TILBURG UNIVERSITY

Bundle methods in combinatorial optimization

Sotirov, R.

Publication date:
2003

Link to publication

Citation for published version (APA):
Sotirov, R. (2003). *Bundle methods in combinatorial optimization*. Klagenfurt: Department of Mathematics, University of Klagenfurt, Austria.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright, please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 17. Nov. 2019

Renata Sotirov

Bundle Methods in Combinatorial Optimization

DISSERTATION

zur Erlangung des akademischen Grades
Doktorin der Technischen Wissenschaften

Universität Klagenfurt
Fakultät für Wirtschaftswissenschaften und Informatik

1. Begutachter: Univ. Prof. Dipl. Ing. Dr. Franz Rendl
   Institut für Mathematik
2. Begutachter: Univ. Prof. Dr. Johannes Schoißengeier
   Institut für Mathematik

Juni 2003

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, daß ich die vorliegende Schrift verfaßt und alle ihr vorausgehenden oder Sie begleitenden Arbeiten durchgeführt habe. Die in der Schrift verwendete Literatur sowie das Ausmaß der mir im gesamten Arbeitsvorgang gewährten Unterstützung sind ausnahmslos angegeben. Die Schrift is noch keiner anderen Prüfungsbehörde vorgelegt worden.

Dipl.-Ing. Renata Sotirov

Klagenfurt, Juni 2003.

Abstract

Semidefinite Programming (SDP) has recently turned out to be a very powerful tool for approximating some NP-hard problems. The nature of the Quadratic Assignment Problem (QAP) suggests SDP as a way to derive tractable relaxation.

We present several Semidefinite Programming relaxations of the QAP with the increasing levels of complexity, that are formulated in matrix spaces of different dimensions. We also use a representation of a permutation matrix in the lifted space, which allows to exploit sparsity in a simple way, and which is smaller than one for the standard SDP relaxations for QAP. The trade off between strength of the bounds and time needed for solving them is presented.

All results are computed using the Interior Point Method and/or the Bundle Method. The Bundle Method turns out to be a very favorable method for solving large combinatorial optimization problems. The method allows the selection of important constraints from the given model, which are treated indirectly using Lagrangian duality.

The computational results demonstrate the efficiency of the approach. Our bounds are the currently strongest ones available for QAP. We investigate their potential for Branch and Bound settings by looking at the bounds in the first and second level of the branching tree.

Acknowledgement

I am extremely grateful to my supervisor Professor Franz Rendl, for his excellent guidance and support throughout my Ph.D. studies. Professor's ideas and advises have been the foundations of the research leading to this thesis. I thank him also for readiness to discuss anytime I needed, as well as for the generously given time. His persistence and patience resulted as well with my German knowledge. Many thanks for that also.

I am grateful to Professor Henry Wolkowicz for the invitations to visit the University of Waterloo. My two visits were, due to many discussions and work, very productive and enjoyable.

I would also like to thank our system engineer Gerald Hochegger for help on numerous occasions concerning university machines, and for making them available for the big computations.

I gratefully acknowledge financial support provided by Prof. Franz Rendl's research grant from the Austrian Science Foundation (Fonds zur Förderung der wissenschaftlichen Forschung FWF) under Project P12660-MAT, and Prof. Gerhard J. Woeginger's research grant from the START program Y43–MAT of the Austrian Ministry of Science. I also gratefully recognize financial support in the form of teaching assistantships (two semesters) from the Department of Mathematics, and University of Klagenfurt for the Dr. Manfred Gehring research fellowship.

Finally, I would like to thank my parents for their encouragement and support through my graduate studies. They were although very far – nevertheless very close.

Renata Sotirov

Contents

List of Tables

List of Figures

List of Symbols

$\mathcal{A}(\cdot)$ ..... the linear operator

$\mathcal{A}^T(\cdot)$ ..... the adjoint of the linear operator $\mathcal{A}(\cdot)$

$A_{i,\cdot}$ ..... the ith row of A

$A_{\cdot,j}$ ..... the jth column of A

$A > B$ ..... $a_{ij} > b_{ij}\ \forall i, j$

$A \geq B$ ..... $a_{ij} \geq b_{ij},\ \forall i, j$

$A \circ B$ ..... the Hadamard product of A and B

$A \otimes B$ ..... the Kronecker product of A and B

$\succeq$ ..... the Löwner partial order

$A \succ B$ ..... $A - B$ is positive definite

$A \succeq B$ ..... $A - B$ is positive semidefinite

arrow$(\cdot)$ ..... the arrow operator

Arrow$(\cdot)$ ..... the adjoint operator of the arrow operator

diag(X) ..... the vector of the diagonal elements of the matrix X

Diag(x) ..... the diagonal matrix with diagonal x

dim(S) ..... dimension of the space S

e_i ..... the ith unit vector

e ..... the vector of ones

E ..... the matrix of ones

$\mathcal{E}$ ..... the set of matrices with row and column sums one

G_J ..... the Gangster operator on $\mathcal{S}_{n^2+1}$

G_S ..... the Gangster operator on $\mathcal{S}_{(n-1)^2+1}$

$\mathcal{G}$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $G_J(\hat{V}R\hat{V}^T) = 0$

$\mathcal{G}_S$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $G_S(R) = 0$

$\text{int}(S)$ ..... the interior of S

I ..... the identity matrix

$\text{Le}(\cdot)$ ..... the operator defined on page 72

$\text{Le}^T(\cdot)$ ..... the adjoint operator of the operator Le

$\mathcal{L}$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $\text{Le}(R) = -(n-1)(n-2)$

$\text{Lm}(\cdot)$ ..... the operator defined on page 70

$\text{Lm}^T(\cdot)$ ..... the adjoint operator of the operator Lm

$\mathcal{M}_k$ ..... the space of $k \times k$ real matrices

$\mathcal{M}_{mk}$ ..... the space of $m \times k$ real matrices

N ..... the operator defined as $(N(Y))_{ij} = Y_{ij}$

$\mathcal{N}$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $N(\hat{V}R\hat{V}^T) \geq 0$

$\mathcal{N}_S$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $N_S(R) \geq 0$

$\mathcal{N}ull(M)$ ..... the null space of M

$\text{offDiag}\,(S)$ ..... operator sets the diagonal of S to zero

$\mathcal{O}$ ..... the set of orthogonal matrices

$\mathcal{P}$ ..... the feasible set of QAP

$\hat{\mathcal{P}}$ ..... the set containing $\mathcal{P}$

$\mathcal{P}_S$ ..... the set defined on page 69

Π ..... the set of permutation matrices

$\mathcal{R}$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $R \succeq 0$ and $\text{arrow}(\hat{V} R \hat{V}^T) = e_0$

$\mathcal{R}_S$ ..... the set of $R \in \mathcal{S}_{(n-1)^2+1}$ s.t. $R \succeq 0$ and $\text{arrow}(R) = e_0$

$\mathbb{R}^k$ ..... the space of k-dimensional vectors

$\mathcal{R}ange(A)$ ..... the range space of A

$\text{rank}(A)$ ..... the rank of a matrix A

$\mathcal{S}_k$ ..... the space of real symmetric $k \times k$ matrices

$\mathcal{S}_k^+$ ..... the set of positive semidefinite matrices

$\mathcal{S}_k^{++}$ ..... the set of positive definite matrices

$\text{tr}(A)$ ..... the trace of a square matrix A

$\langle \cdot, \cdot \rangle$ ..... the trace inner product on $\mathcal{M}_{m,k}$

$||A||_F$ ..... the Frobenius norm of A

$\text{vec}(X)$ ..... the vector formed from the columns of the matrix X

$\mathcal{Z}$ ..... the set of $(0,1)$-matrices

$\begin{pmatrix} n+1 \\ 2 \end{pmatrix}$ ..... the dimension of $\mathcal{S}_n$

Chapter 1

Preliminaries and Notation

In this chapter we give some basic notation and preliminary concepts that will be used throughout the thesis.

1.1 Matrices

We denote the space of $m \times k$ (resp. $k \times k$) real matrices by $\mathcal{M}_{m,k}$ (resp. $\mathcal{M}_k$). We use $\mathrm{tr}(A)$ to denote the *trace* of a square matrix $A \in \mathcal{M}_k$, where

$$\mathrm{tr}(A) = \sum_{i=1}^{k} a_{ii} = \sum_{i=1}^{k} \lambda_i,$$

where λ_i are eigenvalues of $A = (a_{ij})$. For $A \in \mathcal{M}_{m,k},\ B \in \mathcal{M}_{k,m}$ we get

$$\mathrm{tr}(AB) = \mathrm{tr}(BA).$$

The space $\mathcal{M}_{m,k}$ is considered with the *trace inner product*. For $A, B \in \mathcal{M}_{m,k}$ the trace inner product is

$$\langle A, B \rangle = \mathrm{tr}(B^T A) = \sum_{i=1}^{m} \sum_{j=1}^{k} a_{ij} b_{ij}.$$

The norm associated with the trace inner product is the *Frobenius norm*

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

For two matrices $A, B \in \mathcal{M}_k$, $A \geq B$, $(A > B)$ means $a_{ij} \geq b_{ij}$, $(a_{ij} > b_{ij})$ for all i, j.

We say that $A \in \mathcal{M}_k$ is *symmetric* if

$$A = A^T.$$

We denote the space of *real symmetric $k \times k$ matrices* by $\mathcal{S}_k$. The dimension of $\mathcal{S}_k$ is

$$\dim \mathcal{S}_k = \frac{1}{2}k(k+1) =: \left(\begin{array}{c} k+1 \\ 2 \end{array} \right).$$

All eigenvalues of $A \in \mathcal{S}_k$ are real numbers.

Definition 1.1 *(Positive definite and semidefinite matrix)*
$A \in \mathcal{S}_k$ *is* positive semidefinite *$(A \succeq 0)$ if* $x^T A x \geq 0, \ \forall x \in I\!R^k$.
$A \in \mathcal{S}_k$ *is* positive definite *$(A \succ 0)$ if* $x^T A x > 0, \ \forall x \in I\!R^k \backslash \{0\}$.

The matrix $A \in \mathcal{S}_k$ is positive semidefinite if and only if all eigenvalues of A are real and greater than or equal to zero. The matrix $A \in \mathcal{S}_k$ is positive definite if and only if all eigenvalues of A are real and greater than zero. The space $\mathcal{S}_k$ is equipped with the *Löwner partial order*, i.e. $A \succ B$ (resp. $\succeq$) denotes $A - B$ is positive definite (resp. positive semidefinite). We define now the *set of positive semidefinite matrices*

$$\mathcal{S}_k^+ := \{A \in \mathcal{S}_k : A \succeq 0\},$$

and the *set of positive definite matrices*

$$\mathcal{S}_k^{++} := \{A \in \mathcal{S}_k : A \succ 0\}.$$

Remark 1.1 *If $A \in \mathcal{S}_k^+$ and $a_{ii} = 0$ for some $i \in \{1, \ldots, k\}$ then $a_{ij} = 0, \ \forall j \in \{1, \ldots, k\}$.*

More about symmetric matrices can be found in Appendix A1.

We partition a symmetric matrix $Y \in \mathcal{S}_{n^2+1}$ into blocks as follows.

$$Y = \left[\begin{array}{c|c} y_{00} & Y_0^T \\ \hline Y_0 & Z \end{array} \right] = \left[\begin{array}{c|ccc} y_{00} & Y^{01} & \ldots & Y^{0n} \\ \hline Y^{10} & Y^{11} & \ldots & Y^{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Y^{n0} & Y^{n1} & \ldots & Y^{nn} \end{array} \right], \tag{1.1}$$

where we use the index 0 for the first row and column. Hence $Y_0 \in \mathbb{R}^{n^2}$, $Z \in \mathcal{S}_{n^2}$, $Y^{p0} \in \mathbb{R}^n$, and $Y^{pq} \in \mathcal{M}_n$. When referring to entry $r, s \in \{1, 2, \ldots, n^2\}$ of Z, we also use the pairs $(i, j), (k, l)$ with $i, j, k, l \in \{1, 2, \ldots, n\}$. This identifies the element in row $r = (i-1)n + j$ and column $s = (k-1)n + l$ by $Y_{(i,j),(k,l)}$. This notation is going to simplify both the modeling and the presentation of properties of the relaxations. If we consider Z as a matrix consisting of $n \times n$ blocks Y^{ik}, then $Y_{(i,j),(k,l)}$ is just element (j, l) of block (i, k).

We use e_i to denote the ith *unit vector*, e is the *vector of ones*. When there is no confusion with the unit vector, we use e_k to indicate the size of the vector of all ones. The matrix E_k is a $k \times k$ matrix with all its entries being equal to one, and I_k is a $k \times k$ *identity matrix*. We use E and I when there is no ambiguity.

We use $\mathcal{O}$ to denote the *set of orthogonal matrices*, i.e.

$$\mathcal{O} := \{X : \; XX^T = X^T X = I\},$$

$\mathcal{E}$ to denote the *set of matrices with row and column sums one*, called the set of *assignment constraints*, i.e.

$$\mathcal{E} := \{X : \; Xe = X^T e = e\},$$

and $\mathcal{Z}$ the *set of* $(0, 1)$*-matrices*, i.e.

$$\mathcal{Z} := \{X : \; X_{ij} \in \{0, 1\}\}.$$

Definition 1.2 *Let $X = (X_{ij})$ be a $k \times k$ matrix. If the entries x_{ij} fulfill the following conditions*

$$\begin{aligned} \sum_{i=1}^{n} x_{ij} &= 1, \quad 1 \le j \le k \\ \sum_{j=1}^{n} x_{ij} &= 1, \quad 1 \le j \le k \\ x_{ij} &\in \{0, 1\}, \quad 1 \le i, j \le k, \end{aligned}$$

then X is called a permutation matrix. *The* set of all $k \times k$ permutation matrices *is denoted by* Π.

Premultiplication (resp. Postmultiplication) of a matrix with a permutation matrix results in a matrix with rows (resp. columns) permuted.

The following is known for permutation matrices, see [62, 46].

1. $\Pi = \mathcal{E} \cap \mathcal{Z} = \mathcal{O} \cap \mathcal{Z}$.
2. The determinant of a permutation matrix is ± 1.
3. The product of two permutation matrices is again a permutation matrix.
4. A is $k \times k$ doubly stochastic $\Leftrightarrow$ for some $m \in \mathbb{N}$, there exist $k \times k$ permutation matrices $P_1, \ldots, P_m$ and $c_1, \ldots, c_m \in \mathbb{R}$, $c_i \geq 0$ with $c_1 + \ldots + c_m = 1$, such that $A = c_1 P_1 + \ldots + c_m P_m$.

We also need some notation to be able to refer to certain elements or parts of matrices. The notation that we use is similar to the syntax of MATLAB. For $A \in \mathcal{M}_{n,k}$ the ith row of A is denoted by $A_{i,\cdot}$ and accordingly we denote the jth column by $A_{\cdot,j}$.

1.2 Operators

Here we define the most important operators that appear in the thesis. For a linear operator $\mathcal{A} : \mathbb{R}^k \to \mathbb{R}^m$, the *adjoint operator* of $\mathcal{A}$, denoted $\mathcal{A}^*$ is a linear operator mapping from $\mathbb{R}^m$ to $\mathbb{R}^k$ such that for any $x \in \mathbb{R}^k$ and any $y \in \mathbb{R}^m$,

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

The range space of $\mathcal{A}^*$ is orthogonal to the nullspace of $\mathcal{A}$. If $\mathcal{A}$ and $\mathcal{A}^*$ are written as matrices then $\mathcal{A}^* = \mathcal{A}^T$.

For $X \in \mathcal{M}_k$, $\mathrm{vec}(X)$ denotes the vector in $\mathbb{R}^{k^2}$ that is formed from the columns of the matrix X. More precisely, the operator $\mathrm{vec} : \mathcal{M}_k \to \mathbb{R}^{k^2}$ is defined as

$$\mathrm{vec}(X) = \begin{bmatrix} X_{\cdot,1} \\ \vdots \\ X_{\cdot,k} \end{bmatrix}.$$

The connection between operators vec and tr is given with the following relation; see e. g. [32].

$$\mathrm{tr}(AB) = (\mathrm{vec}(A^T))^T \mathrm{vec} B, \quad A, B \in \mathcal{M}_k. \tag{1.2}$$

The operator Diag maps $\mathbb{R}^k$ to $\mathcal{M}_k$, and for some $x \in \mathbb{R}^k$, $\mathrm{Diag}(x)$ is the diagonal matrix with diagonal entries equal to the components of x.

Conversely, $\mathrm{diag}(X)$ is the vector of the diagonal elements of the matrix X. $\mathrm{Diag}(x)$ is the adjoint operator of $\mathrm{diag}(X)$.

The *Hadamard product* is a map $\circ : \mathcal{M}_{m,k} \times \mathcal{M}_{m,k} \to \mathcal{M}_{m,k}$ which is defined as

$$(A \circ B)_{ij} := a_{ij} \cdot b_{ij}, \quad \forall i, j.$$

The *Kronecker product* is a map $\otimes : \mathcal{M}_{m,k} \times \mathcal{M}_{p,q} \to \mathcal{M}_{mp,kq}$ which is defined as

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1k}B \\ a_{21}B & a_{22}B & \dots & a_{2k}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mk}B \end{bmatrix}.$$

Let A, B, C, D be matrices of appropriate size. The following identities are known in matrix analysis, see e. g. [32],

$$(A \otimes B)^T = A^T \otimes B^T \tag{1.3}$$

$$\mathrm{tr}(A \otimes B) = \mathrm{tr}(A)\mathrm{tr}(B) \tag{1.4}$$

$$\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X) \tag{1.5}$$

$$\mathrm{tr}(ABCD) = \mathrm{vec}(D^T)^T (C^T \otimes A)\mathrm{vec}(B). \tag{1.6}$$

Let $J \subset \{(i,j) : 1 \le i, j \le n^2+1\}$. The operator $G_J : \mathcal{S}_{n^2+1} \to \mathcal{S}_{n^2+1}$ defined as

$$(G_J(Y))_{ij} := \begin{cases} Y_{ij} & \text{if } (i,j) \in J \\ 0 & \text{otherwise,} \end{cases} \tag{1.7}$$

is called the *Gangster operator*, and the set J the *set of the gangster indices.* The name of the Gangster operator was introduced in [90]. We keep the name, even though we feel that it is not quite appropriate. We denote the subspace of $(n^2+1) \times (n^2+1)$ symmetric matrices with nonzero index set J with S_J;

$$S_J := \{X \in \mathcal{S}_{n^2+1} : X_{ij} = 0 \text{ if } (i,j) \neq J\}. \tag{1.8}$$

Note that

$$\mathcal{R}ange(G_J(\cdot)) = S_J \text{ and } \mathcal{N}ull(G_{-J}(\cdot)) = S_{-J},$$

where $-J$ denotes the complement of J. The adjoint equation

$$\mathrm{tr}(G_J^*(Z)Y) = \mathrm{tr}(ZG_J(Y))$$

implies that the gangster operator is *self-adjoint*, i.e.,

$$G_J = G_J^*.$$

Chapter 2

Semidefinite Programming

Linearized models appear in many real–world applications, and such models describe key features of a problem quite accurately. Semidefinite programming (SDP) is an extension of linear programming where the nonnegativity constraints are replaced by positive semidefiniteness on the matrix variables. The practice shows that semidefinite models are sometimes significantly stronger than purely linear ones. The algorithmic ideas can be extended quite naturally from linear to semidefinite optimization.

The theory of semidefinite programming has been studied already by Bellman and Fan [11] in the 1960. An explicit use of semidefinite programming in combinatorial optimization appeared in the seminal work of Lovász [61] on the so called theta function, in the late 70's. Lately, there has been much interest in the area of SDP because of applications in combinatorial optimization [59, 58, 97] and in control theory [25, 13], and also because of the development of efficient interior point algorithms.

In Section 2.1 we introduce the standard formulation of a primal semidefinite program (PSDP) and derive its dual (DSDP). In Section 2.2 we explain the duality theory. The condition that implies uniqueness of the primal and dual solution, known as nondegeneracy is described in Section 2.3. In Section 2.4 and 2.6 we describe some primal dual interior point methods and search directions. A short overview on the predictor–corrector interior-point method is given in Section 2.5.

2.1 The Semidefinite Programming Problem

Semidefinite programming is a special case of convex programming where the feasible region is an affine subspace of the cone of positive semidefinite matrices. In comparison to standard linear programming, the vector $x \in \mathbb{R}^n_+$ of variables is replaced by a matrix variable $R \in \mathcal{S}^+_n$. Let L be a given symmetric $n \times n$ matrix, and $a \in \mathbb{R}^m$. We consider the following semidefinite programming problem in the variable $R \in \mathcal{S}_n$.

$$\begin{array}{rll} & \mu^* := \min & \langle L, R\rangle \\ \text{(PSDP)} & & \mathcal{A}(R) = a \\ & & R \succeq 0, \end{array}$$

where $\mathcal{A} : \mathcal{S}_n \to \mathbb{R}^m$ is a linear operator on the space of the symmetric matrices. The linear operator $\mathcal{A}$ acting on $R \in \mathcal{S}_n$ can be expressed explicitly by the following vector

$$\mathcal{A}(R) = \begin{pmatrix} \langle A_1, R\rangle \\ \vdots \\ \langle A_m, R\rangle \end{pmatrix},$$

where $A_i \in \mathcal{S}_n$ for $i = 1, \ldots, m$. The adjoint operator $\mathcal{A}^T$ is satisfying the *adjoint relation*

$$\langle \mathcal{A}(R), w\rangle = \langle R, \mathcal{A}^T(w)\rangle,$$

for all $R \in \mathcal{S}_n$ and $w \in \mathbb{R}^m$. Since

$$\langle \mathcal{A}(R), w\rangle = \sum_{i=1}^m w_i \mathrm{tr}(A_i R) = \mathrm{tr}(R \sum_{i=1}^m w_i A_i) = \langle R, \mathcal{A}^T(w)\rangle,$$

we obtain

$$\mathcal{A}^T(w) = \sum_{i=1}^m w_i A_i.$$

We will derive the dual of (PSDP) by introducing the Lagrange multiplier $w \in \mathbb{R}^m$ for the equality constraint $\mathcal{A}(R) = a$, and by using the the minimax inequality (see Lemma C.1). The Lagrangian for (PSDP) is

$$\mathcal{L}(R, w) = \langle L, R\rangle + \langle a - \mathcal{A}(R), w\rangle$$

and thus

$$\max_w \ \mathcal{L}(R,w) = \begin{cases} \langle L, R\rangle & \text{if} \quad \mathcal{A}(R) = a \\ +\infty & \text{otherwise.} \end{cases}$$

Therefore, the primal problem is equivalent to

$$\mu^* = \min_{R\succeq 0} \max_w \ \mathcal{L}(R,w).$$

By interchanging the max and the min, and using the minimax inequality we get

$$\mu^* \geq \max_w \min_{R\succeq 0} \ \mathcal{L}(R,w).$$

We can rewrite the Lagrangian as $\mathcal{L}(R,w) = \langle a, w\rangle + \langle L - \mathcal{A}^T(w), R\rangle$. From Corollary A.1 it follows that

$$\min_{R\succeq 0} \ \mathcal{L}(R,w) = \begin{cases} \langle a, w\rangle & \text{if} \quad L - \mathcal{A}^T(w) \succeq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

By introducing the dual slack variable Z, the dual problem of (PSDP) is

$$\text{(DSDP)} \qquad \begin{array}{ll} \nu^* := \max & \langle a, w\rangle \\ & L - \mathcal{A}^T(w) = Z \\ & Z \succeq 0, \end{array}$$

for $w \in I\!R^m$.

2.2 Duality Theory

The *gap* between a primal feasible solution R and a dual feasible solution (w, Z), called a *duality gap*, is defined as the difference between the primal objective and the dual objective value,

$$\langle L, R\rangle - \langle a, w\rangle = \langle Z + \mathcal{A}^T(w), R\rangle - \langle \mathcal{A}(R), w\rangle = \langle Z, R\rangle \geq 0. \qquad (2.1)$$

The inequality in (2.1) follows from Lemma A.2.

Lemma 2.1 *(Weak Duality)*
Let $R \in \mathcal{S}_n^+$ and $w \in I\!R^m$ be given with $\mathcal{A}(R) = a$, $L - \mathcal{A}^T(w) \succeq 0$. Then, $\langle L, R\rangle \geq \langle a, w\rangle$.

Weak duality provides *lower bounds* on the optimal value of the primal program. If $\langle Z, R\rangle \geq 0$ turns out to be zero then this primal–dual pair is an *optimal solution*. The crucial issue in duality theory consists in identifying sufficient conditions that insure zero duality gap, also called *strong duality*. The *Slater condition* (see [42]) is an example of a sufficient condition to strong duality.

Definition 2.1 *(Feasibility)*
A point $R \succeq 0$ is feasible for (PSDP) if $\mathcal{A}(R) = a$.
A pair $(w, Z),\ Z \succeq 0$ is feasible for (DSDP) if $L = \mathcal{A}^T(w) + Z$.

Definition 2.2 *(Strict feasibility)*
A point R is strictly feasible for (PSDP) if it is feasible for (PSDP) and satisfies $R \succ 0$.
A pair (w, Z) is strictly feasible for (DSDP) if it is feasible for (DSDP) and satisfies $Z \succ 0$.

Definition 2.3 *(Slater constraint qualification)*
(PSDP) satisfies the Slater condition if there exists a strictly feasible point R for (PSDP).
(DSDP) satisfies the Slater condition if there exists a strictly feasible pair (w, Z) for (DSDP).

If the Slater condition does not hold, then a duality gap $\mu^* > \nu^*$ can exist, and/or the dual (or primal) optimal value may not be attained.

Example 2.1 *Let*

$$\begin{array}{ll} \min & 3x_{23} \\ s.t. & \begin{bmatrix} x_{23}+3 & 0 & 0 \\ 0 & 0 & x_{23} \\ 0 & x_{23} & x_{33} \end{bmatrix} \succeq 0, \end{array}$$

be the primal semidefinite program. We determine the dual program by writ-

ing the cost function and constraints in a matrix form.

$$\begin{aligned}
\min \quad & \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{3}{2} \\ 0 & \frac{3}{2} & 0 \end{bmatrix}, X \right\rangle \\
s.t. \quad & \left\langle \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X \right\rangle = 0 \\
& \left\langle \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, X \right\rangle = 0 \\
& \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X \right\rangle = 0 \\
& \left\langle \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}, X \right\rangle = 3.
\end{aligned}$$

The dual program is

$$\begin{aligned}
\max \quad & 3y_4 \\
\text{s.t.} \quad & Z = \begin{bmatrix} -y_4 & -y_1 & -y_2 \\ -y_1 & -y_3 & \frac{3+y_4}{2} \\ -y_2 & \frac{3+y_4}{2} & 0 \end{bmatrix} \succeq 0.
\end{aligned}$$

Since $x_{22} = 0$, a necessary condition for the primal matrix to be positive semidefinite is that $x_{23} = 0$ (see Remark 1.1). Following the same idea we obtain from $z_{33} = 0$ that $z_{32} = 0$, and hence $y_4 = -3$ in the dual program. Since $\mu^* = 0$ and $\nu^* = -9$, the duality gap is nine.

Duffin [21] shows the following result.

Theorem 2.1 *(i) If (PSDP) satisfies the Slater constraint qualification and μ^* is finite, then $\nu^* = \mu^*$ is attained for (DSDP).*
(ii) If (DSDP) satisfies the Slater constraint qualification and ν^* is finite, $\mu^* = \nu^*$ is attained for (PSDP).
(iii) If (PSDP) and (DSDP) are both strictly feasible, then $\mu^* = \nu^*$ is attained for both problems.

The following theorem is also well known.

Theorem 2.2 *[94]*
Let (PSDP) and (DSDP) satisfy the Slater constraint qualification. If one of the problems is infeasible, then the other is infeasible or unbounded.

Suppose now that (PSDP) and (DSDP) are both strictly feasible. Since the duality gap is zero, it follows that

$$0 = \mathrm{tr}(RZ) = \mathrm{tr}(R^{\frac{1}{2}}ZR^{\frac{1}{2}}) = \langle Z^{\frac{1}{2}}R^{\frac{1}{2}}, Z^{\frac{1}{2}}R^{\frac{1}{2}}\rangle = ||Z^{\frac{1}{2}}R^{\frac{1}{2}}||_F^2.$$

$Z^{\frac{1}{2}}R^{\frac{1}{2}} = 0$ implies that $ZR = 0$.

Definition 2.4 *(Complementarity slackness)*
We say that $(R, Z) \in \mathcal{S}_n \times \mathcal{S}_n$ are complementary, or satisfy complementarity slackness if $ZR = 0$.

The complementarity condition implies that R and Z commute, and hence share an orthonormal system of eigenvectors, say Q. Clearly this results in $\mathrm{rank}(R) + \mathrm{rank}(Z) \leq n$.

Definition 2.5 *(Maximal and Strict Complementarity)*
A primal solution R and a dual solution (w, Z) are said to satisfy maximal complementarity *if R and Z have maximal rank among all solutions. A primal solution R and a dual solution (w, Z) are said to satisfy* strict complementarity *if* $\mathrm{rank}(R) + \mathrm{rank}(Z) = n$.

Let us denote the eigenvalues of R and of Z by

$$\lambda = (\lambda_1, \ldots, \lambda_n) \geq 0 \quad \text{and} \quad \omega = (\omega_1, \ldots, \omega_n) \geq 0$$

respectively. We assume without loss of generality that the components of λ (resp. ω) are arranged in nonincreasing (resp. nondecreasing) order, i.e. $\lambda_1 \geq \ldots \geq \lambda_n$ (resp. $\omega_1 \leq \ldots \leq \omega_n$). Writing the primal solution as (see Theorem A.1)

$$R = Q\mathrm{Diag}(\lambda)Q^T \tag{2.2}$$

and the dual slack solution as

$$Z = Q\mathrm{Diag}(\omega)Q^T, \tag{2.3}$$

we can restate the complementarity condition $ZR = 0$ as $\lambda\omega = 0$, and strict complementarity as $\lambda + \omega > 0$. In the case that strong duality holds, we get the following primal–dual characterization of *optimality conditions* for (PSDP) and (DSDP):

$$\begin{array}{lll} & \mathcal{A}(R) = a & (primal\ feasibility) \\ (\text{OPT}) & Z + \mathcal{A}^T(w) = L & (dual\ feasibility) \\ & ZR = 0 & (complementarity\ slackness), \end{array} \tag{2.4}$$

where $X \in \mathcal{S}_n^+, Z \in \mathcal{S}_n^+$, and $y \in I\!R^m$. The optimality conditions (OPT) are *necessary and sufficient* optimality conditions. These optimality conditions provide the basis for: (i) the primal simplex method (maintain primal feasibility and complementary slackness while striving for dual feasibility); (ii) the dual simplex method (maintain dual feasibility and complementary slackness while striving for primal feasibility), and (iii) the interior point methods (maintain primal and dual feasibility while striving for complementary slackness). Since there are currently no efficient algorithms for primal or dual simplex methods for SDP, we explain interior point methods that have proven to be very successful. Before we describe the interior point algorithm, we derive conditions that imply uniqueness of the primal and dual solution.

2.3 Nondegeneracy and Strict Complementarity

Here we consider the semidefinite program (PSDP) and assume that the matrices $A_k,\ k = 1, \ldots, m$ are linearly independent. Through this section we also assume that there exists a primal feasible point $R \succ 0$, and a dual feasible point (w, Z) with $Z \succ 0$, such that $\text{rank}(R) + \text{rank}(Z) = n$.

We define the set

$$M_r := \{R \in \mathcal{S}_n : \text{rank}(R) = r\}.$$

Since the eigenvalues of a matrix R are continuous functions of R, it is clear that for $r > 0$, the boundary of M_r is

$$\partial M_r = M_0 \cup \ldots \cup M_{r-1}.$$

Let

$$M_r^+ = \mathcal{S}_n^+ \cap M_r = \{R \in \mathcal{S}_n : R \succeq 0 \text{ and } \text{rank}(R) = r\}.$$

Then the boundary of $\mathcal{S}_n^+$ is given by

$$\partial\mathcal{S}_n^+ = M_0^+ \cup \ldots \cup M_{n-1}^+,$$

and the interior of $\mathcal{S}_n^+$ is

$$\text{int}(\mathcal{S}_n^+) = M_n^+.$$

Let R be primal feasible with $\text{rank}(R) = r$ and

$$R = Q\text{Diag}(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)Q^T, \tag{2.5}$$

where $Q^TQ = I$. The tangent space to M_r at R is, see [8]

$$\mathcal{T}_R = \left\{ Q \begin{bmatrix} U & V \\ V^T & 0 \end{bmatrix} Q^T : U \in \mathcal{S}_r, V \in \mathcal{M}_{r,(n-r)} \right\}.$$

Note that

$$\dim \mathcal{T}_R = \begin{pmatrix} r+1 \\ 2 \end{pmatrix} + r(n-r) = \begin{pmatrix} n+1 \\ 2 \end{pmatrix} - \begin{pmatrix} n-r+1 \\ 2 \end{pmatrix}.$$

Remark 2.1 *For $\Delta R \in \mathcal{T}_R$ we have*

$$Q^T(R + \epsilon\Delta R)Q = \begin{bmatrix} \text{Diag}(\lambda_1, \ldots, \lambda_r) + \epsilon U & \epsilon V \\ \epsilon V^T & 0 \end{bmatrix}.$$

Thus $R \pm \epsilon\Delta R$ is not contained in $\mathcal{S}_n^+$, for $\epsilon > 0$, unless $V = 0$ (see Remark 1.1).

Definition 2.6 *R is* primal nondegenerate *if it is primal feasible and*

$$\mathcal{T}_R + \mathcal{N} = \mathcal{S}_n, \tag{2.6}$$

where

$$\mathcal{N} = \{Y \in \mathcal{S}_n : \langle A_k, Y \rangle = 0, \ \forall k\}.$$

Theorem 2.3 *[3] Let R be primal feasible with* $\text{rank}(R) = r$. *A necessary condition for R to be primal nondegenerate is that*

$$\begin{pmatrix} n-r+1 \\ 2 \end{pmatrix} \leq \begin{pmatrix} n+1 \\ 2 \end{pmatrix} - m. \tag{2.7}$$

Furthermore, let $Q_1 \in \mathcal{M}_{n,r}$ and $Q_2 \in \mathcal{M}_{n,(n-r)}$ respectively denote the first r columns and the last $n-r$ columns od Q given by (2.5). Then R is primal nondegenerate if and only if the matrices

$$B_k = \begin{bmatrix} Q_1^T A_k Q_1 & Q_1^T A_k Q_2 \\ Q_2^T A_k Q_1 & 0 \end{bmatrix}, \quad k = 1, \ldots, m \tag{2.8}$$

are linearly independent in $\mathcal{S}_n$.

PROOF: Since $\dim \mathcal{T}_R = \begin{pmatrix} n+1 \\ 2 \end{pmatrix} - \begin{pmatrix} n-r+1 \\ 2 \end{pmatrix}$ and $\dim \mathcal{N} = \begin{pmatrix} n+1 \\ 2 \end{pmatrix} - m$, inequality (2.7) follows directly. Equation (2.6) is equivalent to

$$\dim \mathcal{T}_R^{\perp} \cap \dim \mathcal{N}^{\perp} = \{0\}, \tag{2.9}$$

where $\mathcal{T}_R^{\perp}$ and $\mathcal{N}^{\perp}$ are respectively the orthogonal complements of $\mathcal{T}_R$ and $\mathcal{N}$. Namely,

$$\mathcal{T}_R^{\perp} = \left\{ Q \begin{bmatrix} 0 & 0 \\ 0 & H \end{bmatrix} Q^T : H \in \mathcal{S}_{n-r} \right\}$$

and

$$\mathcal{N}^{\perp} = \text{Span}\{A_k\}.$$

If the B_k are linearly dependent, there exists $\theta \in I\!R^m$, $\theta \neq 0$ such that

$$\sum_{k=1}^{m} \theta_k B_k = 0.$$

This implies

$$\sum_{k=1}^{m} \theta_k A_k \in \mathcal{T}_R^{\perp},$$

which contradicts with (2.9). Conversely, if the B_k are linearly independent then (2.9) holds. ∎

Theorem 2.3 holds for any Q satisfying (2.5).

Theorem 2.4 *[3] Let R be primal nondegenerate and optimal. Then there exists a unique optimal dual solution (w, Z).*

PROOF: From the assumptions of the theorem it follows that a dual optimal solution (w, Z) exists, so that complementarity holds. Let Q_1 and Q_2 respectively denote the first r columns and the last $n-r$ columns of Q given in (2.5). Any $\tilde{Z}$ satisfying the complementarity condition $\tilde{Z}R = 0$ must be of the form

$$\tilde{Z} = Q_2 H Q_2^T,$$

for some $H \in \mathcal{S}_{n-r}$. The dual feasibility condition requires the existence of $\tilde{w} \in I\!R^m$ and $H \in \mathcal{S}_{n-r}$ such that

$$Q_2 H Q_2^T + \mathcal{A}^T(\tilde{w}) = L.$$

Theorem 2.3 guarantees that any solution of this linear system is unique. ■

If we assume Q satisfies (2.3) and (2.5), we find that

$$H = \mathrm{Diag}(w_{r+1}, \ldots, w_n).$$

Let us consider now the *dual nondegeneracy*. Let (w, Z) be dual feasible with rank(Z) = s and

$$Z = Q\mathrm{Diag}(0, \ldots, 0, w_{n-s+1}, \ldots, w_n)Q^T \tag{2.10}$$

with $Q^T Q = I$. The tangent space to $\mathcal{M}_s$ at Z is

$$\mathcal{T}_Z = \left\{ Q \begin{bmatrix} 0 & V \\ V^T & H \end{bmatrix} Q^T : V \in \mathcal{M}_{n-s,s},\ H \in \mathcal{S}_s \right\}.$$

Note that

$$\mathrm{dim}\mathcal{T}_Z = \begin{pmatrix} s+1 \\ 2 \end{pmatrix} + s(n-s) = \begin{pmatrix} n+1 \\ 2 \end{pmatrix} - \begin{pmatrix} (n-s)+1 \\ 2 \end{pmatrix}.$$

Definition 2.7 *The point* (w, Z) *is* dual nondegenerate *if it is dual feasible and* Z *satisfies*

$$\mathcal{T}_Z + \mathrm{Span}\{A_k\} = \mathcal{S}_n. \tag{2.11}$$

Theorem 2.5 *[3] Let* (w, Z) *be dual feasible with* $\mathrm{rank}(Z) = s$. *A necessary condition for* (w, Z) *to be dual nondegenerate is that*

$$\begin{pmatrix} (n-s)+1 \\ 2 \end{pmatrix} \leq m.$$

Furthermore, let $\tilde{Q}_1 \in \mathcal{M}_{n,(n-s)}$ *and* $\tilde{Q}_2 \in \mathcal{M}_{n,s}$ *respectively denote the first* $n-s$ *and the last* s *columns of* Q *given by (2.10). Then* (w, Z) *is dual nondegenerate if and only if the matrices*

$$\tilde{B}_k = [\tilde{Q}_1^T A_k \tilde{Q}_1], \quad k = 1, \ldots, m$$

span $\mathcal{S}_{n-s}$.

PROOF: It is an immediate consequence of the definition. ■

Theorem 2.6 *[3] Let* (w, Z) *be dual nondegenerate and optimal. Then there exists a unique optimal primal solution* R.

PROOF: From the assumptions of the theorem it follows that a primal optimal solution R exists, so that complementarity holds. Let $\tilde{Q}_1$ and $\tilde{Q}_2$ respectively denote the first $n-s$ columns and the last s columns of Q given by (2.10). Any $\tilde{R}$ satisfying complementarity condition $Z\tilde{R} = 0$ must be of the form

$$\tilde{R} = \tilde{Q}_1 U \tilde{Q}_1^T$$

for some $U \in \mathcal{S}_{n-s}$. The primal feasibility condition in (2.4) reduces to

$$\langle \tilde{Q}_1^T A_k \tilde{Q}_1, U \rangle = a_k, \quad k = 1, \ldots, m.$$

Theorem 2.5 guarantees that any solution of this linear system is unique. ■

If we assume Q satisfies (2.5) and (2.10), we find that

$$U = \text{Diag}(\lambda_1, \ldots, \lambda_{n-s}).$$

Note also the distinction between the partitioning of Q used in Theorems 2.3 and 2.5. The former uses $Q = [Q_1, Q_2]$ where Q_1 has r columns and the latter uses $Q = [\tilde{Q}_1, \tilde{Q}_2]$ where $\tilde{Q}_1$ has $n-s$ columns. These partitions are the same if and only if $r + s = n$, i.e. strict complementarity holds.

Theorem 2.7 *[3] Suppose that* R *and* (w, Z) *are respectively primal and dual optimal solutions satisfying strict complementarity. Then if the primal solution* R *is unique, the dual nondegeneracy condition must hold, and if the dual solution* (w, Z) *is unique, the primal nondegeneracy condition must hold.*

PROOF: Let Q satisfy conditions (2.5) and (2.10). Strict complementarity states that $r + s = n$, so the partitioning of Q used in Theorems 2.3 and 2.5 are the same. Thus

$$R = Q_1 \mathrm{Diag}(\lambda_1, \ldots, \lambda_r) Q_1^T, \quad Z = Q_2 \mathrm{Diag}(\omega_1, \ldots, \omega_r) Q_2^T.$$

Suppose first that the dual nondegeneracy assumption (2.11) fails to hold. We show that in this case R can not be a unique primal solution. Since Z is an optimal dual solution, complementarity states that any optimal primal solution $\tilde{R}$ must satisfy

$$\tilde{R} = Q_1 U Q_1^T,$$

for some $U \in \mathcal{S}_r$, and so the primal feasibility condition reduces to

$$\langle Q_1^T A_k Q_1, U \rangle = a_k, \quad k = 1, \ldots, m.$$

Because the dual nondegeneracy assumption does not hold, the solution set of this equation is not unique, but holds on an affine subset of $\mathcal{S}_r$, say $\mathcal{U}$. The condition that $\tilde{R} \succeq 0$ holds if and only if $U \succeq 0$. But the particular choice $U = \mathrm{Diag}(\lambda_1, \ldots, \lambda_r)$ lies in $\mathcal{U}$ and is positive definite, so there is an open set in $\mathcal{U}$ for which the same is true. Every such U defines an $\tilde{R}$ which satisfies the optimality conditions.

Now suppose that the primal nondegeneracy assumption (2.6) fails to hold. We show that in this case (w, Z) cannot be a unique dual solution. Complementarity states that any solution $\tilde{Z}$ must satisfy

$$\tilde{Z} = Q_2 H Q_2^T$$

for some $H \in \mathcal{S}_s$, and so the dual feasibility condition reduces to the solvability of

$$Q_2 H Q_2^T + \mathcal{A}^T(\tilde{w}) = C$$

for some $\tilde{w} \in I\!R^m$ and $H \in \mathcal{S}_s$. Because the primal nondegeneracy assumption does not hold, the solution set of this equation is not unique, but holds on an affine subset of $I\!R^m \times \mathcal{S}_s$, say $\mathcal{W}$. The condition $\tilde{Z} \succeq 0$ if and only if $H \succeq 0$. But the particular choice $(\tilde{w} = w,\ H = \mathrm{Diag}(\omega_{r+1}, \ldots, \omega_n))$ lies in $\mathcal{W}$ with H positive definite, so there is an open set in $\mathcal{W}$ for which the same is true. Every such H defines a $\tilde{Z}$ which satisfies the optimality conditions.

■

2.4 Primal-Dual Interior Point Methods

Semidefinite programs can be solved (more precisely, approximated) in polynomial time within any specified accuracy, either by the ellipsoid algorithm [33] or through interior-point algorithms. More recently, interior point methods [2, 16, 39, 68, 69, 88, 88, 96] have turned out to be the method of choice to solve SDP, since they give faster algorithms than the ellipsoid method. The interior-point algorithms converge very fast and an approximately optimal solution is obtained within a polynomial number of iterations. The computation of a single step is computationally rather expensive for the problems that contain a big number of constraints. Within current technology we are able to solve with these methods problems that contain about 8000 constraints.

The interior point methods for SDP are iterative algorithms which use a Newton-like method to generate search directions to find an approximate solution to the nonlinear system. Below we describe the interior-point approach for SDP, or more precisely the *primal-dual interior point path-following method*. First, we state several assumptions. From Theorem 2.1 follows that a sufficient condition for the attainment of optimal primal and dual solutions is the existence of strictly feasible primal and dual solutions. The concept of the interior-point approach is based on the following assumptions.

Basic Assumption 1 *Both, the primal (PSDP) and the dual (DSDP) problem satisfy the Slater constraint qualification, e.g.,*

$$\exists(R, w, Z) \quad \text{s.t.} \quad R \succ 0,\ Z \succ 0,\ \mathcal{A}(R) = a,\ L - \mathcal{A}^T(w) = Z.$$

Here we also assume that a strictly feasible starting point is known. To avoid trivialities, it is usually to assume the following.

Assumption 2.1 *The linear equations* $\langle A_i, R\rangle = a_i,\ i = 1, \ldots, m$ *are linearly independent.*

The *start* of the interior-point algorithm is in the interior of the feasible region, thus in the cone of the positive definite matrices. In order to stay during the iteration process in that cone, we change the objective function of the primal problem by adding a *barrier term*

$$-\mu \log \det(R).$$

The real number $\mu > 0$ is the so-called *barrier parameter* and $-\log\det(R)$ is the *barrier function*. The value of the barrier function is small in the interior of the feasible region but grows to infinity when the boundary is approached. *Optima* are usually located on the boundary. In order to obtain the convergence of the algorithm towards optima, the effect of the barrier function should be decreased after each iteration of the algorithm. This is obtained by weighting the barrier function with $\mu > 0$, whose value decreases as the algorithm proceeds. We now introduce the associated *barrier problem* for (PSDP), which we call the *primal barrier problem*:

$$\begin{array}{ll} \min & \langle L, R\rangle - \mu \log\det(R) \\ & \mathcal{A}(R) = a, \\ & R \succ 0. \end{array} \tag{2.12}$$

Since the cost function of the barrier problem is strictly convex (see Lemma B.1), the optimal solution exists and is unique.

Remark 2.2 *The* barrier function $-\log\det(R)$ *belongs to the class of so–called* strongly self concordant functions *[68]. If a linear functional is added to a self concordant function the resulting function is a self concordant function. Hence, the objective function in the primal barrier problem is a self concordant function. Nesterov and Nemirovskii [68] showed that for the class of strongly self concordant functions Newton's method works especially well. More precisely, for convex sets having a strongly self concordant barrier function which can be computed efficiently, Newton's method yields a polynomial time interior-point algorithm.*

For each $\mu > 0$, there is a corresponding Lagrangian:

$$\mathcal{L}_\mu(R, w) = \langle L, R\rangle + \langle a - \mathcal{A}(R), w\rangle - \mu \log\det(R),$$

where $w \in I\!R^m$ is a Lagrange multiplier. The first-order optimality conditions for the saddle–point of the Lagrangian $\mathcal{L}_\mu$ are called Karush–Kuhn-Tucker (KKT) conditions (Theorem C.13). For any fixed value $\mu > 0$, the KKT conditions are

$$\frac{\partial}{\partial w}\mathcal{L}_\mu(R, w, Z) = a - \mathcal{A}(R) = 0 \tag{2.13}$$

$$\frac{\partial}{\partial R}\mathcal{L}_\mu(R, w, Z) = L - \mathcal{A}^T(w) - \mu R^{-1} = 0, \tag{2.14}$$

for $R \succ 0$, $Z \succ 0$. Here we use the fact that $\frac{d}{dR} \log\det(R) = R^{-1}$, see Theorem B.1 and Remark B.1. In a primal-dual formulation we set $Z = \mu R^{-1}$. There are several equivalent formulations of this condition. We use $ZR = \mu I$. We now rewrite the KKT conditions in the following form.

$$\begin{array}{rrcll} & \mathcal{A}(R) & = & a, & R \succ 0 \\ \text{(KKT)} & Z + \mathcal{A}^T(w) & = & L, & Z \succ 0 \\ & ZR & = & \mu I. & \end{array} \tag{2.15}$$

The first equality in (2.15) is called the primal feasibility, and the second the dual feasibility. The third equation is called the *perturbed complementarity condition*. For $\mu = 0$ we have the complementarity condition.

Remark 2.3 *The* dual barrier problem *is*

$$\begin{array}{ll} \max & \langle a, w \rangle + \mu \log\det(Z) \\ & L - \mathcal{A}^T(w) = Z, \ Z \succ 0. \end{array}$$

The cost function of this barrier problem is strictly concave, hence the optimal solution exists and is unique. For each $\mu > 0$, there is a corresponding Lagrangian:

$$\bar{\mathcal{L}}_\mu(R, w, Z) = \langle a, w \rangle + \langle R, Z + \mathcal{A}^T(w) - L \rangle + \mu \log\det(Z).$$

From the Lagrangian $\bar{\mathcal{L}}$ we obtain the KKT conditions (2.15).

Under Basic Assumption 1 and Assumption 2.1, for every $\mu > 0$ there exists a unique solution (R_μ, w_μ, Z_μ) of KKT, see [68, 91]. The set

$$\{(R_\mu, w_\mu, Z_\mu) : \mu > 0\}$$

defines a smooth curve parametrized by μ, which is usually called the *primal–dual central path* or *central trajectory*. For each point (R, w, Z) on the central path, it is easy to determine its associated μ value using the last equation of the optimality conditions:

$$\mu = \frac{\mathrm{tr}(ZR)}{n}.$$

The interior-point approach consists in applying an iterative process to follow this curve while μ is decreasing. The central path gets as close to strict

complementarity as possible, actually the point to which the central path converges for $\mu \to 0$ is maximal complementarity (see [51]). This point is called the *analytic center*.

Interior-point algorithms work as follows.
We start with a point $u = (R, w, Z)$ which satisfies $R \succ 0$, $Z \succ 0$. If that point would lie on the central path, the associated parameter would be $\mu = \frac{\mathrm{tr}(ZR)}{n}$. We do not assume that u lies on the central path, but would like to move this triple towards the central path. We head for a point on the central path given by the parameter $\frac{\mu}{2}$. Such an approach performs very well in practice. Next, we attempt to find steps $(\Delta R, \Delta w, \Delta Z)$ such that the new point $(R + \alpha\Delta R, w + \alpha\Delta w, Z + \alpha\Delta Z)$, $\alpha > 0$ becomes close to the point $(R_{\frac{\mu}{2}}, w_{\frac{\mu}{2}}, Z_{\frac{\mu}{2}})$ on the central trajectory. We can find such a step with a *variant* of Newton's method. The equation (2.15) has $(n+1)n+m$ variables, but $\frac{(n+1)n}{2} + n^2 + m$ equations, and therefore Newton's method cannot be directly applied to (2.15). The difference arises from $ZR - \mu I$, which need not be symmetric, even if R and Z are. Therefore some sort of symmetrization of the last equation in (2.15) is necessary to overcome this problem. Many authors have suggested different ways of symmetrizing the third equation in (2.15). Todd [88] analyzes twenty different search directions for SDP. In Section 2.6 we describe three search directions that are used most frequently in practice.

Here we present the variant of the search direction that was independently introduced by Helmberg, Rendl, Vanderbei and Wolkowicz [39]; Kojima, Shindoh and Hara [55], and Monteiro [64]. This search direction is known as the H..K..M direction. It is simple, and yet computationally quite efficient. Here, the *step direction* can be determined by the linearized system

$$\begin{array}{rrcll} & \mathcal{A}(\Delta R) & = & a - \mathcal{A}(R) & =: \quad F_p \\ (\Delta\mathrm{KKT}) & \Delta Z + \mathcal{A}^T(\Delta w) & = & L - \mathcal{A}^T(w) - Z & =: \quad F_d \\ & \Delta ZR + Z\Delta R & = & \mu I - ZR & =: \quad F_{ZR}. \end{array}$$

We first solve for ΔZ and eliminate the second equation of $(\Delta\mathrm{KKT})$

$$\Delta Z = -\mathcal{A}^T(\Delta w) + F_d.$$

Now we solve for ΔR and eliminate the third equation of $(\Delta\mathrm{KKT})$

$$\Delta R = Z^{-1}\mathcal{A}^T(\Delta w)R - Z^{-1}F_dR + Z^{-1}F_{ZR}.$$

By substituting the previous equation into the first equation of the system (ΔKKT), the *final equation* of the system is

$$\mathcal{A}(Z^{-1}\mathcal{A}^T(\Delta w)R) = \mathcal{A}(Z^{-1}F_dR - \mu Z^{-1}) + a. \tag{2.16}$$

In practice it is very important to efficiently find the matrix representing the final system by exploiting the possible structure (see Section 4.3).

Since the matrix $Z^{-1}\mathcal{A}^T(\Delta w)R$ is not symmetric, we need to extend the definition of the operator $\mathcal{A}$.

Remark 2.4 *In order to apply operator $\mathcal{A}$ to unsymmetric matrices, we extend its definition. For any nonsymmetric square matrix X, let*

$$\mathcal{A}(X) = \frac{1}{2}\mathcal{A}(X + X^T).$$

The system (2.16) is positive definite (see [39]) and can therefore be solved quite efficiently by standard methods yielding Δw. In our computations we use the Cholesky factorization (see Section 4.1, 7.2, and 7.3). Back substitution gives ΔR and ΔZ. In general, ΔR can not be assumed to be symmetric, but this search direction always yields a symmetric ΔZ. We symmetrize ΔR by

$$\Delta R \leftarrow \frac{\Delta R + \Delta R^T}{2}.$$

Kojima et al. [55] show that even under this symmetrization the interior point method has polynomial convergence. The new point is

$$\begin{aligned} R^n &= R + \alpha\Delta R \\ w^n &= w + \alpha\Delta w \\ Z^n &= Z + \alpha\Delta Z, \end{aligned}$$

where $\alpha \in \langle 0, 1]$ is the stepsize. Kojima et al. [55] set the condition on α which guarantees the positive semidefiniteness of the updated variables.

Lemma 2.2 *[55] Suppose that $R \in \mathcal{S}_n^{++}$, $\Delta R \in \mathcal{S}_n$ and $\alpha \geq 0$. Let $\lambda_{\min}$ be the minimum eigenvalue of the matrix $R^{-1}\Delta R$ and let*

$$\bar{\alpha} = \sup\{\alpha \text{ s.t. } 1 + \alpha\lambda_{\min} \geq 0\} = \begin{cases} -\frac{1}{\lambda_{\min}} & \text{if } \lambda_{\min} < 0 \\ +\infty & \text{otherwise.} \end{cases}$$

Then $R + \alpha\Delta R \in \mathcal{S}_n^{++}$ if and only if $\alpha < \bar{\alpha}$.

In practice one starts with $\alpha = 1$, which is a full Newton step, and backtracks by multiplying the current α with the factor smaller than 1 (for instance 0.85) until positive definiteness of R^n and Z^n is reached. Once we reach the positive definiteness of R^n and Z^n, we multiply α with 0.95 to make sure the next point is not to close to the boundary.

2.5 Predictor–Corrector

The predictor–corrector approach turns to be very successful for semidefinite programming, see [16, 95]. The step direction is a linear combination of two search directions, the predictor and the corrector one.

The *predictor step* solves the system (ΔKKT) with $\mu = 0$. Hence, the final equation differs from (2.16) only in the right hand side,

$$\mathcal{A}(Z^{-1}\mathcal{A}^T(\delta w^p)R) = \mathcal{A}(Z^{-1}F_dR) + a.$$

The result is the *affine step* direction $\delta s^p = (\delta R^p, \delta w^p, \delta Z^p)$ which is responsible for the progress towards the desired optimum.

The *corrector step* pulls the current iterate closer to the central path, and is often called the *centering step*. The corrector step solves the system (ΔKKT) at the point $(R+\delta R^p, w+\delta w^p, Z+\delta Z^p)$. If higher order terms are neglected for the corrector step, the finally equation of the system is

$$\mathcal{A}(Z^{-1}\mathcal{A}^T(\delta w^c)R) = -\mathcal{A}(Z^{-1}(\mu I - \delta Z^p \delta R^p)).$$

Note that the system again changes only on the right hand side. Hence, we can use the old factorization of the system matrix to solve for δw^c. The result is the centering step direction $\delta s^c = (\delta R^c, \delta w^c, \delta Z^c)$. Finally, the search direction for the line search is $\delta s^p + \delta s^c$.

The predictor–corrector algorithm shows good practical behavior with respect to stability and is proven to converge *superlinearly* [73]. In all our computations we use the predictor–corrector algorithms.

2.6 Search Directions

Here we describe Monteiro-Zhang family [64, 96, 66, 67] of search directions and specify three search directions that are belonging to that family.

For nonsingular matrix $P \in \mathcal{M}_n$ we introduce the mapping $P^{-T} \odot P : \mathcal{S}_n \to \mathcal{S}_n$, see [2]

$$(P^{-T} \odot P)R := \frac{1}{2}(P^{-T}RP^T + PRP^{-1}).$$

Remark 2.5 *This definition can be extended on $\mathcal{M}_n$. For $P, Q, R \in \mathcal{M}_n$ is defined*

$$(P \odot Q)R := \frac{1}{2}(PRQ^T + QR^TP^T).$$

If $P, Q \in \mathcal{M}_n$ and $(P \odot Q)$ is considered as an operator from $\mathcal{S}_n$ to itself then

$$P \odot Q = Q \odot P, \quad (P \odot Q)^T = P^T \odot Q^T.$$

If P is nonsingular matrix then

$$(P \odot P)^{-1} = P^{-1} \odot P^{-1}.$$

We now define

$$H_P := P^{-T} \odot P.$$

The idea is to replace in KKT the complementarity condition $ZR = \mu I$ by

$$H_P(ZR) = H_P(\mu I) = \mu I.$$

Now we rewrite KKT in the following way

$$\begin{array}{rcll} \mathcal{A}(R) & = & a, & R \succ 0 \\ Z + \mathcal{A}^T(w) & = & L, & Z \succ 0 \\ H_P(ZR) & = & \mu I. & \end{array} \tag{2.17}$$

Now the last equation in (2.17) is linearized

$$\frac{1}{2}(P^{-T}\Delta ZRP^T + PR\Delta ZP^{-1}) + \frac{1}{2}(P^{-T}Z\Delta RP^T + P\Delta RZP^{-1}) = F_P, \tag{2.18}$$

where

$$F_P = \mu I - \frac{1}{2}(P^{-T}ZRP^T + PRZP^{-1}).$$

If we define

$$\mathcal{F} := PR \odot P^{-T}, \quad \mathcal{E} := P \odot P^{-T}Z,$$

then the linearization of the system (2.17) is given by

$$\begin{aligned} \mathcal{A}(\Delta R) &= F_p \\ \Delta Z + \mathcal{A}^T(\Delta w) &= F_d \\ \mathcal{F}\Delta Z + \mathcal{E}\Delta R &= F_P. \end{aligned} \tag{2.19}$$

We can alternatively multiply the equation (2.18) by P^T from left and by P from right, and get

$$\begin{aligned} \frac{1}{2}(\Delta Z R P^T P + P^T P R \Delta Z) + \frac{1}{2}(Z \Delta R P^T P + P^T P \Delta R Z) \\ = \mu P^T P - \frac{1}{2}(Z R P^T P + P^T P R Z). \end{aligned}$$

Now is

$$\mathcal{F} := MR \odot I, \quad \mathcal{E} := Z \odot M,$$

and

$$F_P = \mu M - \frac{1}{2}(ZRM + MRZ),$$

with

$$M := P^T P.$$

The choice of P and M often depends on the current iterates Z and R, and hence we sometimes write $P(Z,R)$ or $M(Z,R)$ to highlight this dependence.

i) AHO direction.	(Alizadeh, Haeberly, and Overton [2]) The solution is primal-dual symmetric. $P = M = I$, and $\mathcal{F} = R \odot I$, $\mathcal{E} = Z \odot I$, $F_P = \mu I - \frac{1}{2}(ZR + RZ)$.
ii) H..K..M direction.	(Helmberg, Rendl, Vanderbei and Wolkowicz [39]; Kojima, Shindoh and Hara [55], and Monteiro [64]) The solution of (2.19) is $(\Delta R, w, \Delta Z) \in \mathcal{M}_n \times I\!R^m \times \mathcal{S}_n$. $P = Z^{1/2}$, $M = Z$, and $\mathcal{F} = ZR \odot I$, $\mathcal{E} = Z \odot Z$, $F_P = \mu Z - ZRZ$. Alternatively, so that $\mathcal{E}$ does not need to be inverted: $\mathcal{F} = R \odot Z^{-1}$, $\mathcal{E} = I \odot I$, $F_P = \mu Z^{-1} - R$.

iii) NT direction. (Nesterov and Todd [69, 70, 89])
The solution is primal-dual symmetric.
$P = W^{-1/2}$, $M = W^{-1}$ for the unique scaling matrix:
$W = Z^{-1/2}(Z^{1/2}RZ^{1/2})^{1/2}Z^{-1/2}$, and
$\mathcal{F} = W^{-1}R \odot I$, $\mathcal{E} = Z \odot W^{-1}$,
$F_P = \mu W^{-1} - \frac{1}{2}(W^{-1}RZ + ZRW^{-1})$.

These search directions are currently the most exploited in practical implementation. Beside these three directions Todd in [88] describes seventeen more primal-dual search directions.

Assuming that $\mathcal{E}$ is nonsingular, we find that (2.19) has a unique solution iff the $m \times m$ *Schur complement matrix* $\mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T$ is nonsingular. In this case the solution is obtained from

$$\begin{aligned} (\mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T)\Delta w &= F_p - \mathcal{A}\mathcal{E}^{-1}(F_{RZ} - \mathcal{F}F_d) \\ \Delta Z &= F_d - \mathcal{A}^T(\Delta w) \\ \Delta R &= \mathcal{E}^{-1}(F_{RZ} - \mathcal{F}\Delta Z). \end{aligned} \tag{2.20}$$

The main computational work is the formation and factorization of the Schur complement matrix. The H..K..M and NT directions give a unique search direction for every symmetric positive definite R and Z and surjective operator $\mathcal{A}$. These two directions posses the property that $\mathcal{E}^{-1}\mathcal{F}$ is positive definite and self-adjoint. Hence, the Schur complement matrix is symmetric and the first equation in (2.20) is solved by using a Cholesky factorization of the Schur complement matrix. The AHO direction gives a unique search direction for every symmetric positive definite R and Z if $ZR + RZ$ is symmetric positive definite (see [89]), or if (R, w, Z) lies in a suitable neighborhood of the central path (see [65]). The Schur complement matrix for the AHO direction is not symmetric, but can be shown to be nonsingular if $ZR + RZ \succ 0$, see [84]. For the AHO direction the first equation in (2.20) is solved by using an LU factorization of the Schur complement.

All three methods are described by the following iteration.

INTERIOR POINT ALGORITHM

Basic Iteration.

(i) Choose $0 < \sigma < 1$ and define: $\mu = \sigma \frac{\mathrm{tr}(ZR)}{n}$.
(ii) Determine $(\Delta R, \Delta w, \Delta Z)$ by solving (2.20).
(iii) In the case of the H..K..M direction,
update ΔR by $\Delta R \leftarrow \frac{1}{2}(\Delta R + \Delta R^T)$.
(iv) Choose steplengths $\alpha_P, \alpha_D \in \langle 0, 1]$ so that
$R + \alpha_P \Delta R \succ 0$ and $Z + \alpha_D \Delta Z \succ 0$.
(v) Compute $\alpha = \min\{\alpha_P, \alpha_D\}$ and update
$R \leftarrow R + \alpha \Delta R$, $w \leftarrow w + \alpha \Delta w$, $Z \leftarrow Z + \alpha \Delta Z$.

Stopping Criteria.

If $||\mathcal{A}(R) - a||$, $||\mathcal{A}^T(w) + Z - C||$ and $\langle Z, R \rangle$ are small enough.

Chapter 3

The Quadratic Assignment Problem

3.1 Problem Formulation

The Quadratic Assignment Problem (QAP) was introduced in 1957 by Koopmans and Beckmann [56] as a model for location problems, that takes into account the cost of placing a new facility on a certain site as well as the interaction with other facilities. Nowadays, the QAP is widely considered as a classical combinatorial optimization problem. For the applications of the QAP see Section 3.3.

The Quadratic Assignment Problem can be stated in the following way. For given $A = (a_{ij}), B = (b_{ij})$, and $C = (c_{ij})$ real $n \times n$ matrices find a permutation π of the set $\{1, \ldots, n\}$ which minimizes

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i),\pi(j)} + \sum_{i=1}^{n} c_{i,\pi(i)}. \tag{3.1}$$

This is a *combinatorial formulation* of the QAP. A permutation π_0 which minimizes (3.1) is called an *optimal solution*. The first part in the objective function is called the *quadratic part* while the other is called the *linear term*. The size n of the matrix A (resp. B, C) is the size of the QAP. If the coefficient matrices A and B are symmetric, then the QAP is said to be *symmetric*, otherwise is said to be *unsymmetric*.

As a illustrative example we describe a campus planning model due to Dickey and Hopkins [20]. The university owns a piece of land on which new buildings are to be erected. On the university's land n sites have been identified as possible sites for the buildings. Each of the buildings has a special function, such as library or dormitory. Let a_{ij} be the walking distance between the two sites i and j, where the new buildings can be erected. These distances are collected in the matrix $A = (a_{ij})$, which is often called the *distance matrix*. Let b_{kl} denotes the number of people per week who circulate between buildings k and l. The quantities b_{kl} are collected in the matrix $B = (b_{kl})$, which is often called the *flow matrix*. Note that the diagonal elements of A and B are all zero and both, A and B are symmetric matrices. The product

$$a_{ij}b_{\pi(i),\pi(j)}$$

describes the weekly walking distance of people who travel between buildings $k = \pi(i)$ and $l = \pi(j)$, if building k is erected on site i and building l on site j. The problem of assigning buildings to sites so that the walking distance is minimized corresponds to the following minimization problem

$$\min_{\pi} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{\pi(i),\pi(j)}. \tag{3.2}$$

Suppose now that in addition to the interest of minimizing the walking distance at the campus, the university is also interested in minimizing the total construction cost. Let c_{ij} denotes the cost of erecting the building i on site j. Then the cost construction minimization problem is

$$\min_{\pi} \sum_{i=1}^{n} c_{i,\pi(i)}. \tag{3.3}$$

Note that minimizing the total distance walked by all users of the campus will in general be in conflict with the goal of minimizing construction cost. The minimization problem whose solution fulfills both previously described demands has for the objective function the linear combination of the objective function from (3.2) and (3.3), which is exactly the objective given in (3.1).

3.2 Equivalent Formulations of QAP

The term "quadratic" comes from the formulation of the problem as an optimization problem with a quadratic objective function. There is a one-to-

one correspondence between the set of all permutations of the set $\{1, \ldots, n\}$ and the set of $n \times n$ *permutation matrices* Π (see Definition (1.2)). For the entries of the permutation matrix $X = (x_{ij})$ we specify

$$x_{ik} = \begin{cases} 1 & \text{if and only if } \pi(i) = k \\ 0 & \text{otherwise.} \end{cases}$$

Note that if i is assigned to k and j is assigned to l, i.e. if $\pi(i) = k,\ \pi(j) = l$, we have

$$\sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} = a_{ij} b_{\pi(i),\pi(j)}.$$

Hence, an equivalent formulation of QAP is

$$\min_{X \in \Pi} \ \sum_{ijkl} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{ik} c_{ik} x_{ik}. \tag{3.4}$$

The problem formulation (3.4) is called the *Koopmans-Beckmann formulation* of the QAP.

Another equivalent formulation of the QAP can be obtained using the *trace* of the matrix. Note that for the ik-entry of AXB^T is

$$(AXB^T)_{ik} = \sum_{jl} a_{ij} x_{jl} b_{kl}$$

and ith diagonal entry of CX^T is

$$(CX^T)_{ii} = \sum_{k} c_{ik} x_{ik}.$$

Therefore we have

$$\begin{aligned} \sum_{ijkl} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{ik} c_{ik} x_{ik} &= \sum_{ik} (AXB^T)_{ik} x_{ik} + \sum_{i} (CX^T)_{ii} \\ &= \text{tr } (AXB^T + C)X^T, \end{aligned}$$

and the *trace formulation* of QAP is

$$\min_{X \in \Pi} \ \text{tr } (AXB^T + C)X^T.$$

This formulation of the QAP was introduced by Edwards in [22, 23]. The trace formulation in the space of $n \times n$ matrices truly illustrates the nature

of the problem and is favorable for easy manipulation and relaxation of the model.

From (1.2) and (1.5) it follows that

$$\text{tr } (AXB^T + C)X^T = x^T \text{vec}(AXB^T)x + c^T x = x^T(B \otimes A)x + c^T x,$$

where $x = \text{vec}(X)$ and $c = \text{vec}(C)$. The *Kronecker-product formulation* of the QAP is

$$\min_{X \in \Pi} \; x^T(B \otimes A)x + c^T x.$$

Finally, since $x_{ij} = x_{ij}^2$, we present the *Kronecker-Diag formulation* of QAP

$$\min_{X \in \Pi} \; x^T(B \otimes A + \text{Diag}(c))x.$$

3.3 Applications

There is a large variety of applications of the QAP. Here we supply an overview of published applications of QAP. In 1957 Koopmans and Beckmann [56] derived the QAP as a mathematical model of assigning a set of economic activities to a set of locations. A very important area of applications of QAPs is the "wiring" problem. In a 1961 paper [87], Steinberg described a "backboard wiring" problem. The problem concerns the placement of computer components so as to minimize the total wiring length required to connect them. In the particular instance considered by Steinberg, 34 components with a total of 2625 interconnections are to be placed on a blackboard with 36 open positions. To formulate the wiring problem mathematically it is convenient to add 2 dummy components, with no connections to any others, so that the number of components and locations are both $n = 36$. Steinberg considers 1-norm, 2-norm, and squared 2-norm distances between the blackboard locations. These three norm versions of the Steinberg wiring problem are now known as the Ste36a, Ste36c, and Ste36b problem instances. They are included in QAPLIB [14], a Quadratic Assignment Problem Library, established in 1991 by Burkard, Karisch, and Rendl.

In 1976 Pollatschek, Gershoni, and Radday [72]; and in 1977 Burkard and Offerman [15] investigated the question of assigning letters to a typewriter

keyboard. Suppose the keys of a typewriter are to be arranged on the keyboard such that the time needed to write a text in a certain language is minimal. The set $\{1, \ldots, n\}$ is the set of symbols to be arranged on the keyboard. The matrix A contains the mean frequency of a pair of letters in the considered language. In the study by Burkard and Offerman these quantities were determined by evaluating German, English, and French texts with 100 000 letters and punctuation marks. The entry b_{kl} of the matrix B is the number of times key l is pressed after pressing key k. If the ith symbol is assigned to key k, i.e. $k = \pi(i)$, and jth symbol is assigned to key l, i.e. $l = \pi(j)$, the product $a_{ij}b_{\pi(i),\pi(j)}$ is the time needed to write symbol j after symbol i. In order to minimize the average time for writing a text, the QAP should be solved. Note that this QAP is not a symmetric one.

In 1972, as a part of the design of a German university hospital Klinikum Regensburg in Germany, arose the problem of assigning rooms in a hospital. Krarup [53] models that problem as a QAP. In a 1977 paper [24], Elshafei describes a hospital layout as a QAP. The results show that it is more interesting to minimize the largest distance rather than the sum of all distances. The Krarup and Elshafei instances are also included in QAPLIB [14].

Krarup and Pruzan (1978) [54] model the ranking of archeological data, and Heffley (1976) [41] models the ranking of a team in a relay race as a QAP. Problems as a balancing of turbine runners (see [83] and [57]); analysis of chemical reactions for organic compounds (see [92]), and scheduling [30] leads also to a QAP.

From a graph-theoretical point of view, there are series of graph optimization problems, that can be modeled as a QAP with special structure. The Traveling Salesman Problem (TSP) is a QAP where matrix A is the distance matrix of the problem, and B is the adjacency matrix of a cycle. In the case of the graph partition problem, matrix A is the weighted adjacency matrix of a graph, while the matrix B is the adjacency matrix of two disjoint complete graphs. Also, other types of graph problems, such as Max-Clique, Bandwidth minimization [37], graph isomorphism and the largest common subgraph problem can be reduced to the solution of QAP.

3.4 Computational Complexity of QAP

The QAP is from a worst case computational complexity point of view one of the most difficult combinatorial optimization problems. The QAP is well known to be an NP–hard combinatorial optimization problem, as shown by Sahni and Gonzales [82].

Let

$$z(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{\pi(i),\pi(j)} + \sum_{i=1}^{n} c_{i,\pi(i)}.$$

We introduce the notation of an ϵ-approximation algorithm and ϵ-approximate solution.

Definition 3.1 *[17, pg. 18] Given a real number $\epsilon > 0$, an algorithm Υ for the QAP is said to be an* ϵ-approximation algorithm *if and only if for every instance QAP the following holds:*

$$\left|\frac{z(\pi_\Upsilon) - z(\pi_{opt})}{z(\pi_{opt})}\right| \leq \epsilon,$$

where π_Υ is the solution to QAP computed by algorithm Υ and π_{opt} is an optimal solution to QAP. The solution of QAP produced by an ϵ-approximation algorithm is called an ϵ-approximate solution.

Sahni and Gonzales have also proved that even finding an ϵ-approximate solution for QAP is a NP–hard problem, see [82]. The practice shows that the QAP is extremely difficult to solve to optimality. The computational effort to solve the QAP is very likely to grow exponentially with the problem size. Problems of size $n \geq 20$ are currently considered as huge problems. Christofides and Gerrard [18] show that QAP can be solved in polynomial time using dynamic programming if the matrices A and B are a weighted adjacency matrices of a tree. If only one of these matrices is a weighted adjacency matrix, the problem remains NP-hard.

3.5 A Convex Quadratic Programming Relaxation

Anstreicher et. al [4, 5, 6] made recently a break-through by solving a number of previously unsolved large QAPs. The size of these instances ranges

from $20 \leq n \leq 36$. Their computations are considered to be among the most extensive computations ever performed to solve discrete optimization problems. The key to this break-through lies in the use of a bound for QAP that is both fast to compute, and gives good approximations to the exact value of QAP. Anstreicher-Brixius bounding procedure combines orthogonal, semidefinite and convex quadratic relaxations in a nontrivial way, starting from the Hoffman-Wielandt inequality, see Theorem A.2. Their bounds are known as a convex quadratic programming bounds (QPB), see [4].

We use here the parameterization

$$X = \frac{1}{n}E + V\hat{X}V^T, \tag{3.5}$$

from Lemma 4.2, and assume in addition that $V^TV = I_{n-1}$. By use of (3.5) we get

$$\begin{aligned} AXBX^T &= AV\hat{X}V^TBV\hat{X}^TV^T + \frac{1}{n}(AEBV\hat{X}^TV + AV\hat{X}V^TBE) \\ &\quad + \frac{1}{n^2}AEBE \\ &= AV\hat{X}V^TBV\hat{X}^TV^T + \frac{1}{n}(AEBX^T + AXBE) - \frac{1}{n^2}AEBE. \end{aligned}$$

Hence

$$\mathrm{tr}(AXBX^T) = \mathrm{tr}(\hat{A}\hat{X}\hat{B}\hat{X}^T) + \frac{2}{n}\mathrm{tr}(Aee^TBX^T) - \frac{1}{n^2}s(A)s(B), \tag{3.6}$$

for $\hat{A} = V^TAV,\ \hat{B} = V^TBV$, and

$$s(M) := e^TMe = \sum_{ij} m_{ij}.$$

Using (3.6), we have

$$\begin{aligned} \mathrm{tr}(AXB + C)X^T &= \mathrm{tr}(\hat{A}\hat{X}\hat{B}\hat{X}^T) + \mathrm{tr}(\hat{C} + \frac{2}{n}V^TAee^TBV)\hat{X}^T \\ &\quad + \frac{1}{n^2}s(A)s(B) + \frac{1}{n}s(C), \end{aligned} \tag{3.7}$$

where $\hat{C} = V^TCV$. Hadley et al. [34] use this to bound the quadratic term in $\hat{X}$ by the minimal scalar product of the eigenvalues of $\hat{A}$ and $\hat{B}$, see Theorem

A.2. Anstreicher and Brixius [4] use this observation as a starting point and observe that for any $\hat{S}, \hat{T} \in \mathcal{S}_{n-1}$, and $\hat{X} \in \mathcal{O}_{n-1}$, one has

$$\begin{aligned} 0 &= \text{tr } \hat{S}(I - \hat{X}\hat{X}^T) = \text{tr } \hat{S} - \text{tr } \hat{S}\hat{X}I\hat{X}^T = \text{tr } \hat{S} - \text{tr}(I \otimes \hat{S})(\hat{x}\hat{x}^T) \\ 0 &= \text{tr } \hat{T}(I - \hat{X}^T\hat{X}) = \text{tr } \hat{T} - \text{tr } I\hat{X}\hat{T}\hat{X}^T = \text{tr } \hat{T} - \text{tr}(T \otimes \hat{I})(\hat{x}\hat{x}^T), \end{aligned}$$

where $\hat{x} = \text{vec}(\hat{X})$. We use these results to obtain the following identity

$$\text{tr}(\hat{A}\hat{X}\hat{B}\hat{X}^T) = \text{tr}(\hat{S} + \hat{T}) + \text{tr}(\hat{B} \otimes \hat{A} - I \otimes \hat{S} - \hat{T} \otimes I)(\hat{x}\hat{x}^T). \tag{3.8}$$

Let

$$\hat{Q} := \hat{B} \otimes \hat{A} - I \otimes \hat{S} - \hat{T} \otimes I,$$

and $\hat{D} = \hat{C} + \frac{2}{n}V^T Aee^T BV$. We substitute this into (3.7) and get

$$\text{tr}(AXB + C)X^T = \text{tr}(\hat{S} + \hat{T}) + \hat{x}^T\hat{Q}\hat{x} + \hat{d}^T\hat{x} + \frac{1}{n^2}s(A)s(B) + \frac{1}{n}s(C). \tag{3.9}$$

This relation is true for any orthogonal X and $\hat{X}$ related by (3.5) and symmetric $\hat{S}, \hat{T}$. In order to express the parts in (3.9) containing $\hat{X}$ by the original matrix X we use the following identities:

$$\begin{aligned} \text{tr } \hat{S}(I - V^TV) &= \text{tr } \hat{S}(I - V^TXX^TV) = \text{tr } \hat{S} - \text{tr}(V\hat{S}V^T)XIX^T \\ &= \text{tr } \hat{S} - \text{tr}(I \otimes V\hat{S}V^T)(xx^T) = 0, \\ \text{tr } \hat{T}(I - V^TV) &= \text{tr } \hat{T}(I - V^TX^TXV) = \text{tr } \hat{T} - \text{tr } IX(V\hat{T}V^T)X^T \\ &= \text{tr } \hat{T} - \text{tr}(V\hat{T}V^T \otimes I)(xx^T) = 0. \end{aligned}$$

Hence, for any orthogonal X, and any symmetric $\hat{S}$, $\hat{T}$ we also have

$$\text{tr}(AXB + C)X^T = \text{tr}(\hat{S} + \hat{T}) + x^TQx + c^Tx, \tag{3.10}$$

for

$$Q = B \otimes A - I \otimes (V\hat{S}V^T) - (V\hat{T}V^T) \otimes I.$$

Comparing (3.9) and (3.10) we note that

$$\hat{x}^T\hat{Q}\hat{x} + \hat{d}^T\hat{x} + \frac{1}{n^2}s(A)s(B) + \frac{1}{n}s(C) = x^TQx + c^Tx.$$

Note that Q and $\hat{Q}$ depend on the specific choice of $\hat{S}$, $\hat{T}$. Anstreicher and Brixius use the optimal solution $\hat{S}$, $\hat{T}$ from Theorem A.3 and observe

that dual feasibility yields $\hat{Q} \succeq 0$. Therefore the above problem is a *convex quadratic programming problem*. We denote its optimal solution as the quadratic programming bound.

$$QPB(A, B, C) := \mathrm{tr}(\hat{S} + \hat{T}) + \min\{x^T Q x + c^T x : \ x = \mathrm{vec}(X), \ X \in \mathcal{E}\}.$$

Note that as a consequence of Theorem A.3 matrix Q is positive semidefinite over the set of assignment constraints and that optimal $\hat{S}$, $\hat{T}$ can easily be obtained from the spectral decomposition of $\hat{A}$ and $\hat{B}$. The QP bounds depend on the choice for the dual basis. Anstreicher and Brixius use two different dual basis and obtain QPB0 and QPB1 bounds. In [4] is proved that in general QPB cannot be worse then projected eigenvalue bound (PB) introduced in [34].

Anstreicher et al. [5] solve approximately QPB by using the well-known Frank-Wolfe (FW) algorithm [29]. Although the FW method is known to have poor asymptotic performance, Anstreicher et al. use that method in their computations because of the following reasons. Each iteration of the FW algorithm requires the solution of a linear assignment problem, which can be performed extremely rapidly. The FW algorithm generates dual information that can be used to estimate the effect of fixing an assignment $x_{ij} = 1$ to create a "child" problem of a node in the B&B tree. The branching rules for the branch and bound algorithm are based on this dual informations. In Tables 7.3, 7.4 we list some of the bounds reported in [4], and in Table 7.9 are given QPB results for first level in the branching tree for Had12.

Chapter 4

SDP Relaxations of QAP in $\mathcal{S}_{n^2+1}$

In this Chapter we present three SDP relaxations of increasing complexity, placed in $\mathcal{S}_{n^2+1}$. We present also numerical results obtained by interior point method for the relaxation QAP_{R_2} on Nugent-type instances. We prove the linear dependency of the arrow and gangster constraints.

4.1 Deriving the Relaxations

The natural way (see [47, 97]) of embedding the 0–1 problem into the semidefinite framework is by lifting it into a higher dimensional space of symmetric matrices. For that purpose, we introduce the matrix variable

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 & x^T \end{pmatrix} = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}, \quad x = \text{vec}(X),\ X \in \Pi. \tag{4.1}$$

This matrix $Y \in \mathcal{S}_{n^2+1}$ is positive semidefinite, i. e. $Y \succeq 0$ and satisfies

$$Y_{00} = 1 \tag{4.2}$$

$$Y_{ii} = Y_{0i} = Y_{i0}, \quad i = 1, \ldots, n^2, \tag{4.3}$$

where we use index 0 for the first row and column of the matrix. Constraints (4.3) are equivalent to

$$\text{diag}(xx^T) = x.$$

Equations (4.2) and (4.3) can be formally summarized as $\text{arrow}(Y) = e_0$, where the linear operator $\text{arrow} : \mathcal{S}_{n^2+1} \to I\!R^{n^2+1}$ is defined as

$$\text{arrow}(Y) := \text{diag}(Y) - (0, (Y_{1:n^2,0})), \quad Y \in \mathcal{S}_{n^2+1}. \tag{4.4}$$

The adjoint operator of the operator arrow is $\text{Arrow} : I\!R^{n^2+1} \to \mathcal{S}_{n^2+1}$, given by

$$\text{Arrow}(w) = \begin{pmatrix} w_0 & -\frac{1}{2}w_{1:n^2}^T \\ -\frac{1}{2}w_{1:n^2} & \text{Diag}(w_{1:n^2}) \end{pmatrix}.$$

In order to rewrite the objective function from QAP we use (1.2) and (1.5) and obtain the following form of the objective function

$$\text{tr } (AXB + C)X^T = \langle x, \text{vec}(AXB + C)\rangle = x^T(B \otimes A)x + x^T c,$$

where $x = \text{vec}(X)$ and $c = \text{vec}(C)$. Using the fact that x is 0–1 vector, the objective function of the QAP becomes

$$\text{tr} \begin{pmatrix} 1 & x^T \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{2}\, c^T \\ \frac{1}{2}\, c^T & B \otimes A \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \text{tr } \bar{L}Y, \tag{4.5}$$

where

$$\bar{L} := \begin{pmatrix} 0 & \frac{1}{2}\, c^T \\ \frac{1}{2}\, c^T & B \otimes A \end{pmatrix} \in \mathcal{M}_{n^2+1},$$

and Y is a matrix of the form (4.1).

Remark 4.1 *Note that equivalently we can set*

$$\bar{L} = \begin{pmatrix} 0 & 0 \\ 0 & B \otimes A + \text{Diag}(c) \end{pmatrix}.$$

We define the *feasible set* of QAP

$$\mathcal{P} := \text{conv } \left\{ \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : \; x = \text{vec}(X), \; X \in \Pi \right\}, \tag{4.6}$$

and write QAP in the following form

$$\mu^* = \min\{\text{tr } \bar{L}Y : \; Y \in \mathcal{P}\}.$$

In order to obtain tractable relaxations for QAP we need to approximate the set $\mathcal{P}$ by larger sets containing $\mathcal{P}$. The following lemma collects known results.

Lemma 4.1 $Y \in \mathcal{P} \Leftrightarrow Y \succeq 0,\ \mathrm{arrow}(Y) = e_0$ *and* $\mathrm{rank}(Y) = 1$.

■

We next exploit the fact that the row and column sums of permutation matrices are one.

Lemma 4.2 *[34] Let V be an $n \times (n-1)$ matrix with $V^T e = 0$ and* $\mathrm{rank}(V) = n - 1$. *Then*

$$\left\{X \in \mathcal{M}_n : Xe = X^T e = e\right\} = \left\{\frac{1}{n} ee^T + VMV^T : M \in \mathcal{M}_{n-1}\right\}.$$

■

Matrix V from the previous Lemma could be any basis of $e^{\perp}$. Our choice for V is

$$V = \begin{pmatrix} I_{n-1} \\ -e_{n-1}^T \end{pmatrix}. \tag{4.7}$$

Let us define the $n^2 \times ((n-1)^2 + 1)$ matrix

$$W := \left(\frac{1}{n} e \otimes e, V \otimes V\right), \tag{4.8}$$

and $(n^2 + 1) \times ((n-1)^2 + 1)$ matrix

$$\hat{V} := \begin{pmatrix} e_0^T \\ W \end{pmatrix}. \tag{4.9}$$

Lemma 4.3 *For any $Y \in \mathcal{P}$ there exists a symmetric matrix R of order $(n-1)^2 + 1$, indexed from 0 to $(n-1)^2$, such that*

$$R \succeq 0,\ R_{00} = 1 \text{ and } Y = \hat{V} R \hat{V}^T.$$

PROOF. (See also [97].) First we look at the extreme points of $\mathcal{P}$. Let Y be one of them. Thus

$$Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix},$$

for some permutation matrix X. From Lemma 4.2 it follows that for the permutation matrix X there exists some matrix $M \in \mathcal{M}_{n-1}$ such that $X = \frac{1}{n}ee^T + VMV^T$. With the use of (1.5), we get

$$x = \mathrm{vec}(X) = \frac{1}{n}(e \otimes e) + (V \otimes V)m = Wz,$$

where $m = \mathrm{vec}(M)$, $z = \begin{pmatrix} 1 \\ m \end{pmatrix}$ and W is defined in (4.8). Now

$$Y = \begin{pmatrix} 1 & (Wz)^T \\ Wz & Wzz^TW^T \end{pmatrix} = \begin{pmatrix} e_0^T \\ W \end{pmatrix} zz^T \begin{pmatrix} e_0^T \\ W \end{pmatrix}^T = \hat{V}R\hat{V}^T,$$

with $R = zz^T$ and $\hat{V}$ defined in (4.9). Hence, R is symmetric positive semidefinite matrix and $R_{00} = 1$. The same holds for convex combinations formed from several permutation matrices. ■

Remark 4.2 *The condition $R_{00} = 1$ appears naturally from definitions of Y and $\hat{V}$.*

Lemma 4.4 *[97] Let $R \in \mathcal{S}_{(n-1)^2+1}$ be arbitrary and let*

$$Y = \hat{V}R\hat{V}^T.$$

Then, using the block notation of (1.1), we have
1. $y_{00} = r_{00}$, $Y^{0j}e = r_{00}$, and $\sum_{j=1}^n Y^{0j} = r_{00}e^T$.
2. $Y^{0j} = e^TY^{ij}$, for $i, j = 1, \ldots, n$.
3. $\sum_{i=1}^n Y^{ij} = eY^{0j}$ and $\sum_{i=1}^n \mathrm{diag}(Y^{ij}) = Y^{j0}$, for $j = 1, \ldots, n$.

The Lemma 4.3 suggests the following set $\hat{P}$ containing P.

$$\hat{\mathcal{P}} := \{Y \in \mathcal{S}_{n^2+1} : \exists R \text{ s.t. } R \succeq 0,\ R_{00} = 1,\ Y = \hat{V}R\hat{V}^T,\ \mathrm{arrow}(Y) = e_0\}.$$

In [97] it is shown that $\hat{P}$ has interior points. For instance

$$\hat{R} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{n^2(n-1)}(nI_{n-1}E_{n-1}) \otimes (nI_{n-1}E_{n-1}) \end{pmatrix} \succ 0$$

is such that $\hat{V}\hat{R}\hat{V}^T$ is the barycenter of $\mathcal{P}$, i. e.

$$\hat{V}\hat{R}\hat{V}^T = \frac{1}{n!}\sum_{x\in\Pi}\begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}. \tag{4.10}$$

Theorem 4.1 *[97] Let $\hat{Y} = \hat{V}\hat{R}\hat{V}^T$ be the barycenter of $\mathcal{P}$.*
Then:
1. $\hat{Y}$ has a 1 in the $(0,0)$ 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 0.

$$\hat{Y} = \left(\begin{array}{c|c} 1 & \frac{1}{n}e^T \\ \hline \frac{1}{n}e & (\frac{1}{n^2}E \otimes E) + \frac{1}{n^2(n-1)}(nI - E) \otimes (nI - E) \end{array}\right).$$

2. The rank of $\hat{Y}$ is given by

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

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

$$\left(2, \frac{1}{n-1}e^T_{(n-1)^2}, 0e^T_{2n-1}\right)^T.$$

4. Let

$$T := \left(\frac{I \otimes e^T}{e^T \otimes I}\right).$$

The null space and range space are

$$\mathcal{N}ull(\hat{Y}) = \mathcal{R}ang(\hat{T}^T) \quad \text{and} \quad \mathcal{R}ang(\hat{Y}) = \mathcal{R}ang(\hat{V})$$

(so that $\mathcal{N}ull(\hat{T}) = \mathcal{R}ang(\hat{V})$).

Table 4.1: Computation times to solve the relaxation QAP_{R_1}

n	10	15	20	25	30	35
time (seconds)	0.43	4.16	19.4	69.3	198.8	548

The *basic SDP relaxation* of QAP is

$$(\text{QAP}_{R_1}) \quad \min\{\text{tr } \bar{L}Y : \ Y \in \hat{\mathcal{P}}\}.$$

We can eliminate the $(n^2+1) \times (n^2+1)$ matrix variable Y through the $((n-1)^2+1) \times ((n-1)^2+1)$ matrix variable R in the basic relaxation. For that purpose, we define the following set:

$$\mathcal{R} := \{R \succeq 0 : R \in \mathcal{S}_{(n-1)^2+1}, \ \text{arrow}(\hat{V}R\hat{V}^T) = e_0\}. \tag{4.11}$$

If we define

$$L := \hat{V}^T \bar{L} \hat{V} \in \mathcal{M}_{(n-1)^2+1}, \tag{4.12}$$

then QAP_{R_1} can be written as

$$(\text{QAP}_{R_1}) \quad \min\{\text{tr } LR : \ R \in \mathcal{R}\}.$$

The constraint $R_{00} = 1$ in connection with $\text{arrow}(\hat{V}R\hat{V}^T) = e_0$ is redundant, so it is left out in (4.11). Thus, the basic relaxation contains n^2+1 equality constraints that are coming from the arrow operator. In Table 4.1 we give running times for solving the QAP_{R_1} relaxation of different problem sizes, using a primal-dual path-following interior point method. The running times in seconds are obtained using an Athlon XP with 1800 GHz.

In [97] it is shown that each matrix from $\mathcal{P}$ has a specific zero pattern. For each matrix in $\mathcal{P}$ the off–diagonal elements of the diagonal blocks and the diagonal elements of the off–diagonal blocks are zero. For illustration, let us

look at $\hat{Y} = \hat{V}\hat{R}\hat{V}^T$ for $n = 3$,

$$\hat{Y} = \frac{1}{6}\left(\begin{array}{c|ccc|ccc|ccc} 6 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ \hline 2 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 2 & 0 & 0 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ \hline 2 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ \hline 2 & 0 & 1 & 1 & 0 & 1 & 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 2 & 0 \\ 2 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 \end{array}\right).$$

To express the zero pattern, we index the elements of the matrix $Y \in \mathcal{P}$ by $y_{r,s} = Y_{(i,j)(k,l)}$ for $r, s \in \{0, 1, \ldots, n^2\}$, $i, j, k, l \in \{0, 1, \ldots, n\}$. The zero pattern is covered by the following equalities:

$$y_{rs} = 0 \quad \text{for} \quad r = (i,j),\ s = (i,k), \quad \text{or} \quad r = (j,i),\ s = (k,i),\ j \neq k. \tag{4.13}$$

We collect all these indices (r, s) in the set J, and these equalities in the constraint $G_J(\hat{V}R\hat{V}^T) = 0$. Now we define the set

$$\mathcal{G} := \{R : R \in \mathcal{S}_{(n-1)^2+1},\ G_J(\hat{V}R\hat{V}^T) = 0\}.$$

The operator $G_J : \mathcal{S}_{n^2+1} \to \mathcal{S}_{n^2+1}$ is called the *Gangster operator* (see (1.7)). We strengthen the relaxation QAP_{R_1} by adding this new set of equalities and arrive at the tighter model

$$(\text{QAP}_{R_2}) \quad \min\{\text{tr } LR :\ R \in \mathcal{R} \cap \mathcal{G}\},$$

that contains $O(n^3)$ equations. Model QAP_{R_2} is introduced in [97] as the *Gangster model*.

The relaxation QAP_{R_2} can be further tightened by adding *nonnegativity constraints*

$$(\hat{V}R\hat{V}^T)_{ij} \geq 0, \quad \forall i, j = 0, \ldots, n^2. \tag{4.14}$$

We collect the inequalities (4.14) which are not yet covered by $G_J(\hat{V}R\hat{V}^T) = 0$ in the constraint $N(\hat{V}R\hat{V}^T) \geq 0$. Let us define the set

$$\mathcal{N} := \{R : R \in \mathcal{S}_{(n-1)^2+1},\ N(\hat{V}R\hat{V}^T) \geq 0\}.$$

We arrive at the final relaxation, also introduced in [97]

$$(\mathrm{QAP}_{R_3}) \quad \min\{\mathrm{tr}\ LR :\ R \in \mathcal{R} \cap \mathcal{G} \cap \mathcal{N}\}.$$

The resulting SDP has $O(n^4)$ sign constraints and $O(n^3)$ equality constraints. The relaxation QAP_{R_3} can not be solved straightforward by interior point methods for interesting instances ($n \geq 15$).

4.2 Gangster and Arrow: Linearly Dependent Constraints

Here we eliminate all the linearly dependent constraints that arise from adding the gangster operator to the arrow constraints.

We show first that the system

$$\begin{aligned} \mathrm{arrow}(\hat{V}R\hat{V}^T) &= e_0 \\ G_J(\hat{V}R\hat{V}^T) &= 0, \end{aligned} \tag{4.15}$$

contains linearly dependent equations.

From Lemma 4.4 it follows that

$$Y^{0j} = e^T Y^{jj} \quad \forall j = 1, \ldots, n.$$

Since the off diagonal entries for each Y^{jj} are zero, it follows that all arrow equations except the first one (i.e. $(\hat{V}R\hat{V}^T)_{00} = 1$) are linearly dependent in the system (4.15).

Example 4.1 *For illustration, we consider the case for $n = 4$. The lifted space is $\mathcal{S}_{n^2+1}$. Let $Y \in \mathcal{S}_{n^2+1}$. If Y satisfies the system of equations (4.15) then*

$$Y=\left(\begin{array}{c|cccc|cccc|cccc|cccc}
1 & \alpha_1 & \beta_1 & \gamma_1 & \delta_1 & \alpha_2 & \beta_2 & \gamma_2 & \delta_2 & \alpha_3 & \beta_3 & \gamma_3 & \delta_3 & \alpha_4 & \beta_4 & \gamma_4 & \delta_4 \\ \hline
\alpha_1 & \alpha_1 & 0 & 0 & 0 & 0 & * & * & * & 0 & * & * & * & 0 & * & * & * \\
\beta_1 & 0 & \beta_1 & 0 & 0 & * & 0 & * & * & * & 0 & * & * & * & 0 & * & * \\
\gamma_1 & 0 & 0 & \gamma_1 & 0 & * & * & 0 & * & * & * & 0 & * & * & * & 0 & * \\
\delta_1 & 0 & 0 & 0 & \delta_1 & * & * & * & 0 & * & * & * & 0 & * & * & * & 0 \\ \hline
\alpha_2 & 0 & * & * & * & \alpha_2 & 0 & 0 & 0 & 0 & * & * & * & 0 & * & * & * \\
\beta_2 & * & 0 & * & * & 0 & \beta_2 & 0 & 0 & * & 0 & * & * & * & 0 & * & * \\
\gamma_2 & * & * & 0 & * & 0 & 0 & \gamma_2 & 0 & * & * & 0 & * & * & * & 0 & * \\
\delta_2 & * & * & * & 0 & 0 & 0 & 0 & \delta_2 & * & * & * & 0 & * & * & * & 0 \\ \hline
\alpha_3 & 0 & * & * & * & 0 & * & * & * & \alpha_3 & 0 & 0 & 0 & 0 & * & * & * \\
\beta_3 & * & 0 & * & * & * & 0 & * & * & 0 & \beta_3 & 0 & 0 & * & 0 & * & * \\
\gamma_3 & * & * & 0 & * & * & * & 0 & * & 0 & 0 & \gamma_3 & 0 & * & * & 0 & * \\
\delta_3 & * & * & * & 0 & * & * & * & 0 & 0 & 0 & 0 & \delta_3 & * & * & * & 0 \\ \hline
\alpha_4 & 0 & * & * & * & 0 & * & * & * & 0 & * & * & * & \alpha_4 & 0 & 0 & 0 \\
\beta_4 & * & 0 & * & * & * & 0 & * & * & * & 0 & * & * & 0 & \beta_4 & 0 & 0 \\
\gamma_4 & * & * & 0 & * & * & * & 0 & * & * & * & 0 & * & 0 & 0 & \gamma_4 & 0 \\
\delta_4 & * & * & * & 0 & * & * & * & 0 & * & * & * & 0 & 0 & 0 & 0 & \delta_4
\end{array}\right).$$

On the positions assigned with '$*$' are corresponding elements of the matrix Y. The elements on the arrow positions are $\alpha_i, \beta_i, \gamma_i, \delta_i$ for $i = 1, 2, 3, 4$. Zero elements are on the positions corresponding to the set J.

In [97] is proven that the set

$$\begin{aligned}\hat{J} \;:=\; & \{(i,j) : i = (p-1)n + q, j = (p-1)n + r, q \neq r\} \;\cup \\ & \{(i,j) : i = (p-1)n + q, j = (r-1)n + q, p \neq r, (p, r \neq n) \\ & \qquad ((r,p),(p,r) \neq (n-2, n-1), (n-1, n-2))\},\end{aligned} \tag{4.16}$$

$\hat{J} \subseteq J$, for J defined in (4.13), excludes all linearly dependent indices from J. The set $\hat{J}$ except from the set J the off diagonal block $(n-2, n-1)$ (resp. $(n-1, n-2)$) and the last column (resp. row) of off–diagonal blocks.

Example 4.2 *Let $n = 4$, $Z \in \mathcal{S}_{n^2+1}$, $R \in \mathcal{S}_{(n-1)^2+1}$, and $Y = \hat{V} R \hat{V}^T \in \mathcal{S}_{n^2+1}$. If Z satisfies*

$$\begin{aligned} Z_{00} &= 1 \\ G_{\hat{j}}(Z) &= 0, \end{aligned}$$

then

$$Z = \left(\begin{array}{c|cccc|cccc|cccc|cccc} 1 & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\ \hline * & * & 0 & 0 & 0 & 0 & * & * & * & 0 & * & * & * & * & * & * & * \\ * & 0 & * & 0 & 0 & * & 0 & * & * & * & 0 & * & * & * & * & * & * \\ * & 0 & 0 & * & 0 & * & * & 0 & * & * & * & 0 & * & * & * & * & * \\ * & 0 & 0 & 0 & * & * & * & * & 0 & * & * & * & 0 & * & * & * & * \\ \hline * & 0 & * & * & * & * & 0 & 0 & 0 & * & * & * & * & * & * & * & * \\ * & * & 0 & * & * & 0 & * & 0 & 0 & * & * & * & * & * & * & * & * \\ * & * & * & 0 & * & 0 & 0 & * & 0 & * & * & * & * & * & * & * & * \\ * & * & * & * & 0 & 0 & 0 & 0 & * & * & * & * & * & * & * & * & * \\ \hline * & 0 & * & * & * & * & * & * & * & * & 0 & 0 & 0 & * & * & * & * \\ * & * & 0 & * & * & * & * & * & * & 0 & * & 0 & 0 & * & * & * & * \\ * & * & * & 0 & * & * & * & * & * & 0 & 0 & * & 0 & * & * & * & * \\ * & * & * & * & 0 & * & * & * & * & 0 & 0 & 0 & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * & * & * & * & * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & * & * & * & * & * & * & * & * & 0 & * & 0 & 0 \\ * & * & * & * & * & * & * & * & * & * & * & * & * & 0 & 0 & * & 0 \\ * & * & * & * & * & * & * & * & * & * & * & * & * & 0 & 0 & 0 & * \end{array}\right).$$

On the positions assigned with '$*$' are corresponding elements of the matrix Z. If $Y = \hat{V}R\hat{V}^T$ satisfies

$$\begin{aligned} (\hat{V}R\hat{V}^T)_{00} &= 1 \\ G_{\hat{J}}(\hat{V}R\hat{V}^T) &= 0, \end{aligned}$$

then the matrix $\hat{V}R\hat{V}^T$ is of the form given in Example 4.1. It is due to the properties of the matrix $\hat{V}$.

We define now the following set

$$\hat{J}_0 := \hat{J} \cup (0,0).$$

Remark 4.3 *The relaxation* QAP_{R_2} *can be written as*

$$(\mathrm{QAP}_{R_2}) \qquad \begin{array}{ll} \min & \mathrm{tr}\ LR \\ \text{s.t.} & G_{\hat{J}_0}(\hat{V}R\hat{V}^T) = E_{00} \\ & R \in \mathcal{S}^+_{(n-1)^2+1}. \end{array}$$

The system

$$\begin{aligned} \text{arrow}(\hat{V}R\hat{V}^T) &= e_0 \\ G_{\hat{J}}(\hat{V}R\hat{V}^T) &= 0, \end{aligned} \tag{4.17}$$

contains linearly dependent equations. We want to find linearly dependent $G_{\hat{J}}$ constraints from (4.17). Now we define the set

$$\begin{aligned} \tilde{J} \;:=\; & \{(i,j) : i = (p-1)n+q, j = (p-1)n+r, q \neq r, (q, r \neq n) \\ & \quad ((q,r),(r,q) \neq (n-2,n-1),(n-1,n-2))\} \;\cup \\ & \{(i,j) : i = (p-1)n+q, j = (r-1)n+q, p \neq r, (p, r \neq n) \\ & \quad ((r,p),(p,r) \neq (n-2,n-1),(n-1,n-2))\}. \end{aligned} \tag{4.18}$$

Note that $\tilde{J} \subseteq \hat{J} \subseteq J$. In the following theorem we prove that the system of equations

$$\begin{aligned} \text{arrow}(\hat{V}R\hat{V}^T) &= e_0 \\ G_{\tilde{J}}(\hat{V}R\hat{V}^T) &= 0, \end{aligned}$$

is equivalent to the system of equations

$$G_{\hat{J}_0}(\hat{V}R\hat{V}^T) = E_{00}. \tag{4.19}$$

Example 4.3 *Let* $n = 4$, $Z \in \mathcal{S}_{n^2+1}$, $R \in \mathcal{S}_{(n-1)^2+1}$, *and* $Y = \hat{V}R\hat{V}^T \in \mathcal{S}_{n^2+1}$. *If* Z *satisfies*

$$\begin{aligned} Z_{00} &= 1 \\ G_{\tilde{J}}(Z) &= 0, \end{aligned}$$

then

$$Z = \left(\begin{array}{c|cccc|cccc|cccc|cccc}
* & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\
\hline
* & * & 0 & 0 & \bullet & 0 & * & * & * & 0 & * & * & * & * & * & * & * \\
* & 0 & * & \bullet & \bullet & * & 0 & * & * & * & 0 & * & * & * & * & * & * \\
* & 0 & \bullet & * & \bullet & * & * & 0 & * & * & * & 0 & * & * & * & * & * \\
* & \bullet & \bullet & \bullet & * & * & * & * & 0 & * & * & * & 0 & * & * & * & * \\
\hline
* & 0 & * & * & * & * & 0 & 0 & \bullet & * & * & * & * & * & * & * & * \\
* & * & 0 & * & * & 0 & * & \bullet & \bullet & * & * & * & * & * & * & * & * \\
* & * & * & 0 & * & 0 & \bullet & * & \bullet & * & * & * & * & * & * & * & * \\
* & * & * & * & 0 & \bullet & \bullet & \bullet & * & * & * & * & * & * & * & * & * \\
\hline
* & 0 & * & * & * & * & * & * & * & * & 0 & 0 & \bullet & * & * & * & * \\
* & * & 0 & * & * & * & * & * & * & 0 & * & \bullet & \bullet & * & * & * & * \\
* & * & * & 0 & * & * & * & * & * & 0 & \bullet & * & \bullet & * & * & * & * \\
* & * & * & * & 0 & * & * & * & * & \bullet & \bullet & \bullet & * & * & * & * & * \\
\hline
* & * & * & * & * & * & * & * & * & * & * & * & * & * & 0 & 0 & \bullet \\
* & * & * & * & * & * & * & * & * & * & * & * & * & 0 & * & \bullet & \bullet \\
* & * & * & * & * & * & * & * & * & * & * & * & * & 0 & \bullet & * & \bullet \\
* & * & * & * & * & * & * & * & * & * & * & * & * & \bullet & \bullet & \bullet & * \\
\end{array}\right).$$

On the positions assigned with $'*'$ *and* $'\bullet'$ *are corresponding elements of the matrix* Z*. Elements* $'\bullet'$ *are on the positions from the indices set* $\hat{J} \setminus \tilde{J}$*.*

If $Y = \hat{V}R\hat{V}^T$ *satisfies (4.19) then the matrix* $\hat{V}R\hat{V}^T$ *is of the form given in Example 4.1.*

Theorem 4.2 *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 $G_{\tilde{j}}(\hat{V}R\hat{V}^T) = 0$ *and* $\operatorname{arrow}(\hat{V}R\hat{V}^T) = e_0$ *implies that*

$$Y_{kn+1,(k-1)n+j} = 0 \quad \text{for} \quad k = 1, \dots, n, \ j = 2, \dots, n-1$$

and

$$Y_{k,k-1} = 0 \quad \text{for} \quad k = n, 2n, \dots n^2.$$

PROOF: Let $G_{\tilde{J}}(Y) = 0$ and $\text{arrow}(Y) = e_0$. From Lemma 4.4 it follows that

$$Y^{0j} = e^T Y^{jj} \quad \forall j = 1, \ldots, n.$$

For $i = 1, \ldots, n-3$ we have

$$(Y^{0j})_i = \sum_{k=1}^{n} Y^{jj}_{k,i}.$$

Because of $G_{\tilde{J}}(Y) = 0$ is

$$Y^{jj}_{k,i} = 0 \quad \text{for} \quad k \neq i,\ i, k \neq n.$$

Hence

$$(Y^{0j})_i = Y^{jj}_{i,i} + Y^{jj}_{n,i}.$$

Because of the arrow constraint it follows that

$$Y^{jj}_{i,i} = (Y^{0j})_i,$$

and

$$Y^{jj}_{n,i} = 0.$$

Analogously we get $Y^{jj}_{i,n} = 0$.

For $i = n-2, n-1, n$ we have

$$\begin{aligned} Y^{jj}_{n-1,n-2} + Y^{jj}_{n,n-2} &= 0 \\ Y^{jj}_{n,n-2} + Y^{jj}_{n,n-1} &= 0 \\ Y^{jj}_{n-1,n-2} + Y^{jj}_{n,n-1} &= 0. \end{aligned}$$

Previously system implies that

$$\begin{aligned} Y^{jj}_{n-1,n-2} &= 0 \\ Y^{jj}_{n,n-2} &= 0 \\ Y^{jj}_{n,n-1} &= 0, \end{aligned}$$

and Theorem is proved. ∎

Remark 4.4 *The relaxation* QAP_{R_2} *can be written as*

$$(\text{QAP}_{R_2}) \qquad \begin{array}{ll} \min & \text{tr } LR \\ \text{s.t.} & G_{\tilde{J}}(\hat{V} R \hat{V}^T) = 0 \\ & \text{arrow}(\hat{V} R \hat{V}^T) = e_0 \\ & R \in \mathcal{S}^+_{(n-1)^2+1}. \end{array}$$

4.3 Solving the QAP_{R_2} Relaxation

We solve the QAP_{R_2} relaxation with the interior point method. Solutions of the Nugent-type problems for $n < 20$ are computed on our PC (Athlon XP 1800 or Intel P4), and for $n \geq 20$ by use of the NEOS Server for Optimization. Results are presented in Table 4.2.

The primal program of the primal-dual pair for the model QAP_{R_2} is

$$(\mathrm{QAP}_{R_2}) \qquad \begin{array}{ll} \min & \mathrm{tr}\, LR \\ \text{s.t.} & \mathrm{arrow}(\hat{V}R\hat{V}^T) = e_0 \\ & G_{\tilde{J}}(\hat{V}R\hat{V}^T) = 0 \\ & R \in \mathcal{S}^+_{n^2+1}, \end{array}$$

and the dual program is

$$(\mathrm{DQAP}_{R_2}) \qquad \begin{array}{ll} \max & -w_0 \\ \text{s.t.} & L + \hat{V}^T(\mathrm{Arrow}(w) + G_{\tilde{J}}(Y))\hat{V} - Z = 0 \\ & Z \in \mathcal{S}^+_{n^2+1},\ w \in I\!R^{n^2+1},\ Y \in S_{\tilde{J}}, \end{array}$$

where set $S_{\tilde{J}}$ is defined in (1.8). Here we use the fact that G is a self-adjoint operator.

Writing the QAP_{R_2} relaxation in the general form of Section 2.1 we collect the equations in the operator

$$\mathcal{A}(\cdot) := \begin{pmatrix} \mathrm{arrow}(\cdot) \\ G_{\tilde{J}}(\cdot) \end{pmatrix},$$

and $a = e_0$, the unit vector in $I\!R^{n^2+1+|J|}$. The final system (ΔKKT) for QAP_{R_2} is (see (2.16))

$$\begin{aligned} &\mathcal{A}(\hat{V}Z^{-1}\hat{V}^T\mathcal{A}^T(\Delta w, \Delta Y)\hat{V}R\hat{V}^T) \\ &\qquad = \mathcal{A}(\mu\hat{V}Z^{-1}\hat{V}^T - \hat{V}Z^{-1}(L + \hat{V}^T\mathcal{A}^T(w,Y)\hat{V} - Z)R\hat{V}^T) - a, \end{aligned} \tag{4.20}$$

where $\Delta w \in I\!R^{n^2+1}$ and $\Delta Y \in S_{\tilde{J}}$. We define

$$RT := \hat{V}R\hat{V}^T \quad \text{and} \quad ZT := \hat{V}Z^{-1}\hat{V}^T.$$

The left hand side of the system (4.20) is a 2×2 block,

$$M(\cdot, *) = \begin{bmatrix} \mathrm{arrow}(ZT\mathrm{Arrow}(\cdot)RT) & \mathrm{arrow}(ZTG_{\tilde{J}}(*)RT) \\ G_{\tilde{J}}(ZT\mathrm{Arrow}(\cdot)RT) & G_{\tilde{J}}(ZTG_{\tilde{J}}(*)RT) \end{bmatrix}. \tag{4.21}$$

The QAP_{R_2} relaxation contains $O(n^4)$ constraints. In order to solve the relaxation with the interior point method, we need to efficiently exploit the structure of the problem. We represent the linear operator M by a matrix $S \in \mathcal{M}_{(n^2+1)+|J|}$ whose columns are by definition the images of the basis vectors acting on M. Here we give expressions for all four blocks of the matrix S.

For the jth basis vector we find

$$ZT\mathrm{Arrow}(e_j)RT = \begin{cases} ZT_{\cdot,0}RT_{\cdot,0}^T & \text{if} \quad j = 0 \\ ZT_{\cdot,j}RT_{\cdot,j}^T - \frac{1}{2}(ZT_{\cdot,0}RT_{\cdot,j}^T + ZT_{\cdot,j}RT_{\cdot,0}^T) & \text{if} \quad j > 0. \end{cases} \tag{4.22}$$

Now, the jth column of the block S_{11} is

$$\begin{aligned} &\mathrm{arrow}(ZT\mathrm{Arrow}(e_j)RT) \\ &= \begin{cases} ZT_{\cdot,0} \circ RT_{\cdot,0} - \frac{1}{2}(ZT_{00}RT_{\cdot,0} + RT_{00}ZT_{\cdot,0}) & \text{if} \quad j = 0 \\ ZT_{\cdot,j} \circ RT_{\cdot,j} - \frac{1}{2}(ZT_{0j}RT_{\cdot,j} + RT_{0j}ZT_{\cdot,j}) & \\ -\frac{1}{2}(ZT_{\cdot,0} \circ RT_{\cdot,j} + ZT_{\cdot,j} \circ RT_{\cdot,0}) & \\ +\frac{1}{4}(ZT_{00}RT_{\cdot,j} + RT_{0j}ZT_{\cdot,0} + ZT_{0j}RT_{\cdot,0} + RT_{00}ZT_{\cdot,j}) & \text{if} \quad j > 0. \end{cases} \end{aligned}$$

Remark 4.5 *The definition of* $\mathrm{arrow}(\cdot)$ *was extended such that it can act on the unsymmetric matrix* $ZT\mathrm{Arrow}(\cdot)RT$.

We construct the matrix S_{11} in two steps. First we compute

$$ZT \circ RT - \frac{1}{2}(ZT_{0,\cdot} \circ RT_{\cdot,0} + ZT_{\cdot,0} \circ RT_{\cdot,0}),$$

and then add to each column $j > 0$ of S_{11} the corresponding vector

$$\frac{1}{4}(ZT_{00}RT_{\cdot,j} + RT_{0,j}ZT_{\cdot,0} + ZT_{0,j}RT_{\cdot,0} + RT_{00}ZT_{\cdot,j}).$$

The matrix $S_{11} \in \mathcal{S}_{n^2+1}^{++}$.

Remark 4.6 *Matrix* S_{11} *is also the system matrix for the basic relaxation* $\mathrm{QAP}_{\mathrm{R1}}$.

Now we present the block $M_{21}(\cdot) = G_{\tilde{j}}(ZT\text{Arrow}(\cdot)RT)$. We represent the indices in the vectors P and Q in the following way

$$(i_k, j_k) \in J \quad \Leftrightarrow \quad P(k) = i_k \ \text{ and } \ Q(k) = j_k.$$

We use the expression (4.22) and arrive to

$$S_{21} = \begin{cases} \frac{1}{2}(ZT_{P,0} \circ RT^T_{0,Q} + ZT_{Q,0} \circ RT^T_{0,P}) & \text{if} \quad j = 0 \\ \frac{1}{2}(ZT_{P,j} \circ RT^T_{j,Q} - \frac{1}{2}(ZT_{P,0} \circ RT^T_{j,Q} + ZT_{P,j} \circ RT^T_{0,Q}) & \\ +ZT_{Q,j} \circ RT^T_{j,P} - \frac{1}{2}(ZT_{Q,0} \circ RT^T_{j,P} + ZT_{Q,j} \circ RT^T_{0,P})) & \text{if} \quad j > 0. \end{cases}$$

Remark 4.7 *The definition of the gangster operator was extended such that it can act on the unsymmetric matrix $ZT\text{Arrow}(\cdot)RT$.*

From (4.21) it follows that

$$S_{12} = S^T_{21}.$$

By use of the same notation, the zero column of the block S_{22} is

$$S_{22}(\cdot, 0) = \frac{1}{2}(ZT_{P,0} \circ RT^T_{0,Q} + ZT_{Q,0} \circ RT^T_{0,P}),$$

and jth column for $j > 0$ is

$$\begin{aligned} S_{22}(\cdot, j) = \frac{1}{2}(ZT_{P,j} \circ RT^T_{j,Q} - \frac{1}{2}(ZT_{P,0} \circ RT^T_{j,Q} + ZT_{P,j} \circ RT^T_{0,Q}) \\ +ZT_{Q,j} \circ RT^T_{j,P} - \frac{1}{2}(ZT_{Q,0} \circ RT^T_{j,P} + ZT_{Q,j} \circ RT^T_{0,P})). \end{aligned}$$

The matrix

$$S = \begin{pmatrix} S_{11} & S^T_{21} \\ S_{21} & S_{22} \end{pmatrix}$$

of the final equation (4.20) is positive definite, see [38].

Remark 4.8 *We use a Cholesky factorization for solving the linear system formulation (4.20) with the predictor–corrector interior point method.*

Since the QAP_{R2} relaxation contains $O(n^3)$ equations, we are not able to solve the relaxation on our PC (Athlon XP 1800 or Intel P4) for $n \geq 20$. In

Table 4.2: Optimal solutions of QAP_{R_2} relaxation obtained using NEOS and computation time for one interior point iteration

	exact	QAP_{R_2}	time
Nug12	578	530	40”
Nug14	1014	969	1’ 8”
Nug15	1150	1071	3’ 5”
Nug16a	1610	1528	6’ 5”
Nug16b	1240	1139	6’ 5”
Nug17	1732	1622	12’ 5”
Nug18	1930	1802	21’ 1”
Nug20	2570	2386	1 h 7’
Nug21	2438	2253	1 h 45’
Nug22	3596	3396	2 h 41’
Nug24	3488	3235	6 h
Nug25	3744	3454	8 h 48’
Nug30	6124	5695	39 h

order to solve bigger instances $n \geq 20$ we have used the NEOS Server for Optimization.

In Table 4.2 we give results of some numerical experiments. The first column lists some Nugent instances from QAPLIB [14]. The number in the name of the problem refers to the size of the problem. The second column contains the value of the optimal solution. In the third column we provide the solutions of the relaxation QAP_{R_2}, obtained by use of NEOS Server for Optimization. These results were obtained in collaboration with Henry Wolkowicz[1] in 2001. The fourth column contains the running times required for one single interior point iteration of the algorithm. The results show that QAP_{R_2} is a strong relaxation but the interior point method is not appropriate for solving this relaxation. Because of the system size, the performance of the interior point method requests a serious computational work that includes a huge memory and long running time. The machine that was used at NEOS is Sun E6500 server with 24 processors and 24 GB memory. All processors are 400MHz Sparc2. The Sun server was used because of the 64-bit capability. Nug30 was solved with CSDP solver and the algorithm needed 36 iterations. Thus, the solution was obtained after 1404 hours (!).

4.4 Strictly Feasible Points

Our interior point implementation needs strictly feasible points to start. Here we find strictly feasible points for our relaxations.

Lemma 4.5 *[97] Define the* $((n-1)^2+1) \times ((n-1)^2+1)$ *matrix*

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

Then $\hat{R}$ *is positive definite and feasible for* QAP_{R_1}, QAP_{R_2} *and* QAP_{R_3}.

Proof: First, note that $\hat{R}$ is positive definite, since

$$x^T E_{n-1} x \leq \|E_{n-1}\| \|x\|^2 = (n-1)\|x\|^2 < n\|x\|^2$$

[1] personal communication

implies that $nI_{n-1} - E_{n-1}$ is positive definite.

We complete the proof by showing that

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

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

For simplicity, let us denote $I := I_{n-1}, E := E_{n-1}$.

$$\begin{aligned}
\hat{V}\hat{R}\hat{V}^T &= \left(\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}(e \otimes e) & V \otimes V \end{array}\right) \cdot \\
&\quad \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}(nI - E) \otimes (nI - E) \end{array}\right) \left(\begin{array}{c|c} 1 & \frac{1}{n}(e^T \otimes e^T) \\ \hline 0 & V^T \otimes V^T \end{array}\right) \\
&= \left(\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}(e \otimes e) & V \otimes V \end{array}\right) \left(\begin{array}{c|c} 1 & \frac{1}{n}(e^T \otimes e^T) \\ \hline 0 & \frac{1}{n^2(n-1)}(nI - E) \otimes (nI - E)(V^T \otimes V^T) \end{array}\right) \\
&= \left(\begin{array}{c|c} 1 & \frac{1}{n}(e^T \otimes e^T) \\ \hline \frac{1}{n}(e \otimes e) & \frac{1}{n^2}E \otimes E + \frac{1}{n^2(n-1)}(nVV^T - VEV^T) \otimes (nVV^T - VEV^T) \end{array}\right).
\end{aligned}$$

Now it remains to show that $nVV^T - VE_{n-1}V^T = nI_n - E_n$. We have

$$\begin{aligned}
nVV^T - VE_{n-1}V^T &= n \left(\begin{array}{c|c} I_{n-1} & -e_{n-1} \\ \hline -e_{n-1}^T & (n-1) \end{array}\right) - \left(\begin{array}{c|c} E_{n-1} & -(n-1)e_{n-1} \\ \hline -(n-1)e_{n-1}^T & (n-1)^2 \end{array}\right) \\
&= \left(\begin{array}{c|c} nI_{n-1} - E_{n-1} & -e_{n-1} \\ \hline -e_{n-1}^T & n-1 \end{array}\right) \\
&= nI_n - E_n.
\end{aligned}$$

■

We now provide a strictly dual-feasible point.

Lemma 4.6 *Let* $\hat{w} \in \mathbb{R}^{n^2+1}$, $\hat{w} = (M, 0, \ldots, 0)^T$, *and*

$$\hat{S} = M \left[\begin{array}{c|c} n-1 & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right].$$

Then for $M \in \mathbb{R}$ *large enough,* $(\hat{w}, \hat{S})$ *is strictly feasible for* DQAP_{R_2}.

PROOF: (See also [97].) Since

$$\hat{V}^T(G_{\tilde{f}}(\hat{S}) + \mathrm{Arrow}(\hat{w}))\hat{V} \;=\; M\hat{V}^T \left(\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right) \hat{V},$$

it is sufficient to show that

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

is positive definite.

$$\begin{aligned} \hat{V}^T \left(\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array}\right) \hat{V} &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & (V \otimes V)^T (I_n \otimes (I_n - E_n))(V \otimes V) \end{array}\right) \\ &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & (V^T I_n V) \otimes (V^T (I_n - E_n) V) \end{array}\right) \\ &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & V^T V \otimes V^T V \end{array}\right) \\ &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & (I_{n-1} + E_{n-1}) \otimes (I_{n-1} + E_{n-1}) \end{array}\right). \end{aligned}$$

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. ■

Chapter 5

SDP Relaxations of QAP in $\mathcal{S}_{(n-1)^2+1}$

In this Chapter we use a new representation of a permutation matrix in the lifted space, which allows to exploit sparsity for the SDP relaxation in a simple way. The lifted space is smaller than for the standard SDP relaxations for QAP. We present several SDP relaxations in $\mathcal{S}_{(n-1)^2+1}$, and solve them by use of the NEOS Server for Optimization.

5.1 A New Representation of the Permutation Matrix

The following lemma is well known, see [34, Lemma 3.1]

Lemma 5.1 *Let X be an $n \times n$ matrix with $X \in \mathcal{O} \cap \mathcal{E}$. Then there is an $(n-1) \times (n-1)$ orthogonal matrix $\hat{X}$ such that $X = V\hat{X}V^T + \frac{1}{n}E$, where V is an $n \times (n-1)$ matrix whose columns are an orthonormal basis for the nullspace of e^T. Conversely, if $\hat{X}$ is an $(n-1) \times (n-1)$ orthogonal matrix, then $X = \frac{1}{n}E + V\hat{X}V \in \mathcal{O} \cap \mathcal{E}$.*

In this section we give a new representation of the permutation matrix. First, we prove the following lemma.

Lemma 5.2 *Let*

$$V = \left(\begin{array}{c} I_{n-1} \\ \hline -e_{n-1}^T \end{array} \right) \tag{5.1}$$

be $n \times (n-1)$ *matrix,* P *be an arbitrary matrix of order* $(n-1) \times (n-1)$*, and consider*

$$\hat{P} = \frac{1}{n}(E_n - VE_{n-1}V^T) + VPV^T.$$

Then,
(i)

$$\hat{P} = \left(\begin{array}{c|c} P & e_{n-1} - Pe_{n-1} \\ \hline e_{n-1}^T - e_{n-1}^T P & e_{n-1}^T Pe_{n-1} - (n-2) \end{array} \right).$$

(ii) $\hat{P}e_n = \hat{P}^T e_n = e_n$.
(iii) $\hat{P}$ *is a permutation matrix if and only if* P *is a 0–1 matrix such that the sum of all elements in each row is* ≤ 1*; the sum of all elements in each column is* ≤ 1*, and the sum of all elements* $\geq n-2$.

Proof. (i) From

$$\begin{aligned} \frac{1}{n}(E_n - VE_{n-1}V^T) &= \frac{1}{n}\left(E_n - \left(\begin{array}{c|c} E_{n-1} & -(n-1)e_{n-1} \\ \hline -(n-1)e_{n-1}^T & (n-1)^2 \end{array} \right) \right) \\ &= \frac{1}{n}\left(\begin{array}{c|c} 0 & ne_{n-1} \\ \hline ne_{n-1}^T & -n(n-2) \end{array} \right) \\ &= \left(\begin{array}{c|c} 0 & e_{n-1} \\ \hline e_{n-1}^T & -(n-2) \end{array} \right), \end{aligned} \tag{5.2}$$

and

$$VPV^T = \left(\begin{array}{c} I_{n-1} \\ \hline -e_{n-1}^T \end{array} \right) P \left(\begin{array}{c|c} I_{n-1} & -e_{n-1} \end{array} \right) = \left(\begin{array}{c|c} P & -Pe_{n-1} \\ \hline -e_{n-1}^T P & e_{n-1}^T Pe_{n-1} \end{array} \right),$$

follows the presentation (i) of $\hat{P}$.
(ii)

$$\hat{P}e_n = \left(\begin{array}{c} Pe_{n-1} + e_{n-1} - Pe_{n-1} \\ e_{n-1}^T e_{n-1} - e_{n-1}^T Pe_{n-1} + e_{n-1}^T Pe_{n-1} - (n-2) \end{array} \right) = \left(\begin{array}{c} e_{n-1} \\ 1 \end{array} \right) = e_n.$$

Similarly we prove $P^T e_n = e_n$.
(iii) Let P be a matrix such that $P_{ij} \in \{0, 1\},\ \forall i, j = 1, \ldots, n-1$,

$$\begin{aligned} Pe &\leq 1, \\ e^T P &\leq 1, \end{aligned}$$

and

$$e^T P e \geq n - 2.$$

Then

$$VPV^T = \left(\frac{I_{n-1}}{-e_{n-1}^T}\right) P \left(\frac{I_{n-1}}{-e_{n-1}^T}\right)^T = \left(\begin{array}{c|c} P & -Pe \\ \hline -e^T P & e^T P e \end{array}\right).$$

We define

$$c := -Pe, \quad r := -e^T P \quad \text{and} \quad s := e^T P e.$$

By the initial assumption the elements of the vectors c and r are 0 or -1, and $s \in \{n-2, n-1\}$. If $s = n-1$, then P must be a permutation matrix of order $n-1$, $e - Pe = 0$, $e^T - e^T P = 0$ and $\hat{P} = s - (n-2) = 1$, hence $\hat{P}$ is a permutation matrix. If $s = n-2$, then exactly one of the row sums r, and one of the column sums c, is zero and the corresponding elements $\hat{P}_{n,i} = 1$, $\hat{P}_{j,n} = 1$, $\hat{P}_{n,n} = s - (n-2) = 0$, so $\hat{P}$ is again a permutation matrix. ∎

Following Corollary gives the presentation of the permutation matrix.

Corollary 5.1 *For $X \in \Pi$, and V defined as in Lemma 5.2 it follows that*

$$X = \frac{1}{n}(E_n - VE_{n-1}V^T) + VX(1 : n-1, 1 : n-1)V^T.$$

Proof. Directly from Lemma 5.2. ∎

Remark 5.1 *The matrix V defined in (5.1), is $n \times (n-1)$ matrix containing a basis of the orthogonal complement of the vector of all ones, i.e. $V^T e = 0$. For another choice of the matrix V the presentation of the permutation matrix can be analogously derived.*

5.2 About the Structure of the New Parametrization

In Section 4, Lemma 4.3 we prove that any $Y \in \mathcal{P}$, where $\mathcal{P}$ is the feasible set of QAP, is of the form $Y = \hat{V}R\hat{V}^T$ for $\hat{V}$ defined in (4.9), and $R \in \mathcal{S}_{(n-1)^2+1}$, $R \succeq 0,\ R_{00} = 1$. In this Section we show that for any matrix $\bar{Y} = \bar{V}\bar{R}\bar{V}^T \in \mathcal{P}$, where $\bar{V}$ is defined below, the corresponding matrix $\bar{R} \in \mathcal{S}_{(n-1)^2+1}$ has the structure like the matrix $\bar{Y}$. More precisely, the matrix $\bar{R}$ has zeros on the off-diagonal elements of the diagonal blocks and on the diagonal elements of the diagonal blocks, and $\text{arrow}(\bar{R}) = e_0$.

Lemma 5.3 *Let $\mathcal{P}$ be the feasible set of QAP defined in (4.6), and*

$$\bar{V} := \begin{pmatrix} e_0^T \\ \hline W \end{pmatrix} \in \mathcal{M}_{(n^2+1)\times((n-1)^2+1)}, \tag{5.3}$$

where

$$W := \left(\frac{1}{n}(e_{n^2} - (V \otimes V)e_{(n-1)^2}), V \otimes V\right), \tag{5.4}$$

and V defined in (4.7).

For any $Y \in \mathcal{P}$ there exists a symmetric matrix R of order $(n-1)^2+1$, indexed from 0 to $(n-1)^2$, such that

$$R \succeq 0,\ \text{arrow}(R) = e_0,\ \textit{and}\ Y = \bar{V}R\bar{V}^T.$$

If Y is extreme point of $\mathcal{P}$ then $R_{ij} \in \{0,1\}$, otherwise $R_{ij} \in [0,1]$ for $i,j \in \{0,\ldots,(n-1)^2\}$.

Proof. First we look at the extreme points of $\mathcal{P}$. Let $Y \in \mathcal{P}$ be one of them. Thus

$$Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix},$$

for some $x = \text{vec}(X)$, $X \in \Pi$. From Corollary 5.1 follows that $X = \frac{1}{n}(E_n - VE_{n-1}V^T) + VPV^T$ where $P = X(1:n-1, 1:n-1)$. With the use of (1.5), we get

$$x = \text{vec}(X) = \frac{1}{n}(e_{n^2} - (V \otimes V)e_{(n-1)^2}) + (V \otimes V)p = Wz,$$

where $p = \mathrm{vec}(P)$, $z = \begin{pmatrix} 1 \\ p \end{pmatrix}$, and W is defined in (5.4). Now

$$Y = \begin{pmatrix} 1 & (Wz)^T \\ Wz & Wzz^TW^T \end{pmatrix} = \begin{pmatrix} e_0 \\ \overline{W} \end{pmatrix} zz^T \begin{pmatrix} e_0 \\ \overline{W} \end{pmatrix}^T = \bar{V}R\bar{V}^T,$$

with $R = zz^T$. Hence

$$R = \begin{pmatrix} 1 & p^T \\ p & pp^T \end{pmatrix} \succeq 0, \tag{5.5}$$

and $R_{00} = 1$. Since p is 0–1 vector it follows that $\mathrm{diag}(pp^T) = p$, $R_{ij} \in \{0,1\}$, $\forall i,j \in \{0,\ldots,(n-1)^2\}$, and $\mathrm{arrow}(R) = e_0$. The same holds for any $Y \in \mathcal{P}$ which is the convex combination formed from the several extreme points of $\mathcal{P}$, but $R_{ij} \in [0,1]$. ■

The following statements are direct consequences of the Lemma 5.3.

Corollary 5.2 *For any extreme point $Y \in \mathcal{P}$ exist a unique matrix $R \in \mathcal{S}_{(n-1)^2+1}$ with the properties from Lemma 5.3.*

■

We index now the elements of the matrix $Y \in \mathcal{S}_{(n-1)^2+1}$ by $y_{r,s} = Y_{(i,j)(k,l)}$ for $r,s \in \{0,1,\ldots,(n-1)^2\}$, $i,j,k,l \in \{0,1,\ldots,n-1\}$, and define the set

$$S := \{(r,s) :\ r = (i,j),\ s = (i,k),\ \text{ or }\ r = (j,i),\ s = (k,i),\ j \neq k\}. \tag{5.6}$$

Corollary 5.3 *Let R be a matrix defined as in (5.5), S be a set of indices (5.6) and G_S be the operator on $\mathcal{S}_{(n-1)^2+1}$ defined as in (1.7). Then*

$$G_S(R) = 0.$$

■

Corollary 5.4 *Let*

$$R = \sum_{i=1}^{m} \lambda_i \begin{pmatrix} 1 & p_i^T \\ p_i & p_i p_i^T \end{pmatrix},$$

where $p_i = \mathrm{vec}(P_i)$, $P_i = X_i(1:n-1, 1:n-1)$, $X_i \in \Pi$, and $\lambda_i \geq 0$, $\forall i = 1,\ldots,m$, $\sum_{i=1}^m \lambda_i = 1$. Then $G_S(R) = 0$.

PROOF. Each matrix

$$R_i = \begin{pmatrix} 1 & p_i^T \\ p_i & p_i p_i^T \end{pmatrix},$$

where p_i is defined as in the Corollary, satisfies $G_S(R_i) = 0$. Since the operator G_S is a linear operator follows the statement. ■

5.2.1 Elementwise Matrix Description

In this Subsection we give the componentwise description of the matrix $Y = \bar{V}R\bar{V}^T \in \mathcal{S}_{n^2+1}$, where $R \in \mathcal{S}_{(n-1)^2+1}$, $\text{arrow}(R) = e_0$. Results from this Subsection we need for deriving the relaxations in Section 5.4.

We block the matrix R in the following way

$$R = \left(\begin{array}{c|ccc} 1 & R^{01} & \dots & R^{0(n-1)} \\ \hline R^{10} & R^{11} & \dots & R^{1(n-1)} \\ \dots & \dots & \dots & \dots \\ R^{(n-1)0} & R^{(n-1)1} & \dots & R^{(n-1)(n-1)} \end{array}\right), \tag{5.7}$$

where $R^{k0} = (R^{0k})^T \in I\!R^{n-1}$ and $R^{mk} \in \mathcal{M}_{n-1}$. The matrix $\bar{V}$ is

$$\bar{V} = \left(\begin{array}{c|c|c|c|c} 1_1 & 0_{1\times n-1} & 0_{1\times n-1} & 0 & \dots \\ \hline 0_{n-1\times 1} & I_{n-1} & 0_{n-1} & 0 & \dots \\ 1_1 & -e_{n-1}^T & 0_{1\times n-1} & 0 & \dots \\ \hline 0_{n-1\times 1} & 0_{n-1} & I_{n-1} & 0 & \dots \\ 1_1 & 0 & -e_{n-1}^T & 0 & \dots \\ \hline \dots & \dots & \dots & & \\ \hline e_{n-1} & -I_{n-1} & -I_{n-1} & \dots & \\ \hline -(n-2) & e_{n-1}^T & e_{n-1}^T & \dots & \end{array}\right). \tag{5.8}$$

Since some of the blocks in the matrix $\bar{V}$ contain identity matrices, the corresponding blocks of the matrix R, with the matrix multiplication $\bar{V}R\bar{V}^T$, are shifted in to the space $\mathcal{S}_{n^2+1}$.

Let $Y = \bar{V}R\bar{V}^T$. We now expand the expression for Y;

$$Y = \left(\begin{array}{c|c|c|c|c|c} Y^{00}_{1\times 1} & Y^{01}_{1\times n-1} & Y^{01}_{1\times 1} & \cdots & Y^{0n}_{1\times n-1} & Y^{0n}_{1\times 1} \\ \hline Y^{10}_{n-1\times 1} & Y^{11}_{n-1\times n-1} & Y^{11}_{n-1\times 1} & \cdots & Y^{1n}_{n-1\times n-1} & Y^{1n}_{n-1\times 1} \\ Y^{10}_{1\times 1} & Y^{11}_{1\times n-1} & Y^{11}_{1\times 1} & \cdots & Y^{1n}_{1\times n-1} & Y^{1n}_{1\times 1} \\ \hline Y^{20}_{n-1\times 1} & Y^{21}_{n-1\times n-1} & Y^{21}_{n-1\times 1} & \cdots & \cdots & \cdots \\ Y^{20}_{1\times 1} & Y^{21}_{1\times n-1} & Y^{21}_{1\times 1} & \cdots & \cdots & \cdots \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline Y^{(n-1)0}_{n-1\times 1} & Y^{(n-1)1}_{n-1\times n-1} & Y^{(n-1)1}_{n-1\times 1} & \cdots & \cdots & \cdots \\ Y^{(n-1)0}_{1\times 1} & Y^{(n-1)1}_{1\times n-1} & Y^{(n-1)1}_{1\times 1} & \cdots & \cdots & \cdots \\ \hline Y^{n0}_{n-1\times 1} & Y^{n1}_{n-1\times n-1} & Y^{n1}_{n-1\times 1} & \cdots & Y^{nn}_{n-1\times n-1} & Y^{nn}_{n-1\times 1} \\ Y^{n0}_{1\times 1} & Y^{n1}_{1\times n-1} & Y^{n1}_{1\times 1} & \cdots & Y^{nn}_{1\times n-1} & Y^{nn}_{1\times 1} \end{array}\right).$$

Here the block sizes correspond to the multiplication $\bar{V}R\bar{V}^T$. We use superscripts to denote the block structure and subscripts to denote the block sizes. By abuse of notation, we have the same superscript for two different adjacent blocks that have different sizes.

• *The first column in Y*
From the matrix multiplication $\bar{V}R\bar{V}^T$ follows that

$$\begin{pmatrix} Y^{k0}_{n-1\times 1} \\ Y^{k0}_{1\times 1} \end{pmatrix} = \begin{pmatrix} R^{k0} \\ 1 - e^T R^{k0} \end{pmatrix},$$

for $k = 1, \ldots, n-1$, and

$$Y^{n0}_{n-1\times 1} = \begin{pmatrix} 1 - \sum_{i=1}^{n-1} R^{i0}_1 \\ \vdots \\ 1 - \sum_{i=1}^{n-1} R^{i0}_{n-1} \end{pmatrix}.$$

The last element in the first column of the matrix Y is

$$Y^{n0}_{1\times 1} = -(n-2) + \sum_{i=1}^{n-1} e^T R^{i0}. \tag{5.9}$$

• *Diagonal blocks in Y*
For $k = 1, \ldots, n-1$, from $\bar{V}R\bar{V}^T$ it follows that

$$Y^{kk}_{n-1\times n-1} = R^{kk},$$

$$Y^{kk}_{n-1\times 1} = \begin{pmatrix} R^{k0}_1 - \sum_{i=1}^{n-1} R^{kk}_{1,i} \\ \vdots \\ R^{k0}_{n-1} - \sum_{i=1}^{n-1} R^{kk}_{n-1,i} \end{pmatrix},$$

and

$$Y^{kk}_{1\times 1} = 1 - 2e^T R^{k0} + e^T R^{kk} e. \tag{5.10}$$

Due to the structure of $\bar{V}$, the last diagonal block in the matrix Y differs from the structure of the other diagonal blocks. From $\bar{V}R\bar{V}^T$ it follows that

$$Y^{nn}_{n-1\times n-1} = e(1 - \sum_{i=1}^{n-1} R^{0i}_1, \ldots, 1 - \sum_{i=1}^{n-1} R^{0i}_{n-1}) - \sum_{i=1}^{n-1}(R^{i0}e^T - \sum_{j=1}^{n-1} R^{ij}). \tag{5.11}$$

Hence, the kth diagonal element of the matrix $Y^{nn}_{n-1\times n-1}$ is

$$(Y^{nn}_{n-1\times n-1})_{k,k} = 1 - 2\sum_{i=1}^{n-1} R^{0i}_k + (\sum_{i=1}^{n-1} R^{i0}R^{0i})_{k,k}.$$

For $k = 1, \ldots, n-1$, it follows that

$$(Y^{nn}_{n-1\times 1})_k = -(n-2) + (n-2)\sum_{i=1}^{n-1} R^{0i}_k + e^T \sum_{i=1}^{n-1} R^{0i} - e^T(\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} R^{ij})(:,k),$$

where $\sum_{ij} R^{ij}(:,k)$ means that we from the sum $\sum_{ij} R^{ij}$ pull out the elements from the kth column.

The last element in the last diagonal block in the matrix Y is

$$Y^{nn}_{1\times 1} = (n-2)^2 - (2n-5)\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} R^{0i}_j \tag{5.12}$$

• *Off–diagonal blocks*
For $m \neq k,\ m, k = 1, \ldots, n-1$, it follows that

$$Y^{mk}_{n-1\times n-1} = R^{mk},$$

$$Y^{mk}_{n-1\times 1} = \begin{pmatrix} R^{m0}_1 - \sum_{i=1}^{n-1} R^{mk}_{1,i} \\ \vdots \\ R^{m0}_{n-1} - \sum_{i=1}^{n-1} R^{mk}_{n-1,i} \end{pmatrix},$$

and

$$Y_{1\times 1}^{mk} = 1 - e^T R^{k0} - e^T R^{m0} + e^T R^{mk} e. \tag{5.13}$$

Finally, we look at the off–diagonal blocks

$$\begin{pmatrix} Y_{n-1\times n-1}^{kn} & Y_{n-1\times 1}^{kn} \\ Y_{1\times n-1}^{kn} & Y_{1\times 1}^{kn} \end{pmatrix},$$

for $k = 1, \ldots, n-1$. From $\bar{V} R \bar{V}^T$ we get

$$Y_{n-1\times n-1}^{kn} = R^{k0} e^T - \sum_{i=1}^{n-1} R^{ki}.$$

For the vector $Y_{n-1\times 1}^{kn}$ it follows that

$$(Y_{n-1\times 1}^{kn})_j = -(n-2) R_j^{k0} + \sum_{l=1}^{n-1} \sum_{i=1}^{n-1} R_{jl}^{ki},$$

where $j = 1, \ldots, n-1$.

Finally,

$$Y_{1\times 1}^{kn} = -(n-2) + \sum_{i=1}^{n-1} e^T R^{i0} + (n-2) e^T R^{k0} - \sum_{j=1}^{n-1} \sum_{i=1}^{n-1} R_j^{ki}.$$

5.3 How do $\hat{V}$ and $\bar{V}$ Interrelate?

In [97] is shown that the feasible set for the basic relaxation QAP_{R_1} is

$$\hat{\mathcal{P}} = \{Y \in \mathcal{S}_{n^2+1} : \exists R \ \text{ s.t } R \succeq 0,\ Y = \hat{V} R \hat{V}^T,\ \mathrm{arrow}(Y) = e_0\}.$$

where $\hat{V}$ is an $(n^2+1) \times ((n-1)^2+1)$ matrix,

$$\hat{V} = \left(\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n} e_{n^2} & V \otimes V \end{array} \right),$$

and V is an $n \times (n-1)$ matrix defined in (5.1). Note that the matrix $\bar{V}$ defined in (5.3);

$$\bar{V} = \left(\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n} e_{n^2} - \frac{1}{n} (V \otimes V) e_{(n-1)^2} & V \otimes V \end{array} \right),$$

is obtained by subtracting the sum of all the columns except the first from the first column of the matrix $\hat{V}$. This subtraction does not change the range space. Hence the matrix $\bar{V}$ is obtained from the matrix $\hat{V}$ by the following matrix multiplication

$$\bar{V} := \hat{V}M,$$

where the nonsingular $((n-1)^2+1) \times ((n-1)^2+1)$ matrix M is defined by

$$M := \left(\begin{array}{c|c} 1 & 0 \\ \hline -\frac{1}{n}e & I \end{array}\right), \tag{5.14}$$

i.e. we subtract ($\frac{1}{n}$ times) all the columns from the first column but leave the other columns unchanged. Now we define the set

$$\begin{aligned}\bar{\mathcal{P}} := \{Y \in \mathcal{S}_{n^2+1} : \exists R \text{ s.t} \quad & R \succeq 0,\ R_{ij} \in [0,1],\ G_S(R) = 0, \\ & \text{arrow}(R_{00}) = 1,\ Y = \bar{V}R\bar{V}^T,\ \text{arrow}(Y) = e_0\}.\end{aligned}$$

Note that $\mathcal{P} \subset \bar{\mathcal{P}}$, and the set $\bar{\mathcal{P}}$ is equal to the set $\hat{\mathcal{P}}$. In Section 4.4 is given a feasible point for the basic relaxation which corresponds to the matrix $\hat{V}$. Now we derive the feasible point for the basic relaxation which corresponds to the matrix $\bar{V}$.

Lemma 5.4 *The feasible point for the basic relaxation* QAP_{R_1} *corresponding to the matrix* $\bar{V}$ *is*

$$\bar{R} = \left(\begin{array}{c|c} 1 & \frac{1}{n}e^T \\ \hline \frac{1}{n}e & \frac{1}{n^2}E_{(n-1)^2} + \frac{1}{n^2(n-1)}(nI_{n-1} - E_{n-1}) \otimes (nI_{n-1} - E_{n-1}) \end{array}\right).$$

Proof. The feasible point for the basic relaxation QAP_{R_1} with respect to the set $\bar{\mathcal{P}}$ is (see [97])

$$\bar{R} = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}(nI_{n-1} - E_{n-1}) \otimes (nI_{n-1} - E_{n-1}) \end{array}\right).$$

Since $\bar{V} = \hat{V}M$ for M defined in (5.14) follows that

$$\bar{R} = M^{-1}\hat{R}(M^{-1})^T.$$

By recognizing

$$M^{-1} = \left(\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n}e & I \end{array}\right),$$

the expression for $\bar{R}$ follows directly. ∎

5.4 Deriving the Relaxations

In this Section we derive several SDP relaxations of QAP in $\mathcal{S}_{(n-1)^2+1}$, and compare them with the relaxations of QAP in $\mathcal{S}_{n^2+1}$. The presented relaxations are considerable weaker than the relaxations given in the Chapter 4, but they motivate us to combine constraints on the variables from both spaces. At the end of this Section we arrive to such model.

In order to derive the tractable relaxations for QAP in $\mathcal{S}_{(n-1)^2+1}$ we need to approximate the set

$$\mathcal{P}_S := \text{conv}\left\{ \begin{pmatrix} 1 \\ x_S \end{pmatrix} \begin{pmatrix} 1 \\ x_S \end{pmatrix}^T : x_S = \text{vec}(X(1:n-1, 1:n-1)),\ X \in \Pi \right\},$$

by larger sets containing $\mathcal{P}_S$. We define the set:

$$\mathcal{R}_S := \{R \succeq 0 : R \in \mathcal{S}_{(n-1)^2+1},\ \text{arrow}(R) = e_0\}, \tag{5.15}$$

where $e_0 \in I\!R^{(n-1)^2+1}$ is a unit vector. Note that $\mathcal{P}_S \subset \mathcal{R}_S$. The *ground SDP relaxation* of QAP is

$$(\text{QAP}_{Rs_1}) \quad \min\{\text{tr } LR :\ R \in \mathcal{R}_S\},$$

where L is (see (4.12))

$$L = \bar{V} \begin{pmatrix} 0 & 0 \\ 0 & B \otimes A + \text{Diag}(c) \end{pmatrix} \bar{V}^T.$$

The ground relaxation is very weak relaxation, see Table 5.1.

Remark 5.2 *In this Section we block the matrix $R \in \mathcal{P}_S$ as in (5.7).*

Now we strengthen the ground relaxation by adding some constraints.

If R is a solution of the ground relaxation, than the matrix $\bar{V}R\bar{V}^T$ does not fulfill the arrow constraint. The last diagonal elements in the first $n-1$ diagonal blocks and all elements on the diagonal of the last diagonal block are not equal to the corresponding elements in the first row (resp. column) of the matrix $\bar{V}R\bar{V}^T$. More precisely, the indices of the elements $(\bar{V}R\bar{V}^T)_{(i,j),(i,j)}$

on the diagonal of the matrix $\bar{V}R\bar{V}^T$ that differ from the elements from the first row (resp. column) are

$$(i,j) = \begin{cases} i = 1,\ldots,n-1 \text{ and } j = n \\ i = n \text{ and } j = 1,\ldots,n. \end{cases}$$

For $k = 1,\ldots,n-1$, we introduce the operator $\mathrm{Lm}_k : \mathcal{S}_{(n-1)^2+1} \to I\!R$,

$$\mathrm{Lm}_k(R) := \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} R_{(k,i),(k,j)} - \sum_{i=1}^{n-1} R_{0,(k,i)}, \tag{5.16}$$

where we use index 0 for the first row and column of the matrix. Now we define the linear operator $\mathrm{Lm} : \mathcal{S}_{(n-1)^2+1} \to I\!R^{n-1}$,

$$\mathrm{Lm}(\cdot) := (\mathrm{Lm}_1(\cdot),\ldots,\mathrm{Lm}_{n-1}(\cdot))^T.$$

We demand

$$\mathrm{Lm}(R) = 0.$$

This operator sets

$$(\bar{V}R\bar{V}^T)_{(i,n),(i,n)} = (\bar{V}R\bar{V}^T)_{0,(i,n)}, \quad i = 1,\ldots,n-1.$$

Note that the operator Lm does not act on the elements in the last diagonal block and corresponding elements in the first row (resp. column) of the matrix $\bar{V}R\bar{V}^T$. For $t \in R^{n-1}$ the adjoint operator of the operator Lm is

$$\mathrm{Lm}^T(t) = \frac{1}{2}\begin{pmatrix} 0 & -t_1e_{n-1}^T & -t_2e_{n-1}^T & \ldots & -t_{n-1}e_{n-1}^T \\ -t_1e_{n-1} & 2t_1E_{n-1} & 0 & \ldots & 0 \\ -t_2e_{n-1} & 0 & 2t_2E_{n-1} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ -t_{n-1}e_{n-1} & 0 & 0 & \ldots & 2t_{n-1}E_{n-1} \end{pmatrix}.$$

We now define the SDP relaxation

$$(\mathrm{QAP}_{Rs_2}) \quad \min\{\mathrm{tr}\ LR :\ R \in \mathcal{R}_S,\ \mathrm{Lm}(R) = 0\}.$$

The relaxation QAP_{Rs_2} contains $(n-1)^2 + n$ constraints. In Table 5.1 we present solutions of some Nugent instances computed for the basic relaxation QAP_{R_1} (see Section 4.1), the ground relaxation QAP_{Rs_1} and the relaxation

Table 5.1: Optimal solutions of the basic, ground and QAP_{Rs_2} relaxation

	QAP_{R_1}	QAP_{Rs_1}	QAP_{Rs_2}
Nug7	43	-869	28
Nug8	-15	-3840	-131
Nug12	-216	-25562	-755
Nug15	-823	-145570	-4191

QAP_{Rs_2}, respectively. Note that the relaxation QAP_{Rs_2} is stronger then the relaxation QAP_{Rs_1}, but still weak in comparison with the basic relaxation.

Now, motivated with the structure of the matrix in the set $\mathcal{P}_S$ we add some more constraints. From Corollary 5.4 follows that each matrix from $\mathcal{P}_S$ has a specific zero pattern, i.e. the off–diagonal elements of the diagonal blocks and the diagonal elements of the off–diagonal blocks are zero. The zero pattern is covered by the operator G_S defined on $\mathcal{S}_{(n-1)^2+1}$ (see 1.7), where the set of indices S is defined in (5.6). Now, we define the set:

$$\mathcal{G}_S := \{R : R \in \mathcal{S}_{(n-1)^2+1},\ G_S(R) = 0\},$$

and strengthen the relaxation QAP_{Rs_1} by adding this new set of equalities and arrive at the tighter model

$$(\mathrm{QAP}_{Rs_3}) \quad \min\{\mathrm{tr}\ LR :\ R \in \mathcal{R}_S \cap \mathcal{G}_S\}.$$

This model we call the *incomplete small gangster model.*

Remark 5.3 *Constraint* $\mathrm{Lm}(R) = 0$ *in connection with* $G_S(0) = 0$ *is a redundant constraint.*

The incomplete small gangster model could be considerably tighten by adding *one more* constraint. The solution matrix $R \in \mathcal{S}_{(n-1)^2+1}$ of the relaxation QAP_{Rs_3} lifted by the matrix multiplication $\bar{V}R\bar{V}^T$ into the space $\mathcal{S}_{n^2+1}$, does not fulfill the condition

$$(\bar{V}R\bar{V}^T)_{0n^2} = (\bar{V}R\bar{V}^T)_{n^2n^2}. \tag{5.17}$$

From (5.12) it follows that,

$$Y_{n^2n^2} = (n-2)^2 - (n-2)\sum_{i=1}^{(n-1)^2}(R_{0,i} + R_{i,0}) + \sum_{i=1}^{(n-1)^2}\sum_{j=1}^{(n-1)^2} R_{ij}, \tag{5.18}$$

Table 5.2: Optimal solutions of the relaxations QAP_{Rs_3} and QAP_{Rs_4} obtained using NEOS

	OPT	QAP_{Rs_3}	QAP_{Rs_4}
Nug12	578	350	510
Nug14	1014	852	942
Nug15	1150	940	1006
Nug16a	1610	1122	1480
Nug16b	1240	768	1105
Nug17	1732	1126	1582
Nug18	1930	1547	1758
Nug20	2570	1524	2322
Nug21	2438	1570	2178
Nug22	3596	1961	3273
Nug24	3488	1846	3125

and from (5.9) it follows that

$$Y_{0,n^2} = -(n-2) + \sum_{i=1}^{n-1} R_{0,i}. \tag{5.19}$$

From (5.18) and (5.19), we derive the linear operator $\mathrm{Le} : \mathcal{S}_{(n-1)^2+1} \to I\!R$,

$$\mathrm{Le}(R) := (\frac{3}{2} - n) \sum_{i=1}^{(n-1)^2} R_{0,i} + \sum_{i=1}^{(n-1)^2} \sum_{j=1}^{(n-1)^2} R_{i,j}. \tag{5.20}$$

The condition (5.17) is satisfied if

$$\mathrm{Le}(R) = -(n-1)(n-2).$$

The adjoint operator of the operator Le is

$$\mathrm{Le}^T(t) = \frac{t}{2} \begin{pmatrix} 0 & e^T \\ e & 2E \end{pmatrix}, \tag{5.21}$$

for $t \in I\!R$. Let us define the set

$$\mathcal{L} := \{R : R \in \mathcal{S}_{(n-1)^2+1},\ \mathrm{Le}(R) = -(n-1)(n-2)\}. \tag{5.22}$$

We now derive the relaxation

$$(\mathrm{QAP}_{Rs_4}) \quad \min\{\mathrm{tr}\ LR :\ R \in \mathcal{R}_S \cap \mathcal{G}_S \cap \mathcal{L}\}.$$

This model we call the *small gangster model*. If $R \in \mathcal{S}_{(n-1)^2+1}$ is a solution of the relaxation QAP_{Rs_4} then

$$\mathrm{arrow}(\bar{V}R\bar{V}^T) = e_0 \in I\!R^{n^2+1}.$$

In Table 5.2 we present results of some numerical experiments. The first column lists some of the Nugent instances from QAPLIB [14]. The number in the name of the problem refers to the size of the problem. The second column contains the value of the optimal solution of the corresponding problem. The rest of the columns list the solutions of the relaxation QAP_{Rs_3} and QAP_{Rs_4}, obtained by use of NEOS Server for Optimization. The results show that the relaxation QAP_{Rs_4} is much stronger relaxation then QAP_{Rs_3}. Note that the relaxation QAP_{Rs_4} contains only *one more* constraint then QAP_{Rs_3} relaxation. In the Section 7.3 we report numerical results for the QAP_{Rs_4} relaxation computed by the Bundle Method.

Remark 5.4 *The relaxation* QAP_{Rs_4} *can be further tightened by adding non-negativity constraints*

All relaxations from this Section are considerably weaker than the relaxations derived in Chapter 4, but they lead us to the model that collects constraints from the both matrix spaces, i.e. from $\mathcal{S}_{(n-1)^2+1}$ and $\mathcal{S}_{n^2+1}$. Finally, we arrive to the following relaxation

$$(\mathrm{QAP}_{Ss}) \qquad \begin{array}{ll} \min & \mathrm{tr}\ LR \\ \text{s.t.} & G_S(R) = 0 \\ & G_{\hat{J}}(\bar{V}R\bar{V}^T) = 0 \\ & N(\bar{V}R\bar{V}^T) \geq 0 \\ & R \in \mathcal{R}_S \cup \mathcal{L}, \end{array}$$

where the set S is defined in (5.6) and $\hat{J}$ in (4.16), the operator $G_{\hat{J}}$ (respectively G_S) in (1.7), and the operator N in (4.14). This model contains $O(n^4)$ constraints and can not be solved straightforward by interior point method, but it seems to be good relaxation for applying the Bundle Method (see Chapter 7).

Chapter 6

The Bundle Method in Combinatorial Optimization

This chapter is based on joint work with I. Fischer, G. Gruber and F. Rendl [27].

6.1 Introduction

Many combinatorial optimization problems have a natural formulation in 0-1 variables, leading to either linear or quadratic problems in binary variables. To get tractable relaxations, one can use a (partial) description of the convex hull of integer solutions (polyhedral approach), or more recently, semidefinite programming. In fact, it is possible to combine the two approaches to get better approximations. This procedure may typically lead to a problem of the following type:

$$(P) \quad \min\{c^T x : \ x \in \mathcal{X}, \ \mathcal{A}x \geq b\}.$$

The set $\mathcal{X}$ is 'nice' in the sense that the problem

$$\min\{c^T x : \ x \in \mathcal{X}\}$$

would be much easier to solve than (P).

The linear constraints $\mathcal{A}x \geq b$ might make it difficult to solve (P) either because their number is (polynomial but) huge, or because they destroy the nice structure of $\mathcal{X}$. We call them the "hard constraints".

Barahona and Anbil [9] and Barahona and Ladanyi [10] have recently revived the idea of working with the Lagrangian dual to deal with $\mathcal{A}x \geq b$ only indirectly.

The Lagrangian dual functional is nonsmooth, so one needs to use the algorithmic machinery from nonsmooth optimization to deal with it.

In [49, 50] a simple version of the subgradient iteration scheme of Polyak is used to carry out the iterations.

6.2 Lagrangian Duality

Given the problem (P),

$$z^* = \min\{c^T x : \ x \in \mathcal{X}, \ \mathcal{A}x \geq b\} \tag{6.1}$$

we introduce the Lagrangian

$$\mathcal{L}(x, \gamma) := c^T x + \gamma^T (b - \mathcal{A}x) \tag{6.2}$$

and the dual functional

$$f(\gamma) := \min_{x \in \mathcal{X}} \ \mathcal{L}(x, \gamma) = b^T \gamma + \min_{x \in \mathcal{X}} (c - \mathcal{A}^T \gamma)^T x. \tag{6.3}$$

The following notation will be useful. We call a pair (γ, x) a *matching pair* for f, if $f(\gamma) = \mathcal{L}(x, \gamma)$. The Lagrangian dual now is

$$z^* = \max_{\gamma \geq 0} f(\gamma).$$

Here we make the implicit assumption that $\mathcal{X}$ is nice in the sense that

$$z^* = \min_{x \in \mathcal{X}} \max_{\gamma \geq 0} \mathcal{L}(x, \gamma) = \max_{\gamma \geq 0} f(\gamma)$$

holds. This will be the case for all the applications that will be given later. (Even if strong duality would not hold, we still get approximations of z^* from below, i.e. $z^* \geq f(\gamma)$ holds for all $\gamma \geq 0$.)

The problem now consists in finding $x^* \in \mathcal{X}$ and γ^* such that

$$|f(\gamma^*) - c^T x^*| \approx 0 \text{ and } \|\max\{0, b - \mathcal{A}x^*\}\| \approx 0.$$

Prosaically speaking, we would like to find x^* which is nearly feasible and nearly optimal for (P). The dual variables γ^* serve to estimate z^* from below.

6.3 The Bundle Method

The bundle method dates at least to the 1970's, see e.g. [93, 63, 43, 85, 98]. It was originally developed to minimize a nonsmooth convex function $f(\gamma)$ over $\gamma \in I\!R^n$. The function f is assumed to be given by an oracle, which, for some input γ returns the function value $f(\gamma)$ and a vector g contained in the subdifferential (see Definition B.1) of f at γ, i.e. $g \in \partial f(\gamma)$. Imposing the sign constraints $\gamma \geq 0$, as required in our situation, does not make the problem much harder. We will use the bundle method in a way adapted for our problem, hence it is useful to briefly recall its basic ideas and practical issues.

6.3.1 Bundle Method: the Basic Idea

If $f(\gamma)$ is given as as before, $f(\gamma) = b^T\gamma + \min_{x\in\mathcal{X}}(c - \mathcal{A}^T\gamma)^T x$, the bundle method is most conveniently explained as follows. For given $\gamma \geq 0$ we assume that the oracle defining f returns $f(\gamma)$ and x, such that (γ, x) is a matching pair. A subgradient $g \in \partial f(\gamma)$ is then implicitly given by $g = b - \mathcal{A}x$, as can easily be verified.

At the start, we guess some initial $\gamma_{start} \geq 0$, for instance $\gamma_{start} = 0$ and call the oracle defining f. Thus initially we have a matching pair $(\gamma_{start}, x_{start})$ with $f(\gamma_{start}) = \mathcal{L}(x_{start}, \gamma_{start})$.

The algorithm is iterative and maintains in each iteration a currently best approximation $\hat{\gamma}$ to the maximizer of f, and a sequence $X = (x_1, \ldots, x_k)$ where each $x_i \in \mathcal{X}$ and $(\hat{\gamma}, x_k)$ is a matching pair.

To describe a general step, we assume to have $X = (x_1, \ldots, x_k)$ and $\hat{\gamma}$, with $(\hat{\gamma}, x_k)$ a matching pair. Given X, we compute the following information:

$$g_i := b - \mathcal{A}x_i,\ G = (g_1, \ldots, g_k),\ \phi_i := c^T x_i \text{ and } F = (\phi_1, \ldots, \phi_k)^T. \quad (6.4)$$

The bundle method now combines two ideas to determine a new trial point γ_{test}.

(i) Approximate model of f.
The function $f(\gamma)$ is approximated by

$$f_{appr}(\gamma) := \min\{\mathcal{L}(x,\gamma) : x \in \mathrm{conv}(x_1,\ldots,x_k)\}.$$

Since $x \in \mathrm{conv}(x_1,\ldots,x_k)$ if and only if $\exists\lambda \in \Lambda := \{\lambda \in I\!R^k : \lambda \geq 0,\ e^T\lambda = 1\}$ with $x = X\lambda$, we can write out the definition of f_{appr} to get

$$f_{appr}(\gamma) = \min_{\lambda\in\Lambda}\ b^T\gamma + (c - \mathcal{A}^T\gamma)^T(X\lambda) = \min_{\lambda\in\Lambda}(F^T\lambda + \gamma^T G\lambda),$$

using the definition of F and G, and noting that $G\lambda = b - AX\lambda$.

(ii) Proximal point idea.
The second ingredient of the bundle method consists in the proximal point idea, which penalizes displacements from the currently best point $\hat{\gamma}$ with a term proportional to $\|\gamma - \hat{\gamma}\|^2$.

In summary therefore, the bundle method asks to find a new trial point $\gamma_{test} \geq 0$ by maximizing, for some prescribed parameter $t > 0$ the function

$$f_{appr}(\gamma) - \frac{1}{2t}\|\gamma - \hat{\gamma}\|^2 \tag{6.5}$$

over the set $\gamma \geq 0$. The choice of t is somewhat of an art. We obtain the following dual problem to be solved;

$$\begin{aligned}\max_{\gamma\geq 0}\min_{\lambda\in\Lambda}\ & F^T\lambda + (G\lambda)^T\gamma - \frac{1}{2t}\|\gamma - \hat{\gamma}\|^2 \\ &= \max_{\gamma}\min_{\lambda\in\Lambda,\eta\geq 0} F^T\lambda + (G\lambda)^T\gamma - \frac{1}{2t}\|\gamma - \hat{\gamma}\|^2 + \gamma^T\eta \\ &= \min_{\lambda\in\Lambda,\eta\geq 0}\max_{\gamma}\ F^T\lambda + (G\lambda)^T\gamma - \frac{1}{2t}\|\gamma - \hat{\gamma}\|^2 + \gamma^T\eta\end{aligned}$$

The inner maximization problem is smooth in γ and the first-order optimality condition is

$$\frac{\partial}{\partial\gamma}(\cdot) = 0 \ \Leftrightarrow\ G\lambda + \eta - \frac{1}{t}(\gamma - \hat{\gamma}) = 0 \ \Leftrightarrow\ \gamma = \hat{\gamma} + t(G\lambda + \eta).$$

Hence,

$$\min_{\lambda\in\Lambda,\eta\geq 0} F^T\lambda + (G\lambda)^T(\hat{\gamma} + t(G\lambda+\eta)) - \frac{t}{2}\|G\lambda+\eta\|^2 + (\hat{\gamma}+t(G\lambda+\eta))^T\eta$$

$$= \min_{\lambda\in\Lambda,\eta\geq 0} (F+G^T\hat{\gamma})^T\lambda + t\langle G\lambda+\eta, G\lambda+\eta\rangle - \frac{t}{2}\langle G\lambda+\eta, G\lambda+\eta\rangle + \hat{\gamma}^T\eta.$$

Therefore maximizing $f_{appr}(\gamma) - \frac{1}{2t}\|\gamma-\hat{\gamma}\|^2$ is dual to the following minimization

$$\min_{\lambda\in\Lambda,\eta\geq 0} (F+G^T\hat{\gamma})^T\lambda + \frac{t}{2}\|G\lambda+\eta\|^2 + \hat{\gamma}^T\eta. \tag{6.6}$$

The final problem in λ and η is easy if one set of the variables is kept constant. We start by keeping $\eta \geq 0$ constant, and solving the following minimization problem for λ:

$$\min_{\lambda\in\Lambda} \frac{t}{2}\langle G\lambda, G\lambda\rangle + t\langle G\lambda,\eta\rangle + \langle F+G^T\hat{\gamma},\lambda\rangle. \tag{6.7}$$

This minimization problem is called the *lambda problem*. We now keep as a constant λ computed from the previous lambda problem, and solve the minimization problem (6.6) for η. This minimization problem is called the *eta problem*:

$$\min_{\eta\geq 0} \frac{t}{2}\langle\eta,\eta\rangle + \langle tG\lambda+\hat{\gamma},\eta\rangle =: \min_{\eta\geq 0}\Theta(\eta). \tag{6.8}$$

The eta problem can be solved coordinatewise. It follows that

$$\text{argmin } \Theta(\eta) = \begin{cases} \eta_i = -\frac{1}{t}(\hat{\gamma}_i + t(G\lambda)_i) & \text{if} \quad \eta_i \geq 0 \\ 0 & \text{if} \quad \eta_i < 0. \end{cases}$$

In the next step we solve a new lambda problem with fixed η from the previous eta problem. In this way we iteratively solve the lambda and eta problems until we reach some prescribed accuracy of λ and η. Using these estimates λ and η, we get the *new trial point*

$$\gamma_{test} = \hat{\gamma} + t(G\lambda+\eta).$$

Note that $\lambda_i = 0$ implies that the column x_i of X has no influence on the minimization. We use this observation and remove any columns of X with corresponding $\lambda_i = 0$.

The bundle method now asks to evaluate f at γ_{test}, producing the function value $f(\gamma_{test})$ and the matching pair $(\gamma_{test}, x_{test})$. In our computations (see Chapter 7) we solve the minimization problem (6.3) with the interior point method. Using some standard criteria, we decide whether or not γ_{test} becomes the currently best point, we update our information and start a new iteration.

6.3.2 Bundle Method: Practical Details

We now describe our version of the bundle algorithm to get an approximate solution of maximizing

$$f(\gamma) = b^T\gamma + \min_{x \in \mathcal{X}}(c - \mathcal{A}^T\gamma)^T x.$$

Input: $\gamma_{start} \geq 0$, an initial guess to the dual variables γ, such as $\gamma_{start} = 0$.

Output: $f_{final}, \gamma_{final}, x_{final}$, the final values of f, with corresponding γ and some approximate primal point x.

Initialization: We evaluate $f(\gamma_{start})$. The oracle returns a matching pair $(\gamma_{start}, x_{start})$. Set $\hat{\gamma} = \gamma_{start}$ and $X = (x_{start})$. Compute G and F using X according to (6.4).

Iteration: In a general step of the algorithm, we have $X = (x_1, \ldots, x_k)$ and $\hat{\gamma}$ such that $(\hat{\gamma}, x_k)$ is a matching pair. We further have G and F computed from X.

- **New trial point:** We determine a new trial point γ_{test} by maximizing (6.5) as follows. We first solve (6.6) yielding $\lambda \in \Lambda$ and $\eta \geq 0$. We set $\gamma_{test} = \hat{\gamma} + t(G\lambda + \eta)$ as the new trial point.

- **Stopping test:** If $f_{appr}(\gamma_{test}) - f(\hat{\gamma}) \leq \epsilon$, then return $\hat{\gamma}, f(\hat{\gamma}), \hat{x} = X\lambda$.

- **Evaluation at γ_{test}:** Evaluate $f(\gamma_{test})$ and let $(\gamma_{test}, x_{test})$ denote the matching pair returned by the oracle.

- **Update X and $\hat{\gamma}$:** if
$$f(\gamma_{test}) > f(\hat{\gamma})$$
then
$$\hat{\gamma} = \gamma_{test}.$$
Use λ to remove any columns x_i of X for which $\lambda_i \leq$ tol where tol is some prescribed tolerance, and add x_{test} to X making sure that the last column of the updated X yields a matching pair with $\hat{\gamma}$.

Remark 6.1 *(Special features of the bundle method)*
The bundle method shows a fast initial convergence but a strong tailing off effects as the iterates approach the optimal solution. This is a characteristic of all subgradient methods.

Regarding convergence of the algorithm the following theorem holds. Its proof can be found in several papers, such as [40].

Theorem 6.1 *Let $\epsilon = 0$ in the algorithm above.*

(1) If the algorithm terminates after a finite number of steps then the final $\hat{\gamma}$ is a maximizer of f.

(2) If the approximate maximizer $\hat{\gamma}$ changes only a finite number of times then the final $\hat{\gamma}$ is a maximizer of f and the sequence of test points γ_{test} converges to this $\hat{\gamma}$.

(3) If the approximate maximizer $\hat{\gamma}$ changes infinitely often then the sequence of the $\hat{\gamma}$'s converges to a maximizer of f.

6.3.3 Modifications to Solve (P)

Contrary to the 'classical' use of the bundle method, where the function $f(\gamma)$ to be maximized is considered to be fixed, we have a more flexible version in mind, where f will change in the course of the algorithm.

To get the initial function $f(\gamma)$ we solve

$$\min\{c^T x : x \in \mathcal{X}\}$$

yielding an initial minimizer x^*. In principle we could now define $f(\gamma)$ by dualizing all the constraints $\mathcal{A}x \geq b$. Recall however, that the number m of constraints may be substantially larger than the dimension n of the problem. To maintain efficiency, we are interested only in those constraints, which are active at the optimum but which are unfortunately not known in advance. We look at $r^* := \mathcal{A}x^* - b$ and set $r_{\min} := \min\{r_i^*\}$. In the unlikely event that $r_{\min} \geq 0$, x^* is optimal for (P) and we are done. Otherwise $r_{\min} < 0$ and we consider now only those constraints from $\mathcal{A}x \geq b$ which are 'badly' violated by x^* to define f.

Specifically, let

$$I := \{i : r_i^* \leq \frac{1}{2} r_{\min}\}.$$

We denote by $\mathcal{A}_I$ the submatrix of $\mathcal{A}$ with rows indexed by I, a similar definition holds for b_I. For notational convenience we define

$$f_I(\gamma) := \min_{x \in \mathcal{X}} \; c^T x + (b_I - \mathcal{A}_I x)^T \gamma,$$

and observe that the dimension of γ depends on I.

Thus we start by maximizing f_I using the bundle method described above. In order to maintain computational efficiency, we include as an additional stopping condition an upper bound on the number of iterations. Therefore, when the method stops, returning some γ and some x, it is likely that γ is still far from the true minimizer of f_I. To continue, we need to decide on the following two issues:

- Which of the constraints in I should be kept?
- Are there additional violated constraints that should be added to I?

We use γ to answer to the first question. A large value γ_i indicates that the constraint i is binding and hence should not be removed. $\gamma_i = 0$ indicates that constraint i may be inactive, and therefore could be removed. In summary, we use γ to purge I by removing constraints i with a value γ_i smaller than a prescribed fraction of $\max(\gamma)$.

To decide whether new constraints should be added to I is less obvious.

Chapter 7

Bounds for the QAP by using the Bundle Method

This chapter is based on joint work with F. Rendl [77, 79].

7.1 Bundle Method to Solve QAP SDP Relaxations

Interior point methods are very useful and reliable solution methods for semidefinite programs of moderate size. In Section 4.3 we showed (see Table 4.2) that interior point methods are not convenient for the relaxations QAP_{R_2} and QAP_{R_3}. In order to efficiently compute lower bounds of these relaxations, we need a method that is capable to deal with a huge number of constraints. The bundle method turns out to be a convenient method for this purpose.

In order to specify the bundle concept from Chapter 6, we derive its main ideas for the relaxation QAP_{R_3} (see Section 4.1). We consider the constraints

$$G_J(\hat{V}R\hat{V}^T) = 0,$$

and the nonnegativity constrains collected in the operator N (see page 45)

$$N(\hat{V}R\hat{V}^T) \geq 0,$$

as the "hard constraints" for the QAP_{R_3} relaxation. We dualize the "hard constraints" and maintain explicitly only the constraints from $\mathcal{R}$ (see(4.11));

$$\mathcal{R} = \{R \succeq 0 : R \in \mathcal{S}_{(n-1)^2+1},\ \text{arrow}(\hat{V}R\hat{V}^T) = e_0\}.$$

Introducing Lagrange multipliers γ' and $\gamma'' \geq 0$ for the equations and non-negativity constraints respectively, the Lagrangian is

$$\mathcal{L}(R, \gamma) = \text{tr}\ LR + (\gamma')^T G(\hat{V}R\hat{V}^T) - (\gamma'')^T N(\hat{V}R\hat{V}^T),$$

where $\gamma = (\gamma', \gamma'')$.

Now we define

$$f(\gamma) := \min_{R \in \mathcal{R}}\ \mathcal{L}(R, \gamma) = \min_{R \in \mathcal{R}}\ \langle L + \hat{V}^T(G^T(\gamma') - N^T(\gamma''))\hat{V}, R\rangle, \tag{7.1}$$

and the relaxation QAP_{R_3} is equivalent to

$$\max_{\gamma \in \Gamma}\ f(\gamma), \tag{7.2}$$

where $\Gamma := \{(\gamma', \gamma'') : \gamma'' \geq 0\}$. The problem (7.2) is also difficult to solve directly, but we can efficiently locally approximate f and deal with the maximization over Γ, see Chapter 6. Note that for some input γ the minimization problem over $\mathcal{R}$ in (7.1), can be easily computed with the interior point method. It includes $n^2 + 1$ constraints for the n-dimensional QAP.

We follow now the idea of the bundle method from Chapter 6 and Fischer et al. [27]. For the start of the algorithm we take initial $\gamma = 0$ and compute R from (7.1). A pair (γ, R) is a *matching pair* for f, if $f(\gamma) = \mathcal{L}(R, \gamma)$. Let

$$\gamma^* = (\gamma'^*, \gamma''^*).$$

If (γ^*, R^*) is a matching pair for f then

$$g^G(\gamma'^*) = G(\hat{V}R^*\hat{V}^T)$$

is a subgradient of f at γ'^*, and

$$g^N(\gamma''^*) = -N(\hat{V}R^*\hat{V}^T)$$

is a subgradient of f at γ''^*. We denote the *currently best approximation* to the maximizer of f with $\hat{\gamma} = (\hat{\gamma}', \hat{\gamma}'')$.

In a general step, we assume to have $\bar{R} = (R_1, \ldots, R_k)$ and $\hat{\gamma} := \gamma_k$, with $(\hat{\gamma}, R_k)$ a matching pair. For each matching pair (γ_i, R_i) we calculate corresponding subgradients g_i^G and g_i^N and form matrices

$$G^G = (g_1^G, \ldots, g_k^G) \quad \text{and} \quad G^N = (g_1^N, \ldots, g_k^N).$$

Let $\lambda = (\lambda_1, \ldots, \lambda_k)^T$, $\Lambda = \{\lambda : \lambda \geq 0,\ e^T\lambda = 1\}$, and $F = (\mathrm{tr}(LR_1), \ldots, \mathrm{tr}(LR_k))^T$. The goal is to approximate the function $f(\gamma)$ in the neighborhood of the current iterates reasonable well. The function $f(\gamma)$ is approximated by

$$\begin{aligned} f_{appr}(\gamma) &= \min_{\lambda\in\Lambda} \langle L + \hat{V}^T(G^T(\gamma') - N^T(\gamma''))\hat{V},\ \sum_{i=1}^k \lambda_i R_i\rangle \\ &= \min_{\lambda\in\Lambda} \sum_{i=1}^k \lambda_i \langle L,\ R_i\rangle + \langle \gamma', \sum_{i=1}^k \lambda_i G(\hat{V}R_i\hat{V}^T)\rangle \\ &\quad -\langle \gamma'', \sum_{i=1}^k \lambda_i N(\hat{V}R_i\hat{V}^T)\rangle \\ &= \min_{\lambda\in\Lambda} F^T\lambda + (\gamma')^T G^G\lambda + (\gamma'')^T G^N\lambda. \end{aligned} \tag{7.3}$$

Since f_{appr} is built of local information from the previous iterates, in order to preserve a reasonable quality of the approximation we should stay in the vicinity of the current point $\hat{\gamma}$. Therefore we use the *proximal point* idea and add the *penalty term* for the displacement from the current point. We now determine a new candidate $\gamma = (\gamma', \gamma'') \in \Gamma$ from the current iterate $\hat{\gamma} = (\hat{\gamma}', \hat{\gamma}'')$ by solving the concave problem

$$\max_{\gamma\in\Gamma}\ f_{appr}(\gamma) - \frac{1}{2t}||\gamma - \hat{\gamma}||^2, \tag{7.4}$$

where $t > 0$ is a parameter. The choice of t is somewhat of an art. Setting (7.3) into the maximization problem (7.4), we obtain the optimization problem

$$\max_{\gamma\in\Gamma}\ \min_{\lambda\in\Lambda}\ F^T\lambda + (\gamma')^T G^G\lambda + (\gamma'')^T G^N\lambda - \frac{1}{2t}||\gamma - \hat{\gamma}||^2$$

$$= \min_{\lambda\in\Lambda,\ \eta\geq 0}\ \max_{\gamma}\ F^T\lambda + (\gamma')^T G^G\lambda + (\gamma'')^T G^N\lambda + (\gamma'')^T\eta - \frac{1}{2t}||\gamma - \hat{\gamma}||^2. \tag{7.5}$$

First–order optimality conditions for the inner maximization in (7.5) are

$$\frac{\partial}{\partial\gamma'}(\cdot) = 0 \Leftrightarrow G^G\lambda - \frac{1}{t}(\gamma' - \hat{\gamma}') = 0 \Leftrightarrow \gamma' = \hat{\gamma}' + tG^G\lambda, \tag{7.6}$$

$$\frac{\partial}{\partial\gamma''}(\cdot) = 0 \Leftrightarrow G^N\lambda - \frac{1}{t}(\gamma'' - \hat{\gamma}'') + \eta = 0 \Leftrightarrow \gamma'' = \hat{\gamma}'' + t(\eta + G^N\lambda). \tag{7.7}$$

We now insert equations for γ' and γ'' obtained in (7.6) and (7.7) respectively, into (7.5) and attain the optimization problem

$$\min_{\substack{\lambda\in\Lambda\\ \eta\geq 0}} \frac{t}{2}||G^G\lambda||^2 + \frac{t}{2}||G^N\lambda + \eta||^2 + \langle F + (\hat{\gamma}')^T G^G + (\hat{\gamma}'')^T G^N, \lambda\rangle + \langle\hat{\gamma}'', \eta\rangle. \tag{7.8}$$

This minimization problem could be easily solved if one set of the variables is kept constant. Keeping η constant results in the *lambda problem* (see also (6.7));

$$\min_{\lambda\in\Lambda} \Big(\frac{t}{2}(||G^G\lambda||^2 + ||G^N\lambda||^2) + t\langle G^N\lambda, \eta\rangle + \langle F + (\hat{\gamma}')^T G^G + (\hat{\gamma}'')^T G^N, \lambda\rangle\Big). \tag{7.9}$$

We solve the lambda problem by the interior point method. Keeping λ constant results in the *eta problem.* The eta problem for the relaxation QAP_{R_3} is (see also (6.8))

$$\min_{\eta\geq 0} \frac{t}{2}\langle\eta,\eta\rangle + t\langle\eta, G^N\lambda\rangle + \langle\hat{\gamma}'', \eta\rangle =: \min_{\eta\geq 0}\Theta(\eta).$$

The eta problem can be solved coordinatewise. It turns out that

$$\operatorname{argmin}\,\Theta = \begin{cases} \eta_i = -\frac{t}{2}(\hat{\gamma}_i'' + t(G^N\lambda)_i) & \text{if} \quad \eta_i \geq 0 \\ 0 & \text{if} \quad \eta_i < 0. \end{cases}$$

Thus we start with $\eta = 0$, solve for λ which we then keep constant to solve for η and iterate this process several times to get (approximate) solutions η, λ of (7.8). Setting these estimates λ and η into (7.6) and (7.7) we obtain the next trial point $\gamma_{test} = (\gamma'_{test}, \gamma''_{test})$. We then evaluate f at γ_{test}, producing the function value $f(\gamma_{test})$ and the matching pair $(\gamma_{test}, R_{test})$. For more detailed explanation of bundle method see Chapter 6.

Table 7.1: Bounds of the QAP_{R_2} relaxation computed with the bundle method and computation time per one iteration of the bundle algorithm

	exact	QAP_{R_2}	time
Nug20	2570	2380	15.11 ”
Nug21	2438	2244	18.56 ”
Nug22	3596	3372	22.01 ”
Nug24	3488	3217	35.44 ”
Nug25	3744	3438	44.49 ”
Nug30	6124	5651	122.35 ”

Table 7.2: Computation time per one iteration of the bundle algorithm for the QAP_{R_3} relaxation

n	20	21	22	30
time	22.86”	28.89”	43.38”	223.56”

Remark 7.1 *We similarly derive the bundle concept for the relaxation* QAP_{R_2}. *Here we keep the set of the gangster constraints unchanged during the iterations, and do not include the nonnegativity constraints. The lower bounds obtained for this relaxation are weaker then the lower bounds for the relaxation* QAP_{R_3}, *but still very strong (see Table 7.3 and 7.4).*

Tables 7.1 and 7.2 present the running times required for the one iteration of the bundle algorithm for the relaxations QAP_{R_2} and QAP_{R_3} respectively. Note that there is not a big difference in the running times concerning these two relaxations. It is due to the method's capability of controlling the size of the set of the nonnegativity constraints.

7.2 Computational Results for the Relaxations in $\mathcal{S}_{n^2+1}$

In this Section we present computational results. First, we compare the lower bounds QAP_{R_2} and QAP_{R_3} obtained with the bundle method, with several existing bounding strategies. We use the same test problems as in [4] and [97]. All instances have no linear term, i. e. they are pure quadratic and they are taken from the current version of QAPLIB [14]. The implementation of our bounds was done in MATLAB and performed on Athlon XP 1800.

Tables 7.3 and 7.4 collect some instances from QAPLIB [14], their optimum values, lower bounds from the literature, and our bounds. More precisely, the Tables 7.3 and 7.4 read as follows. The first column gives the problem instances and their sizes, e. g. Had30 refers to the Hadley instance of the size 30. In the second column we provide the optimum value for each instance. The remaining columns give lower bounds in the following order; GLB is the Gilmore–Lawler bound; KCCEB is the dual LP–based bound from [48]; PB is the projected eigenvalue bound from Hadley, Rendl and Wolkowicz [34], and QPB1 is the quadratic programming bound from Anstreicher and Brixius [4]. The last two columns present the bounds $\mathrm{QAP}_{\mathrm{R}_2}$ and $\mathrm{QAP}_{\mathrm{R}_3}$ that are described in Section 4.1 and computed by the bundle method. 'n. a.' means that the value of the bound is not available for a particular problem. All bounds are rounded up to the next integer.

Tables 7.3 and 7.4 demonstrate the efficiency of $\mathrm{QAP}_{\mathrm{R}_2}$ and $\mathrm{QAP}_{\mathrm{R}_3}$. These two relaxations were already proposed in [97], but we can approximate them within reasonable computation time. The bounds of the weaker $\mathrm{QAP}_{\mathrm{R}_2}$ relaxation, dominate over all listed bounds, obtained by different bounding strategies, for all Hadley and Nugent instances. The $\mathrm{QAP}_{\mathrm{R}_3}$ bounds are better then all presented bounds in all instances except for problems Scr12, Scr15 and Tai12a, in which KCCEB dominate. Note that $\mathrm{QAP}_{\mathrm{R}_3}$ bounds obtain positive values for all Eschermann problems.

Table 7.5 reports the Gilmore–Lawler bound, the projected eigenvalue bound, $\mathrm{QAP}_{\mathrm{R}_2}$ and $\mathrm{QAP}_{\mathrm{R}_3}$ bound for Nug27, Nug28 and Kra32 instances. In the second column of Table 7.5 are given optimal solutions of the problems. We separate these three problems from other presented in Table 7.3 and 7.4, since we are not acquainted with any other lower bounds for these instances.

$\mathrm{QAP}_{\mathrm{R}_2}$ and $\mathrm{QAP}_{\mathrm{R}_3}$ bounds presented in Table 7.3, 7.4 and 7.5 are obtained

after 300 iterations of the bundle algorithm. Although we let our algorithm to run for 300 iteration, the bounds computed after the first 200 iterations are still competitive. Table 7.6 presents $\mathrm{QAP}_{\mathrm{R}_3}$ bounds for the Nugent type instances obtained after 10, 20, 50, 100, 200 and 300 iterations of the bundle algorithm. The results show that after fast initial progress (first 100 iterations), as the bound approaches the optimum there is a strong tailing–off effect. Note that bounds for Nug20 and Nug21 obtained in first 100 iterations of the bundle algorithm are better then all bounds obtained with other bounding strategies listed in the Table 7.3 and 7.4. Table 7.6 also shows that for other listed Nugent instances, 200 iterations were sufficient in order to obtain the best known bounds. Figure 7.1 presents the deviation from the integer optimum of the normalized bounds, obtained after increasing numbers of the bundle iterations (see Table 7.6). Computed deviations are given in percent. Note the similar behavior for all presented instances.

We have chosen Nugent instances for demonstrating previously mentioned results since they are considered as very difficult instances. Similar results are valid for other QAPLIB problems.

The running time of one iteration performed by the bundle algorithm does not differ a lot concerning $\mathrm{QAP}_{\mathrm{R}_2}$ and $\mathrm{QAP}_{\mathrm{R}_3}$ relaxations. The running time of one single iteration of the problem size $n = 30$ for the relaxation $\mathrm{QAP}_{\mathrm{R}_2}$ is two, and for the relaxation $\mathrm{QAP}_{\mathrm{R}_3}$ is three minutes (see Table 7.1 and 7.2). That is due to the capability of controlling the size of the set of important inequalities. The majority of the time is consumed for solving the problem (7.1) by interior point method.

Table 7.3: Comparing bounds for QAPLIB instances I

	OPT	GLB	KCCEB	PB	QPB1	QAP_{R_2}	QAP_{R_3}
Esc16a	68	38	41	47	55	49	59
Esc16b	292	220	274	250	250	275	288
Esc16c	160	83	91	95	95	111	142
Esc16d	16	3	4	- 19	-19	-13	8
Esc16e	28	12	12	6	6	11	23
Esc16g	26	12	12	9	9	10	20
Esc16h	996	625	704	708	708	905	970
Esc16i	14	0	0	-25	-25	-22	9
Esc16j	8	1	2	-6	-6	-5	7
Had12	1652	1536	1619	1573	1592	1639	1643
Had14	2724	2492	2661	2609	2630	2707	2715
Had16	3720	3358	3553	3560	3595	3675	3699
Had18	5358	4776	5078	5104	5143	5282	5317
Had20	6922	6166	6567	6625	6677	6843	6885
Kra30a	88900	68360	75566	63717	68572	68526	77647
Kra30b	91420	69065	76235	63818	69021	71429	81156
Nug12	578	493	521	472	482	528	557
Nug14	1014	852	n.a.	871	891	958	992
Nug15	1150	963	1033	973	996	1069	1122
Nug16a	1610	1314	1419	1403	1448	1526	1570
Nug16b	1240	1022	1082	1046	1071	1136	1188
Nug17	1732	1388	1498	1487	1529	1619	1669
Nug18	1930	1554	1656	1663	1705	1798	1852
Nug20	2570	2057	2173	2196	2254	2380	2451
Nug21	2438	1833	2008	1979	2055	2244	2323
Nug22	3596	2483	2834	2966	3080	3372	3440
Nug24	3488	2676	2857	2960	3028	3217	3310
Nug25	3744	2869	3064	3190	3272	3438	3535
Nug30	6124	4539	4785	5266	5365	5651	5803

Table 7.4: Comparing bounds for QAPLIB instances II

	OPT	GLB	KCCEB	PB	QPB1	QAP_{R_2}	QAP_{R_3}
Rou12	235528	202272	223543	200024	206102	219018	223680
Rou15	354210	298548	323589	296705	303777	220567	333287
Rou20	725522	599948	641425	597045	607822	641577	663833
Scr12	31410	27858	29538	4727	8585	23844	29321
Scr15	51140	44737	48547	10355	12479	41881	47840
Scr20	110030	86766	94489	16113	23960	82106	94998
Tai12a	224416	195918	220804	193124	199597	215241	219760
Tai15a	388214	327501	351938	325019	330310	349179	358802
Tai17a	491812	412722	441501	408910	416033	440333	451317
Tai20a	703482	580674	616644	575831	585139	617630	637300
Tai25a	1167256	962417	1005978	956657	983456	1008248	1041337
Tai30a	1818146	1504688	1565313	1500407	1518059	1573580	1652186
Tho30	149936	90578	99855	119254	124684	134368	136059

Table 7.5: Bounds for QAPLIB instances

	OPT	GLB	PB	QAP_{R_2}	QAP_{R_3}
Nug27	5234	3701	4493	4887	4965
Nug28	5166	3786	4433	4780	4901
Kra32	88700	67390	59735	75848	79659

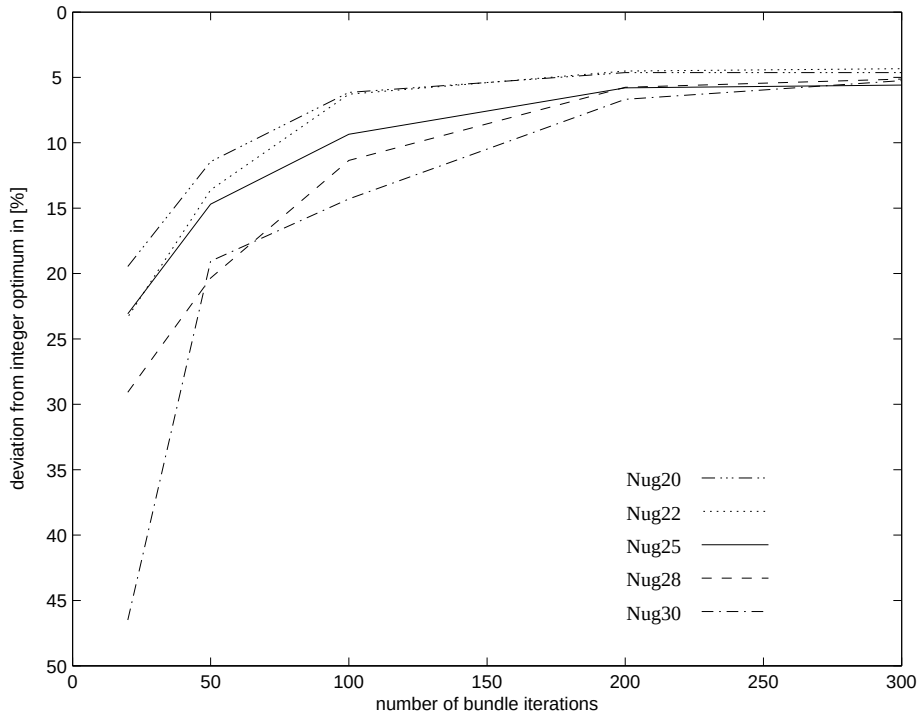

Figure 7.1: Deviation from the integer optimum of the normalized bounds, for different Nugent instances, obtained after increasing numbers of the bundle iterations

Table 7.6: QAP_{R_3} bounds in dependence of number of iterations of the bundle algorithm

	exact	10 it.	20 it.	50 it.	100 it.	200 it.	300 it.
Nug20	2570	1519	2070	2276	2412	2451	2451
Nug21	2438	1163	1935	2122	2253	2320	2323
Nug22	3596	1590	2757	3107	3370	3434	3440
Nug24	3488	1214	2553	2953	3193	3302	3310
Nug25	3744	1994	2880	3194	3394	3527	3535
Nug27	5234	464	3441	4399	4767	4946	4965
Nug28	5166	197	3664	4115	4580	4869	4901
Nug30	6124	416	3277	4957	5249	5715	5803

7.3 Computational Results for the Relaxation in $\mathcal{S}_{(n-1)^2+1}$

In this Section we present the lower bounds for the relaxation QAP_{Rs_4} (see Section 5.2) computed with the bundle method. In this relaxation we maintain explicitly the constraints from the set

$$\mathcal{L}R := \mathcal{R}_S \cup \mathcal{L},$$

where $\mathcal{R}_S$ is (see (5.15))

$$\mathcal{R}_S = \{R \succeq 0 : R \in \mathcal{S}_{(n-1)^2+1},\ \mathrm{arrow}(R) = e_0\},$$

and $\mathcal{L}$ is (see (5.22))

$$\mathcal{L} = \{R : R \in \mathcal{S}_{(n-1)^2+1},\ \mathrm{Le}(R) = -(n-1)(n-2)\}.$$

For the relaxation QAP_{Rs_4} the set $\mathcal{L}R$ is a set of the "easy constraints", in the frame of the bundle method. Hence, in the bundle method appears the following minimization problem (see (6.3))

$$f(\gamma) = \min_{R \in \mathcal{L}R} \mathcal{L}(R, \gamma), \tag{7.10}$$

Table 7.7: The time table for solving the minimization problem (7.10) and (7.1) with the interior point method

n	16	18	20	22	24	26	28	30
$\mathcal{L}R$	4.25"	8.79"	19.08"	25.68"	48.21"	81.84" '	124.81"	185.13"
$\mathcal{R}$	7.01"	13.32"	23.87"	44.07"	77.41" '	123.41"	206.56"	290.55"

where

$$\mathcal{L}(R,\gamma) = \langle L + G_S^T(\gamma'), R\rangle,$$

and $G_S(\cdot)$ denotes the gangster constraints on $\mathcal{S}_{(n-1)^2+1}$. For the relaxation QAP_{Rs_4} the gangster constraints are the "hard constraints".

In Table 7.7 we present the computational times for solving the minimization problem (7.10) (respec. (7.1)) over the set $\mathcal{L}R$ (respec. $\mathcal{R}$) with the interior point method. In the first row are listed dimensions of the QAP problem. In the second (respec. third) row are the running times required for solving the minimization problem (6.3) with the respect to $\mathcal{L}R$ (respec. $\mathcal{R}$). Note that for the bigger instances ($n \geq 22$) the difference in time for solving the same problem is considerable.

If n is the size of the QAP problem, then the minimization over the set $\mathcal{L}R$ includes $(n-1)^2+2$ constraints (i.e. $(n-1)^2+1$ from the arrow operator and one from the Le operator), and the minimization over the set $\mathcal{R}$, n^2+1 constraints (all from the arrow operator acting on the $\mathcal{S}_{n^2+1}$). Therefore, the solutions of the minimization problem (6.3) over the set $\mathcal{L}R$ are weaker then the solutions over the set $\mathcal{R}$.

In Table 7.8 we collect optimum values, lower bounds from literature, and our bounds for some Nugent instances. More, precisely, Table 7.8 reads as follows. The first column gives the problem instances and their size. In the second column we provide the optimum value for each instance. The remaining columns give lower bounds in the following order; GLB is the Gilmore–Lawler bound; PB is the projected eigenvalue bound [34]; $\mathrm{QAP}_{\mathrm{R_2}}$ is the bound presented in Section 4.1, and $\mathrm{QAP}_{\mathrm{Rs_4}}$ is the relaxation from Section 5.2. Lower bounds for relaxations $\mathrm{QAP}_{\mathrm{R_2}}$ and $\mathrm{QAP}_{\mathrm{Rs_4}}$ are computed with the bundle method. Computed lower bounds for the $\mathrm{QAP}_{\mathrm{Rs_4}}$ relaxation are stronger then GLB, and for some instances stronger then PB bounds. Note hat for all listed instances $\mathrm{QAP}_{\mathrm{Rs_4}}$ bounds are weaker then $\mathrm{QAP}_{\mathrm{R_2}}$. This

Table 7.8: $\mathrm{QAP}_{\mathrm{Rs}_4}$ bounds for Nugent instances

	OPT	GLB	PB	$\mathrm{QAP}_{\mathrm{R}_2}$	$\mathrm{QAP}_{\mathrm{Rs}_4}$
Nug12	578	493	472	528	506
Nug14	1014	852	871	958	934
Nug15	1150	963	973	1069	970
Nug16a	1610	1314	1403	1526	1452
Nug16b	1240	1022	1046	1136	1084
Nug17	1732	1388	1487	1619	1552
Nug18	1930	1554	1663	1798	1730
Nug20	2570	2057	2196	2380	2241
Nug21	2438	1833	1979	2244	2093
Nug22	3596	2483	2966	3372	3085
Nug24	3488	2676	2960	3217	2942
Nug25	3744	2869	3190	3438	3184
Nug30	6124	4539	5266	5568	4870

is to be expected according to the difference in the number of the constraints between these two SDP relaxations.

Remark 7.2 *In Section 5.4, we give the relaxation*

$$(\mathrm{QAP}_{Ss}) \qquad \begin{array}{ll} \min & \mathrm{tr}\, LR \\ \text{s.t.} & G_{\hat{j}}(\bar{V}R\bar{V}^T) = 0 \\ & N(\bar{V}R\bar{V}^T) \geq 0 \\ & G_S(R) = 0 \\ & R \in \mathcal{L}R. \end{array}$$

We believe that this relaxation can be efficiently solved then the relaxation QAP_{R_3}*, if we maintain explicitly only the constraints from the set* $\mathcal{L}R$*. The set* $\mathcal{L}R$ *contains* $(n-1)^2+2$ *constrains, and the set* $\mathcal{R}$ *contains* n^2+1 *constrains. Hence, the time for solving the minimization problem (6.3) is shorter (see Table 7.7). We leave this issue for the future work.*

7.4 The Bounds after Branching

In this Section we investigate the lower bounds for some QAPLIB instances in the first and second level of the branching tree. For the purpose of applying $\mathrm{QAP}_{\mathrm{R_3}}$ bounds within a branch and bound framework we investigated the effect on the bounds after fixing an assignment $x_{ij} = 1$. If a considerably large *growth rate* in the branching tree occurs, then a necessary condition for their applicability within Branch and Bound schema is satisfied. In order to evaluate a growth rate of $\mathrm{QAP}_{\mathrm{R_3}}$ bounds, we compare our results for Had12 with results presented in [4]. Table 7.9 gives lower bounds for Had12 in the first level of the branching tree. First column lists all 12 child problems. With Had12.j we denote jth "child" problem obtained by setting $x_{1j} = 1$, $j = 1, \ldots, 12$. The meaning of the rest of the columns is as follows; second column presents exact solutions of the "child" problems; PB and QPB are projected eigenvalue bound and quadratic programming bound respectively, and QAP_{R_3} is the bound presented in Section 2. Table 7.9 shows that the performance of QPB is far superior to that of PB, and that the performance of QAP_{R_3} is far superior to that of QPB. Note that the value of QPB is sufficient to fathom Had12.7 and Had12.12, but the value of QAP_{R_3} is sufficient to fathom all "child" problems except Had12.3.

Table 7.9: Results for the first level in the branching tree for Had12

	exact	PB	QPB	QAP_3
Had12	1652	1573	1592	1643
Had12.1	1674	1593	1629	1673
Had12.2	1690	1590	1639	1680
Had12.3	1652	1573	1607	1652
Had12.4	1662	1585	1616	1656
Had12.5	1696	1608	1647	1694
Had12.6	1706	1616	1649	1696
Had12.7	1714	1601	1656	1705
Had12.8	1654	1566	1610	1653
Had12.9	1660	1573	1617	1655
Had12.10	1672	1605	1628	1670
Had12.11	1694	1601	1641	1690
Had12.12	1700	1618	1656	1699

Table 7.10: Results for the first level in the branching tree for Nug12

	exact	QAP_2	QAP_3
Nug12	578	529	557
Nug12.1	586	551	578
Nug12.2	586	551	577
Nug12.5	578	552	575
Nug12.6	600	556	584

Further branching experiments are done on the Nugent set of problems. These data sets have the following characteristic. The linear term C is equal to 0. The matrix A represents the rectilinear cell distance of a rectangular array of cells. There exists some symmetry in such constructed data. In case of $n = 12$, the resulting rectangular cell array has the following form:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

.

Note that the distance matrix A would not change, if the following cell array would have been used:

5	4	3	2	1
10	9	8	7	6
15	15	13	12	11

.

Mathematically speaking, there exist several permutation matrices X, such that $A = XAX^T$.

Hence, only four subproblems are to be considered in the first level of Nug12 problem (see Table 7.10), and analogously six subproblems in the first level of Nug15 problem (see Table 7.11). With Nugxx.j we denote jth "child" problem obtained by setting $x_{j1} = 1$.

Figure 7.2 shows that in the first level of the branching tree for Nug15, all "child" problems except Nug15.1 are fathomed. Our computations of all "child" problems of Nug15.1 (e.g. 196 since there is no symmetry) resulted with only 14 not fathomed problems (see Figure 7.2). Hence, we have proved

Table 7.11: Results for the first level in the branching tree for Nug15

	exact	QAP_2	QAP_3
Nug15	1150	1071	1122
Nug15.1	1150	1103	1140
Nug15.2	1168	1116	1154
Nug15.3	1164	1121	1157
Nug15.6	1166	1114	1154
Nug15.7	1182	1130	1167
Nug15.8	1184	1134	1169

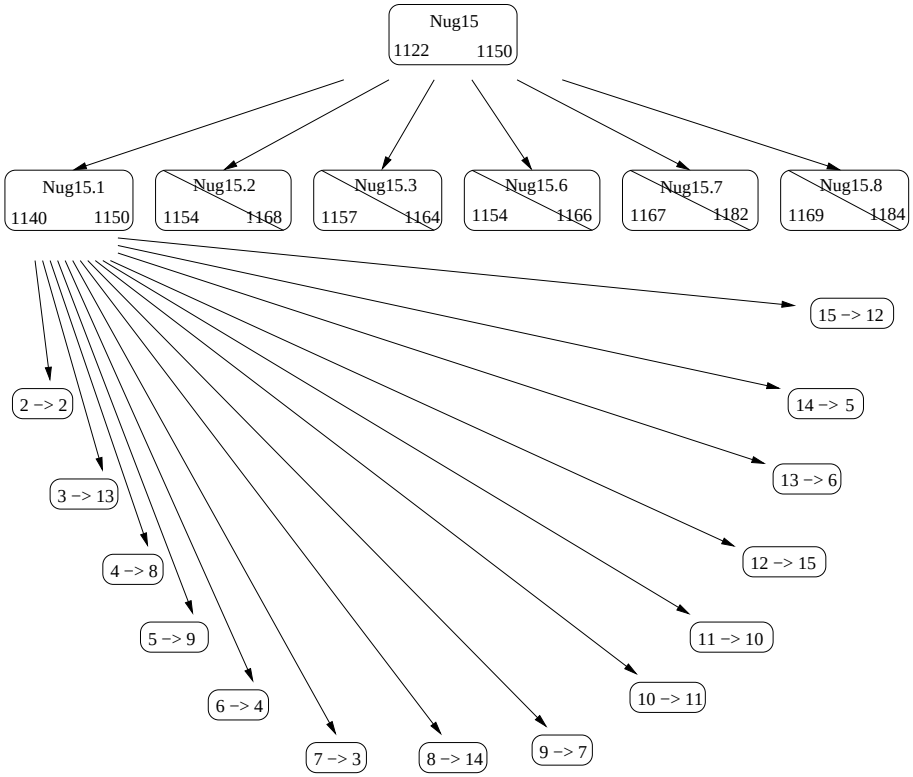

Figure 7.2: First and second level in the branching tree for Nug15

Table 7.12: Results for the first level in the branching tree for Nug20

	exact	QAP_2	QAP_3
Nug20	2570	2380	2451
Nug20.1	2612	2449	2518
Nug20.2	2570	2420	2488
Nug20.3	2586	2421	2487
Nug20.6	2592	2427	2501
Nug20.7	2584	2420	2491
Nug20.8	2604	2419	2502

Table 7.13: Results for the first level in the branching tree for Nug30

	QAP_2	QAP_3
Nug30	5568	5803
Nug30.1	5809	5939
Nug30.2	5771	5895
Nug30.3	5756	5881
Nug30.7	5767	5900
Nug30.8	5750	5885
Nug30.9	5756	5891
Nug30.13	5756	5896
Nug30.14	5750	5883
Nug30.15	5768	5889

the optimal solution of Nug15 problem in the second level of the branching tree.

Table 7.12 presents the bounds in the first level of the branching tree for Nug20, and Table 7.13 bounds in the first level of the branching tree for Nug30.

Our experiments show a promise for implementing $\mathrm{QAP}_{\mathrm{R}_3}$ bounds on Branch and Bound framework.

7.5 The Branch and Bound Tree Estimator

In this Section we present a simple method, introduced by Knuth [52], for estimating the performance of the Branch and Bound algorithm. We apply the algorithm to particular QAP instances. The same testings were done by Anstreicher et.al [6]. It is important to have such estimates before attempting to solve problems since they give a reasonable estimate of the time and/or nodes that the solution process will require. Since we have not jet developed the optimal branching strategies, we are interested in the random exploration of the branching tree.

Let T be a tree rooted at α_0, and α a node in the tree. Let $c(\alpha)$ be a cost associated with the node α. Then, the total cost for the tree is

$$\mathrm{cost}(T) = \sum_{\alpha \in T} c(\alpha). \tag{7.11}$$

If $c(\alpha) = 1$ then then $\mathrm{cost}(T)$ is the number of nodes in the tree. If $c(\alpha)$ is the time required for a search algorithm to process node α then $\mathrm{cost}(T)$ is the total time to search the tree. Let $d(\alpha)$ denote the "children" of α. The goal is to estimate $\mathrm{cost}(T)$ without using any branching strategies. Knuth's estimation procedure is as follows [52, 6].

procedure EstimateCost:
set: $k = 0,\ d_0 = 1,\ c = 0$
for $k = 0, 1, 2, \ldots$
$c \leftarrow c + d_k c(\alpha_k)$
$n_k = |d(\alpha_k)|$
if $n_k = 0$ **return** c
else choose $\alpha_{k+1} \in d(\alpha_k)$ at *random*
$d_{k+1} = d_k n_k$
end

In our algorithm is each child node α_{k+1} *equally likely*. Hence, procedure EstimateCost makes a random walk in the tree (without any backtracking) at each node choosing uniformly from the available children until a terminal node is reached. The procedure computes the estimate

$$C = c(\alpha_0) + d_1 c(\alpha_1) + \ldots + d_k c(\alpha_k) + \ldots \ . \tag{7.12}$$

The validity of estimate (7.12) can be proved as follows.

Theorem 7.1 *[52] The expected value of C, as computed by the procedure* EstimateCost*, is* $\mathrm{cost}(T)$.

Proof: For every k in (7.12) is

$$d_k = d_{k-1} n_{k-1}.$$

Thus

$$d_k = d_{k-1} n_{k-1} = \ldots = n_0 \cdot \ldots \cdot n_{k-1},$$

and the term

$$n_0 \cdot \ldots \cdot n_{k-1} c(\alpha_k) \tag{7.13}$$

occurs in (7.12) with probability $1/(n_0 \cdot \ldots \cdot n_{k-1})$, since this is the chance that the algorithm will consider the node α_k. Hence, the sum of all terms (7.13) has the expected value (7.11). ∎

This theorem demonstrates that C is indeed an appropriate statistic to compute. If the procedure is applied n times, resulting in estimates $C_1, \ldots, C_n$, then the sample mean

$$\bar{C} = \frac{1}{n}(\sum_{i=1}^{n} C_i)$$

Table 7.14: Estimation of nodes for Nug15

level	nodes
0	1
1	4
2	8
3	5
4	0
total	18

should be close to cost(T) if n is sufficient large. Since n_k, the number of children in node α_k, depends on the value of the upper bound, the procedure EstimateCost can most accurately estimate cost(T) when an accurate estimate of the optimal value is known.

We have tested the performance of Knuth's estimator on some Nugent instances. Table 7.14 (7.15, 7.16 respectively) illustrate the performance of the EstimateCost procedure on the Nug15 (Nug20, Nug25 respectively). In Tables are given estimates for the number of nodes only up to the certain level. This occurs since the probability of a random dive reaching deep levels of the tree is very small. If the choice of the child node is not completely random, it is possible to investigate the more difficult parts of the tree and reach deeper levels. Knuth [52] suggested using non–uniform probabilities to choose the child at each node during the dive. The idea is to replace in the procedure EstimateCost the statement $d_{k+1} = d_k n_k$ by $d_{k+1} = d_k/p_k$ where p_k is the probability corresponding to the child picked. We have not done this testings but it would be interesting to make them in the future work.

Table 7.15: Estimation of nodes for Nug20

level	nodes
0	1
1	6
2	19
3	122
4	146
5	78
6	0
total	372

Table 7.16: Estimation of nodes for Nug25

level	nodes
0	1
1	6
2	26
3	374
4	1357
5	3709
6	6061
7	0
total	11534

Appendix A

Symmetric and Positive Semidefinite Matrices

In addition to the characterization $A = A^T$ of symmetric matrices, the following theorem allows a geometrical characterization.

Theorem A.1 *(Spectral Theorem for Symmetric Matrices [46, Thm 4.1.5])*
The real $n \times n$ matrix A is symmetric if and only if there is a real orthogonal matrix $U \in \mathcal{M}_n$ and a real dagonal matrix $\Lambda \in \mathcal{M}_n$ such that $A = U\Lambda U^T$.

The columns u_i of U are the eigenvectors of A, satisfying

$$Au_i = \lambda_i u_i,$$

where λ_i is the ith diagonal entry of Λ.

Remark A.1 *(Product of Two Symmetric Matrices)*
If A and B are real symmetric of the same dimension, then AB is symmetric if and only if $AB = BA$ if and only if there is an orthogonal matrix U that simultaneously diagonalizes A and B, i.e. $A = U^T\Lambda_1 U$ and $B = U^T\Lambda_2 U$.

The semidefiniteness (resp. positive definiteness) of a matrix X can equivalently be expressed as X having only nonnegative (resp. positive) eigenvalues.

Lemma A.1 *(Fejer's Trace Theorem)*
A symmetric matrix A is positive semidefinite if and only if $\langle A, B\rangle \geq 0$ for all $B \in \mathcal{S}_n^+$.

Lemma A.2 *Let $A, B \in \mathcal{S}_n^+$. Then $\langle A, B\rangle \geq 0$, and $\langle A, B\rangle = 0$ if and only if $AB = 0$.*

PROOF: Let $A, B \in \mathcal{S}_n^+$ and $k = \operatorname{rank}(A)$. Let the eigenvalue decomposition of A be given by $A = P\Lambda P^T$ with $\Lambda = \operatorname{Diag}(\lambda)$, $\lambda = (\lambda_1, \ldots, \lambda_k)^T > 0$, $P \in \mathcal{M}_{n,k}$, $P^T P = I_k$. Then,

$$\langle A, B\rangle = \operatorname{tr}(P\Lambda P^T B) = \operatorname{tr}(\Lambda P^T B P) = \sum_{i=1}^{k} \lambda_i(A) \cdot P_{\cdot,i}^T B P_{\cdot,i} \geq 0.$$

As B is positive semidefinite, $P_{\cdot,i}^T B P_{\cdot,i}$ is nonnegative for $i = 1, \ldots, k$. $\langle A, B\rangle = 0$ implies that eigenvectors corresponding to positive eigenvalues of A belong to the null space of B and thus $AB = 0$. ■

The following theorem we use in the Section 3.5.

Theorem A.2 *(Hoffman and Wielandt [45])*
Let A and B be the symmetric matrices of order n with spectral decomposition $A = PDP^T$, $B = QEQ^T$. We assume that the diagonal matrix D contains the eigenvalues of A in nondecreasing order, and E contains the eigenvalues of B in nonincreasing order. Furthermore, $PP^T = QQ^T = I$. Then

$$\min\{\operatorname{tr}(AXBX^T) : \ X^T X = I\} = \operatorname{tr}(DE).$$

Moreover, the minimum is attained for $X = PQ^T$.

A proof of this theorem can be found for instance in [45], the result can be traced back to the work of John von Neumann [71].

Theorem A.3 *[7] Let A and B be symmetric matrices. Then,*

$$\min\{\operatorname{tr}(AXBX^T) : \ XX^T = I\} = \max\{\operatorname{tr} S + \operatorname{tr} T : \ B\otimes A - I\otimes S - T\otimes I \succeq 0\}.$$

Appendix B

Matrix Calculus

We give a survey of the results of Matrix Analysis which are used in this thesis.

Let $F : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable function and and $x \in \mathbb{R}^n$. Then the gradient of $F(x)$ with respect to x is

$$\nabla F(x) = \begin{bmatrix} \frac{\partial F(x)}{\partial x_1} \\ \vdots \\ \frac{\partial F(x)}{\partial x_n} \end{bmatrix}.$$

If $F : \mathbb{R}^n \to \mathbb{R}^m$; $x \mapsto [F_1(x), \ldots, F_m(x)]^T$ is a continuously differentiable function, then

$$\nabla F(x) = [\nabla F_1(x), \ldots, \nabla F_m(x)].$$

In Section 2.4 we introduce the barrier function

$$-\log\det(R).$$

Here we prove that it is a strictly convex function.

Lemma B.1 *The function* $\log\det A$ *is strictly concave on the set of positive definite matrices.*

PROOF: Let $A, B \in \mathcal{S}_n^{++}$. We denote the eigenvalues of $A^{-\frac{1}{2}}BA^{-\frac{1}{2}}$ by λ_i. For $0 < \alpha < 1$ we have

$$\begin{aligned}
\log\det(\alpha A + (1-\alpha)B) &= \log\det(A^{\frac{1}{2}}(\alpha I + (1-\alpha)A^{-\frac{1}{2}}BA^{-\frac{1}{2}})A^{\frac{1}{2}}) \\
&= \log(\det(A)\det(\alpha I + (1-\alpha)A^{-\frac{1}{2}}BA^{-\frac{1}{2}})) \\
&= \log(\det(A)) + \log\prod_{i=1}^{n}(\alpha + (1-\alpha)\lambda_i) \\
&= \log(\det(A)) + \sum_{i=1}^{n}\log(\alpha + (1-\alpha)\lambda_i) \\
&> \log(\det(A)) + (1-\alpha)\log\prod_{i=1}^{n}\lambda_i \\
&= \log(\det(A)) + (1-\alpha)\log\det(A^{-1}B) \\
&= \log(\det(A)) - (1-\alpha)\log\det(A) \\
&\quad +(1-\alpha)\log\det(B) \\
&= \alpha\log(\det(A)) + (1-\alpha)\log\det(B).
\end{aligned}$$

If $A = B$ or $\alpha \in \{0, 1\}$ then holds equality. ∎

Note that $\det(R)$ grows to infinity as $R \succ 0$ approaches the boundary of the positive semidefinite cone. Hence, it acts as a barrier for the iterates. In order to compute the first order optimality conditions for the barrier problem (see (2.12)) we need to derive the derivative of $\log\det(X)$.

Theorem B.1 *For a nonsingular matrix $X \in \mathcal{M}_n$*

$$\nabla_X \det(X) = \det(X)\mathrm{vec}(X^{-T}).$$

PROOF: We denote by $X_{ij} \in \mathcal{M}_n$ the matrix obtained from X by deleting the ith row and jth column. Laplace expansion along row i yields

$$\det(X) = \sum_{j=1}^{n}(-1)^{i+j}x_{ij}\det(X_{ij}).$$

Therefore

$$\frac{\partial\det(X)}{\partial x_{ij}} = (-1)^{i+j}\det(X_{ij}) = \det[X_{\cdot,1}, \ldots, X_{\cdot,j-1}, e_i, X_{\cdot,j+1}, \ldots, X_{\cdot,n}].$$

By Cramer's rule the vector

$$y = \left[\frac{\partial\det(X)}{\partial x_{i1}}, \ldots, \frac{\partial\det(X)}{\partial x_{in}}\right]^T$$

solves $Xy = \det(X)e_i$. The vector y^T forms the ith row of $\nabla_X \log\det(X)$. Consequently, the transpose of the solution Y of $XY = \det(X)I$ yields, after application of the vec-operator, $\nabla_X \det(X)$. ∎

Remark B.1 *We obtain the derivative of* $\log\det(X)$ *for positive definite matrices* X *by the chain rule,*

$$\nabla_X \log\det(X) = \frac{1}{|\det(X)|}\nabla_X \det(X) = \mathrm{vec}(X^{-1}).$$

For scalar matrix functions it is common practice to represent the gradient in the form of a matrix. We keep to this practice and, by a slight abuse of notation, write $\nabla_X \log\det(X) = X^{-1}$.

Very often happens that the function is not smooth everywhere, i.e. the gradient of the function exists almost everywhere. For such functions we introduce the the concept of subdifferential (see e.g. [42]), which generalizes that of gradient. The concept of subdifferential is essentially local.

Definition B.1 *Let* $F : \mathbb{R}^n \to \mathbb{R}$ *be a convex function. The* subdifferential *of* F *at* x *is the nonempty compact convex set of vectors* $s \in \mathbb{R}^n$ *satisfying*

$$\partial F := \{s : F(y) \geq F(x) + \langle s, y - x\rangle, \forall y \in \mathbb{R}^n\}.$$

A vector $s \in \partial F$ *is called a* subgradient *of* F *at* x.

Appendix C

Minimax Problems

C.1 Convex Functions and Convex Sets

In this Section we establish some terms for convex functions. In later Sections we expand these terms on convex–concave functions.

Definition C.1 *A subset M of $\mathbb{R}^n$ is called an* affine *set if $(1-\lambda)x+\lambda y \in M$ for every $x, y \in M$ and $\lambda \in \mathbb{R}$.*

The empty set and $\mathbb{R}^n$ are extreme examples of affine sets. In general, an affine sets has to contain, along with any two different points, the entire line through those points. The intuitive picture is that of an endless uncurved structure, like a line or a plane in space. Let us give some examples of affine sets

Example C.1 *Points, lines and planes are affine sets. Hyperplane of $\mathbb{R}^n$ is also affine set. Every affine subset of $\mathbb{R}^n$ is an intersection of a finite collection of hyperplanes.*

The intersection of an arbitrary collection of affine sets is again affine.

Definition C.2 *For given any $S \subset \mathbb{R}^n$ there exist a unique smallest affine set containing S, that is namely the intersection of the collection of affine sets M such that $S \subset M$. This set is called the* affine hull *of S and is denoted by* aff S.

It can be proved that aff S consist of all the vectors of the form $\lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_m x_m$, such that $x_i \in S$ and $\lambda_1 + \lambda_2 \ldots + \lambda_m = 1$.

Definition C.3 *A subset C of $\mathbb{R}^n$ is said to be* convex *if $(1-\lambda)x + \lambda y \in C$ whenever $x, y \in C$ and $0 < \lambda < 1$.*

According to previous definitions let us now define the following term.

Definition C.4 *The* relative interior *of a convex set C in $\mathbb{R}^n$, which we denote by* ri C*, is defined as the interior which results when C is regarded as a subset of its affine hull* aff C*.*

In other words,

$$\text{ri } C = \{x \in \text{aff } C : \exists \epsilon > 0, B(x, \epsilon) \cap (\text{aff } C) \subset C\},$$

where by B we denote the *Euclidean ball* in $\mathbb{R}^n$:

$$B(x, \epsilon) = \{y : d(x, y) \leq \epsilon\}. \tag{C.1}$$

Obviously, it follows

$$\text{ri } C \subset C \subset \text{cl } C,$$

where cl C is the closure of the set C. Every affine subset of $\mathbb{R}^n$ is an intersection of a finite collection of hyperplanes. So, the empty set may itself be regarded as the intersection of two different parallel hyperplanes, while $\mathbb{R}^n$ may be regarded as the intersection of the empty collection of the hyperplanes of $\mathbb{R}^n$.

Remark C.1 *Every affine set is closed, since every affine set is an intersection of hyperplanes. and every hyperplane can be expressed as the level set of a continuous function.*

Let us now define convex functions.

Definition C.5 *Let C be a nonempty convex set in $\mathbb{R}^n$. A function $f : C \to \mathbb{R}$ is said to be* convex *on C if, for all pairs $(x, y) \in C \times C$ and all $\alpha \in (0, 1)$ there holds*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y). \tag{C.2}$$

We say that f is strictly convex *on C when (C.2) holds as a strict inequality if $x \neq y$.*

Let us define now the following term:

Definition C.6 *Let f be a function whose values are real or $\pm\infty$ and whose domain is a subset S of $\mathbb{R}^n$. The set*

$$\{(x,\mu) : x \in S, \mu \in \mathbb{R}, \mu \geq f(x)\}$$

is called the epigraph *of f and is denoted by* epi$f$.

There are several useful correspondences between convex sets and convex functions. Here are some of the examples.

Example C.2 *The simplest associates with each set C in $\mathbb{R}^n$ is the* indicator function $\delta(\cdot|C)$ *of C, where*

$$\delta(x|C) = \begin{cases} 0 & \text{if} \quad x \in C \\ \\ +\infty & \text{if} \quad x \notin C. \end{cases} \tag{C.3}$$

Clearly C is a convex set if and only if $\delta(\cdot|C)$ is a convex function.

Example C.3 *The* support function $\delta^*(\cdot|C)$ *of a convex set C in $\mathbb{R}^n$ is defined by*

$$\delta^*(x|C) = \sup\{\langle x, y\rangle : y \in C\}. \tag{C.4}$$

The epigraph of the indicator function, defined in *Example* C.2 is a "half–cylinder" with cross–section C.

Remark C.2 *A function f is* convex *function on C if and only if* epi$f$ *is convex subset of $\mathbb{R}^{n+1}$. (A* concave *function on C is a function whose negative is convex.)*

Definition C.7 *The* effective domain *of a convex function f on C, which we denote by* dom$f$, *is the projection of the epigraph of f on $\mathbb{R}^n$:*

$$\text{dom}f = \{x : \exists\mu, (x,\mu) \in \text{epi}f\} = \{x : f(x) < +\infty\}.$$

This is a convex set in $\mathbb{R}^n$, since it is the image of the convex set epi$f$ *under a linear transformation. Its dimension is called the* dimension *of f.*

Definition C.8 *A convex function is said to be* proper *if its epigraph is non–empty and and contains no vertical lines, i. e. if*

$$f(x) < +\infty$$

for at least one x *and*

$$f(x) > -\infty$$

for every x*. A convex function which is not proper is* improper.

According to previous definition, a convex function f is proper if and only if the convex set $C = \mathrm{dom} f$ is non–empty and the restriction of f to C is finite. We can also say that a proper convex function on $\mathbb{R}^n$ is a function obtained by taking a finite convex function f on a non–empty convex set C and then extending it to all of $\mathbb{R}^n$ by setting $f(x) = +\infty$ for $x \notin C$.

Example C.4 *The function*

$$f(x) = \begin{cases} -\infty & \text{if} \quad |x| < 2 \\ 0 & \text{if} \quad |x| = 2 \\ +\infty & \text{if} \quad |x| > 2 \end{cases}$$

is an improper convex function.

A convex function may have discontinuities at the boundary points of its interval of definition $\mathrm{dom} f$, but has a continuous behavior in interior. There is a simple closure operation which makes any proper convex function lower semi–continuous merely by redefining it at certain relative boundary points of its effective domain. Let us first define the lower semi–continuous function.

Definition C.9 *An extended–real–valued function* f *given on a set* $S \subset \mathbb{R}^n$ *is said to be* lower semi–continuous *at a point* x *of* S *if*

$$f(x) \leq \lim_{i \to \infty} f(x_i)$$

for every sequence $x_1, x_2, \ldots$*, in* S *such that* x_i *converges to* x *and the limit of* $f(x_1), f(x_1), \ldots$*, exists in* $[-\infty, +\infty]$*. This condition may be expressed as:*

$$f(x) = \lim_{y \to x} \inf f(y) = \lim_{\epsilon \to 0} (\inf\{f(y) : |x - y| \leq \epsilon\}).$$

Similarly, f is said to be upper semi–continuous *at x if*

$$f(x) = \lim_{y \to x} \sup f(y) = \lim_{\epsilon \to 0} (\sup\{f(y) : |x - y| \leq \epsilon\}).$$

The natural importance of lower semi–continuity in the study of convex functions is apparent from the following result.

Theorem C.1 *[81, pg. 51] Let f be an arbitrary function from $\mathbb{R}^n$ to $[-\infty, +\infty]$. Then the following conditions are equivalent:*
(a) f is lower semi–continuous throughout $\mathbb{R}^n$;
(b) $\{x : f(x) \leq \alpha\}$ is closed for every $\alpha \in \mathbb{R}$;
(c) The epigraph of f is a closed set in $\mathbb{R}^{n+1}$.

PROOF: Lower semi–continuity at x can be re–expressed as the condition that $\mu \geq f(x)$ whenever $\mu = \lim \mu_i$ and $x = \lim x_i$ for sequences $\mu_1, \mu_2, \ldots$ and $x_1, x_2, \ldots$ such that

$$\mu_i \geq f(x_i), \quad \forall i.$$

But this condition is the same as (c).

This also implies (b). Namely, if we take $\alpha = \mu = \mu_1 = \mu_2 = \ldots$. On the other hand, suppose (b) holds. Suppose x_i converges to x and $f(x_i)$ converges to μ. For every real $\alpha > \mu$, $f(x)$ must ultimately be less than α, and hence

$$x \in \text{cl } \{y : f(y) \leq \alpha\} = \{y : f(y) \leq \alpha\}.$$

Hence $f(x) \leq \mu$. this proves (b) implies (a). ∎

For given any function f on $\mathbb{R}^n$ there exist a greatest lower semi–continuous function (not necessarily finite) majorized by f, namely the function whose epigraph is the closure in $\mathbb{R}^{n+1}$ of the epigraph of f.

Definition C.10 *The greatest lower semi–continuous function that is majorized by f is called* lower semi–continuous hull *of f. The* closure *of a convex function f is defined to be the lower semicontinuous hull of f if f nowhere has the value $-\infty$, whereas the closure of f is defined to be the constant function $-\infty$ if f is an improper convex function such that $f(x) = -\infty$ for some x. Either way, the closure of f is another convex function and it is denoted by* cl$f$.

Finally we can define closed function.

Definition C.11 *A convex function is said to be* closed *if* $\mathrm{cl} f = f$.

To get an idea about what the closure operation is like, consider the following examples.

Example C.5 *Let the convex function f on $\mathbb{R}$ is defined by*

$$f(x) = \begin{cases} 0 & x > 0 \\ +\infty & x \leq 0. \end{cases}$$

Here cl f *agrees with f everywhere except at the origin, where the value is* 0 *instead of $+\infty$.*

Example C.6 *Let C be a circular disk in $\mathbb{R}^2$. Let $f(x)$ be* 0 *in the interior of C and $+\infty$ outside of C, and assign arbitrary values in $[0, +\infty]$ to f on the boundary of C. Then f is a proper convex function on $\mathbb{R}^2$. The closure of f is obtained by redefining $f(x)$ to be* 0 *on the boundary of C.*

These examples suggest that the closure operation is a reasonable normalization which makes convex functions more regular by redefinding their values at certain points where there are unnatural discontinues.

Let us observe more examples.

Example C.7 *Let C be a nonempty interval of $\mathbb{R}$. The* indicator function *of C is*

$$I_C(x) := \begin{cases} 0 & \text{if} \quad x \in C \\ +\infty & \text{if} \quad \text{otherwise.} \end{cases}$$

It is a closed convex function if and only if C is closed.

A convex function can be closed without its affective domain being closed.

Example C.8 *The function on $\mathbb{R}$ given by*

$$f(x) := \begin{cases} \frac{1}{x} & \text{if} \quad x > 0 \\ +\infty & \text{if} \quad x \leq 0 \end{cases}$$

is a closed function.

Let f be any closed convex function on $\mathbb{R}^n$. There is a dual way of describing the function f. Namely, one can describe the set F^* consisting of all pairs $(x^*, \mu^*) \in \mathbb{R}^{n+1}$ such that the affine function $h(x) = \langle x, x^* \rangle - \mu^*$ is majorized by f. We have $h(x) \leq f(x)$ for every x if and only if

$$\mu^* \geq \sup\{\langle x, x^* \rangle - f(x) : x \in \mathbb{R}^n\}. \tag{C.5}$$

Thus F^* is actually the epigraph of the function f^* on $\mathbb{R}^n$ defined by

$$f^*(x^*) = \sup_x\{\langle x, x^* \rangle - f(x)\} = -\inf_x\{f(x) - \langle x, x^* \rangle\}. \tag{C.6}$$

Definition C.12 *The function f^*, defined in (C.6) is called the* conjugate *of f.*

It is actually the pointwise supremum of the affine functions $g(x^*) = \langle x, x^* \rangle - \mu$ such that (x, μ) belongs to the set $F = \text{epi } f$. Also, (C.6) actually means

$$f^*(x^*) = \sup_x\{\langle x, x^* \rangle - \mu : (x, \mu) \in \text{epi } f\}. \tag{C.7}$$

Hence f^* is another *convex function*, in fact a closed convex function. Since f is the pointwise supremum of the affine functions $h(x) = \langle x, x^* \rangle - \mu^*$ such that $(x^*, \mu^*) \in F^* = \text{epi } f^*$, we have

$$f(x) = \sup_{x^*}\{\langle x, x^* \rangle - f^*(x^*)\} = -\inf_{x^*}\{f^*(x^*) - \langle x, x^* \rangle\}. \tag{C.8}$$

This says that the conjugate f^{**} of f^* is f. The constant functions $+\infty$ and $-\infty$ are plainly conjugate to each other. Since these are the only improper closed convex functions, all the other conjugate pairs must be proper.

Example C.9 *Let us consider the closed proper convex function $f(x) = e^x$, $x \in \mathbb{R}$. By definition*

$$f^*(x^*) = \sup_x\{xx^* - e^x\}, \quad \forall x^* \in \mathbb{R}.$$

If $x^* < 0$ then $xx^* - e^x$ can be made arbitrarily large by taking $x \to -\infty$, so the supremum is $+\infty$. If $x^* > 0$, then the supremum is $x^* \log x^* - x^*$. If $x^* = 0$, the supremum is 0. Thus the function conjugate to the exponential function is

$$f^*(x^*) = \begin{cases} x^* \log x^* - x^* & \text{if} \quad x^* > 0 \\ 0 & \text{if} \quad x^* = 0 \\ +\infty & \text{if} \quad x^* < 0. \end{cases}$$

Finally, let us introduce an *inner product.*

Definition C.13 *An* inner product *for the conjugate of a convex or concave function f is defined as follows*

$$\langle f, x^* \rangle = \langle x^*, f \rangle = f^*(x^*).$$

Note that $\langle f, x^* \rangle = \langle x, x^* \rangle$ when f is the indicator of the point x, i.e. when

$$f(z) = \begin{cases} 0 & \text{if} \quad z = x \\ \\ +\infty & \text{if} \quad z \neq x. \end{cases}$$

C.2 Directions of Recession and Recessions Function

Closed bounded subsets of $\mathbb{R}^n$ are usually easier to work with than unbounded ones. If sets are convex, difficulties resulting from unboundedness are much smaller, since so many of the sets we need to consider, like epighaphs, are unbounded. Unbounded closed convex sets C have a simple behavior "at infinity". Suppose that x is a point of C. It seems that C must actually contain some entire half–line starting at x, as C is unbounded. The *directions* of such half–lines in C starting at a different point y are apparently just the translates of those starting at x. These directions in which C recedes indefinitely might possibly be thought of as "ideal points" of C lying at infinity. The half–lines in C starting at x could then be interpreted as the segments joining x with such ideal points of C. The objective below is to put these intuitive notions on a sound basis and to apply them to the study of convex functions. Let us first define the following term.

Definition C.14 *A subset K of $\mathbb{R}^n$ is called a* cone *if it is closed under positive scalar multiplication, i. e. $\lambda x \in K$ when $x \in K$ and $\lambda > 0$. A* convex cone *is a cone which is a convex set.*

Example C.10 *Two most important convex cones are the* non–negative orthant *of $\mathbb{R}^n$*

$$\{x = (\xi_1, \ldots, \xi_n) : \xi_1 \geq 0, \ldots, \xi_n \geq 0\}$$

and positive orthant *of* $\mathbb{R}^n$

$$\{x = (\xi_1, \ldots, \xi_n) : \xi_1 > 0, \ldots, \xi_n > 0\}.$$

Definition C.15 *A* direction *of* $\mathbb{R}^n$ *is an equivalence class of the collection of all closed half–lines of* $\mathbb{R}^n$ *under the equivalence relation "half–line* L_1 *is a translate of half–line* L_2*".*

The direction of a the half–line $\{x + \lambda d : \lambda \geq 0\}$, where $d \neq 0$, is then by definition a set of all translates of the half–line, and that is independent of x. Two vectors in $\mathbb{R}^n$ have the same direction if and only if they are positive scalar multiples of each other. Now we can define the following.

Definition C.16 *Let* C *be a non–empty convex set in* $\mathbb{R}^n$*. We say that* C recedes in the direction d*, where* $d \neq 0$*, if* C *includes all the half–lines in the direction* d *which start at points of* C*.*

The previous definition could be interpreted in the following way: C *recedes in the direction* of d if and only if

$$x + \lambda d \in C \quad \forall \lambda \geq 0, x \in C. \tag{C.9}$$

Definition C.17 *The set of all vectors* $d \in \mathbb{R}^n$ *satisfying the latter condition, including* $d = 0$*, will be called the* recession cone *of* C*. The recession cone of* C *will be denoted by* 0^+C*. Directions in which* C *recedes will also be referred to as* directions of recession *of* C*.*

Theorem C.2 *[81, pg. 61] Let* C *be a non–empty convex set. The recession cone* 0^+C *is then a convex cone containing the origin. It is the same as the set of vectors* y *such that* $C + y \subset C$*.*

Example C.11 *As examples of recession cones of convex sets in* $\mathbb{R}^2$ *for*

$$C_1 = \{(x_1, x_2) : x_1 > 0, x_2 \geq \frac{1}{x_1}\},$$

$$C_2 = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\},$$

one has

$$0^+C_1 = \{(x_1, x_2) : x_1 \geq 0, x_2 \geq 0\},$$

$$0^+C_2 = \{(x_1, x_2) : x_1 = 0 = x_2\} = \{(0, 0)\}.$$

Let f be a convex function on $\mathbb{R}^n$ not identically $+\infty$. The epigraph of f, as a non–empty convex set in $\mathbb{R}^{n+1}$, has a *recession cone* $0^+(\text{epi } f)$. By definition, $(y, \nu) \in 0^+(\text{epi } f)$ if and only if

$$(x, \mu) + \lambda(y, \nu) = (x + \lambda y, \mu + \lambda\nu) \in \text{epi } f \tag{C.10}$$

for every $(x, \mu) \in \text{epi } f$ and $\lambda \geq 0$.

Remark C.3 *For a given y, the values of v for which $(y, v) \in 0^+(\text{epi } f)$ will form a closed interval of $\mathbb{R}$ unbounded above, or the empty interval.*

Definition C.18 *The $0^+(\text{epi } f)$ is the epigraph of a certain function. We call that function the* recession function *of f, and we denoted by $f0^+$.*

By the definition, then

$$\text{epi } (f0^+) = 0^+(\text{epi } f).$$

Theorem C.3 *Let f be a proper convex function. The recession function $f0^+$ of f is then a positively homogeneous proper convex function. For every vector y, one has*

$$(f0^+)(y) = \sup\{f(x + y) - f(x) : x \in \text{dom } f\}. \tag{C.11}$$

If f is closed, $f0^+$ is closed too, and for any $x \in \text{dom } f$, $f0^+$ is given by the formula

$$(f0^+)(y) = \sup_{\lambda > 0} \frac{f(x + \lambda y) - f(x)}{\lambda} = \lim_{\lambda \to \infty} \frac{f(x + \lambda y) - f(x)}{\lambda}. \tag{C.12}$$

PROOF: See [81, page 66].

Definition C.19 *The set of all vectors y such that $(f0^+)(y) \leq 0$ will be called the* recession cone *of f.*

The recession cone is is a convex cone containing 0, closed if f is closed. It corresponds to the intersection of $0^+(\text{epi } f)$ with the horizontal hyperplane $\{(y,0) : y \in I\!R^n\}$ in $I\!R^{n+1}$.

Remark C.4 *The directions of the vectors in the recession cone of f will be called* directions in which f recedes, *or* direction of recession *of f.*

Theorem C.4 *Let f be a proper convex function. The support function of* dom f *is then the recession function f^*0^+ of f^*. If f is closed, the support function of* dom f^* *is the recession function $f0^+$ of f.*

PROOF: See [81, page 116].

C.3 Concave Functions

Definition C.20 *A function g from $I\!R^n$ to $[-\infty, +\infty]$ is* concave *if $-g$ is convex.*

For a concave function g, one defines

$$\text{epi } g = \{(x,\mu) : x \in I\!R^n, \mu \in I\!R, \mu \leq g(x)\}, \tag{C.13}$$

as *epigraph* of g and

$$\text{dom } g = \{x : g(x) > -\infty\}, \tag{C.14}$$

as *effective domain* of concave function g. One says that a function g is *proper* if

$$g(x) > -\infty$$

for at least one x and

$$g(x) < +\infty$$

for every x, i.e. if $-g$ is proper. The *closure* cl g of a concave function g is the pointwise infimum of all the affine functions h such that $h \geq g$, i.e. it is $-(\mathrm{cl}(-g))$. If g is proper, or if $x \in \mathrm{cl}(\mathrm{dom}\ g)$, one has

$$(\mathrm{cl}\ g)(x) = \lim_{y \to x} \sup g(y).$$

If g is the constant function $-\infty$, then cl $g = g$; but if g is an improper concave function which has the value $+\infty$ somewhere, then cl g is the constant function $+\infty$. One says that g is *closed* if cl $g = g$ (i.e. if $-g$ is closed). If g is proper, g is closed if and only if it is upper semi–continuous, i. e. if and only if the convex sets

$$\{x : g(x) \geq \alpha\}, \quad \alpha \in I\!R,$$

are all closed.

Definition C.21 *The* conjugate *of concave function g is defined by*

$$g^*(x^*) = \inf_x \{\langle x, x^* \rangle - g(x)\}, \tag{C.15}$$

and one has $g^{**} = \mathrm{cl}\ g$.

Remark C.5 *For the convex function $f = -g$, one has, (instead of $g^*(x^*) = -f^*(x^*)$),*

$$g^*(x^*) = -f^*(-x^*).$$

C.4 Bifunctions

Definition C.22 *We define a* bifunction *from $I\!R^m$ to $I\!R^n$ to be a mapping which assigns to each $u \in I\!R^m$ a function Fu on $I\!R^n$ with values in $[-\infty, +\infty]$. The value of Fu at a point $x \in I\!R^n$ will be denoted by $(Fu)(x)$. The function*

$$(u, x) \to (Fu)(x), \quad (u, x) \in I\!R^m \times I\!R^n = I\!R^{m+n},$$

will be called the graph function *of F.*

It is clear that each extended–real–valued function f on $\mathbb{R}^{m+n}$ is the graph function of exactly one bifunction from $\mathbb{R}^m$ to $\mathbb{R}^n$, namely

$$Fu = f(u, \cdot), \quad \forall u \in \mathbb{R}^m.$$

Thus a bifunction can simply be regarded as the first stage of a function broken down into two stages:

$$F : u \to Fu : x \to (Fu)(x).$$

Definition C.23 *A bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$ will be called* convex *if its graph function is convex on $\mathbb{R}^{m+n}$.*

This implies that Fu is a convex function on $\mathbb{R}^n$ for each $u \in \mathbb{R}^m$.

A convex bifunction will be said to be *closed* or *proper* according to whether its graph function is closed or proper, respectively. The *graph domain* of a convex bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$ is defined to be the effective domain of the graph function of F. The *effective domain* of a bifunction F, denoted by $\mathrm{dom}(F)$, is defined to be the set of all vectors $u \in \mathbb{R}^m$ such that Fu is not the constant function $+\infty$. Thus dom F is the projection on $\mathbb{R}^m$ of the graph domain of F in $\mathbb{R}^{m+n}$ and hence is a convex set in $\mathbb{R}^m$. If F is proper, dom F consists of the vectors u such that the convex function Fu is proper.

Let us give a simple example of a convex bifunction which will be very important to us theoretically.

Example C.12 *The* indicator bifunction *of a linear transformation A from $\mathbb{R}^m$ to $\mathbb{R}^n$, i.e. the F is defined by*

$$(Fu)(x) = \delta(x|Au) = \begin{cases} 0 & \text{if} \quad x = Au \\ +\infty & \text{if} \quad x \neq Au \end{cases} \tag{C.16}$$

This F is convex, because it is the indicator function of the graph of A, which happens to be a convex set in $\mathbb{R}^{m+n}$. Observe that F is closed and proper, and dom $F = \mathbb{R}^m$.

According to the Definition C.13 of the inner product, for any convex or concave bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$, we form

$$\langle Fu, x^* \rangle = \langle x^*, Fu \rangle = (Fu)^*(x^*) \tag{C.17}$$

as a function of (u, x^*) on $\mathbb{R}^m \times \mathbb{R}^n$. Thus, by Definition C.12, and (C.17) it follows

$$\langle Fu, x^* \rangle = \sup_x \{\langle x, x^* \rangle - (Fu)(x)\} \tag{C.18}$$

if F is convex, whereas (see (C.15))

$$\langle Fu, x^* \rangle = \inf_x \{\langle x, x^* \rangle - (Fu)(x)\} \tag{C.19}$$

if F is concave.

Finally, let us define the following term.

Definition C.24 *For any bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$, the* inverse *of F is defined to be the bifunction*

$$F_* : x \to F_* x : u \to (F_* x)(u)$$

from $\mathbb{R}^n$ to $\mathbb{R}^m$ given by

$$(F_* x)(u) = -(Fu)(x), \quad \forall x \in \mathbb{R}^n, \forall u \in \mathbb{R}^m.$$

Note that F_* is concave if F is convex, and vice versa. The inverse operation $F \to F_*$ clearly preserves clossedness and properness of convex or concave bifunctions, and it is involutory, i.e.

$$(F_*)_* = F.$$

C.5 Saddle–Functions

Definition C.25 *Let C and D be subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively, and let K be a function from $C \times D$ to $[-\infty, +\infty]$. We say that K is a* concave–convex *function if $K(u, v)$ is a concave function of $u \in C$ for each $v \in D$ and convex function of $v \in D$ for each $u \in C$.* Convex–concave *functions are defined similarly. We speak of both kinds of functions as* saddle–functions.

The theory of saddle–functions, like the purely convex or concave functions, can be reduced conveniently to the case where the functions are everywhere defined but possibly infinity–valued.

Let K be a concave–convex function on $C \times D$. We can extend the function K beyond sets C and D. There is usually more natural extensions of the function K. Here we give two possible extensions of a concave–convex function K:

$$K_1(u,v) = \begin{cases} K(u,v) & \text{if} \quad u \in C, \quad v \in D, \\ +\infty & \text{if} \quad u \in C, \quad v \notin D, \\ -\infty & \text{if} \quad u \notin C, \end{cases}$$

or

$$K_2(u,v) = \begin{cases} K(u,v) & \text{if} \quad u \in C, \quad v \in D, \\ -\infty & \text{if} \quad u \in C, \quad v \notin D, \\ +\infty & \text{if} \quad u \notin C. \end{cases}$$

We shall call K_1 the *lower simple extension* of K and K_2 the *upper simple extension* of K. Either K_1 or K_2 are adequate for most of the analysis of K.

Since we defined the term of effective domain for the convex functions, let us now define the same term for saddle functions.

For any given concave–convex function K on $\mathbb{R}^m \times \mathbb{R}^n$, we define

$$\begin{aligned} \mathrm{dom}_1 K &= \{u : K(u,v) > -\infty, \forall v\}, \\ \mathrm{dom}_2 K &= \{v : K(u,v) < +\infty, \forall u\}. \end{aligned} \tag{C.20}$$

Observe that $\mathrm{dom}_2 K$ is the intersection of the effective domains of the convex functions $K(u,\cdot)$ as u ranges over $\mathbb{R}^m$, while $\mathrm{dom}_1 K$ is the intersection of the effective domains of the concave functions $K(\cdot,v)$ as v ranges over $\mathbb{R}^n$. In particular, $\mathrm{dom}_1 K$ is a convex set in $\mathbb{R}^m$ and $\mathrm{dom}_2 K$ is a is a convex set in $\mathbb{R}^n$.

Definition C.26 *The product set*

$$\mathrm{dom}\ K = \mathrm{dom}_1 K \times \mathrm{dom}_2 K$$

is called effective domain *of* K.

Since

$$-\infty < K(u,v) < +\infty \quad u \in \mathrm{dom}_1 K,\ v \in \mathrm{dom}_2 K,$$

K is finite on dom K.

Definition C.27 *We say that K is* proper *if* dom $K \neq \emptyset$.

For a given concave–convex function K on $\mathbb{R}^m \times \mathbb{R}^n$ we can apply the convex and concave closure operations to achieve some regularization.

Definition C.28 *Let K be a concave–convex function. The function obtained by closing $K(u,v)$ as a convex function of v for each fixed u is called* convex closure *of K and is denoted by* $\mathrm{cl}_v K$ *or* $\mathrm{cl}_2 K$. *Similarly, the function obtained by closing $K(u,v)$ as a concave function of u for each fixed v is called* concave closure *of K and is denoted by* $\mathrm{cl}_u K$ *or* $\mathrm{cl}_1 K$. *If K coincides with its convex closure, we say K is* convex–closed, *if K coincides with its concave closure, we say K is* concave–closed.

These closure operations preserve concavity–convexity.

Definition C.29 *If K is a concave–convex function on $\mathbb{R}^n \times \mathbb{R}^n$, then* $\mathrm{cl}_2\mathrm{cl}_1 K$ *is called the* lower closure *and* $\mathrm{cl}_1\mathrm{cl}_2 K$ *is called* upper closure *of K. If K is convex–concave, terminology is reversed. We say that K is* lower closed *if and only if it coincides with its lower closure, and we say that K is* upper closed *if and only if it coincides with its upper closure.*

In general, the lower and upper closure operations do not quite produce the same result:

$$\mathrm{cl}_2\mathrm{cl}_1 K \neq \mathrm{cl}_1\mathrm{cl}_2 K. \tag{C.21}$$

Definition C.30 *Two concave–convex functions K and L on $\mathbb{R}^m \times \mathbb{R}^n$ are said to be* equivalent *if* $\mathrm{cl}_1 K = \mathrm{cl}_1 L$ *and* $\mathrm{cl}_2 K = \mathrm{cl}_2 L$.

Definition C.31 *We say that K is* closed *if* $\mathrm{cl}_1 K$ *and* $\mathrm{cl}_2 K$ *are both equivalent to K.*

In the view of the fact that

$$\mathrm{cl}_1\mathrm{cl}_1 K = \mathrm{cl}_1 K, \quad \mathrm{cl}_2\mathrm{cl}_2 K = \mathrm{cl}_2 K,$$

the conditions

$$\mathrm{cl}_1\mathrm{cl}_2 K = \mathrm{cl}_1 K, \quad \mathrm{cl}_2\mathrm{cl}_1 K = \mathrm{cl}_2 K,$$

are *necessary and sufficient* for K to be a closed saddle–function. Trivially, if K is closed and L is equivalent to K, then L is closed.

Let us now write some theorems and corollaries that we will need for proving theorems in minimax theory.

Theorem C.5 *[81, pg. 351] If F is any convex bifunction from $\mathbb{R}^m$ to $\mathbb{R}^n$, then $\langle Fu, x^*\rangle$ is a concave–convex function of (u, x^*) which is convex–closed, and one has*

$$(\mathrm{cl}\ (Fu))(x) = \sup_{x^*}\{\langle x, x^*\rangle - \langle Fu, x^*\rangle\}.$$

On the other hand, given any concave–convex function K on $\mathbb{R}^m \times \mathbb{R}^n$, define the bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$ by

$$(Fu)(x) = \sup_{x^*}\{\langle x, x^*\rangle - K(u, x^*)\}.$$

Then F is convex, Fu is closed on $\mathbb{R}^n$ for each $u \in \mathbb{R}^m$, and one has

$$\langle Fu, x^*\rangle = (\mathrm{cl}_2 K)(u, x^*).$$

(Similarly for concave bifunctions F and convex–concave functions K.)

Corollary C.1 *[81, page 352] If K is any concave–convex function on $\mathbb{R}^n \times \mathbb{R}^n$, then $\mathrm{cl}_1 K$ and $\mathrm{cl}_2 K$ are concave–convex functions such that $\mathrm{cl}_1 K$ is concave–closed and $\mathrm{cl}_2 K$ is convex–closed. (Similarly for convex–concave functions.)*

Theorem C.6 *[81, page 357] The relations*

$$K(u, x^*) = \langle Fu, x^*\rangle, \quad Fu = K(u, \cdot)^*,$$

define a one–to–one correspondence between the lower closed concave–convex functions K on $\mathbb{R}^m \times \mathbb{R}^n$ and the closed convex bifunctions F from $\mathbb{R}^m$ to $\mathbb{R}^n$. Similarly for upper closed saddle–functions and closed concave bifunctions.

Corollary C.2 *[81, page 358] Let $\underline{K}$ and $\overline{K}$ be concave–convex functions on $\mathbb{R}^m \times \mathbb{R}^n$. In order that there exists a closed convex bifunction F (necessarily unique) such that*

$$\underline{K}(u, x^*) = \langle Fu, x^* \rangle, \quad \overline{K}(u, x^*) = \langle u, F^* x^* \rangle,$$

it is necessary and sufficient that $\underline{K}$ and $\overline{K}$ satisfy the relations

$$\mathrm{cl}_1 \underline{K} = \overline{K}, \quad \mathrm{cl}_2 \overline{K} = \underline{K}.$$

These relations imply that $\underline{K}$ is lower closed, $\overline{K}$ is upper closed, and $\underline{K} \leq \overline{K}$.

Theorem C.7 *[81, page 363] Given any closed convex bifunction F from $\mathbb{R}^m$ to $\mathbb{R}^n$, let*

$$\underline{K}(u, x^*) = \langle Fu, x^* \rangle, \quad \overline{K}(u, x^*) = \langle u, F^* x^* \rangle,$$

and let $\Omega(F)$ be the collection of all concave–convex functions K on $\mathbb{R}^m \times \mathbb{R}^n$ such that $\underline{K} \leq K \leq \overline{K}$. Then $\Omega(F)$ is an equivalence class *(containing $\underline{K}$ and $\overline{K}$), and all the functions in $\Omega(F)$ are closed. Conversely, every equivalence class of closed concave–convex functions is of the form $\Omega(F)$ for a unique closed convex bifunction F.*

For any K in $\Omega(F)$, one has

$$\mathrm{cl}_1 K = \overline{K}, \quad \mathrm{cl}_2 K = \underline{K},$$

$$\mathrm{dom}\ K = \mathrm{dom}\ F \times \mathrm{dom}\ F^*,$$

$$(Fu)(x) = \sup_{x^*} \{\langle x, x^* \rangle - K(u, x^*)\},$$
$$(F^* x^*)(u^*) = \inf_{u} \{\langle u, u^* \rangle - K(u, x^*)\}.$$

Moreover,

$$K(u, x^*) = \langle Fu, x^* \rangle = \langle u, F^* x^* \rangle$$

if $u \in \mathrm{ri}\ (\mathrm{dom}\ F)$ or if $x^* \in \mathrm{ri}\ (\mathrm{dom}\ F^*)$.

Theorem C.8 *[81, page 366] Let K be the proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$. Let $C = \mathrm{dom}_1\ K$ and $D = \mathrm{dom}_2\ K$. In order that K be closed, it is necessary and sufficient that K have the following properties.*

(a) For each $u \in \text{ri } C$, $K(u,\cdot)$ *is a closed proper–convex function with effective domain* D.
(b) For each $u \in (C \setminus \text{ri } C)$, $K(u,\cdot)$ *is a proper convex function whose effective domain lies between* D *and* cl D.
(c) For each $u \notin C$, $K(u,\cdot)$ *is an improper convex function which has the value* $-\infty$ *throughout* ri D *(through* D *itself if actually* $u \notin \text{cl } C$*).*
(d) For each $v \in \text{ri } D$, $K(\cdot,v)$ *is a closed proper concave function with effective domain* C.
(e) For each $v \in (D \setminus \text{ri } D)$, $K(\cdot,v)$ *is a proper concave function whose effective domain lies between* C *and* cl C.
(f) For each $v \notin D$, $K(\cdot,v)$ *is an improper concave function which has the value* $+\infty$ *throughout* ri C *(throughout* C *itself if* $v \notin \text{cl } D$*).*

C.6 Minimax Problems

Minimax theory treats a class of extremum problems which involve a combination of minimization and maximization. Let C and D be arbitrary non–empty sets and let K be the function from $C \times D$ to $[-\infty, +\infty]$.

We will observe the following problems:

$$\sup_{u\in C} \inf_{v\in D} K(u,v) \quad \text{and} \quad \inf_{v\in D} \sup_{u\in C} K(u,v).$$

If the "sup inf" and "inf sup" are equal, the common value is called the *minimax* or *saddle–value* of K (with respect to maximizing over C and minimizing over D). One of the tasks of miminax theory is to furnish conditions under which the saddle–value exist and is attained in some suitable sense. In general, "sup inf" and "inf sup' might not be equal, but the certain inequality is at least satisfied.

Lemma C.1 *[81, pg. 379] If* K *is any function from a non–empty product set* $C \times D$ *to* $[-\infty, +\infty]$, *then*

$$\sup_{u\in C} \inf_{v\in D} K(u,v) \le \inf_{v\in D} \sup_{u\in C} K(u,v).$$

PROOF: For each $v \in D$, one has

$$K(u,v) \geq \inf_{v \in D} K(u,v)$$

for every $u \in C$. Consequently,

$$\sup_{u \in C} K(u,v) \geq \sup_{u \in C} \inf_{v \in D} K(u,v).$$

Since previous relation holds for every $v \in D$, one has

$$\inf_{v \in D} \sup_{u \in C} K(u,v) \geq \sup_{u \in C} \inf_{v \in D} K(u,v),$$

and lemma is proved. ■

Let us now define a saddle–point (see [42, page 328]).

Definition C.32 *A point $(\bar{u}, \bar{v}) \in C \times D$ is a* saddle–point *of K with respect to maximizing over C and minimizing over D if*

$$K(u,\bar{v}) \leq K(\bar{u},\bar{v}) \leq K(\bar{u},v), \quad \forall u \in C,\ \forall v \in D. \tag{C.22}$$

From (C.22) follows

$$K(u,\bar{v}) \leq K(\bar{u},v), \quad \forall u \in C,\ \forall v \in D.$$

By the definition of saddle point, $(\bar{u}, \bar{v})$ is a saddle–point of K if and only if

$$\sup_{u \in C} K(u,\bar{v}) = K(\bar{u},\bar{v}) = \inf_{v \in D} K(\bar{u},v), \tag{C.23}$$

where sup and inf must actually be a min and max. The expression (C.23) means that the function $K(\bar{u}, \cdot)$ attains its minimum over D at $\bar{v}$, while $K(\cdot, \bar{v})$ attains its maximum over C at $\bar{u}$. The relationship between saddle–points and saddle–values are as follows.

Lemma C.2 *[81, pg. 380] Let K be any function from a non-empty product set $C \times D$ to $[-\infty, +\infty]$. A point $(\bar{u}, \bar{v})$ is a saddle–point of K (with respect to maximizing over C and minimizing over D) if and only if the supremum in the expression*

$$\sup_{u \in C} \inf_{v \in D} K(u,v)$$

is attained at $\bar{u}$, the infimum in the expression

$$\inf_{v\in D}\sup_{u\in C} K(u,v)$$

is attained at $\bar{v}$, and these two extrema are equal. If $(\bar{u},\bar{v})$ is a saddle–point, the saddle–value of K is $K(\bar{u},\bar{v})$.

PROOF: If $(\bar{u},\bar{v})$ is a saddle–point then because of Lemma C.1 and (C.23) we have

$$K(\bar{u},\bar{v}) = \inf_{v\in D} K(\bar{u},v) \le \sup_{u\in C}\inf_{v\in D} K(u,v) \le \inf_{v\in D}\sup_{u\in C} K(u,v)$$
$$K(\bar{u},\bar{v}) = \sup_{u\in C} K(u,\bar{v}) \ge \inf_{v\in D}\sup_{u\in C} K(u,v) \ge \sup_{u\in C}\inf_{v\in D} K(u,v)$$

Also, from previous inequalities follows

$$\sup_{u\in C}\inf_{v\in D} K(u,v) \le K(\bar{u},\bar{v}) \le \sup_{u\in C}\inf_{v\in D} K(u,v)$$
$$\Rightarrow K(\bar{u},\bar{v}) = \sup_{u\in C}\inf_{v\in D} K(u,v),$$

and

$$\inf_{v\in D}\sup_{u\in C} K(u,v) \le K(\bar{u},\bar{v}) \le \inf_{v\in D}\sup_{u\in C} K(u,v)$$
$$\Rightarrow K(\bar{u},\bar{v}) = \inf_{v\in D}\sup_{u\in C} K(u,v).$$

So,

$$\inf_{v\in D}\sup_{u\in C} K(u,v) = K(\bar{u},\bar{v}) = \sup_{u\in C}\inf_{v\in D} K(u,v),$$

and the three conditions in the lemma are satisfied.

Conversely, if these three conditions are satisfied then the saddle–value α of K exist and one has

$$\sup_{u\in C} K(u,\bar{v}) = \alpha = \inf_{v\in D} K(\bar{u},v).$$

Obviously,

$$\sup_{u\in C} K(u,\bar{v}) \ge K(\bar{u},\bar{v}) \quad \& \quad \inf_{v\in D} K(\bar{u},v) \le K(\bar{u},\bar{v}).$$

From previous inequalities follows:

$$\alpha = \inf_{v\in D} K(\bar{u},v) \le K(\bar{u},\bar{v}) \le \sup_{u\in C} K(u,\bar{v}) = \alpha.$$

Thus $\alpha = K(\bar{u}, \bar{v})$ and because of (C.23), $(\bar{u}, \bar{v})$ is a saddle–point. ∎

Since the problem of minimizing a real–valued function f over a subset S of $\mathbb{R}^n$ can be expressed conveniently as the problem of minimizing f over all of $\mathbb{R}^n$, if one defines $f(x)$ to be $+\infty$ for every $x \notin S$, a similar technical device is useful in the study of minimax problems.

Remark C.6 *We can extend the function K that is defined on $C \times D$, $C \subset \mathbb{R}^m$, $D \subset \mathbb{R}^n$ beyond $C \times D$ by setting*

$$K(u,v) = \begin{cases} K(u,v) & \text{if} \quad u \in C, \quad v \in D, \\ +\infty & \text{if} \quad u \in C, \quad v \notin D, \\ -\infty & \text{if} \quad u \notin C, \quad v \in D, \\ \text{any value in} \quad [-\infty, +\infty] & \text{if} \quad u \notin C, \quad v \notin D. \end{cases}$$

Then obviously

$$\inf_{v \in \mathbb{R}^n} K(u,v) = \inf_{v \in D} K(u,v) < +\infty, \quad \forall u \in \mathbb{R}^m,$$

where the infima are $-\infty$ if $u \notin C$, and hence

$$\sup_{u \in \mathbb{R}^m} \inf_{v \in \mathbb{R}^n} K(u,v) = \sup_{u \in C} \inf_{v \in D} K(u,v).$$

Similarly,

$$\sup_{u \in \mathbb{R}^m} K(u,v) = \sup_{u \in C} K(u,v) > -\infty, \quad \forall v \in \mathbb{R}^n,$$

where the suprema are $+\infty$ if $v \notin D$, and hence

$$\inf_{v \in \mathbb{R}^n} \sup_{u \in \mathbb{R}^m} K(u,v) = \inf_{v \in D} \sup_{u \in C} K(u,v).$$

In particular, if either the saddle–value of K with respect to $\mathbb{R}^m \times \mathbb{R}^n$ or the saddle–value of K with respect to $C \times D$ exists, then *both exist and are equal.* Furthermore, the saddle–points of K with respect to $\mathbb{R}^m \times \mathbb{R}^n$ are the same as the saddle–points of K with respect to $C \times D$ (if any). Indeed, according to what we have just established, $(\bar{u}, \bar{v})$ satisfies the condition

$$\sup_{u \in \mathbb{R}^m} K(u, \bar{v}) = K(\bar{u}, \bar{v}) = \inf_{v \in \mathbb{R}^n} K(\bar{u}, v)$$

if and only if it satisfies

$$\sup_{u\in C} K(u,\bar{v}) = K(\bar{u},\bar{v}) = \inf_{v\in D} K(\bar{u},v),$$

in which case one necessarily has $(\bar{u},\bar{v}) \in C \times D$.

In what follows, we shall be concerned only with saddle–values and saddle–points of concave–convex (or convex–concave) functions on products of *convex sets*.

Remark C.7 *The observations above allow us to reduce almost everything to the case of concave–convex functions defined on all of $\mathbb{R}^m \times \mathbb{R}^n$. The closedness of such functions is imposed as a natural regularity condition.*

Theorem C.9 *[81, pg. 382] Let K be a closed proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$, and let $C = \mathrm{dom}_1 K$ and $D = \mathrm{dom}_2 K$. Then*

$$\begin{aligned} \sup_{u\in\mathbb{R}^m} \inf_{v\in\mathbb{R}^n} K(u,v) &= \sup_{u\in C} \inf_{v\in D} K(u,v), \\ \inf_{v\in\mathbb{R}^n} \sup_{u\in\mathbb{R}^m} K(u,v) &= \inf_{v\in D} \sup_{u\in C} K(u,v). \end{aligned} \tag{C.24}$$

The saddle–value and saddle–points of K with respect to $\mathbb{R}^m \times \mathbb{R}^n$ are the same as those with respect to $C \times D$.

Proof: For a convex function f on $\mathbb{R}^n$, one has

$$\inf\{f(v) : v \in \mathbb{R}^n\} = \inf\{f(v) : v \in D\},$$

for any set D containing $\mathrm{ri}(\mathrm{dom}\ f)$. For a concave function g on $\mathbb{R}^m$, one has

$$\sup\{g(u) : u \in \mathbb{R}^m\} = \sup\{g(u) : u \in C\},$$

for any set C containing $\mathrm{ri}(\mathrm{dom}\ g)$. Obviously, it follows

$$\begin{aligned} \inf_{v\in\mathbb{R}^n} K(u,v) &= \inf_{v\in D} K(u,v) < +\infty, \quad \forall u \in \mathbb{R}^m, \\ \sup_{u\in\mathbb{R}^m} K(u,v) &= \sup_{u\in C} K(u,v) > -\infty, \quad \forall v \in \mathbb{R}^n, \end{aligned}$$

where the infima are $-\infty$ if $u \notin C$ and suprema are $+\infty$ if $v \notin D$. The desired conclusion follow exactly from these facts. ∎

Corollary C.3 *[81, pg. 383] Let K be a closed proper saddle–function on $\mathbb{R}^m \times \mathbb{R}^n$. If K has a saddle–point, this saddle–point lies in* dom K*, and the saddle–value of K is finite.*

PROOF: Let $(\bar{u}, \bar{v})$ be a saddle–point of K (with respect to $\mathbb{R}^m \times \mathbb{R}^n$). By the Theorem C.9, $(\bar{u}, \bar{v})$ is also a saddle–point with respect to the set $C \times D = \text{dom } K$, so that $(\bar{u}, \bar{v}) \in C \times D$. The saddle–value of K is $K(\bar{u}, \bar{v})$ by Lemma C.2, and this is finite because $K(\bar{u}, \bar{v})$ is finite on dom K. ∎

C.7 Conjugate Saddle–Functions and Minimax Theorems

Questions about saddle–values and saddle–points of concave–convex functions can be reduced essentially to questions about convex programs and their associated Lagrangian problems. In this section we will prove minimax theorems for any concave–convex function and in the next section we will applied that theory to Lagrangian problems.

If F is any convex bifunction from $\mathbb{R}^m$ to $\mathbb{R}^n$, the inverse F_* of F is a concave bifunction (see Definition C.24) from $\mathbb{R}^n$ to $\mathbb{R}^m$, and hence $\langle u^*, F_* x\rangle$ is a concave–convex function of (u^*, x) on $\mathbb{R}^m \times \mathbb{R}^n$ (see Theorem C.5, page 127). The next question is: how is $\langle u^*, F_* x\rangle$ related to $\langle Fu, x^*\rangle$, which similarly is concave–convex in (u, x^*). Namely, by Definition C.21, page 122,

$$\begin{aligned} \langle u^*, F_* x\rangle &= (F_* x)^*(u^*) \\ &= \inf_u \{\langle u, u^*\rangle - (F_* x)(u)\} \\ &= \inf_u \{\langle u, u^*\rangle + (Fu)(x)\}. \end{aligned}$$

If F is closed, we have (see Theorem C.5, page 127)

$$(Fu)(x) = \sup_{x_*} \{\langle x, x^*\rangle - \langle Fu, x^*\rangle\},$$

and consequently

$$\langle u^*, F_* x\rangle = \inf_u \sup_{x_*} \{\langle u, u^*\rangle + \langle x, x^*\rangle - \langle Fu, x^*\rangle\}. \tag{C.25}$$

This reasoning, applied also to the bifunctions F^* and F_*^*, leads to the following basic result.

Theorem C.10 *[81, pg. 389] Let F be closed convex bifunction from $\mathbb{R}^m$ to $\mathbb{R}^n$, and let K be any one of the closed concave–convex functions in the equivalence class $\Omega(F)$ corresponding to F, i.e. any concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$ such that*

$$\langle Fu, x^* \rangle \leq K(u, x^*) \leq \langle u, F^* x^* \rangle, \quad \forall u, \forall x^*.$$

Then for every $u^* \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$,

$$\inf_u \sup_{x^*} \{\langle u, u^* \rangle + \langle x, x^* \rangle - K(u, x^*)\} = \langle u^*, F_* x \rangle, \tag{C.26}$$

$$\sup_{x^*} \inf_u \{\langle u, u^* \rangle + \langle x, x^* \rangle - K(u, x^*)\} = \langle F_*^* u^*, x \rangle. \tag{C.27}$$

On the other hand, let K^* be any one of closed concave–convex functions in the equivalence class $\Omega(F_*)$ corresponding to F_*, i. e. any concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$ such that

$$\langle F_*^* u^*, x \rangle \leq K^*(u^*, x) \leq \langle u^*, F_* x \rangle, \quad \forall u^*, \forall x.$$

Then for every $u \in \mathbb{R}^m$ and every $x_* \in \mathbb{R}^n$,

$$\inf_{u^*} \sup_x \{\langle u, u^* \rangle + \langle x, x^* \rangle - K^*(u^*, x)\} = \langle u, F^* x^* \rangle, \tag{C.28}$$

$$\sup_x \inf_{u^*} \{\langle u, u^* \rangle + \langle x, x^* \rangle - K^*(u^*, x)\} = \langle Fu, x^* \rangle. \tag{C.29}$$

PROOF: This follows from the Definition C.24 of inverse operation and Theorem C.7, page 128. ∎

The saddle–function correspondences in Theorem C.10, can be regarded as a generalization of the conjugacy correspondences for convex or concave functions.

Let K be any concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$. For each $u^* \in \mathbb{R}^m$ and $v^* \in \mathbb{R}^n$,

$$\langle u, u^* \rangle + \langle v, v^* \rangle - K(u, v)$$

is a convex–concave function of (u, v), the sort of function one naturally minimizes in u and maximizes in v.

Definition C.33 The lower conjugate $\underline{K}^*$ *of* K *is defined by*

$$\underline{K}^*(u^*, v^*) = \sup_v \inf_u \{\langle u, u^* \rangle + \langle v, v^* \rangle - K(u, v)\},$$

and the upper conjugate $\overline{K}^*$ *of* K *is defined by*

$$\overline{K}^*(u^*, v^*) = \inf_u \sup_v \{\langle u, u^* \rangle + \langle v, v^* \rangle - K(u, v)\},$$

In the view of *Lemma* C.2,

$$\underline{K}^* \leq \overline{K}^*.$$

Corollary C.4 *Let* K *be any closed concave–convex function on* $\mathbb{R}^m \times \mathbb{R}^n$. *The lower conjugate* $\underline{K}^*$ *of* K *is then a lower closed concave–convex function on* $\mathbb{R}^m \times \mathbb{R}^n$, *and the upper conjugate* $\overline{K}^*$ *of* K *is an upper closed concave–convex function on* $\mathbb{R}^m \times \mathbb{R}^n$. *Moreover,* $\underline{K}^*$ *and* $\overline{K}^*$ *are equivalent, and they depend only on the equivalence class containing* K. *If* K^* *is any closed concave–convex function equivalent to* $\underline{K}^*$ *and* $\overline{K}^*$, *the lower and upper conjugates of* K^* *are in turn equivalent to* K.

Proof: By Theorem C.7, page 128, the equivalence classes in Theorem C.10, page 135 are the most general equivalence classes of closed concave–convex functions. The fact that $\underline{K}^*$ is lower closed and $\overline{K}^*$ is upper closed is deduced by applying Theorem C.6, page 127 to F^* and F^*_*, since for $K \in \Omega(F)$ one has

$$\underline{K}^*(u^*, v^*) = \langle F^*_* u^*, v^* \rangle,$$

$$\overline{K}^*(u^*, v^*) = \langle u^*, F_* v^* \rangle,$$

by Theorem C.10, page 135. ∎

Any saddle–function K^* which is equivalent to both the lower and upper conjugates of a given saddle–function K will simply be called conjugate of K.

Example C.13 *The constant functions* $+\infty$ *and* $-\infty$ *on* $\mathbb{R}^m \times \mathbb{R}^n$ *are closed saddle–functions conjugate to each other. Since these are the only improper closed saddle–functions, a saddle–function conjugate to a closed proper saddle–function must be proper.*

Remark C.8 *The importance of Corollary C.4 for minimax theory is that it reduces the possible discrepancies between "*$\sup\inf$*" and "*$\inf\sup$*" to the possible discrepancies between saddle–functions which are equivalent to each other.*

Corollary C.5 *[81, pg. 391] Let K be a closed proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$, and let $C^* \times D^*$ be the common effective domain of the concave–convex functions conjugate to K. If either* ri C^* *contains the origin of $\mathbb{R}^m$ or* ri D^* *contains the origin of $\mathbb{R}^n$ then*

$$\inf_v \sup_u K(u,v) = \sup_u \inf_v K(u,v). \tag{C.30}$$

If both conditions hold, this saddle–value must be finite.

PROOF: Since by definition

$$\underline{K}^*(0,0) = \sup_v \inf_u \{\langle u,0\rangle + \langle v,0\rangle - K(u,v)\},$$
$$\overline{K}^*(0,0) = \inf_u \sup_v \{\langle u,0\rangle + \langle v,0\rangle - K(u,v)\},$$

one has

$$\inf_u \sup_v K(u,v) = -\underline{K}^*(0,0),$$
$$\sup_v \inf_u K(u,v) = -\overline{K}^*(0,0).$$

The existence of the saddle–value of K depends therefore on the position of $(0,0)$ relative to

$$C^* \times D^* = \operatorname{dom} \underline{K}^* = \operatorname{dom} \overline{K}^*,$$

which proves corollary. ∎

In order to get most use out of the minimax criterion in Corollary C.5, we need a direct characterization of sets C^* and D^* in terms of K. This provided next theorem.

Theorem C.11 *[81, pg. 391] Let K be the closed proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$ with effective domain $C \times D$. Let $C^* \times D^*$ be the common effective domain of the concave–convex functions K^*, conjugate to K. The support functions of C^* and D^* are the given by the formulas*

$$\delta^*(w, D^*) = \sup_{u \in \text{ri } C} \sup_{v \in D} \{K(u, v+w) - K(u,v)\},$$
$$-\delta^*(-z, C^*) = \inf_{v \in \text{ri } D} \inf_{u \in C} \{K(u+z, v) - K(u,v)\}.$$

PROOF: Let F be the unique closed proper convex bifunction from $\mathbb{R}^m$ to $\mathbb{R}^n$ such that

$$(\mathrm{cl}_2\ K)(u,v) = \langle Fu, v\rangle, \quad \forall u, \forall v$$

(see Theorem C.7, page 128). We have $C = \mathrm{dom}\ F$. Since the equivalence class of saddle–functions conjugate to K corresponds to F_*, we have $D^* = \mathrm{dom}\ F_*$. Let G be the effective domain of the graph function of F, i.e.

$$G = \{(u,x) : (Fu)(x) < +\infty\}.$$

We have

$$D^* = \{x : \exists u, (u,x) \in G\} = \bigcup\{\mathrm{dom}\ Fu : u \in C\}.$$

In fact,

$$\mathrm{ri}\ D^* = \{x : \exists u, (u,x) \in \mathrm{G}\} = \bigcup\{\mathrm{ri}(\mathrm{dom}\ Fu) : u \in \mathrm{ri}\ C\}.$$

Therefore

$$\begin{aligned}
\delta^*(w|D^*) &= \sup\{\langle x, w\rangle : x \in \mathrm{ri}\ D^*\} \\
&= \sup\{\langle x, w\rangle : x \in \mathrm{ri}(\mathrm{dom}\ Fu), u \in \mathrm{ri}\ C\} \\
&= \sup\{\delta^*(w|\mathrm{dom}\ Fu) : u \in \mathrm{ri}\ C\}.
\end{aligned}$$

On the other hand , for each $u \in \mathrm{ri}\ C$, $K(u,\cdot)$ is a closed proper convex function with effective domain D (see Theorem C.8, page 128), and hence it agrees with $(\mathrm{cl}_2\ K)(u,\cdot)$, which is the conjugate of the closed proper convex function Fu. The support function of dom Fu is the recession function of the conjugate of Fu (see Theorem C.4, page 121). Thus, for each $u \in \mathrm{ri}\ C$, $\delta^*(\cdot, \mathrm{dom}\ Fu)$ is the recession function of $K(u,\cdot)$, and we have

$$\delta^*(w|\mathrm{dom}\ Fu) = \sup\{K(u, v+w) - K(u,v) : v \in D\}$$

by the first recession function formula (see Theorem C.3, page 120). This proves the formula for $\delta^*(\cdot, D^*)$. The proof of the formula for $\delta^*(\cdot, C^*)$ is similar. ■

Corollary C.6 *[81, pg. 392] In the notation of the theorem, one has* $0 \in \mathrm{int}\ D^*$ *if and only if the convex functions* $K(u,\cdot)$ *for* $u \in \mathrm{ri}\ C$ *have no common direction of recession. Similarly, one has* $0 \in \mathrm{int}\ C^*$ *if and only if the convex functions* $-K(\cdot, v)$ *for* $v \in \mathrm{ri}\ D$ *have no common direction of recession.*

PROOF: One has $0 \notin \text{int } D^*$ if and ony if there exist a vector $w \neq 0$ such that $\delta^*(w, D^*) \leq 0$, i.e. (according to the preceding proof)

$$K(u, v + w) - K(u, v) \leq 0, \quad \forall v \in D, \quad \forall u \in \text{ri } C.$$

Since the effective domain of $K(u, \cdot)$ is D for every $u \in \text{ri } C$, the latter condition means (see Definition C.19, page 121) that w belongs to the recession cone of $K(u, \cdot)$ for every $u \in \text{ri } C$. The proof of the other part of the corollary is analogous. ■

The main theorem about the existence of saddle–values may now be stated.

Theorem C.12 *[81, pg. 393] Let K be a closed proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$ with effective domain $C \times D$. Then either of the following conditions implies that the saddle–value of K exists. If both conditions hold, the saddle–value must be finite.*
(a) The convex functions $K(u, \cdot)$ for $u \in \text{ri } C$ have no common direction of recession.
(b) The convex functions $-K(\cdot, v)$ for $v \in \text{ri } D$ have no common direction of recession.

PROOF: This is simply combines Corollary C.5 and Corollary C.6. Namely, if the convex functions $K(u, \cdot)$ for $u \in \text{ri } C$ have no common direction of recession, then from Corollary C.6 follows that $0 \in \text{int } D^*$ and then from Corollary C.5 follows (C.30). Analogous for the convex functions $-K(\cdot, v)$. ■

Corollary C.7 *[81, pg. 393] Let K be a closed proper concave–convex function on $\mathbb{R}^m \times \mathbb{R}^n$ with effective domain $C \times D$. If either C or D is bounded, the saddle–value of K exists.*

PROOF: The effective domain of $K(u, \cdot)$ is D for every $u \in \text{ri } C$ (see Theorem C.8, page 128). The condition (a) is fulfilled when D is bounded. Similarly, condition (b) is fulfilled when C is bounded. ■

The saddle–value of K with respect to $\mathbb{R}^m \times \mathbb{R}^n$ in Theorem C.12 and Corollary C.7 is of course the same as the saddle value of K with respect to $C \times D$. To emphasize this, we state as a special case:

Corollary C.8 *Let C and D be non–empty closed convex sets in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and let K be a continuous finite concave–convex function on $C \times D$. If either C or D is bounded, one has*

$$\inf_{v\in D} \sup_{u\in C} K(u,v) = \sup_{u\in C} \inf_{v\in D} K(u,v).$$

PROOF: Apply the preceding corollary to the lower (or upper) simple extension of K to all of $\mathbb{R}^m \times \mathbb{R}^n$, which is a closed proper concave–convex function with effective domain $C \times D$. ∎

C.8 Saddle–Points of Lagrange Function

Suppose we have the following convex minimization problem:

$$\begin{aligned} \min f(x), &\qquad x \in \mathbb{R}^n, \\ \langle a_i, x\rangle = b_i, &\qquad i = 1,\dots,m \quad \text{or} \quad [Ax = b \in \mathbb{R}^m] \\ c_j(x) \le 0, &\qquad j = 1,\dots,p \quad \text{or} \quad [c(x) \le 0 \in \mathbb{R}^m]. \end{aligned} \tag{C.31}$$

Let us now repeat the following well known theorem:

Theorem C.13 *[42, page 306] For $\bar{x} \in \mathbb{R}^n$ consider the following statements:*
(a) $\bar{x} \in \mathbb{R}^n$ solves the constrained minimization problem (C.31);
(b) there exist $\lambda = (\lambda_1,\dots,\lambda_m) \in \mathbb{R}^m$ and $\mu = (\mu_1,\dots,\mu_p) \in \mathbb{R}^p$ such that

$$0 \in \partial f(\bar{x}) + \sum_{i=1}^{m} \lambda_i a_i(\bar{x}) + \sum_{j=1}^{p} \mu_j \partial c_j(\bar{x}) = \partial_x L(\lambda,\mu,x), \tag{C.32}$$

$$\mu_j \ge 0 \quad \text{and} \quad \mu_j c_j(\bar{x}) = 0 \quad \text{for} \quad j = 1,\dots,p. \tag{C.33}$$

The previous two statements are equivalent.

The condition $0 \in \partial_x L(\lambda,\mu,x)$ is called *the minimality condition*. The existence satisfying (C.32) and (C.33) in Theorem C.13 is called *Lagrange*, or *Karush–Kuhn–Tucker* conditions (KKT). Actually, Lagrange derived them in

the case of equality constraints only, and for differentiable data. The corresponding coefficients $(\lambda, \mu) \in \mathbb{R}^m \times (\mathbb{R}^+)^p$ are called the *Lagrange multipliers.*

After the general study of previous sections, let us consider the particular case where K is the Lagrange function associated with the basic minimization problem (C.31).

According to the previous sections, the notation in this section is the following:

$$\begin{aligned} D &= \mathbb{R}^n, \quad \text{with variable} \quad v = x, \\ C &= \mathbb{R}^m \times (\mathbb{R}^+)^p, \quad \text{with variable} \quad u = (\lambda, \mu), \\ K = L &: \mathbb{R}^m \times (\mathbb{R}^+)^p \times \mathbb{R}^n \to \mathbb{R} \quad \text{is defined by,} \\ L(\lambda, \mu, x) &:= f(x) + \lambda^T (Ax - b) + \mu^T c(x), \end{aligned} \tag{C.34}$$

where L is the *Lagrangian* for convex minimization problem (C.31). In this Section, we are interested in saddle–points of L.

Namely, the definition of a saddle–point (see Definition C.32, page 130), can be reformulated accordingly:

Proposition C.1 *With the system of notation as above, the saddle–points of L on $[\mathbb{R}^m \times (\mathbb{R}^+)^p] \times \mathbb{R}^n$ are those $(\bar\lambda, \bar\mu, \bar x)$ such that:*
(i) $\bar x$ minimizes $L(\bar\lambda, \bar\mu, \cdot)$ on $\mathbb{R}^n$,
(ii) $A\bar x = b$ and $c_j(\bar x) \le 0$ for $j = 1, \dots, p$, ($\bar x$ is feasible in (C.31))
(iii) $\bar\mu_j c_j(\bar x) = 0$ for $j = 1, \dots, p$.

Proof: According to the definition of the saddle point (see Definition C.32), the saddle–point $(\bar\lambda, \bar\mu, \bar x)$ of L satisfies

$$\max_{(\lambda,\mu) \in \mathbb{R}^m \times (\mathbb{R}^+)^p} L(\lambda, \mu, \bar x) = L(\bar\lambda, \bar\mu, \bar x) = \min_{x \in \mathbb{R}^n} L(\bar\lambda, \bar\mu, x). \tag{C.35}$$

(i) is second part of (C.35):

$$L(\bar\lambda, \bar\mu, \bar x) = \min_{x \in \mathbb{R}^n} L(\bar\lambda, \bar\mu, x).$$

The first half of (C.35) expresses that $(\bar\lambda, \bar\mu)$ solves he optimization problem

$$\max\{L(\lambda, \mu, \bar x) : (\lambda, \mu) \in \mathbb{R}^m \times (\mathbb{R}^+)^p\}.$$

Since $L(\cdot,\cdot,\bar{x})$ can be maximized separately with respect to each λ_i and each μ_j, a solution $(\bar{\lambda},\bar{\mu})$ is then characterized as follows:

$$\begin{cases} \langle a_i,\bar{x}\rangle - b_i = 0 \quad i=1,\ldots,m \\ \\ c_j(\bar{x}) = \begin{cases} 0 \quad \text{if} \quad \bar{\mu}_j > 0 \\ \\ \text{nonpositive} \quad \text{if} \quad \bar{\mu}_j = 0 \end{cases} \end{cases}$$

Altogether, we recover (ii) and (iii). ■

Corollary C.9 *[81] If $((\bar{\lambda},\bar{\mu}),\bar{x})$ is a saddle point of L over $[\mathbb{R}^m \times (\mathbb{R}^+)^p] \times \mathbb{R}^n$, then $\bar{x}$ solves the basic minimization problem (C.31).*

PROOF: If $(\bar{\lambda},\bar{\mu},\bar{x})$ is a saddle–point of L, (i), (ii) and (iii) hold in Proposition C.1. Also $\bar{x}$ is feasible. Furthermore, (i), (iii) give:

$$f(\bar{x}) = L(\bar{\lambda},\bar{\mu},\bar{x}) \leq L(\bar{\lambda},\bar{\mu},x) \quad \forall x \in \mathbb{R}^n. \tag{C.36}$$

It suffices to observe that $\bar{\mu} \in (\mathbb{R}^+)^p$ implies

$$L(\bar{\lambda},\bar{\mu},x) = f(x) + \bar{\lambda}^T(Ax-b) + \bar{\mu}^T c(x) \leq f(x) + \bar{\mu}^T c(x) \leq f(x), \tag{C.37}$$

for all feasible x. From (C.36) and (C.37) follows

$$f(\bar{x}) \leq f(x),$$

for all feasible x an the corollary is proved. ■

It is worth nothing that the above two results are very general. Namely, no assumption whatsoever is needed on the data. However, as mentioned already, existence of a saddle point of L is a very restrictive property.

Theorem C.14 *In the basic minimization problem (C.31), assume that the data f and $c_j, j=1,\ldots,p$ are convex functions from $\mathbb{R}^n$ to $\mathbb{R}$. Then two statements below are equivalent:*
(a) $(\bar{\lambda},\bar{\mu},\bar{x})$ is a saddle–point of L over $[\mathbb{R}^m \times (\mathbb{R}^+)^p] \times \mathbb{R}^n$,
(b) $\bar{x}$ solves (C.31) and $(\bar{\lambda},\bar{\mu})$ is a Lagrange multiplier.

PROOF: Use Proposition C.1 and the minimality condition $0 \in \partial_x L(\bar{\lambda},\bar{\mu},\bar{x})$. then recognize in $(\bar{\lambda},\bar{\mu})$ the definition of Lagrange multipliers, as given in Theorem C.13. ■

Remark C.9 *Given that our basic constrained minimization problem (C.31) is assumed to have a solution $\bar{x}$, the existence of Lagrange multipliers is thus equivalent to the existence of a saddle–point of L.*

Summary and Outlook

In this thesis we study Semidefinite Programming (SDP) relaxations for Quadratic Assignment Problem (QAP), and methods for computing optimal and approximative solutions of these relaxations.

In Chapter 1 we gave some basic notation and preliminary concepts.

Chapter 2 covered the main aspects concerning duality and interior point methods in Semidefinite Programming. First, we introduce the standard formulation of a primal semidefinite program and derive its dual. After detailed explanation of the duality theory, nondegeneracy and strict complementarity, we explain the concept of the interior-point approach. We have also specified three search directions from Monteiro–Zhang family of search direction.

Chapter 3 presented the Quadratic Assignment Problem; the nature of the problem, history, computational complexity, and applications. We gave also several equivalent formulations of QAP. In the last Section of Chapter 3 we explained a Convex Quadratic Programming Relaxation that was introduced by Anstreicher and Brixius [4]. By using this relaxations Anstreicher et. al made recently a break-through by solving a number of previously unsolved large QAPs.

In Chapter 4 we derived several SDP relaxations for QAP with the increasing levels of complexity, placed in the matrix space $\mathcal{S}_{n^2+1}$. The presented SDP relaxations were already proposed in the literature, but as it is shown in Chapter 7 we can approximate them within reasonable computational time. In Chapter 4 we proved linearly dependency of some of the constraints in the relaxations. Solutions of the Nugent-type problems for $n < 20$ are computed on our PC (Athlon XP 1800 or Intel P4), and for $n \geq 20$ by use of the NEOS Server for Optimization.

In Chapter 5 we derived the new relaxations in the space $\mathcal{S}_{(n-1)^2+1}$, which

is smaller then for the standard SDP relaxations for QAP. These relaxations are based on the new representation of the permutation matrix described in Section 5.1. Computational results showed that these relaxations are weaker then relaxations in $\mathcal{S}_{n^2+1}$ but faster for computing.

In Chapter 6 is explained the concept of the Bundle Method on the general minimization problem. The differences between 'classical use' of the bundle method and our approach are also explained. The practical details arised form our computational experiences are presented, too.

Chapter 7 explained the bundle approach on the QAP_{R_3} relaxation. Computed bounds for the QAP_{R_3} relaxation are currently the strongest available for QAP. In the Chapter 7 are also presented the lower bounds for some QAPLIB instances in the first and second level of the branching tree. Obtained results show a promise for implementing QAP_{R_3} on B&B framework.

Summing this work up, it proved again that semidefinite programming is very power-fool tool for approximating some NP–hard problems. Even though these relaxations might be computationally very expensive, the quality of the bounds justifies this tradeoff. We showed that the Bundle Method is is a very favorable method for solving large combinatorial optimization problems. It turned out that the combination od SDP relaxations and the Bundle Method is currently the most efficient approach for obtaining strong bounds for QAP.

Further work?
In the Chapter 7 are given some results on utilizing our bounds for Branch and Bound settings. These results are promising, and we believe that further work on this issue is leading to computations of optimal solutions for large Quadratic Assignment Problems. The main problem that here appear, on which we have already worked, is finding the optimal branching strategy.

In the thesis we presented new relaxations that are placed in the lifted space which is smaller then for the standard SDP relaxations. Bounds computed from these relaxations are weak, but these relaxations allows to exploit sparsity in a simple way and can be computed considerably fast. We expect to improve these SDP relaxations by including an additional set of the constraints.

As part of the follow–up research, we would like to investigate the possible application of the new representation of permutation matrices to other problems or relaxations that contain these matrices, and to check the effectives of the approach.

Index of Authors

Index of Topics

Bibliography

[1] W. P. Adams and T. A. Johnson. Improved Linear Programming–Based Lower Bounds for the Quadratic Assignment Problem. In Proceedings of the DIMACS Workshop on Quadratic Assignment Problems, *DIMACS Series in discrete Mathematics and Theoretical Computes Sciences*, American Mathematical Society, 16:43-75, 1994.

[2] F. Alizadeh, J.-P. A. Haeberly, and M.L. Overton. Primal-Dual Interior-Point Methods for Semidefinite Programming: Stability, Convergence, and Numerical Results. *SIAM Journal on Optimization*, 8(3):746-768, 1998.

[3] F. Alizadeh, J.-P. A. Haeberly, and M.L. Overton. Complementarity and Nondegeneracy in Semidefinite Programming. *Mathematical Programming*, 77:111-128, 1997.

[4] K. Anstreicher and N. Brixius. A New Bound for the Quadratic Assignment Problem Based on Convex Quadratic Programming. *Mathematical Programming*, 89:341–357, 2001.

[5] K. Anstreicher and N. Brixius. Solving Quadratic Assignment Problems Using Convex Quadratic Programming Relaxations. *Optimization Methods and Software* 16:49-68, 2001.

[6] K. Anstreicher, N. Brixius, J.–P. Goux, and J. Linderoth. Solving Large Quadratic Assignment Problems on Computational Grids. *Mathematical Programming B*, 91:563–588, 2002.

[7] K. Anstreicher and H. Wolkowicz. On Lagrangian Relaxation of Quadratic Matrix Constraints. *SIAM Journal on Matrix Analysis and its Applications*, 22:41–55, 2000.

[8] V. I. Arnold. On Matrices Depending on Parameters. *Russian Mathematical Surveys*, 26:29–43, 1971.

[9] F. Barahona and R. Anbil. The Volume Algorithm: Producing Primal Solutions with a Subgradient Algorithm. *Mathematical Programming*, 87: 385-399, 2000.

[10] F. Barahona and L. Ladanyi. Branch and Cut Based on the Volume Algorithm: Steiner Trees in Graphs and Max-Cut. IBM report RC22221, October 2001.

[11] R. Bellman and K. Fan. On Systems of Linear Inequalities in Hermitian Matrix Variables. In *Convexity*, volume 7 of *Proceedings of Symposia in Pure Mathematics, American Mathematical Society*, 7:1–11, 1963.

[12] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Optimization Numérique*, Springer–Verlag, 1997.

[13] S. Boyd, L. El Gahaoui, E. Feron, and V. Balakrishnan. Linear Matrix Inequalities in System and Control Theory, volume 15 of *SIAM Studies in Applied Mathematics*. Society of Industrial and Applied Mathematics (SIAM), Philadelphia, PA 19101, USA, 1994.

[14] R. E. Burkard, S. Karisch and F. Rendl. QAPLIB – A Quadratic Assignment Problem Library. *European Journal of Operational Research*, 55:115–119, 1991.

[15] R. E. Burkard and J. Offerman. Entwurf von Schreibmaschinentastaturen mittels Quadratischer Zuordnungsprobleme. *Operations Research*, 21:B121–B132, 1977.

[16] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D.F. Shanno. Higher–Order Predictor Corrector Interior Point Methods with Application to Quadratic Objectives. *SIAM Journal on Optimization*, 3(4):696–725, 1993.

[17] F. Çela. *The Quadratic Assignment Problem: Theory and Algorithms.* Kluwer, Massachessets, USA, 1998.

[18] N. Christofides and M. Gerrard. A Graph Theoretic Analysis of Bounds for the Quadratic Assignment Problem. In P. Haansen, editor, *Studies on Graphs and Discrete Programming*, North-Holland, 61–68, 1981.

[19] J. Clausen and M. Perregaard. Solving Large Quadratic Assignment Problems in Parallel. *Computational Optimization and Applications*, 8:111–127, 1997.

[20] J. W. Dicky and J. W. Hopkins. Campus Building Arrangement Using TOPAZ. *Transportation Research*, 6:59–68, 1972.

[21] R. J. Duffin. Infinite Programs, in: H. W. Kuhn and A. W. Tucker eds., *Linear Inequalities and Related Systems*, Annals of Mathematical Studies vol. 38, Princeton University Press, 157–170, 1956.

[22] C. S. Edwards. The Derivation of a Greedy Approximator for the Koopmans-Beckmann Quadratic Assignment Problem. *Proceedings of the 77-th Combinatorial Programming Conference (CP77)*, 55–86, 1977.

[23] C. S. Edwards. A Branch and Bound Algorithm for the Koopmans-Beckmann Quadratic Assignment Problem. *Mathematical Programming Study*, 13:35–52, 1980.

[24] A. N. Elshafei. Hospital Layout as a Quadratic Assignment Problem. *Operations Research Quarterly*, 28:167–179, 1977.

[25] B. Fares and D. Noll and P. Apkarian. Robust Control via Sequential Semidefinite Programming. *SIAM Journal on Control and Optimization*, 40:1791–1820, 2002.

[26] G. Finke, R. E. Burkard and F. Rendl. Quadratic Assignment Problems. *Annals of Discrete Mathematics*, 31:61–82, 1987.

[27] I. Fischer, G. Gruber, F. Rendl and R. Sotirov. The Bundle Method in Combinatorial Optimization. University of Klagenfurt, Austria. in progress, 2003.

[28] I. Foster and C. Kesselman. Computational Grids. In I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, CA, 1999.

[29] M. Frank and P. Wolfe. An Algorithm for Quadratic Programming. *Naval Research Logistic Quarterly*, 3:95–110, 1956.

[30] A. M. Geoffrion and G. W. Graves. Scheduling Parallel Production Lines with Changeover Costs: Practical Applications of a Quadratic Assignment/LP Approach. *Operations Research*, 24:595–610, 1976.

[31] J. Gondzio. Presolve Analysis of Linear Programs Prior to Applying an Interior Point Method. *INFORMS Journal on Computing*, 9(1):73–91, 1997.

[32] A. Graham. Kronecker Products and Matrix Calculus with Applications, Mathematics and its Applications. *Ellis Horwood Limited, Chichester*, 1981.

[33] M. Grötschel, L. Lovász, and A. Schrijver. The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica*, 1:169–197, 1981.

[34] S. W. Hadley, F. Rendl, and H. Wolkowicz. A New Lower Bound via Projection for the Quadratic Assignment Problem. *Mathematics of Operations Research*, 17:727–739, 1992.

[35] P. M. Hahn, T. Grant and N. Hall. A Branch–and–Bound Algorithm for the Quadratic Assignment Problem Based on the Hungarian Method. *European Journal of Operational Research*, 108:629–640, 1998.

[36] P. M. Hahn, W. L. Hightower, T. A. Johnson, M. Guignard–Spielberg and C. Roucairol. Tree Elaboration Strategies for Branch-and-Bound Solution of the Quadratic Assignment Problem. *Yugoslav Journal of Operations Research*, 11:41-60, 2001.

[37] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl. A Spectral Approach to Bandwidth and Separator Problems in Graphs. *Linear and Multilinear Algebra*, 39: 73–90, 1995.

[38] C. Helmberg. An Interior Point Method for Semidefinite Programming and Max–Cut Bounds. *Technical University*, Graz, dissertation, 1994.

[39] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An Interior-Point Method for Semidefinite Programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[40] C. Helmberg and F. Rendl. A Spectral Bundle Method for Semidefinite Programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[41] D. R. Heffley. Assigning Runners to a Relay Team. In S. P. Ladany und R. E. Machol, editors, *Optimal Strategies in Sports*, North–Holland, Amsterdam, 169–171, 1977.

[42] J. B. Hiriart–Urruty and C. Lemaréchal. Convex Analysis and Minimization Algorithms I. *Springer Verlag*, 1991.

[43] J. B. Hiriart–Urruty and C. Lemaréchal. Convex Analysis and Minimization Algorithms II. *Springer Verlag*, 1991.

[44] J. B. Hiriart–Urruty. *Optimisation et Analyse Convexe*, Presses Universitaires de France, 1998.

[45] A.J. Hoffman and H.W. Wielandt. The Variation of the Spectrum of a Normal Matrix. *Mathematical Journal*, 20:37–39, 1953.

[46] R. A. Horn and C. R. Johnson. Matrix Analysis. *Cambridge University Press*, New York, 1991.

[47] S. E. Karisch. Nonlinear Approaches for Quadratic Assignment and Graph Partition Problems. Disertation. *Technical University Graz*, 1995.

[48] S. E. Karisch, E. Çela, J. Clausen, and T. Espersen. A Dual Framework for Lower Bounds of the Quadratic Assignment Problem Based on Linearization. *Computing*, 63:351–403, 1999.

[49] S. Kim and B. Um. Polyak's Subgradient Method with Simplified Projection for Nondifferentiable Optimization with Linear Constraints. *Optimization*, 20(4):451-456, 1989.

[50] S. Kim and S. Koh. On Polyak's Improved Subgradient Method. *Journal of Optimization Theory and Applications*, 57(2):355-360, 1988.

[51] E. de Klerk, C. Ross, and T. Terlaky. Initialization in Semidefinite Programming via a Self-Dual Skew-Symmetric Embedding. *Operations Research Letters*, 20:213–221, 1997.

[52] Donald E. Knuth. Estimating the Efficiency of Backtrack Programs. *Mathematics of Computation*, 29(129):212–136, 1975.

[53] J. Krarup. Quadratic Assignment. *Saertryk af data*, 3–27, 1972.

[54] J. Krarup and P. M. Pruzan. Computer-Aided Layout Design. *Mathematical Programming Study*, 9:75–94, 1978.

[55] M. Kojima, S. Shindoh, and S. Hara. Interior–Point Methods for the Monotone Semidefinite Linear Complementarity Problem in Symmetric Matrices. *SIAM Journal on Optimization*, 7:86–125, 1997.

[56] T. C. Koopmans and M. J. Beckmann. Assignment Problems and the Location of Economics Activities. *Econometrica*, 25:53–76, 1957.

[57] G. Laporte and H. Mercure. Balancing Hydrulic Turbine Runners: A Quadratic Assignment Problem. *European Journal of Operations Research*, 35:378–382, 1988.

[58] J. B. Lasserre. An Explicit Exact SDP Relaxation for Nonlinear 0-1 Programs. *SIAM Journal on Optimization*, 12:756–769, 2002.

[59] M. Laurent. *Lower Bound for the Number of Iterations in Semidefinite Relaxations for the Cut Polytope.* Preprint, CWI, Amsterdam, The Netherlands, July 2002. To appear in Mathematics of Operations Research.

[60] M. Laurent and F. Rendl. Semidefinite Programming and Integer Programming. Technical report PNA-R0210, CWI, Amsterdaym, 2002. To appear as chapter of the Handbook on Discrete Optimization edited by K.Aardal, G. Nemhauser and R. Weismantel.

[61] L. Lovász. On the Shannon Capacity of a Graph. *IEEE Transactions on Information Theory*, 25(1):1-7, 1979.

[62] H. Lüthkpohl. Hanbook of Matrices. *John Wiley & Sons Ltd* England, 1996.

[63] R. Mifflin. An Algorithm for Constrained Optimization with Semismooth Functions. *Mathematics of Opeartions Research*, 2:191–207, 1977.

[64] R. D. C. Monteiro. Primal-Dual Path Following Algorithm for Semidefinite Programming. *SIAM Journal on Optimization*, 7:663–678, 1997.

[65] R. D. C. Monteiro and P. Zanjacomo. A Note on the Existence of the Alizadeh-Haeberly-Overton Direction for Semidefinite Programming. *Mathematical Programming* 78::393–396, 1997.

[66] R. D. C. Monteiro. Polynomial Convergence of Primal-dual Algorithms for Semidefinite Programming Based on the Monteiro and Zhang Family of Directions. *SIAM Journal on Optimization*, 8:797–812, 1998.

[67] R. D. C. Monteiro and Y. Zhang. A Unified Analysis for a Class of Path-Following Primal-Dual Interior-Point Algorithms for Semidefinite Programming. *Mathematical Programming*, 81:281–299, 1998.

[68] Yu. E. Nesterov and A. S. Nemirovskii. Interior Point Polynomial Algorithms in Convex Programming. *SIAM Publications*, SIAM Philadelphia, USA, 1994.

[69] Yu. E. Nesterov and M. J. Todd. Self-Scaled Barriers and Interior-Point Methods in Convex Programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[70] Yu. E. Nesterov and M. J. Todd. Primal-Dual Interior Point Methods for Self-Scaled Cones. *SIAM Journal on Optimization*, 8:324–364, 1998.

[71] J. von Neumann. Some Matrix Inequalities and Metrization of Matrix Space. *Tomsk Univ. Rev.* 1:286–300, 1937. (reprinted in: John von Neumann: Collected Works, vol 4. A.H. Taub ed., MacMilan, 205–219, 1962.)

[72] M. A. Pollatschek, H. Gershoni, and Y. T. Radday. Optimization of the Typewriter Keyboard by Computer Simulation. *Angewandte Informatik*, 10:438–439, 1976.

[73] F. A. Potra and R. Sheng. Superlinear Convergence of a Predictor–Corrector Method for Semidefinite Programming Without Shrinking Central Path Neighborhood. *Reports on Computational Mathematics*, 91, 1998. Department of Mathematics, University of Iowa.

[74] M. Ramana, L. Tuncel, and H. Wolkowicz. Strong Duality for Semidefinite Programming. *SIAM Journal on Optimization*, 7(3):641-662, 1997.

[75] F. Rendl and H. Wolkowicz. Applications of Parametric Programming and Eigenvalue Maximization to the Quadratic Assignment Problem. *Mathematical Programming*, 53:63–78, 1992.

[76] F. Rendl, R. Sotirov and H. Wolkowicz. A Simplified/Improved HKM Direction for Certain Classes of Semidefinite Programming. Submitted to *European Journal of Operational Research*, 2002.

[77] F. Rendl and R. Sotirov. Bounds for the Quadratic Assignment Problem Using the Bundle Method. In U. Derigs, H. W. Hamacher, and U. Zimmermann, editors, *Discrete Optimization Methods and Applications*. Braunschweig–Kaiserslautern-Köln, 287–305, 2003.

[78] F. Rendl and R. Sotirov. The Bundle Method Applied to QAP. In *SOR'01 Proceedings of the* 6^{th} *International Symposium on Operational Research*, Preddvor, Slovenia, September 26–28, 2001, 95–101.

[79] F. Rendl and R. Sotirov. Bounds for the Quadratic Assignment Problem Using the Bundle Method. University of Klagenfurt, Austria (working paper).

[80] 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. *Operations Research*, 43(5):63–78, 1992.

[81] R. T. Rockafellar. *Convex Analysis*, Princeton University Press, 1970.

[82] S. Sahni and T. Gonzales. P–Complete Approximation Problems. *Journal of the ACM*, 23:555–565, 1976.

[83] D. Schlegel. Die Unwucht-Optimale Verteilung von Turbinenschaufeln als Quadratisches Zuordnungsprobem. Ph.D. Thesis. *ETH Zürich*, Scwitzerland, 1987.

[84] M. Shida, S. Shindoh, and M. Kojima. Existence and Uniqueness of Search Directions in Interior-Point Algorithms for the SDP and the Monotone SDLCP. *SIAM Journal on Optimization*, 8:387–398, 1998.

[85] H. Schramm and J. Zowe. A Version of the Bundle Idea for Minimizing a Nonsmooth Function: Conceptional Idea, Convergence Analysis, Numerical Results. *SIAM Journal on Optimization*, 2:121–152, 1992.

[86] R. Sotirov and H. Wolkowicz. Solving Lovász theta function Problems Using an Inexact Gauss–Newton Method. Technical Report in progress, University of Waterloo, Waterloo, Canada, 2002.

[87] L. Steinberg The Backboard Wiring Problem: A Placement Algorithm *SIAM Review*, 3:37–50, 1961.

[88] M. J. Todd. A Study of Search Directions in Primal–Dual Interior–Point Methods for Semidefinite Programming. *Optimization Methods and Software*, 11:1–46, 1999.

[89] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd Direction in Semidefinite Programming. *SIAM Journal on Optimization*, 8:769–796, 1998.

[90] Ph.L. Toint. On Sparse and Symmetric Matrix Updating Subject to a Linear Equation. *Mathematics of Computation*, 31:954–961, 1977.

[91] L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–95, 1996.

[92] I. Ugi, J. Bauer, J. Brandt, J. Friedrich, J. Gasteiger, C. Jochum, and W. Schubert. Neue Anwendungsgebiete für Computer in der Chemie. *Angewandte Chemie*, 91:99–111, 1979.

[93] P. Wolfe. A Method of Conjugate Subgradients for Minimizing Nondifferentiable Functions. *Mathematical Programming Study*, 3:145–173, 1975.

[94] H. Wolkowicz. Some Applications of Optimization in Matrix Theory. *Linear Algebra and its Applications*, 40:101-118, 1981.

[95] S. J. Wright. Primal–Dual Interior-Point Methods. *SIAM*, Philadelphia, 1997.

[96] Y. Zhang. On Extending Some Primal-Dual Interior-Point Algorithms from Linear Programming to Semidefinite Programming. *SIAM Journal on Optimization*, 8:365–386, 1998.

[97] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite Programming Relaxations for the Quadratic Assignment Problem. *Journal of Combinatorial Optimization*, 2:71-109, 1998.

[98] J. Zowe. Nondifferentiable Optimization – a Motivation and a Short Introduction into the Subgradient – and the Bundle Concept. NATO ASI Series, 15 *Computational Mathematical Programming* edited by K. Schittkowski, Springer–Verlag, Berlin Heidelberg 1985.