BAKER and M. FISHER. Computational results for very large air crew scheduling problems. *OMEGA*, 9(6):613–618, 1981.

[BH90] J. BARUTT and T. HULL. Airline crew scheduling: aupercomputers and algorithms. *SIAM News*, 23(6), 1990.

[BKR91] R.E. BURKARD, S. KARISCH, and F. RENDL. A quadratic assignment problem library. *European Journal of Operations Research*, 55:151–119, 1991.

[BP76] E. BALAS and M.W. PADBERG. Set partitioning: A survey. *SIAM Review*, 18:710–760, 1976.

[BW81] J.M. BORWEIN and H. WOLKOWICZ. Facial reduction for a cone-convex programming problem. *J. Austr. Math. Soc. Series A*, 30:369–380, 1981.

[CB95] P. C. CHU and J. E. BEASLEY. A genetic algorithm for the set partitioning problem. Technical report, Imperial College, The Management School, London, England, 1995. URL: http:/mscmga.ms.ic.ac.uk/pchu/pchu.html.

[CHI95] PAULINA CHIN. *Iterative Algorithm for Solving Linear Programming from Engineering Applications.* PhD thesis, University of Waterloo, 1995.

[CNR68] T.E. VOLLMAN C.E. NUGENT and J. RUML. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[CP86] T.F. COLEMAN and A. POTHEN. The null space problem 1. complexity. *SIAM J. ALG. DISC. METH.*, 7:527–537, 1986.

[CY92] T. J. CHAN and C. A. YANO. A multiplier adjustment approach for the set partitioning problem. *Operations Research*, 40:S40–S47, 1992.

[DES82] R.S DEMBO, S.C. EISENSTAT, and T. STEIHAUG. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[DH73] W.E. DONATH and A.J. HOFFMAN. Lower bounds for the partitioning of graphs. *IBM J. of Research and Development*, 17:420–425, 1973.

[FK90] M. L. FISHER and P. KEDIA. Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36:674–688, 1990.

[FK95] T. FUJIE and M. KOJIMA. Semidefinite programming relaxation for nonconvex quadratic programs. manuscript, 1995.

[FLE95] R. FLETCHER. An optimal positive definite update for sparse Hessian matrices. *SIAM J. on Optimization*, 5(1):192–218, 1995.

[FRW94] J. FALKNER, F. RENDL, and H. WOLKOWICZ. A computational study of graph partitioning. *Mathematical Programming*, 66(2):211–240, 1994.

[GER89] I. GERSHKOFF. Optimizaing flight crew schedules. *Interfaces*, 19:29–43, 1989.

[GIL62] P.C. GILMORE. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal on Applied Mathematics*, 10:305–313, 1962.

[GJ79] M.R. GAREY and D.S. JOHNSON. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[GN69] R.S. GARFINKEL and G.L. NEMHAUSER. The set partitioning problem: Set covering with equality constraints. *Operations Research*, 17:848–856, 1969.

[HEL94] C. HELMBERG. *An interior point method for semidefinite programming and max-cut bounds.* PhD thesis, Graz University of Technology, Graz, Austria, 1994.

[HMV92] S.W. HADLEY, B.L. MARK, and A. VANNELLI. An efficient eigenvector approach for finding netlist partitions. *IEEE Transactions on Computer-aided Design*, 11 No. 7:885–892, 1992.

[HP92] K. L. HOFFMAN and M. PADBERG. Solving large set-partitioning problems with side constraints. *ORSA/TIMS Joint National Meeting, San Francisco*, November 2-4, 1992.

[HP93] K. L. HOFFMAN and M. PADBERG. Solving airline crew-scheduling problems by branch-and-cut. *Management Science*, 6:657–682, 1993.

[HRVW96] C. HELMBERG, F. RENDL, R.J. VANDERBEI, and H. WOLKOWICZ. An interior point method for semidefinite programming. *SIAM J. Optimization*, 6:342–361, 1996.

[HRW92a] S. W. HADLEY, F. RENDL, and H. WOLKOWICZ. A new lower bound via projection the quadratic assignment problem. *Mathematics of Operations Research*, 17:727–739, 1992.

[HRW92b] S. W. HADLEY, F. RENDL, and H. WOLKOWICZ. Symmetrization of nonsymmetric quadratic assignment problems and the Hoffman-Wielandt inequality. *Linear Algebra and its Applications*, 167:53–64, 1992.

[HT94] F. HARCHE and G.L. THOMPSON. The column subtraction algorithm: an exact method for solving weighted set covering, packing and partitioning problems. *Computers & operations Research*, 21(6):689–705, 1994.

[ING89] L. INGBER. Very fast simulated re-annealing. *Mathl. Comput. Modeling*, 12:967–973, 1989.

[KAR95] S. KARISCH. *Nonlinear Approaches for Quadratic Assignment and Graph Partition Problems.* PhD thesis, Graz University of Technology, Graz, Austria, 1995.

[KL70] B.W. KERNIGHAN and S. LIN. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.*, 49:291–307, 1970.

[KLE57] V.L. KLEE. Extemal structure of convex sets. *Arch. Math.*, 8:234–240, 1957.

[KR94] S. KARISCH and F. RENDL. Semidefinite programming and graph equipartition. Technical report, Technical Univerisity of Graz, Austria, 1994.

[KSH94] M. KOJIMA, S. SHINDOH, and S. HARA. Interior-point methods for the monotone linear complementarity problem in symmetric matrices. Technical report, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1994.

[LAW63] E. LAWLER. The quadratic assignment problem. *Management Science*, 9:586–599, 1963.

[LEN90] T. LENGAUER. *Combinatorial algorithms for integrated circuit layout.* John Wiley and Sons, Chicester, 1990.

[LP95] M. LAURENT and S. POLJAK. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224:439–461, 1995.

[LPR93] Y. Li, P.M. PARDALOS, and M.G.C. RESENDE. A greedy randomized adaptive search procedure for the quadratic assignment problem. Technical report, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1993.

[MAR74] R.E. MARSTEN. An algorithm for large set partitioning problems. *Management Science*, 20:774–787, 1974.

[MON95] R.D.C. MONTEIRO. Primal-dual algorithms for semidefinite programming. Technical report, Georgia Tech, Atlanta, GA, 1995.

[NN93] Y. E. NESTEROV and A. S. NEMIROVSKY. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

[NT94] Y. E. NESTEROV and M. TODD. Self-scaled cones and interior-point methods in nonlinear programming. Technical report, School of OR and IE, Cornell University, Ithaca, NY, 1994.

[NT95] Y. E. NESTEROV and M. TODD. Primal-dual interior-point methods for self-scaled cones. Technical Report TR1125, School of OR and IE, Cornell University, Ithaca, NY, 1995.

[NW79] G.L. NEMHAUSER and G. M. WEBER. Optimal set partitioning, matchings and lagrangean duality. *Naval Research Logistics Quarterly*, 26:553–563, 1979.

[PAT94a] G. PATAKI. Algorithms for cone-optimization problems and semidefinite programming. Technical report, Graduate School of Industrial Administration, Carnegie Mellon University, 1994.

[PAT94b] G. PATAKI. On the facial structure of cone-LP's and semi-definite programs. Technical report, Graduate School of Industrial Administration, Carnegie Mellon University, 1994.

[PAT96] G. PATAKI. *Cone programming and eigenvalue optimization: geometry and algorithms*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

[PRVJ94] L.F. PORTUGAL, M.G.C. RESENDE, G. VEIGA, and J.J. JUDICE. A truncated primal-infeasible dual-feasible network interior point method. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, USA, 1994.

[PRW95] S. POLJAK, F. RENDL, and H. WOLKOWICZ. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[RAM93] M.V. RAMANA. *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*. PhD thesis, The Johns Hopkins University, Baltimore, MD, 1993.

[RF88] D.M. RYAN and J.C. FALKNER. On the integer properties of scheduling and set partitioning models. *European Journal of Operational Research*, 35:422–456, 1988.

[ROC70] R.T. ROCKAFELLAR. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[RRD94] M.G.C. RESENDE, K.G. RAMAKRISHNAN, and Z. DREZNER. Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, USA, 1994.

[RTW95] M. RAMANA, L. TUNCEL, and H. WOLKOWICZ. Strong duality for semidefinite programming. Technical Report CORR Report 95-12, University of Waterloo, Waterloo, Canada, 1995. to appear in SIAM J. Optimization 1997.

[RVW95] F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems. *Optimization Methods and Software*, 5:1–16, 1995.

[RW92] F. RENDL and H. WOLKOWICZ. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming*, 53:63–78, 1992.

[RW95a] F. RENDL and H. WOLKOWICZ. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research*, 58:155–180, 1995.

[RW95b] F. RENDL and H. WOLKOWICZ. A semidefinite framework for trust region subproblems with applications to large scale minimization. Technical Report CORR Report 94-32, University of Waterloo, Waterloo, Canada, 1995. accepted in Math. Progr. subject to revision.

[SG76] S. SAHNI and T. GONZALEZ. P-complete approximation problems. *Journal of the Association for Computing Machinery*, 23:555–565, 1976.

[TOI77] PH.L. TOINT. On sparse and symmetric matrix updating subject to a linear equation. *Math. Comp.*, 31:954–961, 1977.

[VB93] L. VANDENBERGHE and S. BOYD. A polynomial-time algorithm for determining quadratic Lyapunov functions for nonlinear systems. In *Eur. Conf. Circuit Th. and Design*, pages 1065–1068, 1993.

[VB95] L. VANDENBERGHE and S. BOYD. A primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming*, 69:205–236, 1995.

[VB96] L. VANDENBERGHE and S. BOYD. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[VC93] R.J. VANDERBEI and T.J. CARPENTER. Symmetric indefinite systems for interior-point methods. *Mathematical Programming*, 58:1–32, 1993.

[WOL62] P. WOLFE. The reduced gradient method. Unpublished manuscript, 1962.

[WOL81] H. WOLKOWICZ. Some applications of optimization in matrix theory. *Linear Algebra and its Applications*, 40:101–118, 1981.

[ZHA95] Y. ZHANG. Basic equalities and inequalities in primal-dual interior-point methods for semidefinite programming. Technical report, Department of Mathematics and Statistics, University of Maryland Baltimore County, 1995.