ADMM for SDP Relaxation of GP

by

Hao Sun

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

Waterloo, Ontario, Canada, 2016

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

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

Abstract

We consider the problem of partitioning the set of nodes of a graph G into k sets of given sizes in order to minimize the cut obtained after removing the k-th set. This is a variant of the well-known *vertex separator problem* that has applications in e.g., numerical linear algebra. This problem is well studied and there are many lower bounds such as: the standard eigenvalue bound; projected eigenvalue bounds using both the adjacency matrix and the Laplacian; quadratic programming (QP) bounds derived from imitating the (QP) bounds for the quadratic assignment problem; and semidefinite programming (SDP) bounds. For the quadratic assignment problem, a recent paper of [8] had great success from applying the ADMM (altenating direction method of multipliers) to the SDP relaxation. We consider the SDP relaxation of the vertex separator problem and the application of the ADMM method in solving the SDP. The main advantage of the ADMM method is that optimizing over the set of doubly non-negative matrices is about as difficult as optimizing over the set of positive semidefinite matrices. Enforcing the non-negativity constraint gives us a clear improvement in the quality of bounds obtained. We implement both a high rank and a nonconvex low rank ADMM method, where the difference is the choice of rank of the projection onto the semidefinite cone. As for the quadratic assignment problem, though there is no theoretical convergence guarantee, the nonconvex approach always converges to a feasible solution in practice.

Acknowledgements

I would like to express my heartfelt thanks to my supervisors Prof. Henry Wolkowicz and Prof. William Cook. Prof. Wolkowicz has given me a lot guidance since my days as an Undergraduate Research Assistant. He has inspired my interest and broadened my horizons in optimization. From him I learned a lot on how to think critically about research and how to ask the right questions. Prof. Cook introduced me to and cultivated my interest in the interplay between polytopes, formulations and optimization. He has been an invaluable guide in the world of academia and helped me make many connections. I would also like to thank the various instructors that have taught me over the years, their passion and enthusiasm seems to have rubbed off on me. I would like to thank my friends and classmates for providing an academic environment and many enlightening mathematical discussions. I would also like to thank the faculty of mathematics and the department of combinatorics and optimization for providing my financial support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

We consider a special variant of the *minimum cut problem, (MC)*, recently studied in [9, 11]. The problem consists in partitioning the node set of a given graph into k sets of given sizes in order to *minimize the cut* obtained by removing the k-th set. To elaborate, we are given an undirected graph $G = (V, E)$ with n vertices and a partition of n into k parts $m_1,\ m_2\ \ldots\ m_k$, and wish to partition the vertices of G into sets $S_1,\ S_2,\ ,..,\ S_k$ with cardinality $|S_i| = m_i$ for all i, such that *the cut* is minimal. The cut refers to the cardinality of the set of edges between the different sets S_1 to S_{k-1}. We omit the edges involving the last set S_k, which will henceforth be referred to as a *vertex separator*.
This problem is known to be NP-hard in general [6, 11]. When $m_k = 0$ we refer to this as the *graph partitioning problem*. This problem has been studied intensely in the literature. It has applications in computer program segmentation, solving symmetric systems of equations, microchip design and circuit board, floor planning and other layout problems [10].
To give a more detailed example, Rendl, Lisser and Piacentini [11] describe an approach to solve sparse symmetric systems of equations that uses vertex separator. Given an $n \times n$ symmetric matrix M, we associate a graph $G = (V, E)$ with vertices $\{1, 2, \ldots, n\}$ and edges $E := \{\{i, j\} : \quad M_{i,j} \neq 0\}$. The algorithm using vertex separator solves the problem on G for m_k small and $m_1, m_2, \ldots, m_{k-1}$ evenly partitioning $n - m_k$. One wishes to get a small cut, particularly if the cut is 0, then it suffices to do the eliminations involving the last set and the blocks $S_1, S_2, \ldots S_{k-1}$ and the blocks themselves. This allows sparse systems to be solved much more quickly. There are various approaches to finding bounds for the vertex separator problem based on relaxations. The standard relaxations find lower bounds using eigenvalue bounds, quadratic programming and semidefinite programming. For the quadratic assignment problem (QAP) a recent paper showed great success in using the al-

ternating direction method of multipliers (ADMM) for solving a semidefinite relaxation of the QAP [8]. Here we discuss the use of the ADMM for solving a semidefinite relaxation of vertex separator. Of particular interest here is that previous SDP codes have had practical difficulty enforcing the non-negativity constraint, so much so that certain vertex separator codes such as the ones in [9] do not enforce non-negativity. In the ADMM code however, this constraint is very cheap to enforce, which gives us a significant improvement in the quality of bounds obtained over codes that do not enforce non-negativity.

1.1 Outline

This thesis is organized as follows: The remainder of this chapter serves as an introduction to semi-definite programming in order to make this thesis self contained. Chapter 2 should serve as an introduction to the vertex separator problem and some of the work that has been done that we do not improve upon in this thesis. Chapter 3 introduces the standard semi-definite programming formulation of vertex separator. Chapter 4 contains the main contribution of this thesis, the application of ADMM to solve the semi-definite relaxation of vertex separator and the use of the low rank ADMM method to generate solutions to the vertex separator problem. Chapter 5 contains the computational results of this thesis. We conclude and describe possible future work in Chapter 6.

1.2 Semidefinite Programming

This section serves to include sufficient background for semidefinite programs so as to make this thesis self contained. A reader who is familiar with this topic may wish to skip this section. A semidefinite program is a max/min function of finitely many variables subject to linear equality constraints, linear inequality constraints and positive semidefiniteness of linear expressions of the variables. This section is loosely based on Levent Tuncel's CO 671 course and the textbook [14].

Denote: $[n]:= \{1,2,3, \dots ,n\}$, I_n identity matrix of size n . For a vector v and matrix M, $\mathrm{Diag}(v)$ is the matrix with v on the diagonal and zeros elsewhere and $\mathrm{diag}(M)$ is the vector of the diagonal entries of M, for $N \in \mathbb{R}^{n\times n}$ $\mathrm{tr}(N) = \sum_{i=1}^{n} N_{i,i}$ is the trace of N.

Definition 1.2.1. *A symmetric matrix $Y \in \mathcal{S}^n$ is called positive semidefinite (PSD) if $\forall v \in \mathbb{R}^n$, we have $v^TYv \geq 0$. We will denote the set of all positive semidefinite matrices by $\mathcal{S}^n_+$. Y is called positive definite (PD) if $\forall x \in \mathbb{R}^n \backslash \{0\}$, we have $v^TYv > 0$. We will denote the set of all positive definite matrices by $\mathcal{S}^n_{++}$.*

Lemma 1.2.2. *Any symmetric matrix $Y \in \mathcal{S}^n$ can be written diagonalized with respect to an orthonormal basis. i.e. $Y = VDV^T$ for D diagonal and $V^TV = I_n$. The diagonal elements of D are the eigenvalues of Y with multiplicities.*

Remark 1.2.3. *If $Y \in \mathcal{S}^n_+$ then for any matrix $B \in \mathbb{R}^{k\times n}$, $BYB^T \in S^n_+$*

Proposition 1.2.4. *For $Y \in \mathcal{S}^n$, the following are equivalent:*

1. *Y is positive semidefinite; that is $Y \in \mathcal{S}^n_+$;*
2. *for some $L \in \mathbb{R}^{n\times n}$, $Y = LL^T$ (It is possible to choose L lower triangular, such that this is true. This is known as a Cholesky decomposition);*
3. *$\lambda_j(Y) \geq 0, \quad \forall j \in \{1, 2, \dots, n\}$;*
4. *there exist $\gamma_i \in \mathbb{R}_+$ and $u^{(i)} \in \mathbb{R}^n$, $\forall i \in \{1, 2, \dots, n\}$ such that*

   $$Y = \sum_{i=1}^{n} \gamma_i u^{(i)} {u^{(i)}}^T;$$

   When the $u^{(i)}$ are orthonormal, this is known as the spectral decomposition of Y.

5. *$\forall S \in \mathcal{S}^n_+, \quad \langle Y, S\rangle \geq 0$;*
6. *for every nonempty $J \subseteq \{1, 2, \dots, n\}$, $\det(Y_J) \geq 0$, where $Y_J := \{[X_{ij}] : i, j \in J\}$;*

Proof. We will prove the equivalence of (1)-(5) and refer the reader to [14] for the technical proof of 6.

1. (1)$\Leftrightarrow$ (3)
   First suppose that Y is positive semidefinite and let $Y = VDV^T$ be a diagonalization (as in Lemma 1.2.2) of Y if $D_{i,i} < 0$ for some i, then $V_{:,i}^T Y V_{:,i} = D_{i,i} < 0$ contradicting positive semidefiniteness of Y.
   Conversely, suppose that $\lambda_j(Y) \geq 0, \forall j \in \{1, 2, \dots, n\}$. This implies that in any diagonalization of $Y = VDV^T$, D has all positive diagonal entries. Thus we can write $D = \text{Diag}([\sqrt{D_1}, \sqrt{D_2} \dots \sqrt{D_n}])^2$.
   Thus $v^TYv = (\text{Diag}([\sqrt{D_1}, \sqrt{D_2} \dots \sqrt{D_n}]V^Tv)^T(\text{Diag}([\sqrt{D_1}, \sqrt{D_2} \dots \sqrt{D_n}]V^Tv) \geq 0$.

2. (1) $\Rightarrow$ (2)
   Let $Y \in \mathcal{S}_+^n$, define a a function $\theta : \mathcal{S}^n \to \{(l,p) : 1 \le l,p \le n\} \cup \{0\}$ as follows:
   If Y is diagonal then define $\theta(Y) = 0$. Otherwise, Y has a nonzero entry $Y_{i,j}$ such that $i > j$ and $Y_{j,1:i-1}$ has no nonzero entries except possibly $Y_{j,j}$ and $\forall l < j$, $Y_{l,:}$ has no non zero entries, except possibly at $Y_{l,l}$. Define $\theta(Y) = (i,j)$. To put it intuitively, $\theta(Y)$ is the next entry we would eliminate in Gaussian elimination on Y. Define $\succ$ on $\{(l,p) : 1 \le l,p \le n\} \cup \{0\}$ by

   (1) $0 \prec (l,p)\ \forall l,p$
   (2) $(l,p) \prec (q,r)$ if $p < r$
   (3) $(l,p) \prec (q,p)$ if $l < q$

   Fix n and we will prove the existence of the Cholesky decomposition by structural induction on $\theta(Y)$ with respect to $\succ$.
   Base case: $\theta(Y) = 0$ in this case Y is diagonal, and we are done.
   Inductive step: here we will apply one row iteration of Gaussian elimination followed by the same column operation. We claim that this preserves symmetry and PSD as well as reducing $\theta(Y)$ with respect to $\succ$.
   Let $\theta(Y) = (i,j)$ then $Y_{i,i}, Y_{j,j} \neq 0$ by PSD of Y. We do the row operation of subtracting $\frac{Y_{i,j}}{Y_{j,j}}$ of row j from row i. Denote the elementary row operation matrix for this operation by P. We then do the column operation of subtracting $\frac{Y_{i,j}}{Y_{j,j}}$ of column j from column i. It is clear the the elementary column operation matrix is given by P^T. Call this new matrix we get $Z = PYP^T$. We can see that after the application of these operations $Z_{j,1:i-1}$ has no nonzero entries except possibly $Z_{j,j}$ and $\forall l < j$, $Y_{l,:}$ has no non zero entries, except possibly at $Y_{l,l}$. As well $Z_{i,j} = 0$, thus we conclude that $\theta(Y) \succ \theta(Z)$. Recall Remark 1.2.3:
   If $Y \in \mathcal{S}_+^n$ then for any matrix B, $BYB^t \in S_+^n$.
   Thus $Z = PYP^T$ is PSD. So by induction Z has a Cholesky factorization $Z = LL^T$. Recall that elementary row operation matrices are invertible and if P is lower triangular, then P^{-1} is upper triangular(and vice versa). Then $Y = (LP^{-1})(LP^{-1})^T$ is a Cholesky factorization of Y.

3. (2) $\Rightarrow$(1) If $Y = LL^T$ then $\forall v \in \mathbb{R}^n$, $v^T Y v = (Lv)^T (Lv) \ge 0$.

4. (4) $\Rightarrow$(1) If $Y = \sum_{i=1}^n \gamma_i u^{(i)} {u^{(i)}}^T$, then $\forall v \in \mathbb{R}^n$, $v^T Y v = \sum_{i=1}^n \gamma_i (v^T u^{(i)})^2 \ge 0$.

5. (3)$\Rightarrow$ (4) Let $Y = VDV^T$ be the spectral decomposition of Y then $Y = \sum_{i=1}^n D_{i,i} V_{:,i}^T V_{:,i}^T$.

6. (5) $\Rightarrow$ (1) Let $v \in \mathbb{R}^n$ Then $v^T Y v = \operatorname{tr} v^T Y v = \operatorname{tr} Y(v^T v V) \geq 0$.
7. (2) $\Rightarrow$ (5) Let $S \in \mathcal{S}$ and $Y = LL^T$ we know $S = BB^T$ for some B then $\operatorname{tr} SY = \operatorname{tr}(B^T L)(B^T L)^T = \sum_{i,j=1}^n (B^T L)_{i,j}^2 \geq 0$.

$\square$

Remark 1.2.5. *Because the determinant of a matrix is continuous, property 6 of Proposition 1.2.4 says that the set of positive semidefinite matrices is a closed set in $\mathbb{R}^{n\times n}$ as it is the intersection of finitely many closed sets (under the usual metric topology).*

Proposition 1.2.6 (Equivalent definitions of PD matrices)**.** *Let $A \in \mathcal{S}^n$. Then the following are equivalent:*

1. *Y is positive definite;*
2. *there exists $L \in \mathbb{R}^{n\times n}$ nonsingular such that $Y = LL^T$ (here, B can be chosen as a lower triangular matrix-the Cholesky decomposition of A);*
3. *$\lambda_i(Y) > 0, \forall i \in \{1, 2, \dots, n\}$;*
4. *there exist $\gamma \in \mathbb{R}_{++}^n$ and $h^{(i)} \in \mathbb{R}^n$, $\forall i \in \{1, 2, \dots, n\}$ linearly independent such that $Y = \sum_{i=1}^n \gamma_i u^{(i)} {u^{(i)}}^T$;*
5. *$\forall S \in \mathcal{S}_+^n \backslash \{0\}, \operatorname{tr} YS > 0$;*
6. *for $[k] := \{1, 2, \dots, k\}, \}, \det(Y_{[k],[k]}) > 0$;*
7. *$Y \succeq 0$ and $\operatorname{rank}(Y) = n$.*

Proof. The proofs of the equivalence of (1)-(5) are basically modified versions of the proofs for their PSD counterparts.
(3) $\iff$ (7)

Remark 1.2.7. *The rank of an n by n PSD matrix Y is the number of positive eigenvalues in the spectral decomposition $Y = VDV^T$, further, by permuting the rows and columns, we may assume $D_{1,1} \geq D_{2,2} \geq .. \geq D_{l,l} \geq 0 = D_{l+1,l+1} = ..D_{n,n}$. Then $Y = V_{:,1:l} D_{1:l,1:l} V_{:,1:l}^T$. This is called the compact spectral decomposition of Y.*

This remark shows (3) $\iff$ (7) we again refer the reader to [14] for the proof of 6. $\square$

Lemma 1.2.8 ([2]). *For a symmetric n by n matrix T*
$\lambda_n(T) = \min_{v \in \mathbb{R}^n:\ \ \|v\|=1} v^t T v.$

Proof. Let $T = VTV^T$ be the spectral decomposition of T with $D_{1,1} \geq D_{2,2} \geq \ldots \geq D_{n.n}$. We see that $\lambda_n(T)$ can be attained by setting $v = V_{:,n}$. Now we prove it is optimal. Since V has full rank $v = \sum_{i=1}^{n} \gamma_i V_{:,i} \quad 1 = \|v\| = v^t v = (i = 1^n \gamma_i V_{:,i})^T (\sum_{i=1}^{n} \gamma_i V_{:,i}) = \sum_{i=1}^{n} \gamma_i^2$ by noting V has orthonormal columns.
Also $v^T T v = \sum_{i=1}^{n} D_i \gamma_i^2 \leq D_n = \lambda_n(T)$ as desired. □

Corollary 1.2.9. *For a symmetric matrix n by n T*
$\lambda_1(T) = \max_{v \in \mathbb{R}^n:\ \ \|v\|=1} v^t T v.$

Theorem 1.2.10 (Gerschgorin Disks)**.** *Let $M \in \mathbb{R}^{n \times n}$. Then the union of disks $B_j(M) := \{\lambda \in \mathbb{C} : |\lambda - M_{j,j}| \leq \sum_{i \neq j} |M_{j,i}|\}$ covers all the eigenvalues of M.*

Proof. Let v be an eigenvector of M with eigenvalue λ. Let v_i be the largest entry of v in magnitude. The equation $Mv = \lambda v$ says on the ith row that

$$\lambda v_i = \sum_{j=1}^{n} M_{i,j} v_j.$$

Solving for v_i

$$(\lambda - M_{i,i}) v_i = \sum_{j \neq i} M_{i,j} v_j.$$

Take absolute value of both sides and note the triangle inequality.

$$\begin{aligned} (|\lambda - M_{i,i}|)|v_i| &= |\textstyle\sum_{j \neq i} M_{i,j} v_j| \\ &\leq \textstyle\sum_{j \neq i} |M_{i,j}||v_j| \\ &\leq \textstyle\sum_{j \neq i} |M_{i,j}||v_i|. \end{aligned}$$

Dividing by $|v_i|$ yields the desired result.

$$(|\lambda - M_{i,i}|) \leq \sum_{j \neq i} |M_{i,j}|.$$

□

Definition 1.2.11. $Y \in \mathcal{S}^n$. *is called* diagonally dominant *if* $Y_{ii} \geq \sum_{j \neq i} |Y_{ij}|$, *for every* $1 \leq i \leq n$. *Similarly,* Y *is called* strictly diagonally dominant *if* $Y_{ii} > \sum_{j \neq i} |Y_{ij}|$, *for every* $1 \leq i \leq n$.

Corollary 1.2.12. *A diagonally dominant matrix is positive semidefinite. A strictly diagonally dominant matrix is positive definite.*

Notice that the Laplacian matrix L of a graph is diagonally dominant.

Corollary 1.2.13. *The Laplacian matrix of a graph is positive semidefinite.*

Lemma 1.2.14. *Let $X \in \mathcal{S}^n$ and $T \in \mathcal{S}^m_{++}$. Then*

$$Y := \begin{pmatrix} T & U^T \\ U & X \end{pmatrix} \succeq 0 \text{ if and only if } X - UT^{-1}U^T \succeq 0.$$

Moreover, $Y \succ 0$ if and only if $X - UT^{-1}U^T \succ 0$.

Proof. Consider the following decomposition of Y:

$$\begin{pmatrix} I & 0 \\ UT^{-1} & I \end{pmatrix} \begin{pmatrix} T & 0 \\ 0 & X - UT^{-1}U^T \end{pmatrix} \begin{pmatrix} I & T^{-1}U^T \\ 0 & I \end{pmatrix} = \begin{pmatrix} T & U^T \\ U & X \end{pmatrix}.$$

Denote

$$P = \begin{pmatrix} I & 0 \\ UT^{-1} & I \end{pmatrix}. \tag{1.1}$$

Since Y is lower triangular and has non-zero entries on the diagonal, P is non-singular. Therefore by noting remark 1.2.3,

$$Y \succeq 0 \iff X - UT^{-1}U^T \succeq 0.$$

Also,

$$Y \succ 0 \iff X - UT^{-1}U^T \succ 0.$$

□

1.2.1 Inner Product and Norms

A (real) inner product $\langle , \rangle : \mathbb{R}^t \times \mathbb{R}^t \to \mathbb{R}$ is a function from $\mathbb{R}^t \times \mathbb{R}^t$ to $\mathbb{R}$ satisfying

1. positivity

$$\langle X, X \rangle \geq 0 \text{ and } \langle X, X \rangle = 0 \text{ if and only if } X = 0.$$

2. linearity:
$$\langle \alpha X, Y\rangle = \alpha\langle X, Y\rangle \text{ and } \langle X+Z, Y\rangle = \langle X, Y\rangle + \langle Z, Y\rangle.$$

3. Symmetry:
$$\langle X, Y\rangle = \langle Y, X\rangle.$$

Let us define our inner product on matrices by $\langle , \rangle : \mathbb{R}^{n\times m} \times \mathbb{R}^{n\times m} \to \mathbb{R}$ by $\langle A, B\rangle = \operatorname{tr} A^T B$

Proposition 1.2.15. *Our inner product $\langle , \rangle$ satisfies (1)-(3) of 1.2.1.*

Proof. We observe that:

1. Positivity follows from $< B, B >= \operatorname{tr} B^T B = \sum_{i,j=1}^n B_{i,j}^2 \geq 0$, and this sum can only be 0 if B is 0.

2. Linearity follows from linearity of the trace function and matrix multiplication.

3. $\operatorname{tr} Y = \operatorname{tr} Y^T$ thus $\langle A, B\rangle = \operatorname{tr} A^T B = \operatorname{tr} B^T A = \langle B, A\rangle$

□

Remark 1.2.16. *It is worth remembering that $\langle A, B\rangle = \sum_{i,j=1}^n A_{i,j} B_{i,j}$. In this sense, it is the standard inner product on $\mathbb{R}^{n^2}$. We will often think of $\mathbb{R}^{n\times n}$ as $\mathbb{R}^{n^2}$ without being explicit about it.*

Definition 1.2.17. *A set $C \subseteq \mathbb{R}^n$ is* convex*, if for every $x, y \in C$ and every $\lambda \in [0, 1]$, we have $\lambda x + (1-\lambda)y \in C$.*

It is canon to refer to the set $\{\lambda x + (1-\lambda)y : \lambda \in [0, 1]\}$ in the above definition as the *line segment* of x and y. Thus another way to define convex is: C is convex, if the line segment of every two points of C also lies in C.

Definition 1.2.18. *A convex set C with the property that $\forall \alpha \geq 0 \quad , \ x \in C$, we have $\alpha x \in C$ is called a cone. A cone is called pointed if $x, -x \in C$ implies that $x = 0$*

Definition 1.2.19. *Given a cone C in an inner product space T with inner product $\langle , \rangle$ we define the dual cone C^* as*
$C^* = \{s \in T \quad \forall x \in C \quad \langle s, x\rangle \geq 0\}$.

Lemma 1.2.20. *Hyperplane Separation Theorem*
For two closed convex sets $X, Y \in \mathbb{R}^n$ with $X \cap Y = \emptyset$ there exists a hyperplane separating them, that is:
$\exists a \in \mathbb{R}^n, \quad b \in \mathbb{R}$ *such that* $\langle a, x\rangle > b > \langle a, y\rangle \quad \forall x \in X \ , \ y \in Y.$

Proof. By closure, there exist points $x \in X, \quad y \in Y$ such that $\|x - y\|$ is minimal. By disjointedness this distance is not zero. Denote $a = y - x$ $b_1 = \langle a, x\rangle, \quad b_2 = \langle a, y\rangle$ $H = \{p : \langle a, p\rangle = \frac{b_1+b_2}{2}\}$ Assume for a contradiction that H does not separate X from Y, then by convexity, H intersects one of X, Y without loss of generality assume it intersects X at a point z. Consider $\frac{d}{dt}\|y-(x+t(z-x)\|^2 = -2\langle(y-x),(z-x)\rangle = -(b_2-b_1) < 0$ which means we can find a closer point to y by moving from x to z a small amount contradicting our assumption that $\|x - y\|$ was minimal. □

Theorem 1.2.21. *For a closed cone C, $(C^*)^* = C$*

Proof. $C \subset C^{**}$ is clear. Let $x \notin C$ then by the Hyperplane Separation Theorem, there exists a, b such that $\langle a, x\rangle < b < \langle a, c\rangle \quad \forall c \in C$. $0 \in C$ so $b \leq 0$ if $\langle a, c\rangle < 0$ for any c, then we can obtain a contradiction by scaling c. So $\langle a, c\rangle \geq 0 \ \forall c \in C$ or $a \in C^*$ and $\langle a, x\rangle < b \leq 0$ so $x \notin C^{**}$ □

Definition 1.2.22. *A cone C is called* self-dual *if C=C^*.*

Remark 1.2.23. *The set of positive semidefinite matrices forms a closed pointed self-dual cone (self dual with respect to our trace inner product). Self dual comes from property 5 of Proposition 1.2.4.*

Theorem 1.2.24. *Let $X, Y \succeq 0$. Then $\langle X, Y\rangle = 0$ if and only if $XY = 0$.*

Proof. Suppose $XY = 0$. Then $\langle X, Y\rangle = \text{trace}(XY) = \text{trace}(0) = 0$.

Now suppose $X, Y \succeq 0$ and $\langle X, Y\rangle = 0$. Then $\langle X, Y\rangle = \text{trace}(XY) = \text{trace}(X^{1/2}YX^{1/2}) = 0$. Since $Y \succeq 0$ and $X^{1/2}$ is symmetric matrix, we have $X^{1/2}YX^{1/2} \succeq 0$. So $\lambda(X^{1/2}YX^{1/2}) \geq 0$. Since $\text{trace}(X^{1/2}YX^{1/2}) = 0$, we have $\lambda(X^{1/2}YX^{1/2}) = 0$. It implies that

$$0 = X^{1/2}YX^{1/2} = X^{1/2}Y^{1/2}\left(X^{1/2}Y^{1/2}\right)^T.$$

So $X^{1/2}Y^{1/2} = 0$. Then

$$XY = X^{1/2}(X^{1/2}Y^{1/2})Y^{1/2} = 0.$$

□

Now we talk about norms on $\mathcal{S}^n$.

Definition 1.2.25. *A norm $\|\|$ on an vector space E is a function $\|\| : E \to \mathbb{R}$ satisfying:*

1. $\|X\| > 0,\ \forall X \neq 0$ *and* $X = 0$ *if and only if* $X = 0$.
2. $\|\alpha X\| = |\alpha|\|X\|$.
3. $\|X + Y\| \leq \|X\| + \|Y\|$ *(triangle inequality).*

We will only be using the *2-norm* $\|v\|_2 = \sqrt{\sum_{i=1}^n v_i^2}$ for vectors in $\mathbb{R}^n$ and *Frobenius norm*
$\|M_F\|_F := \sqrt{\sum_{i,j}(M_{ij})^2}$ for matrices.
When the context is clear, we will omit the subscripts on the norms.

Lemma 1.2.26. *Let $Y \in \mathcal{S}^n$. Then $\|Y_F\| = \|H\|_F^{1/2} = \|\lambda(Y)\|_2$. Where* $\lambda(Y) = \begin{bmatrix} \lambda_1(Y) \\ \lambda_2(Y) \\ \vdots \\ \lambda_n(Y) \end{bmatrix}$

Proof. Consider the spectral decomposition of Y, $Y = VDV^T$
$\|Y\|^2 = \operatorname{tr} Y^T Y = \operatorname{tr} V^T V D V^T V D = \operatorname{tr} DD = \lambda(Y)^T \lambda(Y)$ □

Definition 1.2.27. *For a linear operator $\mathcal{L}$, $\mathcal{L} : U \to \mathbb{V}$, where U and V are vector spaces, we define the adjoint of $\mathcal{A}$ as the linear operator*
$\mathcal{L}^* : \mathbb{V} \to U$ such that

$\langle \mathcal{L}^*(v), u\rangle = \langle v, L(U)\rangle, \forall X \in \mathcal{S}^n, \forall u \in U.$

Example 1.2.28. *Note if we set $U = V = \mathbb{R}^n$ then the adjoint is just the transpose.*

Definition 1.2.29. *Let $X \in \mathbb{R}^{n\times k}$.* $\operatorname{vec}(X)$*, the vector formed by vertically "stacking" the columns of X, is denoted as*

$$\operatorname{vec}(X) = \begin{bmatrix} X_{:,1} \\ X_{:,2} \\ \vdots \\ X_{:,k}, \end{bmatrix}.$$

Definition 1.2.30. *Let $v \in \mathbb{R}^{nk}$ we define*

$$\operatorname{Mat}(v) = \begin{bmatrix} v_{1:n} & v_{n+1:2n} & v_{2n+1:3n} & \dots v_{(k-1)n+1:kn} \end{bmatrix}. \tag{1.2}$$

Mat *maps nk-dimensional vectors to $n \times k$ matrices.*

Example 1.2.31. vec *is a linear mapping. The adjoint, as well as the inverse mapping of* vec() *is* Mat.

1.2.2 Duality

For a cone K let us define the cone optimization problem for K and linear function $c : K \to \mathbb{R}$

$$\begin{array}{lll} \text{(P)} & \inf & \langle c, x\rangle \\ & \text{s.t.} & \mathcal{A}(x) = b \\ & & x \in K. \end{array}$$

$$\begin{array}{lll} \text{(D)} & \sup & b^T y \\ & \text{s.t.} & \mathcal{A}^*(y) + z = c \\ & & z \in K^*. \end{array}$$

Let $C \in \mathcal{S}^n, b \in \mathbb{R}^t$ and a linear transformation $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^t$ be given. Then we define a primal SDP in standard form as:

$$\begin{array}{llccc} \text{(P)} & \inf & \operatorname{tr} CX & & \\ & \text{s.t.} & \mathcal{A}(X) & = & b, \\ & & X & \succeq & 0 \end{array}$$

And the dual as:

$$\begin{array}{llccc} \text{(D)} & \sup & b^T y & & \\ & \text{s.t.} & \mathcal{A}^*(y) + S & = & C, \\ & & S & \succeq & 0 \end{array}$$

Note $\langle U, V\rangle = \operatorname{tr} UV$ is an inner product. Recall that given any inner product $\langle\rangle_t$ and any linear function $\mathcal{L} : \mathcal{S}^n \to \mathbb{R}$, $\mathcal{L}$ can be written as an inner product $\mathcal{L}(S) = \langle W, S\rangle_t$ for some $W \in \mathcal{S}^n$. Thus we can write $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^T$ as

$$\mathcal{A}(X) = \begin{bmatrix} \operatorname{tr} A_1 X \\ \operatorname{tr} A_2 X \\ \vdots \\ \operatorname{tr} A_t X \end{bmatrix}$$

for some matrices $A_1, A_2, \dots A_n \in \mathcal{S}^n$. We can then rewrite the primal and dual SDP as:

$$\begin{array}{llrcl} \text{(P)} & \inf & \langle C, X\rangle & & \\ & \text{s.t.} & \langle A_i, X\rangle & = & b_i, \quad \forall i \in \{1, 2, \cdots, m\} \\ & & X & \succeq & 0. \end{array}$$

$$\begin{array}{llrcl} \text{(D)} & \sup & b^T y & & \\ & \text{s.t.} & \sum_{i=1}^m y_i A_i + S & = & C, \\ & & S & \succeq & 0. \end{array}$$

Theorem 1.2.32. (Weak Duality Theorem for SDP) *If* $(\tilde{X}, (\tilde{y}, \tilde{Z}))$ *are feasible to (P) and (D),respectively, then* $\langle C, \tilde{X}\rangle - b^T \bar{y} = \langle \tilde{X}, \tilde{Z}\rangle \geq 0$.

Proof. By simple algebra

$$\begin{array}{rl} & \langle C, \tilde{X}\rangle - b^T \tilde{y} \\ = & \langle C, \tilde{X}\rangle - \mathcal{A}(\tilde{X})^T \tilde{y} \\ = & \langle C, \tilde{X}\rangle - \langle \mathcal{A}(\tilde{X})\tilde{y}\rangle \\ = & \langle C, \tilde{X}\rangle - \langle \mathcal{A}^*(\tilde{y}), \tilde{X}\rangle \\ = & \langle C - \mathcal{A}^*(\tilde{y}), \tilde{X}\rangle \\ = & \langle \tilde{Z}, \tilde{X}\rangle \geq 0, \end{array}$$

as $\tilde{X}, \tilde{Z} \in \mathcal{S}^n_+$. □

$\langle \tilde{Z}, \bar{X}\rangle$ is called the *duality gap* of $(\tilde{X}, (\tilde{y}, \tilde{S}))$.

Definition 1.2.33. *In 1.2.2* $\hat{X}$ *is called a* Slater point *for* (P) *if it is feasible for* (P) *and* $\hat{X} \succ 0$. *In 1.2.2* $(\bar{y}, \bar{S})$ *is called a Slater point for* (D) *if it is feasible and* $\bar{S} \succ 0$.

Theorem 1.2.34 ([14])**.** (Strong Duality Theorem for SDP) *Suppose (D) has a Slater point. If the objective value of (D) is bounded from above then (P) attains its optimum value and the optimum values of (P) and (D) coincide.*

Corollary 1.2.35. *[14, Corollary 2.17] If both (P) and (D) have Slater points, then both optima are attained and they agree.*

1.2.3 Faces of the SDP cone

A ray of $\mathbb{R}^n$ is a set of the form $\{\lambda v : \lambda \in \mathbb{R}_+\}$ for some $v \in \mathbb{R}^n \backslash \{0\}$.

Definition 1.2.36. *Given a set S we will use* $\text{conv}(S)$ *to denote the smallest convex set containing S.*

Definition 1.2.37. *Given a set S we will use cone(S) to denote the smallest cone containing S.*

Recall that for a convex set P,

Definition 1.2.38. *v is an extreme point of P if $v \in P$ and there do not exist points $u, w \in P$ with $u, w \neq v$ and $0 \leq \lambda \leq 1$ such that $v = \lambda u + (1 - \lambda w)$.*

Another way to state the above is to say that v is an extreme point of P if there do not exist points $u, w \in P$ with $u, w \neq v$ such that $\{v\} \subseteq \text{conv}(\{u, w\})$.

Definition 1.2.39. *An extreme ray of a cone K is a ray $R \subseteq K$ such that there do not exist $R_1, R_2 \subseteq K$ such that $R \subseteq R_1 + (1 - \lambda)R_2$.*

Definition 1.2.40. *Given a cone K we will denote the set of all extreme rays of K by* $\text{ext}(K)$.

Example 1.2.41. *Let H be a hyperplane in $\mathbb{R}^n$ not going through the origin. Let P be a convex set contained in H. $v \in P$ is an extreme point of P if and only if cone$(v) = \{av : a \in \mathbb{R}_+\}$ is an extreme ray of cone(P).*

Definition 1.2.42. *For sets $A_1, A_2 \subseteq \mathbb{R}^n$. We define the* Minkowski Sum *of A_1 and A_2 as $A_1 + A_2 = \{s_1 + s_2 : s_1 \in A_1, s_2 \in A_2\}$.*

Remark 1.2.43. *The Minkowski Sum of two cones C_1, C_2 is a cone*

Theorem 1.2.44. $\text{ext}(\mathcal{S}_+^n) = \{xx^T : x \in \mathbb{R}^n, \|x\| = 1\}$.

Proof. Let $C = \{aY : \ a \in \mathbb{R}_+\}$ be an extreme ray of $\mathcal{S}_+^n$.

Let $Y = \sum_{i=1}^{T} \lambda_i(Y) v_i v_i^T$, be the compact spectral decomposition of Y, where v_i is the normalized eigenvector of X corresponding to the i-th largest eigenvalue α_i.

Suppose for a contradiction that $t := \text{rank}(Y) > 1$. Write

$$Y = \sum_{i=1}^{T} \lambda_i(Y) v_i v_i^T.$$

Let $C_1 = \textit{cone}(v_1 v_1^T)$ and $C_2 = \textit{cone}(\sum_{i=2}^{T} \lambda_i(Y) v_i v_i^T) : \lambda \geq 0\}$. C_1 and C_2 are both rays, and $C \subseteq C_1 + C_2$. But, $C \neq C_1$ and $C \neq C_2$, contradiction. □

Definition 1.2.45. *Let C be a convex set. A face $F \subseteq C$ of C is a set such that $\forall u, v \in C$ $0 < \lambda < 1$ if $\lambda u + (1-\lambda)v \in F$, then $u, v \in F$. We will denote F is a face of C by $F \lhd C$.*

Definition 1.2.46. *A face K of C is a* proper face *of C if $\{0\} \subsetneq K \subsetneq C$.*

Remark 1.2.47. *In the above definition if C is a cone then F is a face of C if and only if $u + v \in F$ implies that $u \in F$ or $v \in F$.*

Definition 1.2.48. *A face F of $C \subseteq \mathbb{R}^n$ is called* exposed *if there exists $a \in \mathbb{R}^n, b \in \mathbb{R}$ such that*

$$F = \{x \in C : \langle a, x \rangle = b\} \text{ and } C \subseteq \{y \in \mathbb{R}^n : \langle a, x \rangle \leq b\}.$$

A set of the form

$$\{y \in \mathbb{R}^n : \langle \alpha, x \rangle \leq \beta\}$$

containing C is called a supporting halfspace *of C and the corresponding set*

$$\{y \in \mathbb{R}^n : \langle \alpha, x \rangle = \beta\}$$

is called the supporting hyperplane *.*
Thus a face F is exposed if it is the intersection of C with one of its supporting hyperplanes.

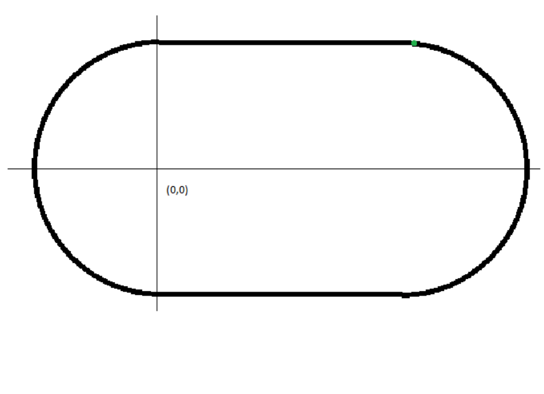
For polytopes and polyhedral cones, every face is exposed, but this is not true for convex sets in general.

Example 1.2.49. *Let*

$$C = \{(x, y) : \quad x^2 + y^2 \leq 1\} \cup \{(x-2)^2 + y^2 \leq 1\} \cup [0, 2].$$

Now we can intuitively see that (2,1) is a face of C, but it is not exposed, see Figure 1.1.

Figure 1.1: Illustration of a nonexposed face in a nonpolyhedral set



Theorem 1.2.50. *The faces of the SDP satisfy the following:*

1. *Any nonempty face F of $\mathcal{S}^n_+$ can be written as*

   $$F = VS^n_+V^T := \{VYV^T : \quad Y \in S^n_+\}$$

   for some $n \times k$ matrix V.

2. *Any nonempty face F of $\mathcal{S}^n_+$ can be written as*
   $\{Z \in S^n_+ : \quad \operatorname{tr} SZ = 0\}$.

Remark 1.2.51. *Every proper face F of $\mathcal{S}^n_+$ is exposed.*

Proof. This follows from (2) of the above theorem. □

Remark 1.2.52 ([14])**.** *While the faces of the SDP cone are all exposed, the feasible set of an SDP program need not be.*

1.2.4 Kronecker Product

Definition 1.2.53. *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$. We define the Kronecker product to be*

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

Proposition 1.2.54. *Let A, B, C, D, X be matrices, and n, k positive integers.*

1. $(A \otimes B)^T = A^T \otimes B^T$.
2. *If the products AC, BD are compatible then so is $(A\otimes B)(C\otimes D)$ and $(A\otimes B)(C\otimes D) = AC \otimes BD$.*
3. *For $A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{k\times k}, \quad X \in \mathbb{R}^{n\times k}$ $\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X)$.*
4. *For $A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{k\times k}, \quad X \in \mathbb{R}^{n\times k}$ $\operatorname{trace}(AXBX^T) = \operatorname{vec}(X)^T(B \otimes A)\operatorname{vec}(X)$.*

Remark 1.2.55. *(2) above says in particular that for vectors $v \in \mathbb{R}^n$, $u \in \mathbb{R}^k$ and matrices*

$$A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{k\times k},$$

we have that

$$A \otimes B(v \otimes u) = (Av) \otimes (Bu).$$

Lemma 1.2.56. *For $u^{(1)}, u^{(2)}, \dots u^{(m)} \in \mathbb{R}^m$ linearly independent, and $v^{(1)}, v^{(2)}, \dots v^{(n)} \in \mathbb{R}^n$ linearly independent,*

$$\{u^{(i)} \otimes v^{(j)} : \quad i \in [m]\ ,\ j \in [n]\}$$

is linearly independent.

Proof. Suppose for a contradiction that this was not the case. Let

$$S := \sum_{i\in[m] j\in[n]} \mu_{i,j} u^{(i)} \otimes v^{(j)} = 0$$

be a nontrivial linear combination that is 0. Without loss of generality $mu_{1,1} \neq 0$.
Define $n^{(i)} = \sum_{j=1}^{n} \mu_{i,j} v^{(j)}$. By linear independence of the $v^{(j)}$ $n^{(1)} \neq 0$ by permuting the columns we may assume $n_1^{(1)}$ is non-zero. Then

$$0 = S_{1:m} = \sum_{i=1}^{m} n_1^{(i)} u^{(i)}$$

which is a nontrivial linear combination of the $u^{(i)}$ that yields 0, a contradiction.
Thus $\{u^{(i)} \otimes v^{(j)} : \quad i \in [m]\ ,\ j \in [n]\}$ is linearly independent. □

Theorem 1.2.57. *Let $A \in \mathcal{S}^n$, $B \in \mathcal{S}^k$, and let $\lambda_1(A), \lambda_2(A), \dots \lambda_n(A)$ be the eigenvalues of A with corresponding eigenvectors $u_1, u_2, \dots u_n$ and $\lambda_1(B), \lambda_2(B), \dots \lambda_n(B)$ be the eigenvalues of B with corresponding eigenvectors $v_1, v_2, \dots v_n$.*
Then $A \otimes B$ has the multi-set of eigenvalues

$$\{\lambda_i(A)\lambda_j(B) : \quad i \in [n]\ ,\ j \in [k]\}$$

with corresponding eigenvectors

$$\{u_i \otimes v_j :\ i \in [n]\ ,\ j \in [k]\}.$$

Proof.
$$(A \otimes B)(u_i \otimes v_j) = (Au_i) \otimes (Bv_j) = \lambda_i(A)\lambda_j(B)u_i \otimes v_j.$$
So the $u_i \otimes v_j$ are eigenvectors of $A \otimes B$ with eigenvalues $\lambda_i(A)\lambda_j(B)$. Also they are linearly independent by the previous lemma. Thus they are all the eigenvectors. □

Corollary 1.2.58. *The Kronecker product of two positive semidefinite matrices A, B is positive semidefinite.*
The Kronecker product of two positive definite matrices A, B is positive definite.

Proof. If A, B are positive semidefinite, then they have non negative eigenvalues. By the previous theorem, this means that $A \otimes B$ has non negative eigenvalues. By Proposition 1.2.4 (3) it means they are positive semidefinite. The argument is similar for positive definiteness. □

Definition 1.2.59. *For two matrices $A, B \in \mathbb{R}^{n\times n}$ the Hadamard product of A, B is defined by*
$$A \circ B \in \mathbb{R}^{n\times n}$$
$$(A \circ B)_{i,j} = A_{i,j}B_{i,j}.$$

Corollary 1.2.60. *The Hadamard product satisfies:*

1. *The Hadamard product of two positive semidefinite matrices A, B is positive semidefinite.*
2. *The Hadamard product of two positive definite matrices A, B is positive definite.*

Proof. The Hadamard product is a submatrix of the Kronecker product. Submatrices of positive semidefinite matrices are positive semidefinite, and submatrices of positive definite matrices are positive definite.

□

Chapter 2

Eigenvalue Bounds for Vertex Separator

2.1 Preliminaries

In order to properly and cleanly write the results and ideas about the vertex separator problem, let us define some notation.

We let A be the *adjacency matrix* of our *graph*, $G = (V, E)$, e the all ones vector of appropriate size, and let

$$B = \begin{bmatrix} ee^T - I_{k-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{k \times k}.$$

For $S = (S_1, S_2, \dots, S_k)$ a partition of the vertices with $|S_i| = m_i$ and $m = (m_1, m_2, .., m_k)$ $n = |V|$, we define a *partition matrix* $X \in \mathbb{R}^{n \times k}$ using

$$X_{i,j} = \begin{cases} 1 & \text{if } \ i \in S_j \\ 0 & \text{otherwise.} \end{cases}$$

Define $\delta(S_i, S_j) = \{\{u, v\} \in E : \quad u \in S_i \ , \ v \in S_j\}$.
We now define the cut which is the objective we mean to minimize:

$$\delta(S) := \bigcup_{i=1}^{k-1} \delta(S_i, S_j).$$

We use the following subsets of matrices. ($m = (m_1, m_2, \dots m_k)^T$ is a partition of n)

- $\mathcal{O}_n := \{Z \in \mathbb{R}^{n\times n} \quad Z^T Z = I_n\}$, *orthogonal*
- $\mathcal{Z} := \{X \in \mathbb{R}^{n\times k} : X_{ij} \in \{0,1\}, \forall i,j\} = \{X \in \mathbb{R}^{n\times k} : (X_{ij})^2 = X_{ij}, \forall i,j\}$
  zero-one
- $\mathcal{N} := \{X \in \mathbb{R}^{n\times k} : X_{ij} \geq 0, \forall i,j\}$ *non-negative*
- $\mathcal{E} := \{X \in \mathbb{R}^{n\times k} : Xe = e, X^T e = m\} = \{X \in \mathbb{R}^{n\times k} : \|Xe - e\|^2 + \|X^T e - m\|^2 = 0\}$
  linear equalities
- $\mathcal{M}_m := \mathcal{Z} \cap \mathcal{E}$ *partition matrices*
- $\mathcal{D} := \{X \in \mathbb{R}^{n\times k} : X \in \mathcal{E} \cap \mathcal{N}\}$ *doubly stochastic type*
- $\mathcal{D}_O := \{X \in \mathbb{R}^{n\times k} : X^T X = \operatorname{Diag}(m)\}$ *m-diagonal orthogonal type*
- $\mathcal{D}_e := \{X \in \mathbb{R}^{n\times k} : \operatorname{diag}(XX^T) = e\}$ *e-diagonal orthogonal type*
- $\mathcal{G} := \{X \in \mathbb{R}^{n\times k} : X_{:i} \circ X_{:j} = 0, \forall i \neq j\}$ *Gangster constraints*

Some preliminary results now follow:

Proposition 2.1.1 ([11]). $|\delta(S)| = \frac{1}{2}\operatorname{tr}((A - \operatorname{Diag}(d))XBX^T), \forall d \in \mathbb{R}^n$.

Proof. We do the simple matrix multiplication

$$(XB)_{i,j} = \begin{cases} 1 & \text{if } i \notin S_j \\ 0 & \text{if } i \in S_j, \end{cases}$$

$$(XBX^T)_{i,j} = \begin{cases} 1 & \text{if } i,j \text{ in different } S_l \\ 0 & \text{if } i,j \text{ are in the same } S_l. \end{cases}$$

Thus

$$\frac{1}{2}\operatorname{tr} AXBX^T = \sum_{i,j=1}^{n} (A + \operatorname{Diag}(d))_{i,j}(XBX^T)_{i,j} = \frac{1}{2} \sum_{\substack{i \in S_{l_i}, j \in S_{l_j} \\ l_i \neq l_j}} A_{i,j}.$$

□

Let us further denote $G := G(d) = A - \mathrm{Diag}(d)$. Recall that we are minimizing $\delta(S)$ over all S with $|S_i| = m_i$. We now show some alternative characterizations of partition matrices.

Proposition 2.1.2 ([9]). *The set of partition matrices in $\mathbb{R}^{n\times k}$ can be expressed as the following.*

$$\begin{aligned} \mathcal{M}_m &= \mathcal{E} \cap \mathcal{Z} \\ &= \mathrm{ext}(\mathcal{D}) \\ &= \mathcal{E} \cap \mathcal{D}_O \cap \mathcal{N} \\ &= \mathcal{E} \cap \mathcal{D}_O \cap \mathcal{D}_e \cap \mathcal{N} \\ &= \mathcal{E} \cap \mathcal{Z} \cap \mathcal{D}_O \cap \mathcal{G} \cap \mathcal{N}. \end{aligned} \tag{2.1}$$

Proof. The first equality follows immediately from the definitions. The second equality we show in lemma 4.1.2. Let us show the third equality. Let $X \in \mathcal{N} \cap \mathcal{E} \cap \mathcal{D}_O$. If $0 < X_{i,j} < 1$ then $m_j = \sum_{l=1}^k X_{l,j} < \sum_{l=1}^k X_{l,j}^2 = m_j$. Thus $X \in \mathcal{Z} \cap \mathcal{E}$, so X is a partition matrix. The fourth and fifth equivalences contain redundant sets of constraints. □

Definition 2.1.3. *Given two vectors $x, y \in \mathbb{R}^n$. Denote $\mathrm{AUT}(n)$ to be the set of permutations of n. The* minimal scalar product *of x, y is defined as $\min_{\phi\in\mathrm{AUT}(n)} \sum_{i=1}^k x_i y_{\phi(i)}$ and will be denoted $\langle x, y\rangle_-$.*

Definition 2.1.4. *Given two vectors $x \in \mathbb{R}^k, y \in \mathbb{R}^n$ with $k < n$ we define the minimal scalar product of x, y to be the minimal scalar product of $\begin{bmatrix} x \\ 0_{n-k} \end{bmatrix}$ and y.*

Remark 2.1.5. *For $x, y \in \mathbb{R}^n$, let $\phi, \psi \in \mathrm{AUT}(n)$ be permutations such that $x_{\phi(1)} \le x_{\phi(2)} \le \ldots \le x_{\phi(n)}$ and $y_{\psi(1)} \ge y_{\psi(2)} \ge \ldots \ge y_{\phi(n)}$. A permutation that yields the minimal scalar product is $\phi^{-1}\psi$ and the sum is equal to $\sum_{i=1}^n x_{\phi(i)} y_{\psi(i)}$.*

Definition 2.1.6. *For a symmetric matrix S let $\lambda_1(S) \ge \lambda_2(S) \ge \ldots \ge \lambda_n(S)$ denote the eigenvalues of S in nonincreasing order.*

Given Proposition 2.1.1 the following theorem should seem relevant.

Theorem 2.1.7 ([5]). *[Hoffman-Wielandt Theorem] For symmetric matrices $\hat{A} \in \mathbb{R}^{n\times n}$, $\hat{B} \in \mathbb{R}^{k\times k}$ the following holds:*

$$\min_{Q^TQ=I_k} \mathrm{tr}\, \hat{A}Q\hat{B}Q^T = \Sigma_{i=1}^k \lambda_{n-i}(A)\lambda_i(B).$$

That is, the minimum is given by the minimal scalar product of $\lambda(A)$ and $\lambda(B)$.

This motivates the following:

2.1.1 Eigenvalue Bounds

A natural thing to do given Proposition 2.1.2 is to relax certain constraints and solve the problem over the relaxation. Suppose we only enforce the $X \in \mathcal{D}_O$ constraint. Then our problem can be written as.

$$\text{cut}(m) \geq \begin{array}{ll} \min & \frac{1}{2}\operatorname{trace} GXBX^T \\ \text{s.t.} & X \in \mathcal{D}_O. \end{array} \tag{2.2}$$

Let us make the following observation:

Proposition 2.1.8. *Define* $\tilde{M} = \text{Diag}([\sqrt{m_1}, \sqrt{m_2}, \ldots, \sqrt{m_k}])$. *For* $X \in \mathbb{R}^{n\times k}$ *define* $Y = X\tilde{M}^{-1}$ *then* $X \in \mathcal{D}_O$ *if and only if* $Y^TY = I_k$.

Proof. Follows from substituting $Y = X\tilde{M}^{-1}$ into $Y^TY = I_k$. □

Then this relaxed problem can be written in the form of the statement of the Hoffman-Wielandt theorem.

$$\text{cut}(m) \geq \begin{array}{ll} \min & \frac{1}{2}\operatorname{trace} GY\tilde{M}B\tilde{M}Y^T \\ \text{s.t.} & Y^TY = I_k. \end{array} \tag{2.3}$$

Lemma 2.1.9 ([11, Lemma 4]). *The k-ordered eigenvalues of the matrix $\tilde{B} := \tilde{M}B\tilde{M}$ satisfy*

$$\lambda_1(\tilde{B}) > 0 = \lambda_2(\tilde{B}) > \lambda_3(\tilde{B}) \geq \ldots \geq \lambda_{k-1}(\tilde{B}) \geq \lambda_k(\tilde{B}). \quad \square$$

By the Hoffman-Wielandt theorem, the minimum of this problem is $\sum_{i=1}^{k} \lambda_{n-i}(A)\lambda_i(B)$. This bound is referred to in the literature as the basic eigenvalue lower bound. For our vertex separator problem, this turns out to always be negative.

Theorem 2.1.10. *Let $d \in \mathbb{R}^n$, $G = A - \text{Diag}(d)$, $\tilde{B} = \tilde{M}B\tilde{M}$. Then*

$$\text{cut}(m) \geq 0 > p^*_{eig}(G) := \frac{1}{2}\left\langle \lambda(G), \begin{pmatrix} \lambda(\tilde{B}) \\ 0 \end{pmatrix} \right\rangle_- = \frac{1}{2}\left(\sum_{i=1}^{k-2} \lambda_{k-i+1}(\tilde{B})\lambda_i(G) + \lambda_1(\tilde{B})\lambda_n(G) \right).$$

Moreover, the function $p^*_{eig}(G(d))$ is concave as a function of $d \in \mathbb{R}^n$.

Proof. We use the substitution $X = Z\tilde{M}$, i.e. $Z = X\tilde{M}^{-1}$, in (2.2). Then the constraint on X implies that $Z^TZ = I$. We now solve the equivalent problem to (2.2):

$$\begin{array}{ll} \min & \frac{1}{2} \operatorname{trace} GZ(\tilde{M}B\tilde{M})Z^T \\ \text{s.t.} & Z^TZ = I. \end{array} \tag{2.4}$$

The optimal value is obtained using the minimal scalar product of eigenvalues as done in the Hoffman-Wielandt result, Theorem 2.1.7. From this we conclude immediately that $\operatorname{cut}(m) \geq p^*_{eig}(G)$. Furthermore, the explicit formula for the minimal scalar product follows immediately from remark 2.1.5.

We now show that $p^*_{eig}(G) < 0$. Note that $\operatorname{tr} \tilde{M}B\tilde{M} = \operatorname{tr} MB = 0$. Thus the sum of the eigenvalues of $\tilde{B} = \tilde{M}B\tilde{M}$ is 0. Let $\widehat{\phi}$ be a permutation of $\{1, \ldots, n\}$ that attains the minimum value $\min\limits_{\phi \in \mathrm{AUT}(n)} \sum_{i=1}^k \lambda_{\phi(i)}(G)\lambda_i(\tilde{B})$. Then for any permutation ψ, we have

$$\sum_{i=1}^k \lambda_{\psi(i)}(G)\lambda_i(\tilde{B}) \geq \sum_{i=1}^k \lambda_{\widehat{\phi}(i)}(G)\lambda_i(\tilde{B}). \tag{2.5}$$

$$\begin{array}{rl} & \sum_{\psi \in \mathrm{AUT}(n)} \left(\sum_{i=1}^k \lambda_{\psi(i)}(G)\lambda_i(\tilde{B}) \right) = \sum_{i=1}^k \left(\sum_{\psi \in \mathrm{AUT}(n)} \lambda_{\psi(i)}(G) \right) \lambda_i(\tilde{B}) \\ = & \left(\sum_{\psi \in \mathrm{AUT}(n)} \lambda_{\psi(1)}(G) \right) \left(\sum_{i=1}^k \lambda_i(\tilde{B}) \right) = 0, \end{array}$$

since $\sum_{\psi \in \mathrm{AUT}(n)} \lambda_{\psi(i)}(G)$ is independent of i. This means that there exists at least one permutation ψ so that $\sum_{i=1}^k \lambda_{\psi(i)}(G)\lambda_i(\tilde{B}) \leq 0$, which implies that the minimal scalar product must satisfy $\sum_{i=1}^k \lambda_{\widehat{\phi}(i)}(G)\lambda_i(\tilde{B}) \leq 0$. Moreover, in view of (2.5) and (2.1.1), this minimal scalar product is zero if, and only if, $\sum_{i=1}^k \lambda_{\psi(i)}(G)\lambda_i(\tilde{B}) = 0$, for all $\psi \in \mathrm{AUT}(n)$. Recall from Lemma 2.1.9 that $\lambda_1(\tilde{B}) > \lambda_k(\tilde{B})$. Moreover, if all eigenvalues of G were equal, then necessarily $G = \beta I$ for some $\beta \in \mathbb{R}$ and A must be diagonal. This implies that $A = 0$, a contradiction. This contradiction shows that $G(d)$ must have at least two distinct eigenvalues, regardless of the choice of d. Therefore, we can change the order and change the value of the scalar product on the left in (2.5). Thus $p^*_{eig}(G)$ is strictly negative.

Finally, the concavity follows by observing from (2.4) that

$$p^*_{eig}(G(d)) = \min_{Z^TZ=I} \frac{1}{2} \operatorname{trace} G(d)Z(\tilde{M}B\tilde{M})Z^T$$

is a function obtained as a minimum of a set of functions affine in d, and recalling that the minimum of affine functions is concave. □

Let us explain our motivation to include the d here by detouring into the similar graph partitioning problem.
Given a graph G and a partition m of n into k pieces, define $\bar{B} = e_k e_k^T - I_k$. We define the graph partitioning problem as

$$\begin{array}{ll} \min & \frac{1}{2} \operatorname{trace} AX\bar{B}X^T \\ \text{s.t.} & X \in \mathcal{M}_m. \end{array}$$

Alternatively if X is the partition matrix for $(S_1, S_2, \dots S_k)$ then the objective we minimize is $\sum_{i<j} \delta(S_i, S_j)$. Consider the analogous basic eigenvalue bound for (2.2).

$$\operatorname{part}(m) := \begin{array}{ll} \min & \frac{1}{2} \operatorname{trace} AX\bar{B}X^T \\ \text{s.t.} & X \in \mathcal{M}_m, \end{array}$$

Theorem 2.1.11. *Let $d \in \mathbb{R}^n$, $G = A - \operatorname{Diag}(d)$. Then*

$$\operatorname{part}(m) \geq 0 > q_{eig}^*(G) := \frac{1}{2}\left\langle \lambda(G), \begin{pmatrix} \lambda(\bar{B}) \\ 0 \end{pmatrix} \right\rangle_- = \frac{1}{2}\left(\sum_{i=1}^{k-2} \lambda_{k-i+1}(\bar{B})\lambda_i(G) + \lambda_1(\bar{B})\lambda_n(G) \right).$$

Proof. The proof here is basically the same as in Theorem 2.1.10 we again notice that $\sum_{i=1}^k \bar{B} = 0$ same as for $\tilde{B}$ and continue as in Theorem 2.1.10. □

Proposition 2.1.12. *If X is the partition matrix for the partition $(S_1, S_2, \dots S_n)$ then*

$$\begin{array}{rcl} \sum_{i<j} \delta(S_i, S_j) & = & \frac{1}{2} \operatorname{trace} AX\bar{B}X^T \\ & = & \frac{1}{2} \operatorname{trace} LXX^T \end{array}$$

where $L := \operatorname{Diag}(Ae) - A$ is the Laplacian of our graph.

This gives us an alternative eigenvalue bound by minimizing $\frac{1}{2} \operatorname{trace} LXX^T$ over $\mathcal{D}_O$.

Proposition 2.1.13. *The value of this bound, $\sum_{i=1}^k \lambda_{n-k}(L)$ is non-negative and is strictly positive if and only if G has fewer than k components.*

Proof. This follows from the fact that the Laplacian of a graph G is positive semidefinite and has number of zero eigenvalues equal to the number of components of G □

This illustrates the power of choosing the objective function carefully. For the vertex separator problem, we do not have anything nearly as good. We emphasize that although the choice of d in our objective function here does not change the value $\frac{1}{2}\operatorname{trace} G(d)X(\tilde{M}B\tilde{M})X^T$ for any partition matrix X the eigenvalue bound we get often is different. The concavity of $p^*(d)$ derived in Theorem 2.1.10 is also true if instead of minimizing over $\mathcal{D}_O$ we minimize over $\mathcal{D}_O \cap \mathcal{E}$. This bound fortunately is not always negative and because it is inexpensive, can be used multiple times to get better d [11].

Now let us consider optimizing $G(d)XBX^T$ over $\mathcal{D}_o \cap \mathcal{E}$. The bound obtained here is referred to in the literature as the projected eigenvalue bound.
Let V, W be orthogonal matrices that form the orthogonal complements to e,m respectively. Define:

$$P := \begin{bmatrix} \frac{1}{\sqrt{n}}e & V \end{bmatrix} \in \mathcal{O}_n, \quad Q := \begin{bmatrix} \frac{1}{\sqrt{n}}\tilde{m} & W \end{bmatrix} \in \mathcal{O}_k. \tag{2.6}$$

Now suppose that $X \in \mathbb{R}^{n\times k}$ and $Z \in \mathbb{R}^{(n-1)\times(k-1)}$ are related by

$$X = P \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} Q^T \tilde{M}. \tag{2.7}$$

Then the following holds:

Lemma 2.1.14 ([11]).

- *(1)* $X \in \mathcal{E}$.
- *(2)* $X \in \mathcal{N} \Leftrightarrow VZW^T \geq -\frac{1}{n}e\tilde{m}^T$.
- *(3)* $X \in \mathcal{D}_O \Leftrightarrow Z^TZ = I_{k-1}$.

□

Remark 2.1.15 ([11]). *Conversely, if* $X \in \mathcal{E}$, *then there exists* Z *such that the representation* (2.7) *holds.*

Let $\mathcal{Q} : \mathbb{R}^{(n-1)\times(k-1)} \to \mathbb{R}^{n\times k}$ be the linear transformation defined by $\mathcal{Q}(Z) = VZW^T\tilde{M}$ and define $\widehat{X} = \frac{1}{n}em^T \in \mathbb{R}^{n\times k}$. Then $\widehat{X} \in \mathcal{E}$, and Lemma 2.1.14 states that $\mathcal{Q}$ is an invertible transformation between $\mathbb{R}^{(n-1)\times(k-1)}$ and $\mathcal{E} - \widehat{X}$. More precisely it says that $X \in \mathcal{E}$ if, and only if, $X = P \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} Q^T\tilde{M}$ for some Z.

Since

$$\begin{aligned} & P \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} Q^T \tilde{M} \\ = & \begin{bmatrix} \frac{e}{\sqrt{n}} & V \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{n}} \tilde{m}^T \\ W^T \end{bmatrix} \tilde{M} \\ = & \frac{1}{n} e m^T + V Z W^T \tilde{M} \\ = & \widehat{X} + V Z W^T \tilde{M}, \end{aligned}$$

we get that $X \in \mathcal{E}$ if and only if X is equal to $\widehat{X} + VZW^T\tilde{M}$ for some Z. Thus, the set $\mathcal{E}$ can be parametrized using $\widehat{X} + VZW^T\tilde{M}$ and the set $\mathcal{E} \cap \mathcal{D}_O$ can be parametrized by orthogonal Z. [11, 12].

Denote

$$\begin{array}{ll} \widehat{X} = \frac{1}{n} e m^T, & \hat{G} = V^T G V \\ \hat{B} = W^T \tilde{M} B \tilde{M} W, & . \end{array}$$

Then we can rewrite the above as:

$$\begin{aligned} \operatorname{tr} GXBX^T &= \operatorname{tr} G(\hat{X} + VZW^T\tilde{M})B(\hat{X} + VZW^T\tilde{M})^T \\ &= \operatorname{tr} G\widehat{X}B\widehat{X}^T + \operatorname{tr}(V^TGV)Z(W^T\tilde{M}B\tilde{M}W)Z^T + \operatorname{tr} 2V^TG\widehat{X}B\tilde{M}WZ^T. \end{aligned}$$

The standard method here is to choose the above d such that $Ge = 0$ so that the linear and constant terms disappear. Then after we relax the non-negativity constraint, we obtain following relaxation of our vertex separator problem.

$$p^* := \min_{Z^TZ = I_k} \operatorname{tr} \hat{G} Z \hat{B} Z^T. \tag{2.8}$$

Now we can apply the Hoffman-Wielandt theorem to get

$$p^* = \Sigma_{i=1}^k \lambda_{n-i}(\hat{G})\lambda_i(\hat{B}).$$

The value p^* is referred to in the literature as the projected eigenvalue bound, and provides an inexpensive lower bound for large problems.

Chapter 3

SDP for graph partitioning

3.1 Semidefinite Relaxation Derivation

Let us now derive the Semidefinite Relaxation for our vertex separator problem. In the optimization literature it is standard to derive relaxations by taking the Lagrangian dual twice [15]. We will follow in the fashion of [15], except that we will enforce another redundant constraint $X_{:,i} \circ X_{:,j} = 0$. It turns out that doing this derives the Gangster constraint that was originally added separately in [15]. We take the Lagrangian dual of

$$\begin{array}{ll} \min & \operatorname{tr} AXBX^T \\ \text{s.t.} & \|Xe - e\|^2 = 0, \\ & \|X^Te - m\|^2 = 0 \\ & X^TX = M \\ & \operatorname{diag}(XX^T) = e \\ & X_{:,i} \circ X_{:,i} = X_{:,i} \quad X_{:,i} \circ X_{:,j} = 0 \quad \forall i,j \end{array}$$

to get

$$\begin{aligned} d^* = \max\nolimits_{D_1,D_2,S,s,t_i,t_{i,j}} \min\nolimits_X \operatorname{tr} AXBX^T + \operatorname{tr} D_1(Xee^TX^T - Xee^T + ee^TX + ee^T) + \\ \operatorname{tr} D_2(X^Tee^TX - X^Tem^T + me^TX + mm^T) + \operatorname{tr} S(M - X^TX) + \\ s^T(e - \operatorname{diag}(XX^T)) + \Sigma_{i=1}^k (X_{:,i} \circ X_{:,i}^T - X_{:,i})^T \operatorname{Diag}(t_i) + \Sigma_{i \neq j} \operatorname{tr}(X_{:,i}X_{:,j}^T) \operatorname{Diag}(t_{i,j}). \end{aligned}$$

Introduce another variable x_0 with constraint $x_0^2 = 1$ to homogenize the system.

$\max_{D_1,D_2,S,s,t_i,T_{i,j}} \min_X \operatorname{tr} y^T \mathcal{L}(D_1, D_2, S, s, t_i, T_{i,j}) y + \operatorname{tr} D_1 ee^T + \operatorname{tr} D_2 mm^T + \operatorname{tr} SM + e^T t$

Where T is the matrix with $T_{i,j}$ on the diagonal of the off diagonal blocks. And the following variables are defined:

$$\mathcal{L}(D_1, D_2, S, s, t_i, T_{i,j})$$
$$:= \begin{bmatrix} 1 & C \\ C & B \otimes A + ee^T \otimes D_1 + D_2 \otimes ee^T + S \otimes I_n + I_k \otimes \mathrm{Diag}(s) + \mathrm{Diag}\left(\begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_k \end{bmatrix} \right) + T \end{bmatrix},$$

$$C = \tfrac{1}{2} \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_n \end{bmatrix} + \mathrm{vec}(ee^T D_1) + \mathrm{vec}(D_2 me^T),$$

$$y = \begin{bmatrix} a_0 \\ \mathrm{vec}(X) \end{bmatrix}.$$

Now let us take the Lagrangian dual

$$\begin{array}{ll} \max & \mathrm{tr}\, D_1 ee^T + \mathrm{tr}\, D_2 mm^T + \mathrm{tr}\, SM + e^T t \\ & \text{s.t.} \\ & \mathcal{L}(D_1, D_2, S, s, t_i, T_{i,j}) \succeq 0 \end{array}$$

Let us take the dual of the above to get:

$$\min_Y \max_{D_1,D_2,S,s,t_i,T_{i,j} \quad C} \quad \mathrm{tr}\, D_1 ee^T + \mathrm{tr}\, D_2 mm^T + \mathrm{tr}\, SM + e^T t + \mathrm{tr}\, \mathcal{L}(D_1, D_2, S, s, t_i, T_{i,j}) Y$$

$$= \min_Y \max_{D_1,D_2,S,t,t_i,T_{i,j}} \quad \mathrm{tr}(Y \begin{bmatrix} 1 & 0 \\ 0 & B \otimes A \end{bmatrix} + Y \begin{bmatrix} 1 & \mathrm{vec}(ee^T D_1)^T \\ \mathrm{vec}(ee^T D_1) & ee^T \otimes D_1 \end{bmatrix} +$$
$$Y \begin{bmatrix} 1 & \mathrm{vec}(ee^T D_1)^T \\ \mathrm{vec}(D_2 me^T) & D_2 \otimes ee^T \end{bmatrix} + Y \begin{bmatrix} 1 & 0 \\ 0 & \otimes ee^T + S \otimes I_n + I_k \otimes \mathrm{Diag}(t) \end{bmatrix} +$$
$$Y \begin{bmatrix} 1 & \frac{1}{2} \begin{bmatrix} t_1 & t_2 & \dots & t_n \end{bmatrix} \\ \frac{1}{2} \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_n \end{bmatrix} & \mathrm{Diag}\left(\begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_k \end{bmatrix} \right) \end{bmatrix})$$

$$
\begin{array}{rlrcl}
= & \min & \frac{1}{2}\operatorname{tr} L_G Y & & \\
& \text{s.t.} & \operatorname{arrow}(Y) & = & e_0, \\
& & \operatorname{tr} D_1 Y & = & 0, \\
& & \operatorname{tr} D_2 Y & = & 0, \\
& & \mathcal{G}_J(Y) & = & e_0 e_0^T, \\
& & \mathcal{D}_O(Y) & = & M, \\
& & \mathcal{D}_e(Y) & = & e, \\
& & Y_{00} & = & 1, \\
& & Y \succeq 0, & &
\end{array}
$$

Where

$$
L_G := \begin{bmatrix} 1 & 0 \\ 0 & B \otimes G \end{bmatrix} \qquad D_1 := \begin{bmatrix} n & -e_k^T \otimes e_n^T \\ -e_k \otimes e_n & (e_k e_k^T) \otimes I_n \end{bmatrix}
$$

$$
D_2 := \begin{bmatrix} m^T m & -m^T \otimes e_n^T \\ -m \otimes e_n & I_k \otimes (e_n e_n^T) \end{bmatrix} \qquad \operatorname{arrow}(Y) := \operatorname{diag}(Y) - (0, Y_{0,1:kn})^T.
$$

3.2 Semidefinite Lifting

Alternatively we consider the function $\operatorname{tr} G(d) X B X^T$ as a function of $\begin{bmatrix} 1 \\ \operatorname{vec}(X) \end{bmatrix}$ and use the standard semidefinite lifting with $Y = \begin{bmatrix} 1 \\ \operatorname{vec}(X) \end{bmatrix} \begin{bmatrix} 1 \\ \operatorname{vec}(X) \end{bmatrix}^T$ [7, 15, 17]. This gives us the semidefinite relaxation:

$$
\begin{array}{rlrcl}
\operatorname{cut}(m) \geq p^*_{SDP}(G) := & \min & \frac{1}{2}\operatorname{tr} L_G Y & & \\
& \text{s.t.} & \operatorname{arrow}(Y) & = & e_0, \\
& & \operatorname{tr} D_1 Y & = & 0, \\
& & \operatorname{tr} D_2 Y & = & 0, \\
& & \mathcal{G}_J(Y) & = & e_0 e_0^T, \\
& & \mathcal{D}_O(Y) & = & M, \\
& & \mathcal{D}_e(Y) & = & e, \\
& & Y_{00} & = & 1, \\
& & Y \succeq 0, & &
\end{array}
$$

$\operatorname{tr} D_1 Y = 0$, $\operatorname{tr} D_2 Y = 0$, are the lifting of the constraints $\|Xe - e\|^2 = 0$, $\|X^T e - m\|^2 = 0$. The mapping $\mathcal{G}_J : \mathcal{S}^{nk+1} \to \mathcal{S}^{nk+1}$ is commonly referred to as the *Gangster operator* and defined by the following.

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

where

$$J := \{(i,j) : i = (p-1)n + q,\ j = (r-1)n + q, \text{ for all } p, r \text{ with } p < r, \\ p, r \in \{1, \ldots, k\}\ q \in \{1, \ldots, n\}\}.$$

Write Y as $Y = \begin{bmatrix} Y_{00} & Y_{0,:} \\ Y_{:,0} & \bar{Y} \end{bmatrix}, \quad \bar{Y} = \begin{bmatrix} \bar{Y}_{(11)} & \bar{Y}_{(12)} & \cdots & \bar{Y}_{(1k)} \\ \bar{Y}_{(21)} & \bar{Y}_{(22)} & \cdots & \bar{Y}_{(2k)} \\ \vdots & \ddots & \ddots & \vdots \\ \bar{Y}_{(k1)} & \ddots & \ddots & \bar{Y}_{(kk)} \end{bmatrix},$

where $Y_{i,j} \in \mathbb{R}^{n \times n}$ for $1 \leq i, j \leq k$.

Then $\mathcal{D}_O(Y)_{i,j} := \operatorname{tr} Y_{i,j}$ for $i, j \in \{1, 2, .., k\}$ and $\mathcal{D}_e(Y)_i = \Sigma_{j=1}^k Y_{j,j_i}$. These represent the constraints $X^T X = \operatorname{Diag}(m), \quad XX^T = e$ respectively.

Note that $D_1,\ D_2$ are positive semidefinite, yet $\operatorname{tr} D_1 Y = 0 \quad \operatorname{tr} D_2 Y = 0$, thus the problem can be facially reduced [1, 3]. Let

$$V_j := \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & \ldots & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 1\ldots & \ldots & \ldots & \ldots & 1 \\ -1 & \ldots & \ldots & -1 & -1 \end{bmatrix}$$

and let

$$\hat{V} := \begin{bmatrix} 1 & 0 \\ \frac{1}{n} m \otimes e_n & V_k \otimes V_n \end{bmatrix}.$$

Then the columns of $\hat{V}$ are in the nullspace of $D_1, \quad D_2$ thus Y can be written as $Y = \hat{V} R \hat{V}^T$. Then our new facially reduced SDP is given by

$$\begin{array}{rcll} \operatorname{cut}(m) \geq p^*_{SDP}(G) & = & \min & \frac{1}{2} \operatorname{tr} \widehat{V}^T L_G \widehat{V} Z \\ & & \text{s.t.} & \operatorname{arrow}(\widehat{V} Z \widehat{V}^T) = e_0 \\ & & & \mathcal{G}_{\bar{J}}(\widehat{V} Z \widehat{V}^T) = \mathcal{G}_{\bar{J}}(e_0 e_0^T) \\ & & & \mathcal{D}_O(\widehat{V} Z \widehat{V}^T) = M \\ & & & \mathcal{D}_e(\widehat{V} Z \widehat{V}^T) = e \\ & & & Z \succeq 0, \quad Z \in \mathcal{S}^{(k-1)(n-1)+1}. \end{array} \tag{3.1}$$

Surprisingly it turns out, under this facial reduction the only non-redundant constraint in the above is $\mathcal{G}_J(Y) = e_0 e_0^T$. [9]

Lemma 3.2.1 ([15])**.** *The arrow constraint can be derived from*

$$\begin{array}{rcl} \mathcal{G}_{\bar{J}}(\widehat{V} Z \widehat{V}^T) &=& \mathcal{G}_{\bar{J}}(e_0 e_0^T) \\ \operatorname{tr} D_1 Y &=& 0, \\ \operatorname{tr} D_2 Y &=& 0, \\ Y &\geq& 0. \end{array}$$

In particular, it means that (3.2) *satisfies the arrow constraint.*

Proof. $Y \geq 0$ and $\operatorname{tr} D_i Y = 0$ imply $D_i Y = 0$. $(D_1)_{l,:} Y = 0$ implies that for $1 \leq l \leq nk$,

$$Y_{l,0} = \sum_{\substack{j=l \mod n \\ 1 \leq j \leq nk}} Y_{l,j}.$$

Note that the Gangster constraint says that for $a < b$ and $a = b$ mod n $Y_{a,b} = Y_{b,a} = 0$. Thus the above equation becomes $Y_{0,l} = Y_{l,l}$ □

Theorem 3.2.2 ([9])**.** *Under the facial reduction $Y = \hat{V} R \hat{V}^T$ our problem can be formulated as*

$$\begin{array}{rcll} \operatorname{cut}(m) \geq p^*_{SDP}(G) &=& \min & \frac{1}{2} \operatorname{tr} \left(\widehat{V}^T L_G \widehat{V} \right) Z \\ && \text{s.t.} & \mathcal{G}_{\bar{J}}(\widehat{V} Z \widehat{V}^T) = \mathcal{G}_{\bar{J}}(e_0 e_0^T) \\ &&& Z \succeq 0, \quad Z \in \mathcal{S}^{(k-1)(n-1)+1}. \end{array} \tag{3.2}$$

The dual program is

$$\begin{array}{ll} \max & \frac{1}{2} W_{00} \\ \text{s.t.} & \widehat{V}^T \mathcal{G}_{\bar{J}}(W) \widehat{V} \preceq \widehat{V}^T L_G \widehat{V}. \end{array} \tag{3.3}$$

Both primal and dual satisfy Slater's constraint qualification and the objective function is independent of the $d \in \mathbb{R}^n$ chosen to form G.

Proof. Lemma 3.2.1 shows us that the arrow constraint is satisfied by 3.2. It only remains to show that the last two equality constraints in (3.1) are redundant. The Gangster constraint using the linear transformation $\mathcal{G}_{\bar{J}}$ implies that the blocks in $Y = \widehat{V} Z \widehat{V}^T$ satisfy $\operatorname{diag} \bar{Y}_{(ij)} =$

0 for all $i \neq j$, where $\bar{Y}$ respects the block structure described in 3.2. Next, we note that $D_i \succeq 0$, $i = 1, 2$ and $Y \succeq 0$. Therefore, the Schur complement of Y_{00} implies that

$$Y \succeq Y_{0:kn,0} Y_{0:kn,0}^T.$$

Writing $v_1 := Y_{0:kn,0}$ and $X = \mathrm{Mat}(Y_{1:kn,0})$, we see further that

$$0 = \mathrm{trace}(D_i Y) \geq \mathrm{trace}(D_i v_1 v_1^T) = \begin{cases} \|Xe - e\|^2 & \text{if } i = 1, \\ \|X^T e - m\|^2 & \text{if } i = 2. \end{cases}$$

This together with the arrow constraints show that $\mathrm{trace}\, \bar{Y}_{(ii)} = \sum_{j=(i-1)n+1}^{ni} Y_{j0} = m_i$. Thus, $\mathcal{D}_O(\widehat{V} Z \widehat{V}^T) = M$ holds. Similarly, one can see from the above and the arrow constraint that $\mathcal{D}_e(\widehat{V} Z \widehat{V}^T) = e$ holds.

The conclusion about Slater's constraint qualification for (3.2) follows from [15, Theorems 4.1], which discussed the primal SDP relaxations of the GP. That relaxation has the same feasible set as (3.2). In fact, it is shown in [15] that

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

where $\bar{m}_{k-1}^T = (m_1, \ldots, m_{k-1})$ and E_{n-1} is the $n-1$ square matrix of ones, is a strictly feasible point for (3.2). The right-hand side of the dual (3.3) differs from the dual of the SDP relaxation of the GP. Let

$$\hat{W} = \begin{bmatrix} 0 & 0 \\ 0 & (E_k - I_k) \otimes I_n \end{bmatrix}.$$

Since $\hat{W}$ has all non-zero entries in $\bar{J}$,

$$\mathcal{G}_{\bar{J}}(\hat{W}) = \hat{W}$$

so

$$-\widehat{V}^T \mathcal{G}_{\bar{J}}(\hat{W}) \widehat{V} = \widehat{V}^T(-\hat{W})\widehat{V}.$$

We note that this is positive definite if and only if $-\hat{W}$ is positive definite on the range of $\widehat{V}$. We can see from the properties of the Kronecker product (Theorem 1.2.57) that the spectral decomposition of W yields e_0 with eigenvalue 0 and $l_i := \begin{bmatrix} 0 \\ e_k \otimes e_n^{(i)} \end{bmatrix}$ with negative eigenvalues for $i = 1, 2., , , n$, where $e_n^{(i)}$ is the standard unit vector of length n with one 1 in ith entry. We note that since $V_k \otimes V_n$ is orthogonal to all l_i and $e_0 - \widehat{V}_{:,0}$, l_i and e_0 are not in the range of $\widehat{V}$ thus $-\widehat{V}^T \mathcal{G}_{\bar{J}}(\hat{W}) \geq 0$. Therefore $\widehat{V}^T \mathcal{G}_{\bar{J}}(\beta \hat{W}) \widehat{V} \prec \widehat{V}^T L_G \widehat{V}$ for sufficiently large β, i.e. Slater's constraint qualification holds for the dual (3.3).

To show that the choice of d does not matter, denote

$$L_D = \begin{bmatrix} 0 & 0 \\ 0 & B \otimes Diag(d) \end{bmatrix}.$$

So

$$L_G = \begin{bmatrix} 1 & 0 \\ 0 & B \otimes A \end{bmatrix} + L_D,$$

Then notice that any feasible Y has zero entries in the nonzero positions of $B \otimes Diag(d)$ due to the Gangster constraint. Thus $\operatorname{tr} L_D Y = 0$ and the choice of d does not matter. □

We have our problem cut(m) with relaxation $P^*_{SDP}(G)$:

$$\text{cut}(m) \geq p^*_{SDP}(G) =$$

$$\begin{array}{ll} \min & \frac{1}{2} \operatorname{tr}(\hat{V}^T L_G \hat{V} R) \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(\hat{V} R \hat{V}^T) = e_0 e_0^T \\ & R \succeq 0, \\ & R \in \mathcal{S}^{(k-1)(n-1)+1}. \end{array} \tag{3.4}$$

With dual

$$\begin{array}{ll} \max & \frac{1}{2} W_{00} \\ \text{s.t.} & \hat{V}^T \mathcal{G}_{\bar{J}}(W) \hat{V} \preceq \hat{V}^T L_G \widehat{V}. \end{array} \tag{3.5}$$

In standard SDP algorithms, it is often important to start at a Slater point. We end this chapter with the following result of [15] which gives us an easy way to find a Slater Point.

Remark 3.2.3 ([15]). *The following matrix $\hat{R}$ defined by*

$$\hat{R} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{n^2(n-1)} (n \operatorname{Diag}(\bar{m}_{k-1}) - \bar{m}_{k-1} \bar{m}_{k-1}^T) \otimes (n I_{n-1} - E_{n-1}) \end{bmatrix}$$

is strictly feasible for (3.4).

Chapter 4

Algorithms

The augmented Lagrangian method is a method to solve constrained optimization problems by forming the Lagrangian and adding a penalty term for violating the constraints. The ADMM (alternating direction method of multipliers) is a type of augmented Lagrangian method that solves the problem of solving the Lagrangian by solving it over each variable separately. The advantage of this method is that the subproblems are often easy to solve as is the case with the vertex separator problem.

4.1 ADMM Algorithms

We will do three algorithms here, the ADMM high rank, ADMM high rank with non-negativity, and the ADMM low rank. Recall the final SDP formulation 3.2:

$$\begin{array}{ll} \min & \frac{1}{2}\operatorname{tr} L_G Y \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(Y) = \mathcal{G}_{\bar{J}}(e_0 e_0^T) \\ & Y = \widehat{V} R \widehat{V}^T \\ & Z \in \mathcal{S}^{(k-1)(n-1)+1} \quad Y \in \mathcal{S}^{kn+1}. \end{array} \tag{4.1}$$

The dual program is

$$\begin{array}{ll} \max & \frac{1}{2} Z_{00} \\ \text{s.t.} & \widehat{V}^T \mathcal{G}_{\bar{J}}(Z) \widehat{V} \preceq \widehat{V}^T L_G \widehat{V}. \end{array} \tag{4.2}$$

For the rest of this section, R_0, Y_0, Z_0 will be matrices so that $R = R_0, \quad Y = Y_0, \quad Z = Z_0$ are feasible for 4.1, 4.2 and $\mu > 0$ will be the acceptable tolerance.

4.1.1 Preliminary Concepts

Define:

$\mathcal{Y} = \{Y \in \mathcal{S}^{nk+1} : \mathcal{G}_{\bar{J}}(Y) = E_{00}, \quad 0 \leq Y \leq 1\}$
$\mathcal{P}_1 = \{Y \in \mathcal{S}^{nk+1} : \quad \mathcal{G}_J(Y) = E_{00}\}$
$\mathcal{Z} := \{Z \in \mathcal{S}_+^{nk+1} : \; \hat{V}^T Z \hat{V} \preceq 0\}$

We start with $R = R_0, \quad Y = Y_0, \quad Z = Z_0$ feasible for 4.1, 4.2.
and define the Lagrangian
$\mathcal{L}(R, Y, Z) = \langle L_G, Y\rangle + \langle Z, Y - \hat{V}R\hat{V}^T\rangle + \frac{\beta}{2}\|Y - \hat{V}R\hat{V}^T\|_F^2.$
Then we iterate:

$$R_+ = \operatorname{argmin}_{R \in \mathcal{S}^(k-1)(n-1)_+} \mathcal{L}(R, Y, Z)$$
$$Y_+ = \operatorname{argmin}_{Y \in \mathcal{P}_t} \mathcal{L}(R, Y, Z)$$
$$Z_+ = Z + \gamma\beta(Y_+ - \hat{V}R\hat{V}^T)$$

Let us work out the formulas explicitly. One of the main benefits of using ADMM for graph partitioning is that the sub-problems above can be computed efficiently.

$$\begin{aligned}
R_+ &= \operatorname{argmin}_{R \succeq 0} \|Y - \hat{V}R\hat{V}^T + \tfrac{1}{\beta}Z\|^2 \\
&= \|R - \hat{V}^T(Y + \tfrac{1}{\beta}Z)\hat{V}\|^2 \\
&= \mathcal{P}_{\mathcal{S}_+}(\hat{V}^T(Y + \tfrac{1}{\beta}Z)\hat{V}) \\
Y_+ &= \operatorname{argmin}_{\mathcal{G}_J(Y) = E_{00}} \|Y - \hat{V}R_+\hat{V}^T + \tfrac{L_G + Z}{B}\|^2 \\
&= E_{00} + \mathcal{G}_{J^c}(\hat{V}R_+\hat{V}^T + \tfrac{L_G + Z}{B})
\end{aligned}$$

Where $\mathcal{P}_{\mathcal{S}_+}$ is the projection onto the semidefinite cone. So our algorithm is:

Algorithm 4.1.1: ADMM high rank algorithm for vertex separator

Data: Input: Graph $G = (V, E)$ vector m

Result: Output: Solution for ADMM High rank

Input: $R_0, \quad Y_0, \quad Z_0$

feasible for 4.1, 4.2.

We also define a tolerance μ for the acceptable primal and dual residuals.

Initialize:

$$\begin{array}{l} R = R_0, \quad R_+ = R_0 \\ Y = Y_0, \quad Y_+ = Y_0 \\ Z = Z_0, \quad Z_+ = Z_0 \end{array}$$

while $\|Y - Y_+\| + \|Y - \hat{V}R\hat{V}^T\| > \mu$ **do**

Iterate by

$$\begin{array}{ll} R & = R_+, \quad Y = Y_+, \quad Z = Z_+ \\ R_+ & = \mathcal{P}_{\mathcal{S}_+}(\hat{V}^T(Y + \frac{1}{\beta}Z)\hat{V}) \\ Y_+ & = E_{00} + \mathcal{G}_{J^c}(\hat{V}R_+\hat{V}^T + \frac{L_G + Z}{B}) \\ Z_+ & = Z + \gamma\beta(Y_+ - \hat{V}R_+\hat{V}^T) \end{array}$$

end while

Now we consider adding the constraints $0 \leq Y \leq 1$ to 4.1 and call this following new problem (4.3)

$$\begin{array}{ll} \min & \frac{1}{2}\operatorname{tr}(\hat{V}^T L_G \hat{V} R) \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(\hat{V}R\hat{V}^T) = e_0 e_0^T \\ & Y = \hat{V}R\hat{V}^T \geq 0 \\ & Y, R \in \mathcal{S}^{(k-1)(n-1)+1}. \end{array} \tag{4.3}$$

This changes the Y update to:

$$\begin{array}{rl} Y_+ = & \operatorname{argmin}_{Y \in \mathcal{Y}} \mathcal{L}(R, Y, Z) \\ = & \mathcal{P}_{[0,1]^{J^c}}(\mathcal{G}_{J^c}(\hat{V}R_+\hat{V}^T + \frac{L_G + Z}{B})) \\ = & E_{00} + \min(1, \max(0, \mathcal{G}_{J^c}(\hat{V}R_+\hat{V}^T + \frac{L_G + Z}{B}))) \end{array}$$

($\mathcal{P}_Q$ stands for the projection of the set Q) and we get the following algorithm for ADMM with non-negativity:

Algorithm 4.1.2: ADMM high rank algorithm for vertex separator with non-negativity

Data: Input: Graph $G = (V, E)$ vector m
Result: Output: Solution for ADMM High rank
Input: $R_0, \quad Y_0, \quad Z_0$
feasible for 4.1, 4.2 .
We also define a tolerance μ for the acceptable primal and dual residuals.
Initialize:

$$R = R_0, \quad R_+ = R_0$$
$$Y = Y_0, \quad Y_+ = Y_0$$
$$Z = Z_0, \quad Z_+ = Z_0$$

while $\|Y - Y_+\| + \|Y - \hat{V}R\hat{V}^T\| > \mu$ **do**
Iterate by

$$\begin{aligned} R &= R_+, \quad Y = Y_+, \quad Z = Z_+ \\ R_+ &= \mathcal{P}_{\mathcal{S}_+}(\hat{V}^T(Y + \tfrac{1}{\beta}Z)\hat{V}) \\ Y_+ &= \mathcal{P}_{[0,1]^{J^c}}(\mathcal{G}_{J^c}(\hat{V}R_+\hat{V}^T + \tfrac{L_G+Z}{B})) \\ Z_+ &= Z + \gamma\beta(Y_+ - \hat{V}R\hat{V}^T) \end{aligned}$$

end while

We would like to remark that in the standard SDP solvers such as SDPT3 the $0 \leq Y$ constraint is very expensive and is often not used for bounds.

Lemma 4.1.1. *Let*

$$\mathcal{R} := \{R \succeq 0\},$$
$$\mathcal{Y} := \{Y : \mathcal{G}_J(Y) = E_{00}, \quad 0 \leq Y \leq 1\},$$
$$\mathcal{Z} := \{Z \in \mathbb{S}_+^{nk+1} : \ \hat{V}^T Z \hat{V} \preceq 0\}.$$

Define the ADMM dual function

$$g(Z) := \min_{Y \in \mathcal{Y}} \langle L_G + Z, Y \rangle.$$

Then the dual problem of ADMM (4.3) is defined as follows and satisfies weak duality.

$$d_Z^* := \max_{Z \in \mathcal{Z}} g(Z) \leq p_R^*.$$

Proof. The dual problem can be written as

$$
\begin{array}{rcl}
d_Z^* & := & \max_Z \ \min_{R\in\mathcal{R},\ Y\in\mathcal{Y}} \langle L_G, Y\rangle + \langle Z, Y - \hat{V}R\hat{V}^T\rangle \\
& = & \max_Z \ \min_{Y\in\mathcal{Y}} \langle L_G, Y\rangle + \langle Z, Y\rangle + \ \min_{R\in\mathcal{R}} \langle Z, -\hat{V}R\hat{V}^T\rangle \\
& = & \max_Z \ \min_{Y\in\mathcal{Y}} \langle L_G, Y\rangle + \langle Z, Y\rangle + \ \min_{R\in\mathcal{R}} \langle \hat{V}^T Z\hat{V}, -R\rangle \\
& = & \max_{Z\in\mathcal{Z}} \ \min_{Y\in\mathcal{Y}} \langle L_G + Z, Y\rangle \\
& = & \max_{Z\in\mathcal{Z}} g(Z).
\end{array}
$$

□

For any $Z \in \mathcal{Z}$, we have g(Z) is a lower bound to (3.6) and thus the original QAP. We use the dual function value of the projection $g(\mathcal{P}_{\mathcal{Z}}(Z^{out})$ as the lower bound, and next we show how to get $\mathcal{P}_{\mathcal{Z}}(\tilde{Z})$ for any symmetric matrix $\tilde{Z}$.

Let $\hat{V}_\perp$ be the orthogonal complement to $\hat{V}$ so that $\bar{V} := (\hat{V}, \hat{V}_\perp)$ is orthogonal.

Let $\bar{V}Z\bar{V} = W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$ then we have

$\bar{V}Z\bar{V} \preceq 0$ if and only if $W_{11} \preceq 0$.

Thus

$$
\begin{array}{rcl}
\mathcal{P}_{\mathcal{Z}}(\tilde{Z}) = & \operatorname{argmin}_{Z\in\mathcal{Z}} \|Z - \tilde{Z}\| \\
= & \operatorname{argmin}_{W_{11}\preceq 0} \|\bar{V}W\bar{V}^T - \tilde{Z}\| \\
= & \operatorname{argmin}_{Z\in\mathcal{Z}} \|W - \bar{V}^T\tilde{Z}\bar{V}\| \\
= & \begin{bmatrix} -\mathcal{P}_{\mathcal{S}_+}(-W'_{11}) & W'_{12} \\ W'_{21} & W'_{22} \end{bmatrix}
\end{array}
$$

where

$$
\bar{V}^T\tilde{Z}\bar{V} = \begin{bmatrix} W'_{11} & W'_{12} \\ W'_{21} & W'_{22} \end{bmatrix}.
$$

Let $(R^{out}, Y^{out}, Z^{out})$ be the output of the ADMM for (3.6). Denote the largest eigenvalue and corresponding eigenvector of Y by λ and v respectively. Let

$$
X^{out} = \operatorname{Mat}(\lambda v v_{2:nk,0}^T) \tag{4.4}
$$

We then solve the nearest matrix problem

$$\begin{array}{ll} \min_X & \|X - X^{out}\| \\ \text{s.t} & X \in \mathcal{M}_m \end{array} . \tag{4.5}$$

We can do this very efficiently as seen from the following (and recalling $X^T X = \text{Diag}(m)$):

$$\begin{array}{rl} & \|X - X^{out}\| \\ = & \text{tr}\, X^T X + 2X^T X^{out}(X^{out})^T X^{out} \\ = & \text{constant} + \text{tr}\, 2X^T X^{out}. \end{array}$$

Thus it becomes equivalent to solve the following linear program:

$$\begin{array}{ll} \max_X & \langle X^{out}, X \rangle \\ \text{s.t.} & X \in \text{conv}(\mathcal{M}_m). \end{array}$$

In the following Lemma 4.1.2 we show that this is equivalent to the linear program (4.6).

$$\begin{array}{ll} \max_X & \langle X^{out}, X \rangle \\ \text{s.t.} & Xe = e, X^T e = m \\ & X \geq 0. \end{array} \tag{4.6}$$

This solution is labelled as the eig upper bound in table 1.

Lemma 4.1.2 ([13])**.** *The extreme points of the linear program* (4.6) *are partition matrices.*

Proof. Assume $\tilde{X}$ is an extreme point and $\tilde{X}$ has a non-integer entry in position (i_1, j_1). Since the row sums are integer, there is another non-integer entry in a position (i_2, j_2), in the same row or in the same column. Let us define (i_p, j_p) for $p > 1$ in the following manner: We choose (i_{p+1}, j_{p+1}) to be a non-integer entry in the same row as (i_p, j_p), if p is odd, and in the same column as (i_p, j_p) if p is even. Consider the path $P = (i_1, j_1)(i_2, j_2), ..$ Since we have finitely many edges, $((i_p, j_p), (i_{p+1}, j_{p+1})$ in our matrix this must cycle edgewise. To be clear, by cycling edgewise, we mean the path has a repeated edge. In this way, without loss of generality, we can construct a circuit on the non-integer entries $(i_1, j_1), .., (i_k, j_k)$ where k is even.(The path alternates vertical and horizontal edges so k must be even.) Now define a matrix $J(\delta)$ such that

$$J(\delta)_{a,b} = \begin{cases} 0 & \text{if}(a, b) \neq (i_p, j_p)\ \forall p, \quad 1 \leq p \leq k, \\ \delta(-1)^p & \text{else.} \end{cases}$$

Since we have a circuit this is unique. Now we claim $J(\delta)e = 0$ and $J(\delta)^T e = 0$. Indeed for any row a, we look at horizontal edges (in our cycle) $((i_p, j_p), (i_{p+1}, j_{p+1}))$ with $i_p = i_{p+1} = a$ (since our path alternates vertical and horizontal edges for any (i_p, j_p) we have exactly one of $((i_p, j_p), (i_p, j_{p+1}), ((i_p, j_{p-1}), (i_p, j_p))$ in our cycle that is each place is contained in exactly one horizontal edge. Then the sum along our row is the sum over all horizontal edges along row a of the sum of each horizontal edge with is 0.

$$\sum_{i_p = a} J(\delta)_{(i_p, j_p)} = \sum_{e=((a,j_p),(a,j_{p+1}))} \delta(-1)^p + \delta(-1)^{p+1} = 0$$

So the row sums of $J(\delta)$ are 0 and likewise the column sums of $J(\delta)$ are 0. Thus $Y(\delta) := \tilde{X} + J(\delta)$ satisfies $Y(\delta)e = e, Y(\delta)^T e = m$ now pick $\delta_0 = \min\{\tilde{X}_{(i_p,j_p)} \;\; p = 1, .., k\} > 0$ and the line segment $\{Y(\delta) : \delta \in [-\delta_0, \delta_0]\} \subset \{X \in \mathbb{R}^{n \times t} : \;\; Xe = e, X^T e = m, X \geq 0\}$. Thus $\tilde{X}$ is not an extreme point. We can thus conclude that $\{X \in \mathbb{R}^{n \times t} : \;\; Xe = e, X^T e = m, X \geq 0\} = \operatorname{conv}(M_m)$ □

Proposition 4.1.3. *Suppose that Y is feasible for (3.2). Let $v_1 = Y_{1:kn,0}$ and $\begin{pmatrix} v_0 & v_2^T \end{pmatrix}^T$ denote a unit eigenvector of Y corresponding to the largest eigenvalue. Then $X_1 := \operatorname{Mat}(v_1) \in \mathcal{E} \cap \mathcal{N}$. Moreover, if $v_0 \neq 0$, then $X_2 := \operatorname{Mat}(\frac{1}{v_0} v_2) \in \mathcal{E}$. Furthermore, if, $Y \geq 0$, then $v_0 \neq 0$ and $X_2 \in \mathcal{N}$.*

Proof. $X_1 \in \mathcal{E}$ was shown in the proof of Theorem 3.2.2. From the arrow constraint and $Y \geq 0$, $X_1 \in \mathcal{N}$. Let us now prove the results for X_2.

Note that for the spectral decomposition, $Y = \sum_{i=1}^{nk+1} \lambda_i u_i u_i^T$ $\operatorname{tr} D_i Y = 0$. Since D_j are PSD, $\lambda \operatorname{tr} D_j u_i u_i^T \geq 0$ and thus $\sum_{i=1}^{nk+1} \lambda \operatorname{tr} D_j u_i u_i^T = 0$ implies $\operatorname{tr} D_j u_i u_i^T = 0$. Since D_j are PSD this implies $\operatorname{tr} D_j u_i = 0 \;\; \forall i$.

We can see from simple algebra that,

$$0 = \operatorname{trace}(D_i [v_0 v_2]^T [v_0, v_2] = \begin{cases} \lambda_1(Y) v_0^2 \|X_2 e - e\|^2, & \text{if } i = 1, \\ \lambda_1(Y) v_0^2 \|X_2^T e - m\|^2, & \text{if } i = 2. \end{cases} \tag{4.7}$$

It thus follows that $X_2 \in \mathcal{E}$.

Finally, suppose that $Y \geq 0$. We claim that any eigenvector $\begin{pmatrix} v_0 & v_2^T \end{pmatrix}^T$ corresponding to the largest eigenvalue must satisfy:

1. $v_0 \neq 0$;

2. all entries have the same sign, i.e., $v_0 v_2 \geq 0$.

From these claims, it would follow immediately that $X_2 = \mathrm{Mat}(v_2/v_0) \in \mathcal{N}$.

Recall lemma 1.2.8: For a symmetric matrix $L \in \mathbb{R}^{n\times n}$ with g the eigenvector corresponding to the largest eigenvalue,
$g \in \mathrm{argmax}_{h:\ h^T h=1} h^T L h$.

This means that if $[v_0\ v_2]^T$ is an eigenvector corresponding to the largest eigenvalue of Y, then since Y has all positive entries, so is $[|v_0|\ |v_2|]^T$ (absolute value taken entry-wise). Thus $D_1[|v_0|\ |v_2|]^T = 0$ and the first row says $\sum_{i=1}^{nk} |(v_2)_i| = n|v_0|$ thus $v_0 \neq 0$
without loss of generality, let $v_0 > 0$. Then the first row of $D_1[|v_0|\ |v_2|]^T = 0 = D_1[v_0\ v_2]^T$ says that $\sum_{i=1}^{nk}(v_2)_i = nv_0 = \sum_{i=1}^{nk} |(v_2)_i|$ and thus $v_2 \geq 0$.

Now we prove $v_0 \neq 0$. Assume for a contradiction that $v_0 = 0$, Then the first row of $D_1[v_0 v_2]^T = 0$ says $\sum_{i=1}^{nk}(v_2)_i = 0$ since $v_2 \geq 0$, $v_2 = 0$, and our eigenvector $[v_0 v_2] = 0$ contradiction so $v_2 \neq 0$.

This completes the proof. □

Let Y feasible for (3.2) and define

$$X_1 := \mathrm{Mat}(Y_{1:nk,1}) \tag{4.8}$$

Likewise we can round this solution to the nearest partition matrix, this will be referred to as the row1 upper bound in table 1.

Low-rank solution

Note that in the SDP relaxation any rank 1 solution will be a partition matrix. Thus a naive idea is to modify ADMM to make R rank 1.We have no theoretical guarantee that this method will find a good solution. In fact, since our feasible reason is no longer convex, we don't even have any convergence guarantee for our algorithm. However, this idea turns out to be quite good in practice and despite no theoretical convergence guarantee, we have always had convergence in our examples.
Define:
$\mathcal{R}_1 = \{R \succeq 0 \quad \mathrm{rank}(R) = 1\}$

$$\begin{aligned} R_+ &= \mathrm{argmin}_{R\in\mathcal{R}_1} \mathcal{L}(R, Y, Z) \\ &= \mathcal{P}_{\mathcal{S}\cap\mathcal{R}_1}(\hat{V}^T(Y + \tfrac{Z}{B})\hat{V}) = \lambda_1 w w^T. \end{aligned}$$

Where λ_1 is the largest eigenvalue and w the corresponding eigenvector of $\hat{V}^T(Y + \frac{Z}{B})\hat{V}$.

Note the projection of a matrix M onto $\mathcal{R}_1$ is $\lambda w w^t$ where λ is the largest eigenvalue of

M and w the corresponding eigenvector. We now get the following algorithm for ADMM low rank.

Algorithm 4.1.3: ADMM low rank

Data: Input: Graph $G = (V, E)$ vector m

Result: Output: Solution for ADMM low rank

Input: $R_0, \quad Y_0, \quad Z_0$

feasible for 4.1, 4.2.

We also define a tolerance μ for the acceptable primal and dual residuals.

Initialize:

$$R = R_0, \quad R_+ = R_0$$
$$Y = Y_0, \quad Y_+ = Y_0$$
$$Z = Z_0, \quad Z_+ = Z_0$$

while $\|Y - Y_+\| + \|Y - \hat{V} R \hat{V}^T\| > \mu$ *&* **do**

Iterate by

$$\begin{aligned} R &= R_+, \quad Y = Y_+, \quad Z = Z_+ \\ R_+ &= \mathcal{P}_{\mathcal{R}_1}(\hat{V}^T (Y + \tfrac{1}{\beta} Z) \hat{V}) \\ Y_+ &= \mathcal{P}_{[0,1]^{J^c}}(\mathcal{G}_{J^c}(\hat{V} R_+ \hat{V}^T + \tfrac{L_G + Z}{B})) \\ Z_+ &= Z + \gamma\beta(Y_+ - \hat{V} R_+ \hat{V}^T) \end{aligned}$$

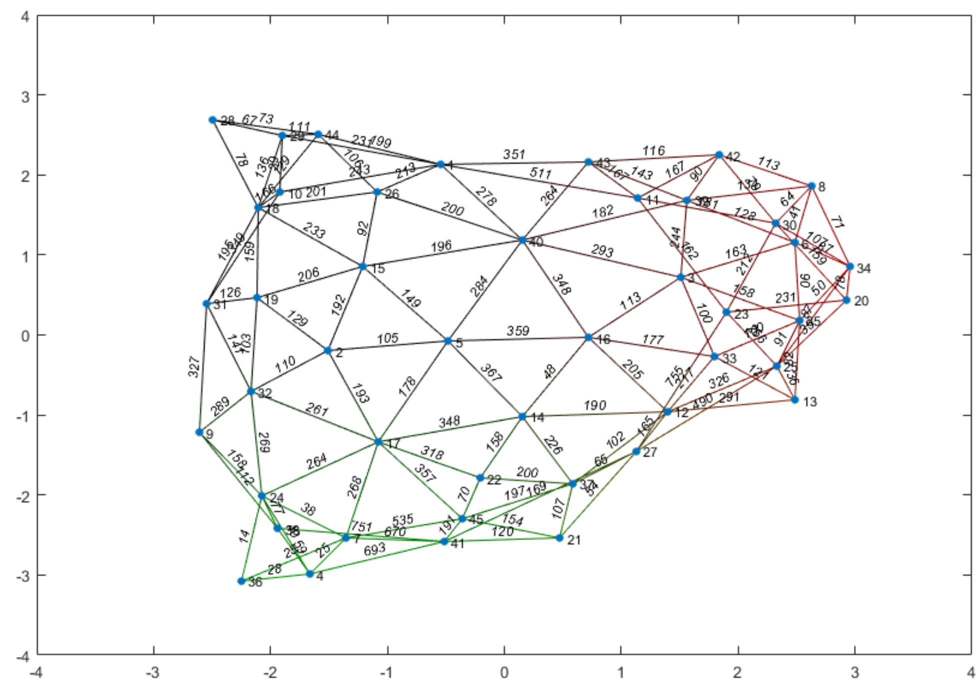
end while

Chapter 5

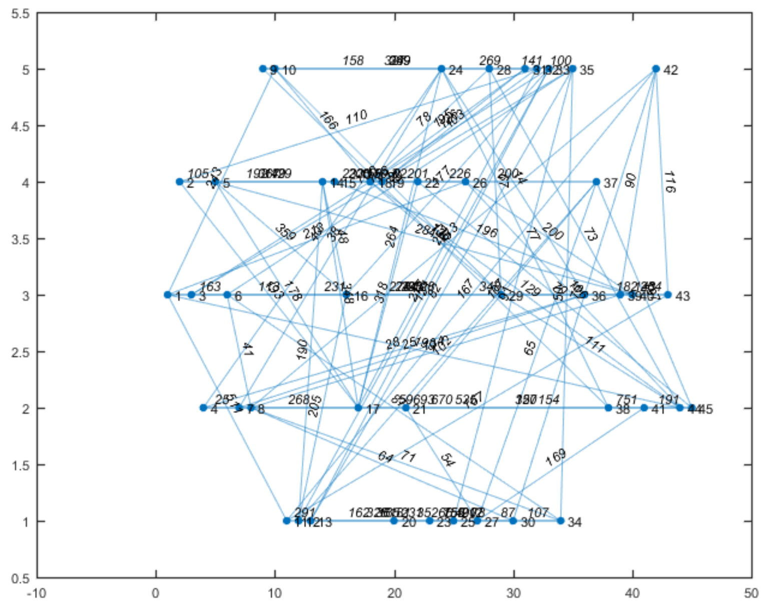
Numerics

5.1 Delaunay Triangulation Example

To give a more concrete feel, lets first do a simple example with the following weighted graph drawn below:



our ADMM code generated the following solution (low rank) with cut 8962:



5.2 ADMM Comparisons

In this thesis we generate problems with the MATLAB code called *newV0sparse* from [9] and we label the i-th problem using R$i(n, k)$; n, k being the number of vertices and number of sets, respectively. We then compare our ADMM code (which enforces the non-negativity constraint) with the *mysdp* (SDP bound) code (which does not enforce the non-negativity constraint) and *newV0sparse* (projected bound) code in [9] . We now present two tables with numerical experiments and report the bounds and the times.(Windows 10 machine, Intel i7 2.40 GHz, 8GB memory) We omitted the quadratic programming bounds as they are theoretically worse than the SDP bounds and in practice by a lot. Table 1 presents the comparisons for the lower bounds. Recall that low rank ADMM is a heuristic for finding good partition matrices and does not give a lower bound. Table 2 presents the comparisons for the upper bounds which are obtained by rounding to a partition matrix (4.5). Times are not shown for these because the codes in [9] do not display time for this rounding process. The eig upper bound and row1 upper bound refer to what is obtained by rounding (4.4) , (4.8) to the nearest partition matrix respectively.

	SDP & ADMM		Proj. Eig.		SDP & SDPT3	
Prob./size (vert.,sets)	High rank Lower bound	High rank cpu-sec	lower bound	cpu-sec	lower bound	cpu-sec
R1(20,5)	31	3	23	0.0347	27	0.49
R2(36,6)	47	16	19	0.0899	33	1.98
R3(32,6)	17	19	-7	0.37e	4	1.62
R4(81,8)	310	639	220.85	0.4	257	53.2
R5(26,6)	16	7	1.48	0.0892	9	0.84
R6(37,6)	54	28	22.67	0.371	37	1.62
R7(26,5)	7	5	-8.30	0.209	2	0.45
R8(50,6)	48	48	15.74	0.0366	36	3.30

Table 1: comparisons of our ADMM with: Projection and SDP bounds; 8 instances.

	SDP with ADMM			Projected upper bnd	SDPT3
Problem	High rank	High rank	Low rank	projected	SDP
	Eig upper bnd	Row1 upper bnd	upper bnd	upper bnd	upper bnd
R1(20,5)	40	37	35	38	40
R2(36,6)	61	61	57	80	67
R3(32,6)	19	19	27	33	34
R4(81,8)	390	385	366	408	415
R5(26,6)	22	22	21	31	25
R6(37,6)	61	65	67	86	74
R7(26,5)	8	8	10	21	12
R8(50,6)	59	62	60	91	75

Table 2: comparisons of our ADMM with: Projection, SDP bounds; 8 instances;

Let us define the relative gap as

$$\frac{1}{2} * \frac{(\text{best upper bound} - \text{ best lower bound})}{((\text{best upper bound} + \text{best lower bound} + 1)}$$

and we replace the lower bound by 0 if it is negative.

Conclusions from the numerics: On a random set of 200 problems (see Appendices section) generated with the newV0sparsetest.m from [9] (k=8, imax=8)high rank yields an average relative gap of 0.094, low rank yields an average gap of 0.1092 standard SDP yields a gap of 0.7058. If we take the better of low rank and high rank, we get a relative gap of 0.0870.

Lower bounds from ADMM beats all others significantly, 2.1826 times as much over the 200 random examples. Upper bounds from high rank ADMM are on average 0.7563 times the upper bound from mysdp.m code. Upper bound from low rank ADMM is on average 0.7897 times the upper bounds from the mysdp.m code. If we take the better of the high rank upper bound and low rank upper bound we can get on average 0.7332 the upper bound from the mysdp.m code.

Chapter 6

Conclusion

In this paper we showed how to extend [9] by the method used in [8], namely the use of ADMM to solve the SDP relaxation. We have analyzed the effectiveness of the ADMM method in solving the vertex separator problem and have seen significant improvements in the average lower bounds obtained through minimizing over the doubly non-negative matrices as well as improvements in the feasible solution obtained on average. We have also seen average improvements in the upper bound obtained. We also implemented the low rank ADMM method to solve the vertex separator problem. Similar to the QAP we always had convergence to a feasible solution that on average beats the feasible solution from the mysdp.m code in [9]. Computation times have not been so great.

As for applications, we hope that our algorithm could be used as a subroutine for the moats and control zones mentioned in [4].

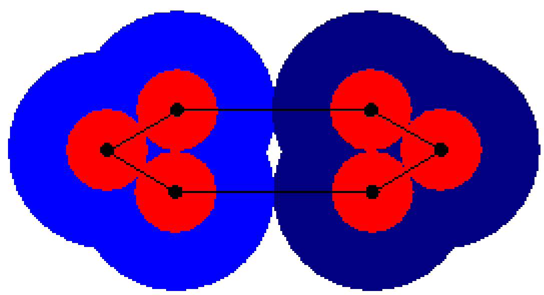
The control zone and moat problem is given by (for a graph $G = (V, E)$ with distances $d_{i,j}$ for edge $\{ i, j \}$):

$$\begin{array}{c} \max \sum_{v \in V} r_v + \sum_{i=1}^{k} \gamma(S_i) \\ s.t \\ r_i + r_j + \gamma(S_{\phi(i)}) + \gamma(S_{\phi(j)}) \leq d_{i,j} \quad \forall i, j \in V \\ S_1,\ S_2,\ ,\ldots,\ S_k \ \text{ form a partition of } \ V \\ k \in \mathbb{Z} \end{array} \tag{6.1}$$

Given a graph $G = (V, E)$ with distances $d_{i,j}$ for edge $\{ i, j \}$ the traveling salesman problem is the problem of finding a Hamiltonian cycle with minimum sum of edge distances. For simplicity let us assume that our graph is in the plane and the distances are given by the euclidean distances. Given a solution to (6.1), $r, \gamma(S_1), \gamma(S_2), \ldots \gamma(S_k)$, call the sets

$\{x \in \mathbb{R}^2 : \|x - v\| \leq r_v\}$ control zones. Define $Q_i := \cup_{v \in S_i} \{x \in \mathbb{R}^2 : \|x - v\| \leq r_v\}$. For $x \in \mathbb{R}^2$, $S \subset \mathbb{R}^2$ denote $d(x, S)$ as the distance between x and S. Define the ith moat to be $\{x \in \mathbb{R}^2 : d(x, Q_i) \leq \gamma(S_i)\} \backslash Q_i$. We can intuitively see that any tour of the vertices must "cross" each control zone twice and each moat at least twice. It thus follows that the minimum tour must be at least $2(\sum_{v \in V} r_v + \sum_{i=1}^{k} \gamma(S_i))$.

Figure 6.1: Moats in blue and control zones in red



Although this problem is not graph partitioning, we think that a good partition may be a good heuristic to choose the control zones and moats in the above problem. This is known in the literature as the clustering problem for the TSP.

6.1 Future Work

Recall in the standard ADMM method:
We are given an optimization problem:

$$\begin{gathered} \min F(x) \\ Ax = b \\ A \in \mathcal{R}^{m \times n} \end{gathered}$$

A recent paper of Xu. [16] proposes the following modification of ADMM:
They assume $F(x) = f(x) + g(x)$ where f is a convex Lipschitz differentiable function, and g is closed, convex, but not necessarily differentiable. Define the Lagrangian

$$\mathcal{L}(x, \lambda) = F(x) - \langle \lambda, Ax - b \rangle + \frac{\beta}{2} \|Ax = b\|.$$

Recall in classical ADMM, we do:
$x_+ \in \text{argmin}_{x\in\mathbb{R}^n} \mathcal{L}(x,\lambda)$
$\lambda + = \text{argmin}_{\lambda\in\mathcal{R}^m} \mathcal{L}(x,\lambda)$

They do two modifications, one of which is to "replace f by a quadratic function that dominates f around x" and replace the x iteration by:

$$\bar{x}^{(k+1)} \in \text{argmin}_{x\in\mathbb{R}^n}\langle \nabla f(x^{(k)})A^T\lambda, x\rangle + g(x) + \frac{\beta}{2}\|Ax-b\|^2 + \frac{1}{2}\|x - x^{(k)}\|$$

where $x^{(k)}$ is the previous x iterate. The other is to shorten the step taken by a parameter $0 < \alpha_k < 1$ and set $x^{(k+1)} = (1-\alpha_k)x^{(k)} + \alpha_k\bar{x}^{(k+1)}$. In [16] Xu shows the improved theoretical guarantee of convergence as well as faster convergence in practice. It may be of interest to investigate the application of this method for the vertex separator problem.

Appendix A

APPENDICES

The following 3 tables are the computational results of the mysdpbd.m code in [9] and our ADMM high rank and ADMM low rank codes run on the 200 random problems described at the end of chapter 5. The times for the three tables in this chapter are on linux machine (Four AMD Opteron 6168 12-core 2.3 GHz processors 256 GB memory) running Matlab 2015a. The primal and dual columns in the high rank ADMM & SDP tables show the primal and dual objective for the SDP relaxation of vertex separator, we see that the relaxations are solved to optimality. For the low rank ADMM the primal is the value of the solution returned, low rank does not solve the SDP relaxation to optimality. The eig upper bound and row1 upper bound refer to what is obtained by rounding (4.4) , (4.8) to the nearest partition matrix respectively. It is worth noticing that the objectives of the low rank solution and of the low rank solution rounded to the nearest partition matrix are the same. This is because practically low rank always gives a partition matrix as a solution.

A.1 SDP Table results

primal	dual	feasible solution value	time (sec)	relative gap
25.6056	25.6056	76	21.7881	0.49598
16.0102	16.0101	63	17.236	0.59473
14.906	14.9059	61	17.588	0.60725
6.0597	6.0597	46	16.6154	0.7672
-2.2	-2.2	36	13.4862	1.1302
34.7853	34.7853	79	16.7325	0.38858

primal	dual	feasible solution value	time (sec)	relative gap
6.7723	6.7723	46	16.8101	0.74334
13.289	13.2889	46	7.0255	0.55172
20.7797	20.7796	59	11.0698	0.47907
28.4773	28.4773	85	22.7246	0.4981
6.0027	6.0027	48	12.557	0.77769
3.757	3.757	40	9.519	0.82828
7.3684	7.3684	56	13.3145	0.76744
14.3961	14.3961	65	12.6046	0.63736
18.4327	18.4327	52	8.9001	0.47659
20.1462	20.1461	86	21.8965	0.62041
12.6877	12.6877	39	6.5709	0.50906
10.0164	10.0164	37	6.8117	0.57392
2.4957	2.4957	29	6.1804	0.84152
10.0878	10.0878	30	5.1542	0.49671
25.875	25.875	78	12.1977	0.5018
20.4926	20.4926	62	13.0747	0.50317
11.3253	11.3253	54	10.8998	0.65326
14.7707	14.7707	47	11.057	0.52176
17.8729	17.8729	59	12.0628	0.535
9.5508	9.5508	52	12.3869	0.68966
4.6589	4.6589	49	10.7309	0.82635
-0.23136	-0.23136	26	4.7517	1.018
15.8714	15.8714	48	7.8242	0.50302
19.9528	19.9528	53	8.3366	0.453
42.3511	42.351	93	13.7716	0.3742
6.1952	6.1952	34	7.3443	0.69175
6.7542	6.7542	40	12.3068	0.71107
2.9285	2.9285	36	10.9569	0.84954
-3.694	-3.694	16	6.9573	1.6004
12.3986	12.3986	38	8.1048	0.50798
8.2161	8.2161	51	15.3626	0.7225
25.166	25.166	86	18.3824	0.54724
19.1292	19.1292	55	10.2382	0.4839
7.7316	7.7315	40	9.1742	0.67604
37.3643	37.3642	86	12.6038	0.39425
7.6787	7.6787	56	17.0019	0.75883
19.7996	19.7995	76	15.9387	0.58665

primal	dual	feasible solution value	time (sec)	relative gap
31.9132	31.9132	84	13.0175	0.44936
12.5833	12.5832	35	6.1395	0.47111
30.7355	30.7355	71	12.47	0.39578
18.8378	18.8378	50	33008.7737	0.45269
23.8454	23.8454	59	14.3255	0.42434
3.9896	3.9896	45	11.1811	0.83712
-1.1021	-1.1021	38	12.0491	1.0597
2.2594	2.2594	40	11.3373	0.89307
6.3419	6.3419	34	10.3558	0.68559
6.2159	6.2159	40	9.0007	0.73101
15.4277	15.4277	44	7.6487	0.48079
26.6375	26.6375	67	19.5589	0.43105
6.1512	6.1512	42	12.9484	0.7445
9.5422	9.5422	31	7.8594	0.52927
7.0922	7.0922	48	12.6407	0.74253
19.0537	19.0536	83	17.1412	0.6266
9.2959	9.2959	43	12.219	0.64449
7.967	7.967	51	16.4913	0.72978
4.541	4.541	35	10.9512	0.77031
18.0038	18.0038	43	12.0752	0.40975
3.6515	3.6515	41	16.6829	0.83645
22.6228	22.6228	60	14.1623	0.45238
1.1287	1.1287	37	11.2685	0.9408
8.5467	8.5467	63	15.1189	0.76109
6.3639	6.3638	40	11.6117	0.72548
22.2574	22.2573	54	17.3522	0.41626
3.4047	3.4047	43	20.6531	0.85326
32.3944	32.3944	77	15.1939	0.40775
3.7159	3.7159	26	4.7137	0.74991
28.3601	28.3601	96	22.7295	0.5439
25.7132	25.7132	84	19.0446	0.53126
13.9214	13.9214	65	19.9934	0.64721
3.9448	3.9448	29	11.8062	0.76052
0.66771	0.6677	42	17.3504	0.9687
11.1774	11.1774	42	15.0126	0.57962
12.067	12.067	61	23.845	0.6697
11.4661	11.4661	60	21.337	0.67912

primal	dual	feasible solution value	time (sec)	relative gap
12.8236	12.8236	57	14.7693	0.63269
0.43701	0.43701	43	25.1326	0.97988
-2.6219	-2.6219	29	21.4077	1.1988
-0.19521	-0.19521	35	14.1603	1.0112
11.9371	11.9371	56	13.7564	0.64858
12.6499	12.6499	40	13.3298	0.51947
9.987	9.987	53	14.9278	0.68289
14.2441	14.2441	44	14.9058	0.51088
9.9282	9.9282	50	27.0636	0.66866
9.3992	9.3992	70	22.3315	0.76324
4.6652	4.6652	49	21.9119	0.82614
10.3108	10.3108	41	10.7249	0.5981
8.2098	8.2098	49	17.9481	0.71299
15.4428	15.4428	71	24.8277	0.6427
9.3992	9.3992	70	22.8693	0.76324
4.6652	4.6652	49	18.2819	0.82614
10.3108	10.3108	41	10.4506	0.5981
8.2098	8.2098	49	13.8565	0.71299
15.4428	15.4428	71	24.092	0.6427
22.7772	22.7772	60	12.3021	0.44967
-9.6574	-9.6574	18	8.6287	3.3152
8.6983	8.6983	50	20.3984	0.70363
15.1029	15.1028	55	13.0481	0.56912
5.5296	5.5295	22	4.6212	0.59828
14.3453	14.3453	57	14.0054	0.59786
13.5875	13.5875	68	17.6528	0.66692
2.2889	2.2889	55	15.454	0.92009
15.6981	15.6981	53	12.0118	0.54298
20.8527	20.8527	71	2644.0701	0.54595
6.9587	6.9586	37	14.8746	0.6834
12.6247	12.6246	58	15.2361	0.64249
6.6499	6.6499	38	8.7401	0.70213
42.5277	42.5276	99	18.4385	0.39902
28.0267	28.0267	71	15.6856	0.43396
28.5789	28.5789	92	25.7509	0.52597
13.7508	13.7508	64	15.4688	0.64628
4.8019	4.8019	43	9.5302	0.79909

primal	dual	feasible solution value	time (sec)	relative gap
26.2699	26.2699	83	19.9757	0.51917
8.4662	8.4662	51	16.6526	0.71526
48.4374	48.4374	101	16.6843	0.35174
11.495	11.495	55	13.5695	0.65426
5.3944	5.3944	26	6.7656	0.65635
12.5837	12.5837	46	9.0162	0.5704
-1.8463	-1.8463	36	10.6775	1.1081
0.039403	0.039403	25	8.2983	0.99685
2.4833	2.4832	38	11.1425	0.87732
29.2682	29.2682	75	12.1516	0.4386
14.6344	14.6344	70	22.4988	0.65417
-1.4886	-1.4886	35	10.8329	1.0888
-5.1986	-5.1986	18	6.6466	1.8122
9.7511	9.7511	56	14.96	0.70339
-2.9534	-2.9534	27	8.024	1.2456
9.823	9.823	46	10.2655	0.64807
1.0617	1.0617	47	12.7293	0.95582
19.7648	19.7648	76	21.0999	0.58722
17.089	17.089	68	15.6207	0.59833
-1.9116	-1.9117	39	12.2174	1.1031
-6.2933	-6.2933	24	5.8458	1.7108
8.0646	8.0646	50	11.1135	0.72222
20.0574	20.0574	54	9.2162	0.45833
26.1917	26.1917	65	10.106	0.42557
7.296	7.296	26	5.5811	0.56175
11.9759	11.9759	66	14.3899	0.69283
6.1512	6.1512	42	11.3333	0.7445
9.5422	9.5422	31	8.1894	0.52927
7.0922	7.0922	48	11.7986	0.74253
19.0537	19.0536	83	18.5613	0.6266
9.2959	9.2959	43	11.3158	0.64449
7.967	7.967	51	13.0724	0.72978
4.541	4.541	35	7.0071	0.77031
18.0038	18.0038	43	8.66	0.40975
3.6515	3.6515	41	12.9519	0.83645
22.6228	22.6228	60	11.4747	0.45238
1.1287	1.1287	37	10.7647	0.9408

primal	dual	feasible solution value	time (sec)	relative gap
8.5467	8.5467	63	12.7661	0.76109
6.3639	6.3638	40	10.0819	0.72548
22.2574	22.2573	54	9.7545	0.41626
3.4047	3.4047	43	13.6411	0.85326
32.3944	32.3944	77	12.5399	0.40775
28.3601	28.3601	96	20.8057	0.5439
25.7132	25.7132	84	17.1459	0.53126
13.9214	13.9214	65	17.6388	0.64721
3.9448	3.9448	29	8.4512	0.76052
0.66771	0.6677	42	10.4266	0.9687
11.1774	11.1774	42	10.5029	0.57962
12.067	12.067	61	15.9824	0.6697
11.4661	11.4661	60	17.1851	0.67912
0.43701	0.43701	43	13.8669	0.97988
-2.6219	-2.6219	29	12.7933	1.1988
-0.19521	-0.19521	35	9.4966	1.0112
11.9371	11.9371	56	11.4145	0.64858
12.6499	12.6499	40	11.5234	0.51947
9.987	9.987	53	8.3994	0.68289
14.2441	14.2441	44	7.7422	0.51088
9.9282	9.9282	50	15.308	0.66866
9.3992	9.3992	70	18.0932	0.76324
4.6652	4.6652	49	15.6296	0.82614
10.3108	10.3108	41	8.5881	0.5981
8.2098	8.2098	49	11.5335	0.71299
15.4428	15.4428	71	17.8176	0.6427
16.1695	16.1695	61	15.6613	0.58093
22.7772	22.7772	60	10.3401	0.44967
-9.6574	-9.6574	18	7.0812	3.3152
8.6983	8.6983	50	15.4019	0.70363
15.1029	15.1028	55	10.6644	0.56912
14.3453	14.3453	57	12.5569	0.59786
13.5875	13.5875	68	16.9515	0.66692
2.2889	2.2889	55	15.3838	0.92009
15.6981	15.6981	53	11.0877	0.54298
20.8527	20.8527	71	18.7013	0.54595
6.9587	6.9586	37	10.3716	0.6834

primal	dual	feasible solution value	time (sec)	relative gap
12.6247	12.6246	58	14.4722	0.64249
6.6499	6.6499	38	8.6082	0.70213
42.5277	42.5276	99	16.9126	0.39902
28.0267	28.0267	71	16.2909	0.43396
28.5789	28.5789	92	19.8273	0.52597
13.7508	13.7508	64	14.1735	0.64628
4.8019	4.8019	43	9.4063	0.79909
26.2699	26.2699	83	20.9013	0.51917
8.4662	8.4662	51	18.1648	0.71526

Table 1: mysdpbd.m output

A.2 ADMM Low Rank Table

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
56.000	43.106	56.000	56.000	8.2531	0.057051
50.000	33.520	50.000	50.000	7.5607	0.081583
50.000	32.952	50.000	50.000	11.692	0.084398
33.000	21.241	33.000	33.000	11.352	0.087754
21.000	13.416	21.000	21.000	6.5363	0.088181
70.000	50.457	70.000	70.000	9.7734	0.069300
32.000	20.889	32.000	32.000	41.728	0.085473
34.000	26.156	34.000	34.000	31.027	0.056838
50.000	35.510	50.000	50.000	37.034	0.071733
79.000	50.196	79.000	79.000	68.343	0.090580
39.000	24.532	39.000	39.000	42.416	0.091572
29.000	17.089	29.000	29.000	36.744	0.10094
55.000	26.168	55.000	55.000	46.629	0.12987
41.000	30.312	41.000	41.000	53.240	0.064383
42.000	29.449	42.000	42.000	31.790	0.073828
69.000	45.939	69.000	69.000	59.080	0.082953
33.000	24.679	33.000	33.000	28.644	0.062096
55.000	26.168	55.000	55.000	26.981	0.12987
41.000	30.312	41.000	41.000	26.937	0.064383
42.000	29.449	42.000	42.000	23.643	0.073828
69.000	45.939	69.000	69.000	38.442	0.082953
33.000	24.679	33.000	33.000	43.283	0.062096
32.000	22.362	32.000	32.000	39.149	0.074141
20.000	12.842	20.000	20.000	36.110	0.087293
25.000	18.756	25.000	25.000	44.382	0.061216
55.000	42.490	55.000	55.000	46.321	0.056353
50.000	34.592	50.000	50.000	36.813	0.076275
40.000	27.150	40.000	40.000	23.934	0.079323
40.000	25.656	40.000	40.000	28.058	0.088541
53.000	34.775	53.000	53.000	32.995	0.085165
33.000	22.386	33.000	33.000	47.037	0.079212
33.000	19.752	33.000	33.000	32.804	0.098863
19.000	10.470	19.000	19.000	42.919	0.10936

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
40.000	26.788	40.000	40.000	39.040	0.081553
43.000	30.736	43.000	43.000	26.867	0.070481
78.000	59.818	78.000	78.000	27.929	0.057904
27.000	19.908	27.000	27.000	47.328	0.064475
36.000	20.247	36.000	36.000	51.864	0.10790
30.000	16.662	30.000	30.000	34.995	0.10933
16.000	5.4917	16.000	16.000	34.368	0.15922
32.000	22.892	32.000	32.000	41.213	0.070062
44.000	25.701	44.000	44.000	48.843	0.10281
65.000	45.171	65.000	65.000	49.118	0.075682
58.000	34.706	58.000	58.000	42.362	0.099547
36.000	20.338	36.000	36.000	27.957	0.10728
73.000	55.002	73.000	73.000	41.243	0.061219
42.000	25.118	42.000	42.000	34.948	0.099303
68.000	40.140	68.000	68.000	38.191	0.10168
60.000	46.757	60.000	60.000	36.388	0.054723
28.000	21.106	28.000	28.000	38.708	0.060472
61.000	45.266	61.000	61.000	37.479	0.063960
48.000	32.272	48.000	48.000	38.788	0.081074
50.000	34.511	50.000	50.000	39.063	0.076677
31.000	18.915	31.000	31.000	33.437	0.095910
24.000	14.624	24.000	24.000	39.727	0.095676
28.000	17.109	28.000	28.000	36.869	0.095535
31.000	18.444	31.000	31.000	32.455	0.099649
31.000	18.292	31.000	31.000	40.456	0.10085
35.000	24.914	35.000	35.000	51.548	0.071026
56.000	38.704	56.000	56.000	37.553	0.076532
28.000	17.698	28.000	28.000	44.312	0.090371
27.000	18.665	27.000	27.000	30.715	0.075773
33.000	22.631	33.000	33.000	34.081	0.077381
59.000	40.408	59.000	59.000	42.644	0.078118
41.000	22.239	41.000	41.000	44.467	0.11302
30.000	23.823	30.000	30.000	36.007	0.050629
28.000	15.448	28.000	28.000	42.470	0.11011
38.000	27.281	38.000	38.000	37.863	0.069606
30.000	16.673	30.000	30.000	39.630	0.10924

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
47.000	34.586	47.000	47.000	43.108	0.065335
25.000	16.139	25.000	25.000	50.972	0.086877
41.000	26.361	41.000	41.000	27.479	0.088184
33.000	20.316	33.000	33.000	62.929	0.094656
42.000	34.833	42.000	42.000	57.839	0.042157
29.000	17.355	29.000	29.000	47.781	0.098685
59.000	46.428	59.000	59.000	33.339	0.052824
20.000	12.657	20.000	20.000	41.949	0.089545
76.000	49.441	76.000	76.000	37.030	0.086794
59.000	44.964	59.000	59.000	47.978	0.058974
59.000	31.532	59.000	59.000	48.933	0.11541
25.000	14.743	25.000	25.000	47.952	0.10056
28.000	13.896	28.000	28.000	45.646	0.12372
35.000	24.195	35.000	35.000	38.116	0.076095
50.000	29.071	50.000	50.000	33.600	0.10361
49.000	30.345	49.000	49.000	40.906	0.094219
45.000	29.702	45.000	45.000	41.742	0.084057
34.000	18.599	34.000	34.000	31.966	0.11160
20.000	10.338	20.000	20.000	31.138	0.11782
23.000	13.715	23.000	23.000	45.411	0.098772
40.000	27.672	40.000	40.000	58.497	0.076098
39.000	26.735	39.000	39.000	50.261	0.077629
34.000	25.120	34.000	34.000	34.687	0.064345
35.000	23.802	35.000	35.000	38.878	0.078858
42.000	26.043	42.000	42.000	65.528	0.093863
43.000	28.886	43.000	43.000	38.063	0.081113
41.000	22.716	41.000	41.000	35.297	0.11014
34.000	21.539	34.000	34.000	53.364	0.090299
35.000	21.091	35.000	35.000	37.243	0.097947
53.000	35.439	53.000	53.000	20.540	0.082061
56.000	38.067	56.000	56.000	48.823	0.079349
16.000	4.0589	16.000	16.000	51.294	0.18093
43.000	24.840	43.000	43.000	48.080	0.10437
45.000	30.725	45.000	45.000	44.269	0.078432
19.000	12.495	19.000	19.000	56.517	0.083399
46.000	32.362	46.000	46.000	40.259	0.073321

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
58.000	34.288	58.000	58.000	53.715	0.10133
42.000	21.926	42.000	42.000	33.443	0.11808
39.000	27.135	39.000	39.000	55.292	0.075095
66.000	40.165	66.000	66.000	53.539	0.097125
28.000	21.201	28.000	28.000	62.576	0.059638
44.000	26.316	44.000	44.000	46.117	0.099348
28.000	17.156	28.000	28.000	39.528	0.095125
75.000	59.603	75.000	75.000	61.635	0.050984
63.000	45.084	63.000	63.000	54.062	0.070534
72.000	47.656	72.000	72.000	52.251	0.083946
43.000	30.495	43.000	43.000	50.690	0.071868
30.000	17.847	30.000	30.000	26.084	0.099612
77.000	48.917	77.000	77.000	37.865	0.090590
39.000	25.138	39.000	39.000	38.489	0.087735
97.000	65.564	97.000	97.000	34.247	0.080606
42.000	28.634	42.000	42.000	36.128	0.078626
25.000	14.664	25.000	25.000	40.419	0.10134
35.000	25.345	35.000	35.000	61.881	0.067991
37.000	14.821	37.000	37.000	38.140	0.14786
24.000	11.653	24.000	24.000	27.714	0.12599
27.000	17.123	27.000	27.000	53.189	0.089787
50.000	41.951	50.000	50.000	32.421	0.039847
54.000	33.187	54.000	54.000	39.133	0.095471
27.000	12.978	27.000	27.000	44.271	0.12748
13.000	4.7049	13.000	13.000	59.840	0.15361
43.000	27.604	43.000	43.000	52.921	0.088483
22.000	8.9428	22.000	22.000	43.819	0.14508
33.000	21.388	33.000	33.000	28.474	0.086655
30.000	15.165	30.000	30.000	39.793	0.12160
60.000	38.563	60.000	60.000	34.687	0.088581
53.000	32.560	53.000	53.000	40.237	0.095515
28.000	15.666	28.000	28.000	26.112	0.10819
17.000	5.9473	17.000	17.000	46.578	0.15790
38.000	24.510	38.000	38.000	38.360	0.087599
42.000	32.268	42.000	42.000	33.777	0.057247
61.000	40.831	61.000	61.000	43.423	0.081987

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
23.000	16.902	23.000	23.000	61.544	0.064870
42.000	29.219	42.000	42.000	34.773	0.075180
25.000	17.698	25.000	25.000	44.219	0.071591
27.000	18.665	27.000	27.000	27.563	0.075773
33.000	22.631	33.000	33.000	31.489	0.077381
59.000	40.408	59.000	59.000	46.246	0.078118
41.000	22.239	41.000	41.000	45.024	0.11302
30.000	23.823	30.000	30.000	42.015	0.050629
28.000	15.448	28.000	28.000	45.297	0.11011
39.000	27.281	39.000	39.000	34.647	0.074173
30.000	16.673	30.000	30.000	36.164	0.10924
47.000	34.586	47.000	47.000	41.520	0.065335
25.000	16.139	25.000	25.000	45.251	0.086877
41.000	26.361	41.000	41.000	56.431	0.088184
33.000	20.316	33.000	33.000	59.950	0.094656
42.000	34.833	42.000	42.000	49.695	0.042157
29.000	17.355	29.000	29.000	33.002	0.098685
59.000	46.428	59.000	59.000	39.138	0.052824
76.000	49.441	76.000	76.000	34.450	0.086794
59.000	44.964	59.000	59.000	50.685	0.058974
56.000	31.532	56.000	56.000	45.334	0.10827
25.000	14.743	25.000	25.000	44.041	0.10056
28.000	13.896	28.000	28.000	38.835	0.12372
35.000	24.195	35.000	35.000	36.032	0.076095
50.000	29.071	50.000	50.000	39.011	0.10361
52.000	30.345	52.000	52.000	39.932	0.10312
34.000	18.599	34.000	34.000	35.748	0.11160
20.000	10.338	20.000	20.000	34.540	0.11782
23.000	13.715	23.000	23.000	53.619	0.098772
40.000	27.672	40.000	40.000	57.283	0.076098
39.000	26.735	39.000	39.000	49.986	0.077629
34.000	25.120	34.000	34.000	29.775	0.064345
35.000	23.802	35.000	35.000	37.931	0.078858
42.000	26.043	42.000	42.000	57.874	0.093863
43.000	28.886	43.000	43.000	51.426	0.081113
41.000	22.716	41.000	41.000	37.915	0.11014

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
34.000	21.539	34.000	34.000	29.795	0.090299
34.000	21.091	34.000	34.000	52.863	0.093540
53.000	35.439	53.000	53.000	44.583	0.082061
50.000	33.479	50.000	50.000	45.244	0.081786
56.000	38.067	56.000	56.000	53.537	0.079349
16.000	4.0589	16.000	16.000	52.329	0.18093
43.000	24.840	43.000	43.000	43.098	0.10437
45.000	30.725	45.000	45.000	59.806	0.078432
46.000	32.362	46.000	46.000	40.559	0.073321
59.000	34.288	59.000	59.000	47.203	0.10383
42.000	21.926	42.000	42.000	35.299	0.11808
39.000	27.135	39.000	39.000	48.811	0.075095
66.000	40.165	66.000	66.000	49.266	0.097125
28.000	21.201	28.000	28.000	58.847	0.059638
44.000	26.316	44.000	44.000	48.014	0.099348
28.000	17.156	28.000	28.000	35.552	0.095125
75.000	59.603	75.000	75.000	58.184	0.050984
63.000	45.084	63.000	63.000	54.326	0.070534
72.000	47.656	72.000	72.000	28.890	0.083946
43.000	30.495	43.000	43.000	11.879	0.071867
30.000	17.847	30.000	30.000	10.987	0.099612
77.000	48.917	77.000	77.000	22.969	0.090590
39.000	25.138	39.000	39.000	15.295	0.087734

Table 2: ADMM Low Rank Table

A.3 ADMM High Rank Table Results

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
43.106	43.106	65.000	61.000	122.90	0.085121
33.520	33.520	51.000	47.000	43.761	0.082678
32.952	32.952	48.000	48.000	47.179	0.091813
21.241	21.241	32.000	34.000	31.327	0.099178
13.416	13.416	21.000	25.000	11.381	0.10706
50.457	50.457	80.000	73.000	29.054	0.090563
20.888	20.889	32.000	41.000	85.115	0.10310
26.156	26.156	39.000	40.000	99.637	0.097071
35.510	35.510	47.000	46.000	46.829	0.063568
50.196	50.196	79.000	78.000	194.13	0.10761
24.532	24.532	35.000	34.000	156.06	0.079523
17.089	17.089	27.000	25.000	114.71	0.091795
26.168	26.168	42.000	44.000	177.01	0.11444
30.312	30.312	45.000	47.000	119.14	0.096234
29.449	29.449	46.000	43.000	85.340	0.092246
45.939	45.939	72.000	76.000	318.07	0.10956
24.679	24.679	29.000	29.000	73.549	0.039511
26.168	26.168	42.000	44.000	33.821	0.11444
30.312	30.312	45.000	45.000	60.116	0.096234
29.449	29.449	45.000	44.000	31.564	0.097722
45.939	45.939	70.000	76.000	58.942	0.10288
24.679	24.679	29.000	29.000	133.91	0.039511
22.362	22.362	36.000	35.000	111.57	0.10828
12.842	12.842	25.000	24.000	101.75	0.14743
18.756	18.756	25.000	25.000	57.999	0.069757
42.490	42.490	59.000	56.000	49.204	0.067898
34.592	34.592	49.000	51.000	49.354	0.085159
27.150	27.150	35.000	35.000	34.652	0.062157
25.656	25.656	37.000	36.000	94.310	0.082543
34.775	34.775	53.000	54.000	49.858	0.10265
22.386	22.386	42.000	44.000	146.99	0.14999
19.752	19.752	33.000	34.000	54.168	0.12323
10.470	10.470	17.000	17.000	77.430	0.11468

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
26.788	26.788	36.000	36.000	71.399	0.072204
30.736	30.736	44.000	49.000	31.773	0.087566
59.818	59.818	81.000	89.000	97.499	0.074680
19.908	19.908	28.000	32.000	152.15	0.082730
20.247	20.247	37.000	36.000	197.88	0.13759
16.662	16.662	35.000	34.000	49.807	0.16780
5.4917	5.4917	10.000	13.000	62.200	0.13668
22.892	22.892	26.000	35.000	113.24	0.031148
25.701	25.701	37.000	47.000	138.92	0.088692
45.171	45.171	65.000	66.000	92.254	0.089181
34.706	34.706	49.000	54.000	117.55	0.084374
20.338	20.338	32.000	35.000	29.164	0.10932
55.002	55.002	71.000	69.000	70.990	0.055992
25.118	25.118	35.000	33.000	57.525	0.066659
40.140	40.140	82.000	72.000	87.112	0.14080
46.757	46.757	71.000	66.000	149.57	0.084579
21.106	21.106	31.000	35.000	125.03	0.093151
45.266	45.266	65.000	68.000	59.718	0.088681
32.272	32.272	56.000	58.000	146.52	0.13290
34.511	34.511	51.000	52.000	131.96	0.095298
18.915	18.915	28.000	35.000	57.136	0.094798
14.624	14.624	22.000	22.000	70.303	0.098026
17.109	17.109	40.000	35.000	141.55	0.16844
18.444	18.444	29.000	29.000	108.64	0.10895
18.292	18.292	26.000	30.000	78.564	0.085086
24.914	24.914	31.000	32.000	192.02	0.053464
38.704	38.704	61.000	60.000	147.29	0.10680
17.698	17.698	31.000	28.000	237.80	0.11031
18.665	18.665	26.000	31.000	82.406	0.080314
22.631	22.631	41.000	41.000	61.423	0.14211
40.408	40.408	65.000	61.000	84.365	0.10054
22.239	22.239	28.000	28.000	167.14	0.056216
23.823	23.823	33.000	36.000	117.54	0.079352
15.448	15.448	20.000	20.000	122.13	0.062446
27.281	27.281	41.000	40.000	109.53	0.093140
16.673	16.673	28.000	30.000	142.31	0.12400

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
34.586	34.586	45.000	45.000	161.36	0.064611
16.139	16.139	26.000	28.000	146.09	0.11430
26.361	26.361	35.000	39.000	23.217	0.069262
20.316	20.316	34.000	34.000	213.51	0.12369
34.833	34.833	47.000	51.000	159.82	0.073440
17.355	17.355	36.000	34.000	103.05	0.15896
46.428	46.428	52.000	52.000	59.731	0.028020
12.657	12.657	20.000	16.000	153.76	0.056356
49.441	49.441	80.000	69.000	138.12	0.081877
44.964	44.964	66.000	58.000	147.56	0.062693
31.532	31.532	57.000	62.000	209.75	0.14223
14.743	14.743	22.000	22.000	183.16	0.096137
13.896	13.896	19.000	19.000	180.01	0.075296
24.195	24.195	34.000	35.000	344.77	0.082824
29.071	29.071	46.000	46.000	127.99	0.11127
30.345	30.345	54.000	58.000	122.87	0.13859
29.702	29.702	40.000	43.000	163.30	0.072829
18.599	18.599	29.000	28.000	56.161	0.098753
10.338	10.338	15.000	12.000	56.809	0.035598
13.715	13.715	21.000	20.000	187.79	0.090515
27.672	27.672	43.000	45.000	203.92	0.10693
26.735	26.735	35.000	34.000	190.41	0.058844
25.120	25.120	34.000	40.000	93.465	0.073849
23.802	23.802	29.000	29.000	73.423	0.048305
26.043	26.043	43.000	45.000	223.17	0.12105
28.886	28.886	48.000	46.000	211.73	0.11276
22.717	22.716	42.000	45.000	52.148	0.14672
21.539	21.539	24.000	25.000	87.250	0.026443
21.091	21.091	45.000	43.000	72.804	0.16829
35.439	35.439	56.000	52.000	23.835	0.093630
38.067	38.067	50.000	53.000	132.25	0.066987
4.0588	4.0589	8.0000	10.000	211.24	0.15090
24.840	24.840	44.000	42.000	204.48	0.12647
30.725	30.725	47.000	49.000	83.788	0.10336
12.495	12.495	16.000	17.000	237.53	0.059420
32.362	32.362	43.000	45.000	42.812	0.069653

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
34.288	34.288	51.000	56.000	205.33	0.096840
21.926	21.926	37.000	40.000	63.725	0.12577
27.135	27.135	43.000	49.000	212.84	0.11151
40.165	40.165	64.000	59.000	202.98	0.094021
21.201	21.201	33.000	31.000	245.65	0.092092
26.316	26.316	35.000	39.000	192.28	0.069677
17.156	17.156	27.000	27.000	150.76	0.10900
59.603	59.603	77.000	76.000	257.06	0.060018
45.084	45.084	57.000	52.000	220.56	0.035253
47.656	47.656	72.000	64.000	184.19	0.072541
30.495	30.495	48.000	50.000	201.26	0.11010
17.847	17.847	26.000	29.000	24.537	0.090894
48.917	48.917	72.000	72.000	86.043	0.094666
25.138	25.138	44.000	43.000	54.116	0.12918
65.564	65.564	93.000	94.000	132.08	0.085972
28.634	28.634	40.000	40.000	88.296	0.081615
14.664	14.664	20.000	19.000	52.959	0.062549
25.345	25.345	37.000	35.000	187.47	0.078691
14.821	14.821	34.000	32.000	98.030	0.17961
11.653	11.653	13.000	15.000	30.531	0.026258
17.123	17.123	25.000	21.000	58.952	0.049542
41.951	41.951	66.000	63.000	77.084	0.099335
33.187	33.187	50.000	48.000	50.719	0.090115
12.978	12.978	28.000	26.000	102.59	0.16287
4.7049	4.7049	7.0000	8.0000	265.37	0.090325
27.604	27.604	39.000	39.000	158.15	0.084285
8.9428	8.9428	13.000	14.000	117.57	0.088421
21.388	21.388	35.000	29.000	34.135	0.074061
15.165	15.165	25.000	29.000	101.68	0.11945
38.564	38.563	59.000	61.000	100.28	0.10367
32.560	32.560	51.000	43.000	90.267	0.068184
15.666	15.666	29.000	28.000	31.345	0.13807
5.9473	5.9473	12.000	11.000	137.43	0.14076
24.510	24.510	32.000	35.000	96.328	0.065121
32.268	32.268	43.000	44.000	78.231	0.070357
40.831	40.831	57.000	62.000	58.483	0.081800

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
16.902	16.902	27.000	27.000	177.69	0.11244
29.219	29.219	42.000	43.000	95.635	0.088485
17.698	17.698	32.000	28.000	142.38	0.11031
18.665	18.665	26.000	31.000	69.099	0.080314
22.631	22.631	41.000	41.000	35.000	0.14211
40.408	40.408	65.000	61.000	50.433	0.10054
22.239	22.239	28.000	28.000	93.228	0.056216
23.823	23.823	33.000	36.000	86.966	0.079352
15.448	15.448	20.000	20.000	120.00	0.062446
27.281	27.281	41.000	40.000	95.168	0.093140
16.673	16.673	28.000	30.000	88.866	0.12400
34.586	34.586	45.000	45.000	119.44	0.064611
16.139	16.139	26.000	29.000	106.83	0.11430
26.361	26.361	35.000	39.000	150.55	0.069262
20.316	20.316	34.000	34.000	149.98	0.12369
34.833	34.833	47.000	51.000	81.081	0.073440
17.355	17.355	36.000	34.000	43.535	0.15896
46.428	46.428	52.000	52.000	106.48	0.028020
49.441	49.441	80.000	69.000	96.043	0.081877
44.964	44.964	66.000	58.000	124.13	0.062693
31.532	31.532	57.000	62.000	141.97	0.14223
14.743	14.743	22.000	22.000	118.76	0.096137
13.896	13.896	19.000	19.000	213.86	0.075296
24.195	24.195	34.000	35.000	79.791	0.082824
29.071	29.071	46.000	46.000	102.99	0.11127
30.345	30.345	55.000	56.000	120.28	0.14277
18.599	18.599	29.000	28.000	44.027	0.098753
10.338	10.338	15.000	12.000	41.432	0.035598
13.715	13.715	21.000	20.000	148.17	0.090515
27.672	27.672	43.000	45.000	181.84	0.10693
26.735	26.735	35.000	34.000	154.85	0.058844
25.120	25.120	34.000	40.000	67.098	0.073849
23.802	23.802	29.000	29.000	40.283	0.048305
26.043	26.043	43.000	45.000	166.00	0.12105
28.886	28.886	48.000	46.000	123.92	0.11276
22.717	22.716	42.000	45.000	152.10	0.14672

			feasible solution value		
primal	dual	Largest Eig	Row1	time (sec)	relative gap
21.539	21.539	24.000	25.000	34.813	0.026443
21.091	21.091	45.000	43.000	84.098	0.16829
35.439	35.439	56.000	52.000	56.498	0.093630
33.479	33.479	47.000	47.000	125.36	0.082970
38.067	38.067	49.000	53.000	156.22	0.062071
4.0588	4.0589	8.0000	10.000	159.47	0.15090
24.840	24.840	44.000	42.000	59.468	0.12647
30.725	30.725	47.000	49.000	181.24	0.10336
32.362	32.362	43.000	45.000	54.823	0.069653
34.288	34.288	53.000	55.000	230.39	0.10597
21.926	21.926	37.000	37.000	40.910	0.12577
27.135	27.135	43.000	49.000	124.85	0.11151
40.165	40.165	64.000	59.000	135.33	0.094021
21.201	21.201	33.000	31.000	193.63	0.092092
26.316	26.316	36.000	39.000	108.08	0.076473
17.156	17.156	27.000	27.000	92.775	0.10900
59.603	59.603	79.000	81.000	185.86	0.069473
45.084	45.084	57.000	52.000	174.90	0.035253
47.656	47.656	77.000	66.000	164.65	0.079998
30.495	30.495	48.000	49.000	157.58	0.11010
17.847	17.847	27.000	26.000	49.861	0.090893
48.917	48.917	72.000	71.000	204.16	0.091314
25.138	25.138	39.000	39.000	60.715	0.10641

Table 3: ADMM High Rank Table

Index

References

[1] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81.

[2] R.A. Brualdi and H.J. Ryser. *Combinatorial Matrix Theory.* Cambridge University Press, New York, 1991.

[3] Y-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein's 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013.

[4] W. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation.* Princeton University Press, 2012.

[5] A.J. Hoffman and H.W. Wielandt. The variation of the spectrum of a normal matrix. *Duke Mathematics*, 20:37–39, 1953.

[6] R.H. Lewis. Yet another graph partitioning problem is NP-Hard. Report arXiv:1403.5544, [cs.CC], 2014.

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

[8] D.E. Oliveira, H. Wolkowicz, and Y. Xu. ADMM for the SDP relaxation of the QAP. Technical report, University of Waterloo, Waterloo, Ontario, 2015. arXiv:1512.05448.

[9] T.K. Pong, H. Sun, N. Wang, and H. Wolkowicz. Eigenvalue, quadratic programming, and semidefinite programming relaxations for a cut minimization problem. *Comput. Optim. Appl.*, 63(2):333–364, 2016.

[10] Alex Pothen. Graph partitioning algorithms with applications to scientific computing. In *Parallel numerical algorithms (Hampton, VA, 1994)*, volume 4 of *ICASE/LaRC Interdiscip. Ser. Sci. Eng.*, pages 323–368. Kluwer Acad. Publ., Dordrecht, 1997.

[11] F. Rendl, A. Lisser, and M. Piacentini. Bandwidth, vertex separators and eigenvalue optimization. In *Discrete Geometry and Optimization*, volume 69 of *The Fields Institute for Research in Mathematical Sciences, Communications Series*, pages 249–263. Springer, 2013.

[12] F. Rendl and H. Wolkowicz. A projection technique for partitioning the nodes of a graph. *Ann. Oper. Res.*, 58:155–179, 1995. Applied mathematical programming and modeling, II (APMOD 93) (Budapest, 1993).

[13] A Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication.

[14] L Tunçel. *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*, volume 27 of *Fields Institute Monographs.* American Mathematical Society, Providence, RI, 2010.

[15] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. Selected for the special Editors' Choice, Edition 1999.

[16] Y. Xu. Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *ArXiv e-prints*, June 2016.

[17] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996).