Eigenvalue, Quadratic Programming, and Semidefinite Programming Relaxations for a Cut Minimization Problem *

Ting Kei Pong † Hao Sun ‡ Ningchuan Wang § Henry Wolkowicz ¶

November 18, 2014

University of Waterloo
Department of Combinatorics & Optimization
Waterloo, Ontario N2L 3G1, Canada
Research Report

Key words and phrases: vertex separators, eigenvalue bounds, semidefinite programming bounds, graph partitioning, large scale.

AMS subject classifications: 05C70, 15A42, 90C22, 90C27, 90C59

Abstract

We consider the problem of partitioning the node set of a graph into k sets of given sizes in order to *minimize the cut* obtained using (removing) the k-th set. If the resulting cut has value 0, then we have obtained a vertex separator. This problem is closely related to the graph partitioning problem. In fact, the model we use is the same as that for the graph partitioning problem except for a different *quadratic* objective function. We look at known and new bounds obtained from various relaxations for this NP-hard problem. This includes: the standard eigenvalue bound, projected eigenvalue bounds using both the adjacency matrix and the Laplacian, quadratic programming (QP) bounds based on recent successful QP bounds for the quadratic assignment problems, and semidefinite programming bounds. We include numerical tests for large and *huge* problems that illustrate the efficiency of the bounds in terms of strength and time.

*Presented at Retrospective Workshop on Discrete Geometry, Optimization and Symmetry, November 24-29, 2013, Fields Institute, Toronto, Canada.

†Department of Applied Mathematics, the Hong Kong Polytechnic University, Hung Hom, Hong Kong. This author was also supported as a PIMS postdoctoral fellow at Department of Computer Science, University of British Columbia, Vancouver, during the early stage of the preparation of the manuscript. Email: tk.pong@polyu.edu.hk

‡Department of Combinatorics and Optimization, University of Waterloo, Ontario N2L 3G1, Canada. Research supported by an Undergraduate Student Research Award from The Natural Sciences and Engineering Research Council of Canada. Email: hao_sun@live.com

§Research supported by The Natural Sciences and Engineering Research Council of Canada and by the U.S. Air Force Office of Scientific Research. Email: wangningchuan1987@hotmail.com

¶Research supported in part by The Natural Sciences and Engineering Research Council of Canada and by the U.S. Air Force Office of Scientific Research. Email: hwolkowicz@uwaterloo.ca

Contents

List of Tables

List of Figures

1 Introduction

We consider a special type of *minimum cut problem, MC*. The problem consists in partitioning the node set of a graph into k sets of given sizes in order to *minimize the cut* obtained by removing the k-th set. This is achieved by minimizing the number of edges connecting distinct sets after removing the k-th set, as described in [20]. This problem arises when finding a re-ordering to bring the sparsity pattern of a large sparse positive definite matrix into a block-arrow shape so as to minimize fill-in in its Cholesky factorization. The problem also arises as a subproblem of the *vertex separator problem, VS*. In more detail, a vertex separator is a set of vertices whose removal from the graph results in a disconnected graph with $k-1$ components. A typical VS problem has $k=3$ on a graph with n nodes, and it seeks a vertex separator which is optimal subject to some constraints on the partition size. This problem can be solved by solving an MC for each possible partition size. Since there are at most $\binom{n-1}{2}$ 3-tuple integers that sum up to n, and it is known that VS is NP-hard in general [16, 20], we see that MC is also NP-hard when $k \geq 3$.

Our MC problem is closely related to the *graph partitioning problem, GP*, which is also NP-hard; see the discussions in [16]. In both problems one can use a model with a *quadratic* objective function over the set of *partition matrices*. The model we use is the same as that for GP except that the quadratic objective function is different. We study both existing and new bounds and provide both theoretical properties and empirical results. Specifically, we adapt and improve known techniques for deriving lower bounds for GP to derive bounds for MC. We consider eigenvalue bounds, a convex quadratic programming, QP, lower bound, as well as lower bounds based on semidefinite programming, SDP, relaxations.

We follow the approaches in [12, 20, 22] for the eigenvalue bounds. In particular, we replace the standard quadratic objective function for GP, e.g., [12, 22] with that used in [20] for MC. It is shown in [20] that one can equally use either the adjacency matrix A or the negative Laplacian $(-L)$ in the objective function of the model. We show in fact that one can use $A - \mathrm{Diag}(d), \forall d \in \mathbb{R}^n$, in the model, where $\mathrm{Diag}(d)$ denotes the diagonal matrix with diagonal d. However, we emphasize and show that this is no longer true for the eigenvalue bounds and that using $d = 0$ is, empirically, stronger. Dependence of the eigenvalue lower bound on diagonal perturbations was also observed for the quadratic assignment problem, QAP, and GP, see e.g., [10, 21]. In addition, we find a new projected eigenvalue lower bound using A that has three terms that can be found explicitly and efficiently. We illustrate this empirically on large and huge scale sparse problems.

Next, we extend the approach in [1, 2, 5] from the QAP to MC. This allows for a QP bound that is based on SDP duality and that can be solved efficiently. The discussion and derivation of this lower bound is new even in the context of GP. Finally, we follow and extend the approach in [28] and derive and test SDP relaxations. In particular, we answer a question posed in [28] about redundant constraints. This new result simplifies the SDP relaxations even in the context of GP.

1.1 Outline

We continue in Section 2 with preliminary descriptions and results on our special MC. This follows the approach in [20]. In Section 3 we outline the basic eigenvalue bounds and then the projected eigenvalue bounds following the approach in [12, 22]. Theorem 3.7 includes the projected bounds along with our new three part eigenvalue bound. The three part bound can be calculated explicitly and efficiently by finding $k-1$ eigenvalues and a minimal scalar product, and making use of the result in Section 3.2.1. The QP bound is described in Section 4. The SDP bounds are presented in

Section 5.

Upper bounds using feasible solutions are given in Section 6. Our numerical tests are in Section 7. Our concluding remarks are in Section 8.

2 Preliminaries

We are given an undirected graph $\mathsf{G} = (N, E)$ with a nonempty node set $N = \{1, \dots, n\}$ and a nonempty edge set E. In addition, we have a positive integer vector of set sizes $m = (m_1, \dots, m_k)^T \in \mathbb{Z}_+^k$, $k > 2$, such that the sum of the components $m^T e = n$. Here e is the vector of ones of appropriate size. Further, we let $\mathrm{Diag}(v)$ denote the diagonal matrix formed using the vector v; the adjoint $\mathrm{diag}(Y) = \mathrm{Diag}^*(Y)$ is the vector formed from the diagonal of the square matrix Y. We let $\mathrm{ext}(K)$ represent the extreme points of a convex set K. We let $x = \mathrm{vec}(X) \in \mathbb{R}^{nk}$ denote the vector formed (columnwise) from the matrix X; the adjoint and inverse is $\mathrm{Mat}(x) \in \mathbb{R}^{n\times k}$. We also let $A \otimes B$ denote the Kronecker product; and $A \circ B$ denote the Hadamard product.

We let

$$P_m := \left\{(S_1, \dots, S_k) : S_i \subset N, |S_i| = m_i, \forall i, S_i \cap S_j = \emptyset, \text{ for } i \neq j, \cup_{i=1}^k S_i = N\right\}$$

denote the set of all *partitions of* N with the appropriate sizes specified by m. The partitioning is encoded using an $n \times k$ *partition matrix* $X \in \mathbb{R}^{n\times k}$ where the column $X_{:j}$ is the incidence vector for the set S_j

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

Therefore, the set cardinality constraints are given by $X^T e = m$; while the constraints that each vertex appears in exactly one set is given by $Xe = e$.

The set of partition matrices can be represented using various linear and quadratic constraints. We present several in the following. In particular, we phrase the linear equality constraints as quadratics for use in the Lagrangian relaxation below in Section 5.

Definition 2.1. *We denote the set of zero-one, nonnegative, linear equalities, doubly stochastic type, m-diagonal orthogonality type, e-diagonal orthogonality type, and gangster constraints as, respectively,*

$$\begin{array}{rcl}
\mathcal{Z} & := & \{X \in \mathbb{R}^{n\times k} : X_{ij} \in \{0,1\}, \forall ij\} = \{X \in \mathbb{R}^{n\times k} : (X_{ij})^2 = X_{ij}, \forall ij\} \\
\mathcal{N} & := & \{X \in \mathbb{R}^{n\times k} : X_{ij} \geq 0, \forall ij\} \\
\mathcal{E} & := & \{X \in \mathbb{R}^{n\times k} : Xe = e, X^T e = m\} = \{X \in \mathbb{R}^{n\times k} : \|Xe - e\|^2 + \|X^T e - m\|^2 = 0\} \\
\mathcal{D} & := & \{X \in \mathbb{R}^{n\times k} : X \in \mathcal{E} \cap \mathcal{N}\} \\
\mathcal{D}_O & := & \{X \in \mathbb{R}^{n\times k} : X^T X = \mathrm{Diag}(m)\} \\
\mathcal{D}_e & := & \{X \in \mathbb{R}^{n\times k} : \mathrm{diag}(XX^T) = e\} \\
\mathcal{G} & := & \{X \in \mathbb{R}^{n\times k} : X_{:i} \circ X_{:j} = 0, \forall i \neq j\}
\end{array}$$

There are many equivalent ways of representing the set of all partition matrices. Following are a few.

Proposition 2.2. *The set of partition matrices in $\mathbb{R}^{n\times k}$ can be expressed as the following.*

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

Proof. The first equality follows immediately from the definitions. The second equality follows from the transportation type constraints and is a simple consequence of Birkhoff and Von Neumann theorems that the extreme points of the set of doubly stochastic matrices are the permutation matrices, see e.g., [23]. The third equality is shown in [20, Prop. 1]. The fourth and fifth equivalences contain redundant sets of constraints. □

We let $\delta(S_i, S_j)$ denote the set of edges between the sets of nodes S_i, S_j, and we denote the set of edges with endpoints in distinct partition sets $S_1, \ldots, S_{k-1}$ by

$$\delta(S) = \cup_{i<j<k}\delta(S_i, S_j). \tag{2.2}$$

The minimum of the cardinality $|\delta(S)|$ is denoted

$$\operatorname{cut}(m) = \min\{|\delta(S)| : S \in P_m\}. \tag{2.3}$$

The graph G has a *vertex separator* if there exists an $S \in P_m$ such that the removal of set S_k results in the sets $S_1, \ldots, S_{k-1}$ being pairwise disjoint. This is equivalent to $\delta(S) = \emptyset$, i.e., $\operatorname{cut}(m) = 0$. Otherwise, $\operatorname{cut}(m) > 0$.[1]

We define the $k \times k$ matrix

$$B := \begin{bmatrix} ee^T - I_{k-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{S}^k,$$

where $\mathcal{S}^k$ denotes the vector space of $k \times k$ symmetric matrices equipped with the trace inner-product, $\langle S, T\rangle = \operatorname{trace} ST$. We let A denote the adjacency matrix of the graph and let $L := \operatorname{Diag}(Ae) - A$ be the Laplacian.

In [20, Prop. 2], it was shown that $|\delta(S)|$ can be represented in terms of a quadratic function of the partition matrix X, i.e., as $\frac{1}{2}\operatorname{trace}(-L)XBX^T$ and $\frac{1}{2}\operatorname{trace} AXBX^T$, where we note that the two matrices A and $-L$ differ only on the diagonal. From their proof, it is not hard to see that their result can be slightly extended as follows.

Proposition 2.3. *For a partition $S \in P_m$, let $X \in \mathcal{M}_m$ be the associated partition matrix. Then*

$$|\delta(S)| = \frac{1}{2}\operatorname{trace}\left(A - \operatorname{Diag}(d)\right)XBX^T, \quad \forall d \in \mathbb{R}^n. \tag{2.4}$$

In particular, setting $d = 0, Ae$, respectively yields $A, -L$.

[1] A discussion of the relationship of $\operatorname{cut}(m)$ with the bandwidth of the graph is given in e.g., [8,18,20]. Particularly, for $k = 3$, if $\operatorname{cut}(m) > 0$, then $m_3 + 1$ is a lower bound for the bandwidth.

Proof. The result for the choices of $d = 0, Ae$, equivalently $A, -L$, respectively, was proved in [20, Prop. 2]. Moreover, as noted in the proof of [20, Prop. 2], $\operatorname{diag}(XBX^T) = 0$. Consequently,

$$\frac{1}{2}\operatorname{trace} AXBX^T = \frac{1}{2}\operatorname{trace}(A - \operatorname{Diag}(d))XBX^T, \quad \forall d \in \mathbb{R}^n.$$

□

In this paper we focus on the following problem given by (2.3) and (2.4):

$$\begin{array}{rcl} \operatorname{cut}(m) = & \min & \frac{1}{2}\operatorname{trace}(A - \operatorname{Diag}(d))XBX^T \\ & \text{s.t.} & X \in \mathcal{M}_m; \end{array} \tag{2.5}$$

here $d \in \mathbb{R}^n$. We recall that if $\operatorname{cut}(m) = 0$, then we have obtained a vertex separator, i.e., removing the k-th set results in a graph where the first $k-1$ sets are disconnected. On the other hand, if we find a positive lower bound $\operatorname{cut}(m) \geq \alpha > 0$, then no vertex separator can exist for this m. This observation can be employed in solving some classical vertex separator problems, which look for an "optimal" vertex separator in the case $k = 3$ under constraints on (m_1, m_2, m_3). Specifically, since there are at most $\binom{n-1}{2}$ 3-tuple integers summing up to n, one only needs to consider at most $\binom{n-1}{2}$ different MC problems in order to find the *optimal* vertex separator.

Though any choice of $d \in \mathbb{R}^n$ is equivalent for (2.5) on the feasible set $\mathcal{M}_m$, as we shall see repeatedly throughout the paper, this does *not* mean that they are equivalent on the relaxations that we look at below. We would also like to mention that similar observations concerning diagonal perturbation were previously made for the QAP, the GP and their relaxations, see e.g., [10, 21]. Finally, note that the feasible set of (2.5) is the same as that of the GP, see e.g., [22, 28] for the projected eigenvalue bound and the SDP bound, respectively. Thus, the techniques for deriving bounds for MC can be adapted to obtain new results concerning lower bounds for GP.

3 Eigenvalue Based Lower Bounds

We now present bounds on $\operatorname{cut}(m)$ based on $X \in \mathcal{D}_O$, the m-diagonal orthogonality type constraint $X^TX = \operatorname{Diag}(m)$. For notational simplicity, from now on, we define $M := \operatorname{Diag}(m)$, $\tilde{m} := \left(\sqrt{m_1}, \ldots, \sqrt{m_k}\right)^T$ and $\tilde{M} := \operatorname{Diag}(\tilde{m})$. For a real symmetric matrix $C \in \mathcal{S}^t$, we let

$$\lambda_1(C) \geq \lambda_2(C) \geq \cdots \geq \lambda_t(C)$$

denote the eigenvalues of C in nonincreasing order, and set $\lambda(C) = (\lambda_i(C)) \in \mathbb{R}^t$.

3.1 Basic Eigenvalue Lower Bound

The Hoffman-Wielandt bound [14] can be applied to get a simple eigenvalue bound. In this approach, we solve the relaxed problem

$$\begin{array}{rcl} \operatorname{cut}(m) \geq & \min & \frac{1}{2}\operatorname{trace} GXBX^T \\ & \text{s.t.} & X \in \mathcal{D}_O, \end{array} \tag{3.1}$$

where $G = G(d) = A - \operatorname{Diag}(d)$, $d \in \mathbb{R}^n$. We first introduce the following definition.

Definition 3.1. *For two vectors x, $y \in \mathbb{R}^n$, the minimal scalar product is defined by*

$$\langle x, y\rangle_- := \min\left\{\sum_{i=1}^{n} x_{\phi(i)}y_i : \ \phi \text{ is a permutation on } N\right\}.$$

We will also need the following two auxiliary results.

Theorem 3.2 (Hoffman and Wielandt [14])**.** *Let C and D be symmetric matrices of orders n and k, respectively, with $k \le n$. Then*

$$\min\left\{\operatorname{trace} CXDX^T : X^T X = I_k\right\} = \left\langle \lambda(C), \begin{pmatrix}\lambda(D)\\ 0\end{pmatrix}\right\rangle_- . \tag{3.2}$$

The minimum on the left is attained for $X = \begin{bmatrix} p_{\phi(1)} & \dots & p_{\phi(k)}\end{bmatrix} Q^T$, where $p_{\phi(i)}$ is a normalized eigenvector to $\lambda_{\phi(i)}(C)$, the columns of $Q = \begin{bmatrix} q_1 & \dots & q_k\end{bmatrix}$ consist of the normalized eigenvectors q_i of $\lambda_i(D)$, and ϕ is the permutation of $\{1, \dots, n\}$ attaining the minimum in the minimal scalar product. □

Lemma 3.3 ([20, Lemma 4])**.** *The k-ordered eigenvalues of the matrix $\tilde{B} := \tilde{M}B\tilde{M}$ satisfy*

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

We now present the basic eigenvalue lower bound, which turns out to always be negative.

Theorem 3.4. *Let $d \in \mathbb{R}^n$, $G = A - \operatorname{Diag}(d)$. Then*

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

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 (3.1). Then the constraint on X implies that $Z^T Z = I$. We now solve the equivalent problem to (3.1):

$$\begin{array}{ll} \min & \frac{1}{2}\operatorname{trace} GZ(\tilde{M}B\tilde{M})Z^T \\ \text{s.t.} & Z^T Z = I. \end{array} \tag{3.3}$$

The optimal value is obtained using the minimal scalar product of eigenvalues as done in the Hoffman-Wielandt result, Theorem 3.2. From this we conclude immediately that $\operatorname{cut}(m) \ge p^*_{eig}(G)$. Furthermore, the explicit formula for the minimal scalar product follows immediately from Lemma 3.3.

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

$$\sum_{i=1}^{k} \lambda_{\psi(i)}(G)\lambda_i(\tilde{B}) \ge \sum_{i=1}^{k} \lambda_{\widehat{\phi}(i)}(G)\lambda_i(\tilde{B}). \tag{3.4}$$

Now if $\mathcal{T}$ is the set of all permutations of $\{1,2,\dots,n\}$, then we have

$$\sum_{\psi\in\mathcal{T}}\left(\sum_{i=1}^{k}\lambda_{\psi(i)}(G)\lambda_i(\tilde{B})\right)=\sum_{i=1}^{k}\left(\sum_{\psi\in\mathcal{T}}\lambda_{\psi(i)}(G)\right)\lambda_i(\tilde{B})=\left(\sum_{\psi\in\mathcal{T}}\lambda_{\psi(1)}(G)\right)\left(\sum_{i=1}^{k}\lambda_i(\tilde{B})\right)=0, \tag{3.5}$$

since $\sum_{\psi\in\mathcal{T}}\lambda_{\psi(i)}(G)$ is independent of i. This means that there exists at least one permutation ψ so that $\sum_{i=1}^{k}\lambda_{\psi(i)}(G)\lambda_i(\tilde{B})\le 0$, which implies that the minimal scalar product must satisfy $\sum_{i=1}^{k}\lambda_{\widehat{\phi}(i)}(G)\lambda_i(\tilde{B})\le 0$. Moreover, in view of (3.4) and (3.5), 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 $\psi\in\mathcal{T}$. Recall from Lemma 3.3 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 (3.4). Thus $p^*_{eig}(G)$ is strictly negative.

Finally, the concavity follows by observing from (3.3) 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. □

Remark 3.5. *We emphasize here that the eigenvalue bounds depend on the choice of $d\in\mathbb{R}^n$. Though the d is irrelevant in Proposition 2.3, i.e., the function is equivalent on the feasible set of partition matrices $\mathcal{M}_m$, the values are no longer equal on the relaxed set $\mathcal{D}_O$. Of course the values are negative and not useful as a bound. We can fix $d=Ae\in\mathbb{R}^n$ and consider the bounds*

$$\operatorname{cut}(m)\ge 0>p^*_{eig}(A-\gamma\operatorname{Diag}(d))=\frac{1}{2}\left\langle\lambda(A-\gamma\operatorname{Diag}(d)),\begin{pmatrix}\lambda(\tilde{B})\\0\end{pmatrix}\right\rangle_-,\quad\gamma\ge 0.$$

From our empirical tests on random problems, we observed that the maximum occurs for γ closer to 0 than 1, thus illustrating why the bound using $G=A$ is better than the one using $G=-L$. This motivates our use of $G=A$ in the simulations below for the improved bounds.

3.2 Projected Eigenvalue Lower Bounds

Projected eigenvalue bounds for the QAP, and for GP are presented and studied in [10,12,22]. They have proven to be surprisingly stronger than the basic eigenvalue bounds. (Seen to be <0 above.) These are based on a special parametrization of the affine span of the linear equality constraints, $\mathcal{E}$. Rather than solving for the basic eigenvalue bound using the program in (3.1), we include the linear equality constraints $\mathcal{E}$, i.e., we consider the problem

$$\begin{array}{ll}\min & \frac{1}{2}\operatorname{trace}GXBX^T\\ \text{s.t.} & X\in\mathcal{D}_O\cap\mathcal{E},\end{array} \tag{3.6}$$

where $G=A-\operatorname{Diag}(d)$, $d\in\mathbb{R}^n$.

We define the $n\times n$ and $k\times k$ orthogonal matrices P,Q with

$$P=\begin{bmatrix}\frac{1}{\sqrt{n}}e & V\end{bmatrix}\in\mathcal{O}_n,\quad Q=\begin{bmatrix}\frac{1}{\sqrt{n}}\tilde{m} & W\end{bmatrix}\in\mathcal{O}_k. \tag{3.7}$$

Lemma 3.6. *[22, Lemma 3.1] Let P, Q, V, W be defined in* (3.7). *Suppose that $X \in \mathbb{R}^{n\times k}$ and $Z \in \mathbb{R}^{(n-1)\times(k-1)}$ are related by*

$$X = P \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} Q^T \tilde{M}. \tag{3.8}$$

Then the following holds:

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

Conversely, if $X \in \mathcal{E}$, then there exists Z such that the representation (3.8) *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 3.6 states that $\mathcal{Q}$ is an invertible transformation between $\mathbb{R}^{(n-1)\times(k-1)}$ and $\mathcal{E} - \widehat{X}$. Moreover, from (3.8), we see that $X \in \mathcal{E}$ if, and only if,

$$\begin{array}{rcl} X & = & P \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} Q^T \tilde{M} \\ & = & \begin{bmatrix} \frac{e}{\sqrt{n}} & V \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{n}}\tilde{m}^T \\ W^T \end{bmatrix} \tilde{M} \\ & = & \frac{1}{n}em^T + VZW^T\tilde{M} \\ & = & \widehat{X} + VZW^T\tilde{M}, \end{array} \tag{3.9}$$

for some Z. Thus, the set $\mathcal{E}$ can be parametrized using $\widehat{X} + VZW^T\tilde{M}$.

We are now ready to describe our two projected eigenvalue bounds. We remark that (3.11) and the first inequality in (3.14) were already discussed in Proposition 3, Theorem 1 and Theorem 3 in [20]. We include them for completeness. We note that the notation in Lemma 3.6, equation (3.9) and the next theorem will also be used frequently in Section 4 when we discuss the QP lower bound.

Theorem 3.7. *Let $d \in \mathbb{R}^n$, $G = A - \operatorname{Diag}(d)$. Let V, W be defined in* (3.7) *and $\widehat{X} = \frac{1}{n}em^T \in \mathbb{R}^{n\times k}$. Then:*

1. *For any $X \in \mathcal{E}$ and $Z \in \mathbb{R}^{(n-1)\times(k-1)}$ related by* (3.9), *we have*

$$\begin{array}{rcl} \operatorname{trace} GXBX^T & = & \alpha + \operatorname{trace} \widehat{G}Z\widehat{B}Z^T + \operatorname{trace} CZ^T \\ & = & -\alpha + \operatorname{trace} \widehat{G}Z\widehat{B}Z^T + 2\operatorname{trace} G\widehat{X}BX^T, \end{array} \tag{3.10}$$

and

$$\operatorname{trace}(-L)XBX^T \quad = \quad \operatorname{trace} \widehat{L}Z\widehat{B}Z^T, \tag{3.11}$$

where

$$\widehat{G} = V^TGV, \widehat{L} = V^T(-L)V, \ \ \widehat{B} = W^T\tilde{M}B\tilde{M}W, \ \ \alpha = \frac{1}{n^2}(e^TGe)(m^TBm), \ \ C = 2V^TG\widehat{X}B\tilde{M}W. \tag{3.12}$$

2. *We have the following two lower bounds:*

(a)

$$
\begin{aligned}
\operatorname{cut}(m) &\geq p^*_{projeig}(G) := \frac{1}{2}\left\{-\alpha + \left\langle \lambda(\widehat{G}), \begin{pmatrix}\lambda(\widehat{B})\\ 0\end{pmatrix}\right\rangle_- + 2\min_{X\in\mathcal{D}} \operatorname{trace} G\widehat{X}BX^T\right\}\\
&= \frac{1}{2}\left\{\alpha + \left\langle \lambda(\widehat{G}), \begin{pmatrix}\lambda(\widehat{B})\\ 0\end{pmatrix}\right\rangle_- + \min_{0\leq \widehat{X}+VZW^T\tilde{M}} \operatorname{trace} CZ^T\right\}\\
&= \frac{1}{2}\left\{-\alpha + \sum_{i=1}^{k-2}\lambda_{k-i}(\widehat{B})\lambda_i(\widehat{G}) + \lambda_1(\widehat{B})\lambda_{n-1}(\widehat{G}) + 2\min_{X\in\mathcal{D}} \operatorname{trace} G\widehat{X}BX^T\right\}.
\end{aligned}
\tag{3.13}
$$

(b)

$$
\operatorname{cut}(m) \geq p^*_{projeig}(-L) := \frac{1}{2}\left\langle \lambda(\widehat{L}), \begin{pmatrix}\lambda(\widehat{B})\\ 0\end{pmatrix}\right\rangle_- \geq p^*_{eig}(-L). \tag{3.14}
$$

3. The eigenspaces of V^TLV correspond to the eigenspaces of L that are orthogonal to e.

Proof. After substituting the parametrization (3.9) into the function $\operatorname{trace} GXBX^T$, we obtain a constant, quadratic, and linear term:

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

and

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

These together with (3.12) yield the two equations in (3.10). Since $Le = 0$ and hence $L\widehat{X} = 0$, we obtain (3.11) on replacing G with $-L$ in the above relations. This proves Item 1.

We now prove (3.13), i.e., Item 2a. To this end, recall from (2.5) and (2.1) that

$$
\operatorname{cut}(m) = \min\left\{\frac{1}{2}\operatorname{trace} GXBX^T : \ X \in \mathcal{D}\cap\mathcal{D}_O\right\}.
$$

Combining this with (3.10), we see further that

$$
\begin{aligned}
\operatorname{cut}(m) &= \frac{1}{2}\left(-\alpha + \min_{X\in\mathcal{D}\cap\mathcal{D}_O}\left\{\operatorname{trace}\widehat{G}Z\widehat{B}Z^T + 2\operatorname{trace} G\widehat{X}BX^T\right\}\right)\\
&\geq \frac{1}{2}\left(-\alpha + \min_{X\in\mathcal{E}\cap\mathcal{D}_O}\operatorname{trace}\widehat{G}Z\widehat{B}Z^T + 2\min_{X\in\mathcal{D}}\operatorname{trace} G\widehat{X}BX^T\right)\\
&= \frac{1}{2}\left(-\alpha + \left\langle \lambda(\widehat{G}), \begin{pmatrix}\lambda(\widehat{B})\\ 0\end{pmatrix}\right\rangle_- + 2\min_{X\in\mathcal{D}}\operatorname{trace} G\widehat{X}BX^T\right) = p^*_{projeig}(G),
\end{aligned}
\tag{3.15}
$$

where Z and X are related via (3.9), and the last equality follows from Lemma 3.6 and Theorem 3.2. Furthermore, notice that

$$\begin{aligned} &-\alpha + 2\min_{X\in\mathcal{D}} \operatorname{trace} G\widehat{X}BX^T = \alpha + 2\min_{X\in\mathcal{D}} \operatorname{trace} G\widehat{X}B(X-\widehat{X})^T \\ &= \alpha + 2\min_{0\le \widehat{X}+VZW^T\tilde{M}} \operatorname{trace} G\widehat{X}B(VZW^T\tilde{M})^T = \alpha + \min_{0\le \widehat{X}+VZW^T\tilde{M}} \operatorname{trace} CZ^T, \end{aligned} \tag{3.16}$$

where the second equality follows from Lemma 3.6, and the last equality follows from the definition of C in (3.12). Combining this last relation with (3.15) proves the first two equalities in (3.13). The last equality in (3.13) follows from the fact that

$$\lambda_k(\tilde{B}) \le \lambda_{k-1}(\widehat{B}) \le \lambda_{k-1}(\tilde{B}) \le \cdots \le \lambda_2(\tilde{B}) = 0 \le \lambda_1(\widehat{B}) \le \lambda_1(\tilde{B}), \tag{3.17}$$

which is a consequence of the eigenvalue interlacing theorem [15, Corollary 4.3.16], the definition of $\widehat{B}$ and Lemma 3.3.

Next, we prove (3.14). Recall again from (2.5) and (2.1) that

$$\operatorname{cut}(m) = \min\left\{\frac{1}{2}\operatorname{trace}(-L)XBX^T : \ X\in\mathcal{D}\cap\mathcal{D}_O\right\}.$$

Using (3.11), we see further that

$$\begin{aligned} \operatorname{cut}(m) &\ge \frac{1}{2}\min\left\{\operatorname{trace}(-L)XBX^T : \ X\in\mathcal{E}\cap\mathcal{D}_O\right\} \\ &= \frac{1}{2}\min\left\{\operatorname{trace}\widehat{L}Z\widehat{B}Z^T : \ X\in\mathcal{E}\cap\mathcal{D}_O\right\} \\ &= \frac{1}{2}\left\langle \lambda(\widehat{L}), \begin{pmatrix}\lambda(\widehat{B})\\ 0\end{pmatrix}\right\rangle_- \ (= p^*_{projeig}(-L)) \\ &\ge \min\left\{\frac{1}{2}\operatorname{trace}(-L)XBX^T : \ X\in\mathcal{D}_O\right\}, \end{aligned}$$

where Z and X are related via (3.9). The last inequality follows since the constraint $X\in\mathcal{E}$ is dropped.

Since $Le = 0$ and the columns of V are orthogonal to e, the last conclusion of the theorem follows immediately. □

Remark 3.8. *Let $Q\in\mathbb{R}^{(k-1)\times(k-1)}$ be the orthogonal matrix with columns consisting of the eigenvectors of $\widehat{B}$, defined in* (3.12), *corresponding to eigenvalues of $\widehat{B}$ in nondecreasing order; let $P_G, P_L\in\mathbb{R}^{(n-1)\times(k-1)}$ be the matrices with orthonormal columns consisting of $k-1$ eigenvectors of $\widehat{G}, \widehat{L}$, respectively, corresponding to the largest $k-2$ in nonincreasing order followed by the smallest. From* (3.17) *and Theorem 3.2, the minimal scalar product terms in* (3.13) *and* (3.14), *respectively, are attained at*

$$Z_G = P_G Q^T, \quad Z_L = P_L Q^T, \tag{3.18}$$

respectively, and two corresponding points in $\mathcal{E}$ are given, according to (3.9), *respectively, by*

$$X_G = \widehat{X} + VZ_GW^T\tilde{M}, \quad X_L = \widehat{X} + VZ_LW^T\tilde{M}. \tag{3.19}$$

The linear programming problem, LP *, in* (3.13) *can be solved explicitly; see Lemma 3.10 below. Since the condition number for the symmetric eigenvalue problem is* 1*, e.g., [9], the above shows that we can find the projected eigenvalue bounds very accurately. In addition, we need only find* $k-1$ *eigenvalues of* $\widehat{G}$*,* $\widehat{B}$*. Hence, if the number of sets* k *is small relative to the number of nodes* n *and the adjacency matrix* A *is sparse, then we can find bounds for large problems both efficiently and accurately; see Section 7.2.*

Remark 3.9. *We emphasize again that although the objective function in* (2.5) *is equivalent for all* $d \in \mathbb{R}^n$ *on the set of partition matrices* $\mathcal{M}_m$*, this is not true once we relax this feasible set. Though there are advantages to using the Laplacian matrix as shown in [20] in terms of simplicity of the objective function, our numerics suggest that the bound* $p^*_{projeig}(A)$ *obtained from using the adjacency matrix* A *is stronger than* $p^*_{projeig}(-L)$*. Numerical tests confirming this are given in Section 7.*

3.2.1 Explicit Solution for Linear Term

The constant term α and eigenvalue minimal scalar product term of the bound $p^*_{projeig}(G)$ in (3.13) can be found efficiently using the two quadratic forms for $\widehat{G}$, $\widehat{B}$ and finding $k-1$ eigenvalues from them. We now show that the third term, i.e., the linear term, can also be found efficiently. Precisely, we give an explicit solution to the linear optimization problem in (3.13) in Lemma 3.10, below.

Notice that in (3.13), the minimization is taken over $X \in \mathcal{D}$, which is shown to be the convex hull of the set of partition matrices $\mathcal{M}_m$. As mentioned above, this essentially follows from the Birkhoff and Von Neumann theorems, see e.g., [23]. Thus, to solve the linear programming problem in (3.13), it suffices to consider minimizing the same objective over the nonconvex set $\mathcal{M}_m$ instead.

Lemma 3.10. *Let* $d \in \mathbb{R}^n$*,* $G = A - \mathrm{Diag}(d)$*,* $\widehat{X} = \frac{1}{n} e m^T \in \mathcal{M}_m$ *and*

$$v_0 = \begin{bmatrix} (n - m_k - m_1) e_{m_1} \\ (n - m_k - m_2) e_{m_2} \\ \vdots \\ (n - m_k - m_{k-1}) e_{m_{k-1}} \\ 0 e_{m_k} \end{bmatrix},$$

where $e_j \in \mathbb{R}^j$ *is the vector of ones of dimension* j*. Then*

$$\min_{X \in \mathcal{M}_m} \operatorname{trace} G \widehat{X} B X^T = \frac{1}{n} \langle Ge, v_0 \rangle_- .$$

Proof. Let X_0 denote the feasible partition matrix

$$X_0 = \begin{bmatrix} e_{m_1} & 0 & \cdots & 0 \\ 0 & e_{m_2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & e_{m_k} \end{bmatrix} \in \mathcal{M}_m. \tag{3.20}$$

Then it is clear that $X \in \mathcal{M}_m$ if, and only if, there exists a permutation matrix P on $\{1, \ldots, n\}$ so that $X = P X_0$. Using this observation and letting S_N denote the set of permutation matrices on

$\{1,\dots,n\}$, we have

$$\min_{X\in\mathcal{M}_m} \operatorname{trace} G\widehat{X}BX^T = \frac{1}{n}\min_{P\in S_N} \operatorname{trace} Gem^T BX_0^T P^T$$
$$= \frac{1}{n}\min_{P\in S_N} \operatorname{trace} Ge(X_0Bm)^T P^T$$
$$= \frac{1}{n}\min_{P\in S_N} \operatorname{trace} Gev_0^T P^T = \frac{1}{n}\langle Ge, v_0\rangle_-,$$

where the last equality follows from the definition of minimal scalar product. □

4 Quadratic Programming Lower Bound

A new successful and efficient bound used for the QAP is given in [1,5]. In this section, we adapt the idea described there to obtain a lower bound for $\operatorname{cut}(m)$. This bound uses a relaxation that is a *convex* QP, i.e., the minimization of a quadratic function that is convex on the feasible set defined by linear inequality constraints. Approaches based on nonconvex QPs are given in e.g., [13] and the references therein.

The main idea in [1,5] is to use the zero duality gap result for a homogeneous QAP [2, Theorem 3.2] on an objective obtained via a suitable reparametrization of the original problem. Following this idea, we consider the parametrization in (3.10) where our main objective in (2.5) is rewritten as:

$$\frac{1}{2}\operatorname{trace} GXBX^T = \frac{1}{2}\left(\alpha + \operatorname{trace}\widehat{G}Z\widehat{B}Z^T + \operatorname{trace} CZ^T\right) \tag{4.1}$$

with X and Z related according to (3.8), and $G = A - \operatorname{Diag}(d)$ for some $d \in \mathbb{R}^n$. We next look at the homogeneous part:

$$\begin{array}{rl} v_r^* := \min & \frac{1}{2}\operatorname{trace}\widehat{G}Z\widehat{B}Z^T \\ \text{s.t.} & Z^TZ = I. \end{array} \tag{4.2}$$

Notice that the constraint $ZZ^T \preceq I$ is redundant for the above problem. By adding this redundant constraint, the corresponding Lagrange dual problem is given by

$$\begin{array}{rl} v_{dsdp} := \max & \frac{1}{2}\operatorname{trace} S + \frac{1}{2}\operatorname{trace} T \\ \text{s.t.} & I_{k-1}\otimes S + T\otimes I_{n-1} \preceq \widehat{B}\otimes\widehat{G}, \\ & S \preceq 0, \\ & S\in\mathcal{S}^{n-1},\ T\in\mathcal{S}^{k-1}, \end{array} \tag{4.3}$$

where the variables S and T are the dual variables corresponding to the constraints $ZZ^T \preceq I$ and $Z^TZ = I$, respectively. It is known that $v_r^* = v_{dsdp}$; see [19, Theorem 2]. This latter problem (4.3) can be solved efficiently. For example, as in the proofs of [2, Theorem 3.2] and [19, Theorem 2], one can take advantage of the properties of the Kronecker product and orthogonal diagonalizations of $\widehat{B},\widehat{G}$, to reduce the problem to solving the following LP with $n+k-2$ variables,

$$\begin{array}{rl} \max & \frac{1}{2}e^Ts + \frac{1}{2}e^Tt \\ \text{s.t.} & t_i + s_j \le \lambda_i\sigma_j, \quad i=1,\dots,k-1, \quad j=1,\dots,n-1, \\ & s_j \le 0, \quad j=1,\dots,n-1, \end{array} \tag{4.4}$$

where

$$\widehat{B} = U_1 \operatorname{Diag}(\lambda) U_1^T \quad \text{and} \quad \widehat{G} = U_2 \operatorname{Diag}(\sigma) U_2^T \tag{4.5}$$

are eigenvalue orthogonal decompositions of $\widehat{B}$ and $\widehat{G}$, respectively. From an optimal solution (s^*, t^*) of (4.4), we can recover an optimal solution of (4.3) as

$$S^* = U_2 \operatorname{Diag}(s^*) U_2^T \qquad T^* = U_1 \operatorname{Diag}(t^*) U_1^T. \tag{4.6}$$

Next, suppose that the optimal value of the dual problem (4.3) is attained at (S^*, T^*). Let Z be such that the X defined according to (3.8) is a partition matrix. Then we have

$$\begin{aligned}
\frac{1}{2} \operatorname{trace}(\widehat{G} Z \widehat{B} Z^T) &= \frac{1}{2} \operatorname{vec}(Z)^T (\widehat{B} \otimes \widehat{G}) \operatorname{vec}(Z) \\
&= \frac{1}{2} \operatorname{vec}(Z)^T \underbrace{(\widehat{B} \otimes \widehat{G} - I \otimes S^* - T^* \otimes I)}_{\widehat{Q}} \operatorname{vec}(Z) + \frac{1}{2} \operatorname{trace}(Z Z^T S^*) + \frac{1}{2} \operatorname{trace}(T^*) \\
&= \frac{1}{2} \operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z) + \frac{1}{2} \operatorname{trace}([Z Z^T - I] S^*) + \frac{1}{2} \operatorname{trace}(S^*) + \frac{1}{2} \operatorname{trace}(T^*) \\
&\geq \frac{1}{2} \operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z) + \frac{1}{2} \operatorname{trace}(S^*) + \frac{1}{2} \operatorname{trace}(T^*),
\end{aligned}$$

where the last inequality uses $S^* \preceq 0$ and $Z Z^T \preceq I$.

Recall that the original nonconvex problem (2.5) is equivalent to minimizing the right hand side of (4.1) over the set of all Z so that the X defined in (3.8) corresponds to a partition matrix. From the above relations, the third equality in (2.1) and Lemma 3.6, we see that

$$\begin{array}{rll}
\operatorname{cut}(m) \geq \min & \frac{1}{2}(\alpha + \operatorname{trace} C Z^T + \operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z)) + \frac{1}{2} \operatorname{trace}(S^*) + \frac{1}{2} \operatorname{trace}(T^*) \\
\text{s.t.} & Z^T Z = I_{k-1}, \quad V Z W^T \tilde{M} \geq -\widehat{X}.
\end{array} \tag{4.7}$$

We also recall from (4.3) that $\frac{1}{2} \operatorname{trace}(S^*) + \frac{1}{2} \operatorname{trace}(T^*) = v_{dsdp} = v_r^*$, which further equals

$$\frac{1}{2} \left\langle \lambda(\widehat{G}), \begin{pmatrix} \lambda(\widehat{B}) \\ 0 \end{pmatrix} \right\rangle_-$$

according to (4.2) and Theorem 3.2.

A lower bound can now be obtained by relaxing the constraints in (4.7). For example, by dropping the orthogonality constraints, we obtain the following lower bound on $\operatorname{cut}(m)$:

$$\begin{array}{rl}
p_{QP}^*(G) := \min & q_1(Z) := \frac{1}{2} \left(\alpha + \operatorname{trace} C Z^T + \operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z) + \left\langle \lambda(\widehat{G}), \begin{pmatrix} \lambda(\widehat{B}) \\ 0 \end{pmatrix} \right\rangle_- \right) \\
\text{s.t.} & V Z W^T \tilde{M} \geq -\widehat{X},
\end{array} \tag{4.8}$$

Notice that this is a QP with $(n-1)(k-1)$ variables and nk constraints.

As in [1, Page 346], it is possible to reformulate (4.8) into a QP in variables $X \in \mathcal{D}$. Note that $\tilde{Q}$ defined in (4.10) is not positive semidefinite in general. Nevertheless, the QP is implicitly convex.

Theorem 4.1. *Let S^*, T^* be optimal solutions of* (4.3) *as defined in* (4.6). *A lower bound on* $\operatorname{cut}(m)$ *is obtained from the following QP:*

$$\operatorname{cut}(m) \geq p^*_{QP}(G) = \min_{X \in \mathcal{D}} \frac{1}{2} \operatorname{vec}(X)^T \tilde{Q} \operatorname{vec}(X) + \frac{1}{2} \left\langle \lambda(\widehat{G}), \begin{pmatrix} \lambda(\widehat{B}) \\ 0 \end{pmatrix} \right\rangle_- \tag{4.9}$$

where

$$\tilde{Q} := B \otimes G - M^{-1} \otimes VS^*V^T - \tilde{M}^{-1} W T^* W^T \tilde{M}^{-1} \otimes I_n. \tag{4.10}$$

The QP in (4.9) *is implicitly convex since $\tilde{Q}$ is positive semidefinite on the tangent space of $\mathcal{E}$.*

Proof. We start by rewriting the second-order term of q_1 in (4.8) using the relation (3.8). Since $V^T V = I_{n-1}$ and $W^T W = I_{k-1}$, we have from the definitions of $\widehat{B}$ and $\widehat{G}$ that

$$\begin{aligned} \widehat{Q} &= \widehat{B} \otimes \widehat{G} - I_{k-1} \otimes S^* - T^* \otimes I_{n-1} \\ &= W^T \tilde{M} B \tilde{M} W \otimes V^T G V - I_{k-1} \otimes S^* - T^* \otimes I_{n-1} \\ &= (\tilde{M} W \otimes V)^T [B \otimes G - M^{-1} \otimes V S^* V^T - \tilde{M}^{-1} W T^* W^T \tilde{M}^{-1} \otimes I_n] (\tilde{M} W \otimes V) \end{aligned} \tag{4.11}$$

On the other hand, from (3.9), we have

$$\operatorname{vec}(X - \widehat{X}) = \operatorname{vec}(V Z W^T \tilde{M}) = (\tilde{M} W \otimes V) \operatorname{vec}(Z).$$

Hence, the second-order term in q_1 can be rewritten as

$$\operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z) = \operatorname{vec}(X - \widehat{X})^T \tilde{Q} \operatorname{vec}(X - \widehat{X}), \tag{4.12}$$

where $\tilde{Q}$ is defined in (4.10). Next, we see from $V^T e = 0$ that

$$(M^{-1} \otimes V S^* V^T) \operatorname{vec}(\widehat{X}) = \frac{1}{n} (M^{-1} \otimes V S^* V^T)(m \otimes I_n) e = \frac{1}{n} (e \otimes V S^* V^T) e = 0.$$

Similarly, since $W^T \tilde{m} = 0$, we also have

$$\begin{aligned} (\tilde{M}^{-1} W T^* W^T \tilde{M}^{-1} \otimes I_n) \operatorname{vec}(\widehat{X}) &= \frac{1}{n} (\tilde{M}^{-1} W T^* W^T \tilde{M}^{-1} \otimes I_n)(m \otimes I_n) e \\ &= \frac{1}{n} (\tilde{M}^{-1} W T^* W^T \tilde{m} \otimes I_n) e = 0. \end{aligned}$$

Combining the above two relations with (4.12), we obtain further that

$$\begin{aligned} & \operatorname{vec}(Z)^T \widehat{Q} \operatorname{vec}(Z) \\ &= \operatorname{vec}(X)^T \tilde{Q} \operatorname{vec}(X) - 2 \operatorname{vec}(\widehat{X})^T [B \otimes G] \operatorname{vec}(X) + \operatorname{vec}(\widehat{X}) [B \otimes G] \operatorname{vec}(\widehat{X}) \\ &= \operatorname{vec}(X)^T \tilde{Q} \operatorname{vec}(X) - 2 \operatorname{trace} G \widehat{X} B X^T + \alpha. \end{aligned}$$

For the first two terms of q_1, proceeding as in (3.16), we have

$$\alpha + \operatorname{trace} C Z^T = -\alpha + 2 \operatorname{trace} G \widehat{X} B X^T.$$

Furthermore, recall from Lemma 3.6 that with X and Z related by (3.8), $X \in \mathcal{D}$ if, and only if, $V Z W^T \tilde{M} \geq -\widehat{X}$.

The conclusion in (4.9) now follows by substituting the above expressions into (4.8).

Finally, from (4.11) we see that $\tilde{Q}$ is positive semidefinite when restricted to the range of $\tilde{M} W \otimes V$. This is precisely the tangent space of $\mathcal{E}$. ∎

Although the dimension of the feasible set in (4.9) is slightly larger than the dimension of the feasible set in (4.8), the former feasible set is much simpler. Moreover, as mentioned above, even though $\tilde{Q}$ is not positive semidefinite in general, it is when restricted to the tangent space of $\mathcal{E}$. Thus, as in [5], one may apply the Frank-Wolfe algorithm on (4.9) to approximately compute the QP lower bound $p^*_{QP}(G)$ for problems with huge dimension.

Since $\widehat{Q} \succeq 0$, it is easy to see from (4.8) that $p^*_{QP}(G) \geq p^*_{projeig}(G)$. This inequality is not necessarily strict. Indeed, if $G = -L$, then $C = 0$ and $\alpha = 0$ in (4.8). Since the feasible set of (4.8) contains the origin, it follows from this and the definition of $p^*_{projeig}(-L)$ that $p^*_{QP}(-L) = p^*_{projeig}(-L)$. Despite this, as we see in the numerics Section 7, we have $p^*_{QP}(A) > p^*_{projeig}(A)$ for most of our numerical experiments. In general, we still do not know what conditions will guarantee $p^*_{QP}(G) > p^*_{projeig}(G)$.

5 Semidefinite Programming Lower Bounds

In this section, we study the SDP relaxation constructed from the various equality constraints in the representation in (2.1) and the objective function in (2.4).

One way to derive an SDP relaxation for (2.5) is to start by considering a suitable Lagrangian relaxation, which is itself an SDP. Taking the dual of this Lagrangian relaxation then gives an SDP relaxation for (2.5); see [29] and [28] for the development for the QAP and GP cases, respectively. Alternatively, we can also obtain the *same* SDP relaxation directly using the well-known *lifting process*, e.g., [3,17,24,28,29]. In this approach, we start with the following equivalent quadratically constrained quadratic problems to (2.5):

$$\begin{array}{rlll} \mathrm{cut}(m) = & \min & \frac{1}{2}\operatorname{trace} GXBX^T = & \min & \frac{1}{2}\operatorname{trace} GXBX^T \\ & \text{s.t.} & X \circ X = X, & \text{s.t.} & X \circ X = x_0 X, \\ & & \|Xe - e\|^2 = 0, & & \|Xe - x_0 e\|^2 = 0, \\ & & \|X^T e - m\|^2 = 0, & & \|X^T e - x_0 m\|^2 = 0, \\ & & X_{:i} \circ X_{:j} = 0,\ \forall i \neq j, & & X_{:i} \circ X_{:j} = 0,\ \forall i \neq j, \\ & & X^T X - M = 0, & & X^T X - M = 0, \\ & & \operatorname{diag}(XX^T) - e = 0. & & \operatorname{diag}(XX^T) - e = 0, \\ & & & & x_0^2 = 1. \end{array} \tag{5.1}$$

Here: $G = A - \operatorname{Diag}(d)$, $d \in \mathbb{R}^n$; the first equality follows from the fifth equality in (2.1), and we add x_0 and the constraint $x_0^2 = 1$ to *homogenize* the problem. Note that if $x_0 = -1$ at the optimum, then we can replace it with $x_0 = 1$ by changing the sign $X \leftarrow -X$ while leaving the objective value unchanged. We next linearize the quadratic terms in (5.1) using the matrix

$$Y_X := \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} (1 \quad \operatorname{vec}(X)^T).$$

Then $Y_X \succeq 0$ and is rank one. The objective function becomes

$$\frac{1}{2}\operatorname{trace} GXBX^T = \frac{1}{2}\operatorname{trace} L_G Y_X,$$

where

$$L_G := \begin{bmatrix} 0 & 0 \\ 0 & B \otimes G \end{bmatrix}. \tag{5.2}$$

By removing the rank one restriction on Y_X and using a general symmetric matrix variable Y rather than Y_X, we obtain the following SDP relaxation and its properties:

$$
\begin{array}{rll}
\operatorname{cut}(m) \geq p^*_{SDP}(G) := & \min & \frac{1}{2}\operatorname{trace} L_G Y \\
& \text{s.t.} & \operatorname{arrow}(Y) = e_0, \\
& & \operatorname{trace} D_1 Y = 0, \\
& & \operatorname{trace} D_2 Y = 0, \\
& & \mathcal{G}_J(Y) = 0, \\
& & \mathcal{D}_O(Y) = M, \\
& & \mathcal{D}_e(Y) = e, \\
& & Y_{00} = 1, \\
& & Y \succeq 0,
\end{array}
\tag{5.3}
$$

where the rows and columns of $Y \in \mathcal{S}^{kn+1}$ are indexed from 0 to kn. We now describe the constraints in detail.

1. The *arrow linear transformation* acts on $\mathcal{S}^{kn+1}$,

$$
\operatorname{arrow}(Y) := \operatorname{diag}(Y) - (0, Y_{0,1:kn})^T, \tag{5.4}
$$

   $Y_{0,1:kn}$ is the vector formed from the last kn components of the first row (indexed by 0) of Y. The arrow constraint represents $X \in \mathcal{Z}$, and e_0 is the first (0th) unit vector.

2. The norm constraints for $X \in \mathcal{E}$ are represented by the constraints with the two $(kn+1) \times (kn+1)$ matrices

$$
D_1 := \begin{bmatrix} n & -e_k^T \otimes e_n^T \\ -e_k \otimes e_n & (e_k e_k^T) \otimes I_n \end{bmatrix},
$$

$$
D_2 := \begin{bmatrix} m^T m & -m^T \otimes e_n^T \\ -m \otimes e_n & I_k \otimes (e_n e_n^T) \end{bmatrix}.
$$

3. We let $\mathcal{G}_J$ represent the gangster operator on $\mathcal{S}^{kn+1}$, i.e., it shoots *holes* in a matrix,

$$
(\mathcal{G}_J(Y))_{ij} := \begin{cases} Y_{ij} & \text{if } (i,j) \text{ or } (j,i) \ \in J \\ 0 & \text{otherwise,} \end{cases} \tag{5.5}
$$

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

   The gangster constraint represents the (Hadamard) orthogonality of the columns. The zeros are the diagonal elements of the off-diagonal blocks $\bar{Y}_{(ij)}, 1 < i < j$, of Y; see the block structure in (5.6) below.

4. Again, by abuse of notation, we use the symbols for the sets of constraints $\mathcal{D}_O, \mathcal{D}_e$ to represent the linear transformations in the SDP relaxation (5.3). Note that

$$
\langle \Psi, X^T X \rangle = \operatorname{trace} I X \Psi X^T = \operatorname{vec}(X)^T (\Psi \otimes I) \operatorname{vec}(X).
$$

   Therefore, the adjoint of $\mathcal{D}_O$ is made up of a zero row/column and k^2 blocks that are multiples of the identity:

$$
\mathcal{D}_O^*(\Psi) = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \Psi \otimes I_n \end{array}\right].
$$

If Y is blocked appropriately as

$$Y = \left[\begin{array}{c|c} Y_{00} & Y_{0,:} \\ \hline Y_{:,0} & \bar{Y} \end{array}\right], \quad \bar{Y} = \begin{bmatrix} \bar{Y}_{(11)} & \bar{Y}_{(12)} & \cdots & \bar{Y}_{(1k)} \\ \bar{Y}_{(21)} & \bar{Y}_{(22)} & \cdots & \bar{Y}_{(2k)} \\ \vdots & \ddots & \ddots & \vdots \\ \bar{Y}_{(k1)} & \ddots & \ddots & \bar{Y}_{(kk)} \end{bmatrix}, \tag{5.6}$$

with each $\bar{Y}_{(ij)}$ being a $n \times n$ matrix, then

$$\mathcal{D}_O(Y) = \left(\operatorname{trace} \bar{Y}_{(ij)}\right) \in \mathcal{S}^k. \tag{5.7}$$

Similarly,

$$\langle \phi, \operatorname{diag}(XX^T) \rangle = \langle \operatorname{Diag}(\phi), XX^T \rangle = \operatorname{vec}(X)^T \left(I_k \otimes \operatorname{Diag}(\phi) \right) \operatorname{vec}(X).$$

Therefore we get the sum of the diagonal parts

$$\mathcal{D}_e(Y) = \sum_{i=1}^{k} \operatorname{diag} \bar{Y}_{(ii)} \in \mathbb{R}^n. \tag{5.8}$$

5.1 Final SDP Relaxation

We present our final SDP relaxation (SDP$_{final}$) in Theorem 5.1 below and discuss some of its properties. This relaxation is surprisingly simple/strong with many of the constraints in (5.3) redundant. In particular, we show that the problem is independent of the choice of $d \in \mathbb{R}^n$ in constructing G. We also show that the two constraints using $\mathcal{D}_O, \mathcal{D}_e$ are redundant in the SDP relaxation (SDP$_{final}$). This answers affirmatively the question posed in [28] on whether these constraints were redundant in the SDP relaxation for the GP.

Since both D_1 and D_2 are positive semidefinite and trace $D_i Y = 0, i = 1, 2$, we conclude that the feasible set of (5.3) has no strictly feasible (positive definite) points, $Y \succ 0$. Numerical difficulties can arise when an interior-point method is directly applied to a problem where strict feasibility, Slater's condition, fails. Nonetheless, we can find a very simple structured matrix in the relative interior of the feasible set to project (and regularize) the problem into a smaller dimension. As in [28], we achieve this by finding a matrix V with range equal to the intersection of the nullspaces of D_1 and D_2. This is called *facial reduction*, [4, 7]. Let $V_j \in \mathbb{R}^{j \times (j-1)}$, $V_j^T e = 0$, e.g.,

$$V_j := \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ -1 & \dots & \dots & -1 & -1 \end{bmatrix}_{j \times (j-1)} .$$

and let

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

Then the range of $\widehat{V}$ is equal to the range of (any) $\widehat{Y} \in \operatorname{relint} F$, the relative interior of the minimal face. And, we can facially reduce (5.3) using the substitution

$$Y = \widehat{V} Z \widehat{V}^T \in \mathcal{S}^{kn+1}, \quad Z \in \mathcal{S}^{(k-1)(n-1)+1}.$$

The facially reduced SDP is then

$$\begin{array}{rcll} \operatorname{cut}(m) \geq p^*_{SDP}(G) & = & \min & \frac{1}{2} \operatorname{trace} \widehat{V}^T L_G \widehat{V} Z \\ & & \text{s.t.} & \operatorname{arrow}(\widehat{V} Z \widehat{V}^T) = e_0 \\ & & & \mathcal{G}_J(\widehat{V} Z \widehat{V}^T) = 0 \\ & & & (\widehat{V} Z \widehat{V}^T)_{00} = 1 \\ & & & \mathcal{D}_O(\widehat{V} Z \widehat{V}^T) = M \\ & & & \mathcal{D}_e(\widehat{V} Z \widehat{V}^T) = e \\ & & & Z \succeq 0, \quad Z \in \mathcal{S}^{(k-1)(n-1)+1}. \end{array} \tag{5.9}$$

We let $\bar{J} := J \cup (0,0)$. Our main, simplified, SDP relaxation is as follows.

Theorem 5.1. *The facially reduced SDP* (5.9) *is equivalent to the single equality constrained problem*

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

The dual program is

$$\begin{array}{ll} \max & \frac{1}{2} W_{00} \\ \text{s.t.} & \widehat{V}^T \mathcal{G}_{\bar{J}}(W) \widehat{V} \preceq \widehat{V}^T L_G \widehat{V} \end{array} \tag{5.10}$$

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. It is shown in [28] that the second and third constraint in (5.9) along with $Z \succeq 0$ implies that the arrow constraint holds, i.e. the arrow constraint is redundant. It only remains to show that the last two equality constraints in (5.9) are redundant. First, the gangster constraint implies that the blocks in $Y = \widehat{V} Z \widehat{V}^T$ satisfy $\operatorname{diag} \bar{Y}_{(ij)} = 0$ for all $i \neq j$. Next, notice that $D_i \succeq 0$, $i = 1, 2$. Moreover, using $Y \succeq 0$ and considering the Schur complement of Y_{00}, we have

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

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

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

This together with the arrow constraints show that $\operatorname{trace} \bar{Y}_{(ii)} = \sum_{j=(i-1)n+1}^{ni} Y_{j0} = m_i$. 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 (SDP$_{final}$) follows from [28, Theorems 4.1], which discussed the primal SDP relaxations of the GP. That relaxation has the same feasible set as (SDP$_{final}$). In fact, it is shown in [28] 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 (SDP$_{final}$). The right-hand side of the dual (5.10) differs from the dual of the SDP relaxation of the GP. However, let

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

From the proof of [28, Theorems 4.2] we see that $\mathcal{G}_{\bar{J}}(\hat{W}) = \hat{W}$ and

$$\begin{aligned} -\widehat{V}^T \mathcal{G}_{\bar{J}}(\hat{W})\widehat{V} &= \widehat{V}^T(-\hat{W})\widehat{V} \\ &= \begin{bmatrix} 1 & m^T \otimes e^T/n \\ 0 & V_k^T \otimes V_n^T \end{bmatrix} \begin{bmatrix} -\alpha & 0 \\ 0 & ((I_k - E_k) \otimes I_n \end{bmatrix} \begin{bmatrix} 1 & 0 \\ m \otimes e/n & V_k \otimes V_n \end{bmatrix} \\ &= \begin{bmatrix} -\alpha + m^T(I_k - E_k)m/n & (m^T(I_k - E_k)V_k) \otimes (e^T V_n)/n \\ (V_k^T(I_k - E_k)m) \otimes (V_n^T e)/n & (V_k^T(I_k - E_k)V_k) \otimes (V_n^T V_n) \end{bmatrix} \\ &= \begin{bmatrix} -\alpha + m^T(I_k - E_k)m/n & 0 \\ 0 & (I_{k-1} + E_{k-1}) \otimes (I_{n-1} + E_{n-1}) \end{bmatrix} \\ &\succ 0, \qquad \text{for sufficiently large } -\alpha. \end{aligned}$$

Therefore $\widehat{V}^T \mathcal{G}_{\bar{J}}(\beta\hat{W})\widehat{V} \prec \widehat{V}^T L_G \widehat{V}$ for sufficiently large $-\alpha, \beta$, i.e., Slater's constraint qualification holds for the dual (5.10).

Finally, we let $Y = \widehat{V} Z \widehat{V}^T$ with Z feasible for (SDP$_{final}$). Then Y satisfies the gangster constraints, i.e., $\mathrm{diag}\,\bar{Y}_{(ij)} = 0$ for all $i \neq j$. On the other hand, if we restrict $D = \mathrm{Diag}(d)$, then the objective matrix L_D has nonzero elements only in the same diagonal positions of the off-diagonal blocks from the application of the Kronecker product $B \otimes \mathrm{Diag}(d)$. Thus, we must have $\mathrm{trace}\, L_D Y = 0$. Consequently, for all $d \in \mathbb{R}^n$,

$$\mathrm{trace}\left(\widehat{V}^T L_G \widehat{V}\right) Z = \mathrm{trace}\, L_G \widehat{V} Z \widehat{V}^T = \mathrm{trace}\, L_G Y = \mathrm{trace}\, L_A Y = \mathrm{trace}\, \widehat{V} L_A \widehat{V}^T Z.$$

□

The above Theorem 5.1 also answers a question posed in [28], i.e., whether the two constraints $\mathcal{D}_O, \mathcal{D}_e$ are redundant in the corresponding SDP relaxation of GP. Surprisingly, the answer is yes, they are both redundant.

We next present two useful properties for finding/recovering approximate solutions X from a solution Y of (SDP$_{final}$).

Proposition 5.2. *Suppose that Y is feasible for* (SDP$_{final}$). *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 $X_1 := \mathrm{Mat}(v_1) \in \mathcal{E} \cap \mathcal{N}$. Moreover, if $v_0 \neq 0$, then $X_2 := \mathrm{Mat}(\frac{1}{v_0} v_2) \in \mathcal{E}$. Furthermore, if, $Y \geq 0$, then $v_0 \neq 0$ and $X_2 \in \mathcal{N}$.*

Proof. The fact that $X_1 \in \mathcal{E}$ was shown in the proof of Theorem 5.1. That $X_1 \in \mathcal{N}$ follows from the arrow constraint. We now prove the results for X_2. Suppose first that $v_0 \neq 0$. Then

$$Y \succeq \lambda_1(Y) \begin{pmatrix} v_0 \\ v_2 \end{pmatrix} \begin{pmatrix} v_0 \\ v_2 \end{pmatrix}^T.$$

Using this and the definitions of D_i and X_2, we see further that

$$0 = \operatorname{trace}(D_i Y) \geq \begin{cases} \lambda_1(Y) v_0^2 \|X_2 e - e\|^2, & \text{if } i = 1, \\ \lambda_1(Y) v_0^2 \|X_2^T e - m\|^2, & \text{if } i = 2. \end{cases} \tag{5.11}$$

Since $\lambda_1(Y) \neq 0$ and $v_0 \neq 0$, it 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 $X_2 = \operatorname{Mat}(v_2/v_0) \in \mathcal{N}$.

To prove these claims, we note first from the classical Perron-Fröbenius theory, e.g., [6], that the vector $\begin{pmatrix} |v_0| & |v_2|^T \end{pmatrix}^T$ is also an eigenvector corresponding to the largest eigenvalue.[2] Letting $\chi := \operatorname{Mat}(v_2)$ and proceeding as in (5.11), we conclude that

$$\|\chi e - v_0 e\|^2 = 0 \quad \text{and} \quad \||\chi| e - |v_0| e\|^2 = 0.$$

The second equality implies that $v_0 \neq 0$. If $v_0 > 0$, then for all $i = 1, \cdots, n$, we have

$$\sum_{j=1}^{k} \chi_{ij} = v_0 = \sum_{j=1}^{k} |\chi_{ij}|,$$

showing that $\chi_{ij} \geq 0$ for all i, j, i.e., $v_2 \geq 0$. If $v_0 < 0$, one can show similarly that $v_2 \leq 0$. Hence, we have also shown $v_0 v_2 \geq 0$. This completes the proof. □

6 Feasible Solutions and Upper Bounds

In the above we have presented several approaches for finding lower bounds for $\operatorname{cut}(m)$. In addition, we have found matrices X that approximate the bound and satisfy some of the graph partitioning constraints. Specifically, we obtain two approximate solutions $X_A, X_L \in \mathcal{E}$ in (3.19), an approximate solution to (4.8) which can be transformed into an $n \times k$ matrix via (3.9), and the X_1, X_2 described in Proposition 5.2. We now use these to obtain feasible solutions (partition matrices) and thus obtain upper bounds.

We show below that we can find the closest feasible partition matrix X to a given approximate matrix $\bar{X}$ using linear programming, where $\bar{X}$ is found, for example, using the projected eigenvalue, QP or SDP lower bounds. Note that (6.1) is a *transportation problem* and therefore the optimal X in (6.1) can be found in strongly polynomial time ($O(n^2)$), see e.g., [25, 26].

[2] Indeed, if Y is irreducible, the top eigenspace must be the span of a positive vector. Hence the conclusion follows. For a reducible Y, the top eigenspace must then be a direct product of the top eigenspaces of each irreducible block. The conclusion follows similarly.

Theorem 6.1. *Let $\bar{X} \in \mathcal{E}$ be given. Then the closest partition matrix X to $\bar{X}$ in Fröbenius norm can be found by using the simplex method to solve the linear program*

$$\begin{array}{rc} \min & -\operatorname{trace} \bar{X}^T X \\ \text{s.t.} & Xe = e, \\ & X^T e = m, \\ & X \geq 0. \end{array} \tag{6.1}$$

Proof. Observe that for any partition matrix X, $\operatorname{trace} X^T X = n$. Hence, we have

$$\min_{X \in \mathcal{M}_m} \|\bar{X} - X\|_F^2 = \operatorname{trace}(\bar{X}^T \bar{X}) + n + 2 \min_{X \in \mathcal{M}_m} \operatorname{trace}\left(-\bar{X}^T X\right).$$

The result now follows from this and the fact that $\mathcal{M}_m = \operatorname{ext}(\mathcal{D})$, as stated in (2.1). (This is similar to what is done in [29].) □

7 Numerical Tests

In this section, we provide empirical comparisons for the lower and upper bounds presented above. All the numerical tests are performed in MATLAB version R2012a on a *single* node of the *COPS* cluster at University of Waterloo. It is an SGI XE340 system, with two 2.4 GHz quad-core Intel E5620 Xeon 64-bit CPUs and 48 GB RAM, equipped with SUSE Linux Enterprise server 11 SP1.

7.1 Random Tests with Various Sizes

In this subsection, we compare the bounds on two kinds of randomly generated graphs of various sizes:

1. Structured graphs: These are formed by first generating k disjoint cliques (of sizes $m_1, \ldots, m_k$, randomly chosen from $\{2, ..., \mathsf{imax}+1\}$). We join the first $k-1$ cliques to every node of the kth clique. We then add u_0 edges between the first $k-1$ cliques, chosen uniformly at random from the complement graph. In our tests, we set $u_0 = \lfloor e_c p \rfloor$, where e_c is the number of edges in the complement graph and $0 \leq p < 1$. By construction, $u_0 \geq \operatorname{cut}(m)$.

2. Random graphs: We start by fixing positive integers k, imax and generating integers $m_1, \ldots, m_k$, each chosen randomly from $\{2, ..., \mathsf{imax}+1\}$. We generate a graph with $n = e^T m$ nodes. The incidence matrix is generated with the MATLAB command:

```
A = round(rand(n)); A = round((A + A')/2); A = A - diag(diag(A));
```

   Consequently, an edge is chosen with probability 0.75.

First, we note the following about the eigenvalue bounds. Figures 1 and 2 show the difference in the projected eigenvalue bounds from using $A - \gamma \operatorname{Diag}(d)$ for a random $d \in \mathbb{R}^n$ on two structured graphs. This is typical of what we saw in our tests, i.e. that the maximum bound is near $\gamma = 0$. We had similar results for the specific choice $d = Ae$. This empirically suggests that using A would yield a better projected eigenvalue lower bound. This phenomenon will also be observed in subsequent tests.

Figure 1: Negative value for optimal γ

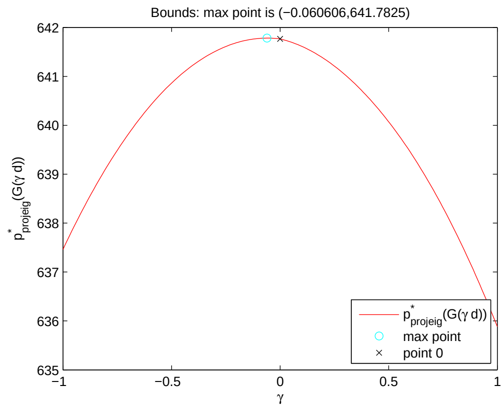
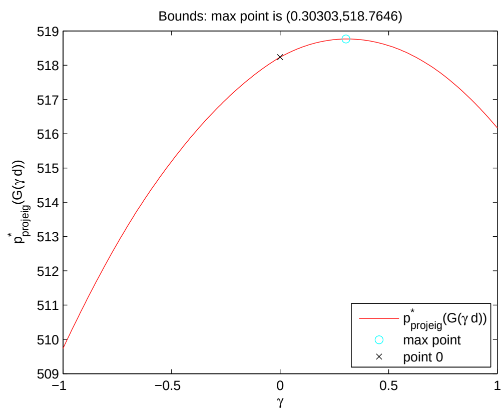


Figure 2: Positive value for optimal γ



In Tables 1 and 2, we consider small instances where $k = 4, 5$, $p = 20\%$ and $\mathsf{imax} = 10$. We consider the projected eigenvalue bounds with $G = -L$ (eig_{-L}) and $G = A$ (eig_A), the QP bound with $G = A$, the SDP bound and the doubly nonnegative programming (DNN) bound.[3] For each approach, we present the lower bounds (rounded up to the nearest integer) and the corresponding upper bounds (rounded down to the nearest integer) obtained via the technique described in Section 6.[4] We also present the relative gap (Rel. gap), defined as

$$\text{Rel. gap} = \frac{\text{best upper bound} - \text{best lower bound}}{\text{best upper bound} + \text{best lower bound}}. \tag{7.1}$$

In terms of lower bounds, the DNN approach usually gives the best lower bounds. While the SDP approach gives better lower bounds than the QP approach for random graphs, they are comparable

[3]The doubly nonnegative programming relaxation is obtained by imposing the constraint $\widehat{V}Z\widehat{V}^T \geq 0$ onto (SDP_{final}). Like the SDP relaxation, the bound obtained from this approach is independent of d. In our implementation, we picked $G = A$ for both the SDP and the DNN bounds.

[4]The SDP and DNN problems are solved via SDPT3 (version 4.0), [27], with tolerance gaptol set to be $1e{-6}$ and $1e{-3}$ respectively. The problems (4.4) and (4.8) are solved via SDPT3 (version 4.0) called by CVX (version 1.22), [11], using the default settings. The problem (6.1) is solved using simplex method in MATLAB, again using the default settings.

for structured graphs. Moreover, the projected eigenvalue lower bounds with A always outperforms the ones with $-L$. On the other hand, the DNN approach usually gives the best upper bounds.

Data				Lower bounds					Upper bounds					Rel. gap
n	k	$\|E\|$	u_0	eig_{-L}	eig_A	QP	SDP	DNN	eig_{-L}	eig_A	QP	SDP	DNN	
31	4	362	25	21	22	24	23	25	68	102	25	36	25	0.0000
18	4	86	16	13	14	15	16	16	22	35	16	19	16	0.0000
29	5	229	44	32	37	40	39	44	76	74	44	53	44	0.0000
41	5	453	91	76	84	86	86	91	159	162	101	125	102	0.0521

Table 1: Results for small structured graphs

Data			Lower bounds					Upper bounds					Rel. gap
n	k	$\|E\|$	eig_{-L}	eig_A	QP	SDP	DNN	eig_{-L}	eig_A	QP	SDP	DNN	
25	4	231	53	59	64	67	71	80	79	74	75	72	0.0070
23	4	189	7	9	12	14	18	25	24	22	22	20	0.0526
32	5	379	101	112	119	123	134	152	151	141	141	137	0.0111
28	5	266	77	89	95	100	106	124	132	111	115	112	0.0230

Table 2: Results for small random graphs

We consider medium-sized instances in Tables 3 and 4, where $k = 8,\ 10,\ 12$, $p = 20\%$ and $\mathsf{imax} = 20$. We do not consider DNN bounds due to computational complexity. We see that the lower bounds always satisfy $\mathrm{eig}_{-L} \leq \mathrm{eig}_A \leq \mathrm{QP}$. In particular, we note that the (lower) projected eigenvalue bounds with A always outperform the ones with $-L$. However, what is surprising is that the lower projected eigenvalue bound with A (for structured graphs) sometimes outperforms the SDP lower bound. This illustrates the strength of the heuristic that replaces the quadratic objective function with the sum of a quadratic and linear term and then solves the linear part exactly over the partition matrices.

Data				Lower bounds				Upper bounds				Rel. gap
n	k	$\|E\|$	u_0	eig_{-L}	eig_A	QP	SDP	eig_{-L}	eig_A	QP	SDP	
69	8	1077	317	249	283	290	281	516	635	328	438	0.0615
114	8	3104	834	723	785	794	758	1475	1813	834	1099	0.0246
85	8	2164	351	262	319	327	320	809	384	367	446	0.0576
116	10	3511	789	659	725	737	690	1269	2035	796	1135	0.0385
104	10	2934	605	500	546	554	529	1028	646	631	836	0.0650
78	10	1179	455	358	402	413	389	708	625	494	634	0.0893
129	12	3928	1082	879	988	1001	965	1994	1229	1233	1440	0.1022
120	12	3102	1009	833	913	926	893	1627	1278	1084	1379	0.0786
126	12	2654	1305	1049	1195	1218	1186	1767	1617	1361	1736	0.0554

Table 3: Results for medium-sized structured graphs

In Tables 5 and 6, we consider larger instances with $k = 35,\ 45,\ 55$, $p = 20\%$ and $\mathsf{imax} = 100$. We do not consider SDP and DNN bounds due to computational complexity. We see again that the projected eigenvalue lower bounds with A always outperforms the ones with $-L$.

Data			Lower bounds				Upper bounds				Rel. gap
n	k	$\|E\|$	eig_{-L}	eig_A	QP	SDP	eig_{-L}	eig_A	QP	SDP	
96	8	3405	1982	2103	2126	2146	2357	2353	2354	2368	0.0460
96	8	3403	2264	2420	2439	2451	2668	2652	2658	2696	0.0394
94	8	3292	1795	1885	1910	1930	2128	2141	2092	2130	0.0403
90	10	3009	1533	1622	1649	1659	1867	1886	1850	1873	0.0544
114	10	4823	2218	2394	2443	2459	2759	2780	2725	2777	0.0513
110	10	4542	3021	3160	3185	3201	3487	3491	3484	3492	0.0423
168	12	10502	7523	7860	7894	7912	8509	8504	8494	8594	0.0355
126	12	5930	4052	4292	4318	4330	4706	4687	4672	4735	0.0380
134	12	6616	4402	4523	4557	4577	4955	5004	4963	5011	0.0397

Table 4: Results for medium-sized random graphs

Data				Lower bounds		Upper bounds		Rel. gap
n	k	$\|E\|$	u_0	eig_{-L}	eig_A	eig_{-L}	eig_A	
2012	35	575078	361996	345251	356064	442567	377016	0.0286
1545	35	351238	210375	193295	205921	258085	219868	0.0328
1840	35	439852	313006	295171	307139	371207	375468	0.0944
1960	45	532464	346838	323526	339707	402685	355098	0.0222
2059	45	543331	393845	369313	386154	469219	483654	0.0971
2175	45	684405	419955	396363	412225	541037	581416	0.1351
2658	55	924962	651547	614044	638827	780106	665760	0.0206
2784	55	1063828	702526	664269	690186	853750	922492	0.1059
2569	55	799319	624819	586527	612605	721033	713355	0.0760

Table 5: Results for larger structured graphs

Data			Lower bounds		Upper bounds		Rel. gap
n	k	$\|E\|$	eig_{-L}	eig_A	eig_{-L}	eig_A	
1608	35	969450	837200	851686	875955	875521	0.0138
1827	35	1250683	1066083	1083048	1112377	1112523	0.0134
1759	35	1159454	1032413	1048350	1075600	1074945	0.0125
2250	45	1897480	1669309	1694456	1735583	1734965	0.0118
2287	45	1959760	1808192	1838114	1879230	1877722	0.0107
2594	45	2522071	2183560	2212241	2263249	2264242	0.0114
2660	55	2651856	2481928	2516160	2568521	2566434	0.0099
2715	55	2763486	2503729	2535541	2589999	2589202	0.0105
2661	55	2652743	2413321	2442960	2495530	2495115	0.0106

Table 6: Results for larger random graphs

We now briefly comment on the computational time (measured by MATLAB tic-toc function) for the above tests. For lower bounds, the eigenvalue bounds are fastest to compute. Computational time for small, medium and larger problems are usually less than 0.01 seconds, 0.1 seconds and 0.5 minutes, respectively. The QP bounds are more expensive to compute, taking around 0.5 to 2 seconds for small instances and 0.5 to 15 minutes for medium-sized instances. The SDP bounds are even more expensive to compute, taking 0.5 to 3 seconds for small instances and 2 minutes to 2 hours for medium-sized instances. The DNN bounds are the most expensive to compute. Even for small instances, it can take 20 seconds to 40 minutes to compute a bound. For upper bounds, using the MATLAB simplex method, the time for solving (6.1) is usually less than 1 second for small and medium-sized problems; while for the larger problems in Tables 5 and 6, it takes 1 to 5 minutes.

Finding a Vertex Separator. Before ending this subsection, we comment on how the above bounds can possibly be used in finding vertex separators when m is not explicitly known beforehand. Since there can be at most $\binom{n-1}{k-1}$ k-tuples of integers summing up to n, theoretically, one can consider all possible such m and estimate the corresponding $\mathrm{cut}(m)$ with the bounds above.

As an illustration, we consider a concrete instance of a structured graph, generated with $n = 600$, $m_1 = m_2 = m_3 = 200$ and $p = 0$. Thus, we have $k = 3$, and, by construction, $\mathrm{cut}(m) = 0$.

Suppose that the correct size vector m is not known in advance. Therefore we now consider a range of estimated vectors m'. In Table 7, we consider sizes m'_1 and m'_2 with values taken between 180 to 220, with $m'_3 = 600 - m'_1 - m'_2$. We report on the eigenvalue bounds, the QP bounds and the SDP bounds for each m'. Observe that the SDP lower bounds are usually the largest while the QP upper bounds are usually the smallest. The existence of a vertex separator when $m_1 = m_2 = m_3 = 200$ is identified by the QP and SDP bounds.[5] Furthermore, the QP upper bound being zero for the cases $(m'_1, m'_2) = (180, 180)$, $(180, 200)$ or $(200, 180)$ also indicates the existence of a vertex separator.

Data		Lower bounds				Upper bounds			
m'_1	m'_2	eig_{-L}	eig_A	QP	SDP	eig_{-L}	eig_A	QP	SDP
180	180	-3600	-2400	-2400	-1800	2520	32400	0	540
180	200	-1922	-1281	-1270	-949	2538	36000	0	3240
180	220	-99	-66	-16	0	3600	39600	3600	4312
200	180	-1922	-1281	-1270	-949	2538	36000	0	1440
200	200	0	0	1	0	2200	39801	0	0
200	220	2074	2716	2759	4000	4000	40000	4398	11832
220	180	-99	-66	-16	0	3600	39600	3958	19768
220	200	2074	2716	2759	4000	4000	40000	11518	11200
220	220	4400	5867	5867	8400	8400	40241	8400	12916

Table 7: Results for medium-sized graph without an explicitly known m

7.2 Large Sparse Projected Eigenvalue Bounds

We assume that $n \gg k$. The projected eigenvalue bound in Theorem 3.7 in (3.13) is composed of a constant term, a minimal scalar product of $k-1$ eigenvalues and a linear term. The constant term and linear term are trivial to evaluate and essentially take no CPU time. The evaluation of the $k-1$ eigenvalues of $\widehat{B}$ is also efficient and accurate as the matrix is small and symmetric. The only significant cost is the evaluation of the largest $k-2$ eigenvalues and the smallest eigenvalue

[5] The QP lower bound of 1 in this case actually corresponds to an objective value in the order of $1e-5$. We obtain the 1 since we always truncate the lower bound to the smallest integer exceeding it.

of $\widehat{G}$. In our test below, we use $G = A$ for simplicity. This choice is also justified by our numerical results in the previous subsection and the observation from Figures 1 and 2.

We use the MATLAB *eigs* command for the $k-1$ eigenvalues of V^TAV for the lower bound. Since the corresponding (6.1) has much larger dimension than we considered in the previous subsection, we turn to IBM ILOG CPLEX version 12.4 (MATLAB interface) with default settings to solve for the upper bound. We use the MATLAB tic-toc function to time the routine for finding the lower bound, and report `output.time` from the function `cplexlp.m` as the cputime for finding the upper bound.

We use two different choices V_0 and V_1 for the matrix V in (3.7).

1. We choose the following matrix V_0 with mutually orthogonal columns that satisfies $V_0^T e = 0$.[6]

$$V_0 = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ -1 & 1 & 1 & \dots & 1 \\ 0 & -2 & 1 & \dots & 1 \\ 0 & 0 & -3 & \dots & 1 \\ \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & \dots & -(n-1) \end{bmatrix}$$

   Let $s = \big(\|V_0(:,i)\|\big) \in \mathbb{R}^{n-1}$. Then the operation needed for the MATLAB large sparse eigenvalue function *eigs* is ($*$ denotes multiplication and $\cdot'$ denotes transpose, ./ denotes elementwise division)

$$\widehat{A} * v = V' * (A * (V * v)) = V_0' * (A * (V_0 * (v./s)))./s. \tag{7.2}$$

   Thus we never form the matrix $\widehat{A}$ and we preserve the structure of V_0 and sparsity of A when doing the matrix-vector multiplications.

2. An alternative approach uses

$$V_1 = \left[\begin{bmatrix} \left[I_{\lfloor \frac{n}{2} \rfloor} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right] \\ 0_{(n-2\lfloor \frac{n}{2} \rfloor), \lfloor \frac{n}{2} \rfloor} \end{bmatrix} \begin{bmatrix} \left[I_{\lfloor \frac{n}{4} \rfloor} \otimes \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \right] \\ 0_{(n-4\lfloor \frac{n}{4} \rfloor), \lfloor \frac{n}{4} \rfloor} \end{bmatrix} \left[\dots \right] \left[\widehat{V} \right] \right]_{n \times n-1}$$

   i.e., the block matrix consisting of t blocks formed from Kronecker products along with one block $\widehat{V}$ to complete the appropriate size so that $V^TV = I_{n-1}$, $V^Te = 0$. We take advantage of the 0, 1 structure of the Kronecker blocks and delay the scaling factors till the end. Thus we use the same type of operation as in (7.2) but with V_1 and the new scaling vector s.

The results on large scale problems using the two choices V_0 and V_1 are reported in Tables 8, 9 and 10. For simplicity, we only consider random graphs, with various `imax` and k. We generate m as described before and use the commands

[6] Choosing a sparse V in the orthogonal matrix in (3.7) would speed up the calculation of the eigenvalues. Choosing a sparse V would be easier if V did not require orthonormal columns but just linearly independent columns, i.e., if we could arrange for a parametrization as in Lemma 3.6 without P orthogonal.

```
A=sprandsym(n,dens); A(1:n+1:end)=0; A(abs(A)>0)=1;
```

to generate a random incidence matrix, with dens $= 0.05/i$, for $i = 1, \ldots, 10$. In the tables, we present the number of nodes, sets, edges (n, k, $|E|$), the true density of the random graph $density := 2|E|/(n(n-1))$, the lower and upper projected eigenvalue bounds, the relative gap (7.1), and the cputime (in seconds) for computing the bounds.

The results using the matrix V_0 are in Tables 8. Here the cost for finding the lower bound using the eigenvalues becomes significantly higher than the cost for finding the upper bound using the simplex method.

n	k	$\|E\|$	density	lower	upper	Rel. gap	cpu (low)	cpu (up)
13685	68	4566914	4.88×10^{-2}	3958917	4271928	0.0380	409.4	7.1
13599	65	2282939	2.47×10^{-2}	1967979	2181778	0.0515	330.1	6.1
13795	68	1572487	1.65×10^{-2}	1314033	1495421	0.0646	316.2	7.9
13249	66	1090447	1.24×10^{-2}	832027	985375	0.0844	265.6	7.4
12425	66	767961	9.95×10^{-3}	589226	710093	0.0930	253.2	6.0
13913	66	803074	8.30×10^{-3}	591486	726783	0.1026	304.9	7.1
14144	65	711936	7.12×10^{-3}	543017	666721	0.1023	274.4	7.1
13667	67	581930	6.23×10^{-3}	427464	538291	0.1148	254.9	6.5
12821	68	455329	5.54×10^{-3}	329902	422417	0.1230	244.5	7.4
12191	69	370595	4.99×10^{-3}	262521	343426	0.1335	211.1	6.3

Table 8: Large scale random graphs; imax 400; $k \in [65, 70]$, using V_0

The results using the matrix V_1 are shown in Tables 9 and 10. We can see the obvious improvement in cputime when finding the lower bounds using V_1 compared to using V_0, which becomes more significant when the graph gets sparser.

n	k	$\|E\|$	density	lower	upper	Rel. gap	cpu (low)	cpu (up)
14680	69	5254939	4.88×10^{-2}	4586083	4955524	0.0387	262.9	6.4
14464	65	2583109	2.47×10^{-2}	2133187	2397098	0.0583	135.5	6.0
14974	69	1852955	1.65×10^{-2}	1555718	1776249	0.0662	98.2	6.9
13769	65	1177579	1.24×10^{-2}	956260	1124729	0.0810	44.4	5.9
13852	69	954632	9.95×10^{-3}	775437	924265	0.0876	51.3	6.0
12516	65	650028	8.30×10^{-3}	475477	598372	0.1144	34.0	4.3
13525	66	651025	7.12×10^{-3}	508512	630663	0.1072	33.3	5.8
13622	66	578111	6.23×10^{-3}	414786	535755	0.1273	34.6	6.0
13004	65	468437	5.54×10^{-3}	328925	434795	0.1386	29.1	5.2
14659	69	535899	4.99×10^{-3}	380571	501082	0.1367	27.2	5.9

Table 9: Large scale random graphs; imax 400; $k \in [65, 70]$, using V_1

In all three tables, we note that the relative gaps deteriorate as the density decreases. Also, the cputime for the eigenvalue bound is significantly better when using V_1 suggesting that sparsity of V_1 is better exploited in the MATLAB *eigs* command.

n	k	$\|E\|$	density	lower	upper	Rel. gap	cpu (low)	cpu (up)
22840	80	12721604	4.88×10^{-2}	11548587	12262688	0.0300	782.4	12.5
16076	77	3190788	2.47×10^{-2}	2754650	3053622	0.0515	199.1	8.9
20635	77	3519170	1.65×10^{-2}	2916188	3287657	0.0599	228.5	10.1
19408	79	2339682	1.24×10^{-2}	1989278	2272340	0.0664	147.3	10.6
17572	76	1536161	9.95×10^{-3}	1188933	1417085	0.0875	83.6	9.0
18211	80	1376087	8.30×10^{-3}	1127696	1336407	0.0847	90.7	11.2
21041	80	1575333	7.12×10^{-3}	1232501	1482463	0.0921	93.6	10.5
20661	77	1329856	6.23×10^{-3}	1023056	1251437	0.1004	74.5	11.8
19967	77	1104350	5.54×10^{-3}	831335	1035126	0.1092	74.0	9.6
20839	78	1082982	4.99×10^{-3}	831672	1034104	0.1085	73.9	11.0

Table 10: Large scale random graphs; imax 500; $k \in [75, 80]$, using V_1

8 Conclusion

In this paper, we presented eigenvalue, projected eigenvalue, QP, and SDP lower and upper bounds for a minimum cut problem. In particular, we looked at a variant of the projected eigenvalue bound found in [20] and showed numerically that our variant is stronger. We also proposed a new QP bound following the approach in [1], making use of a duality result presented in [19]. In addition, we studied an SDP relaxation and demonstrated its strength by showing the redundancy of quadratic (orthogonality) constraints. We emphasize that these techniques for deriving bounds for our cut minimization problem can be adapted to derive new results for the GP. Specifically, one can easily adapt our derivation and obtain a QP lower bound for the GP, which was not previously known in the literature. Our derivation of the simple facially reduced SDP relaxation (SDP_{final}) can also be adapted to simplify the existing SDP relaxation for the GP studied in [28].

We also compared these bounds numerically on randomly generated graphs of various sizes. Our numerical tests illustrate that the projected eigenvalue bounds can be found efficiently for large scale sparse problems and that they compare well against other more expensive bounds on smaller problems. It is surprising that the projected eigenvalue bounds using the adjacency matrix A are both cheap to calculate and strong.

Index

References

[1] K.M. Anstreicher and N.W. Brixius. A new bound for the quadratic assignment problem based on convex quadratic programming. *Math. Program.*, 89(3, Ser. A):341–357, 2001. 3, 13, 14, 29

[2] K.M. Anstreicher and H. Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM J. Matrix Anal. Appl.*, 22(1):41–55, 2000. 3, 13

[3] E. Balas, S. Ceria, and G. Cornuejols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Programming*, 58:295–324, 1993. 16

[4] 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. 18

[5] N.W. Brixius and K.M. Anstreicher. Solving quadratic assignment problems using convex quadratic programming relaxations. *Optim. Methods Softw.*, 16(1-4):49–68, 2001. Dedicated to Professor Laurence C. W. Dixon on the occasion of his 65th birthday. 3, 13, 16

[6] R. A. Brualdi and H. J. Ryser. *Combinatorial Matrix Theory.* Cambridge University Press, New York, 1991. 21

[7] 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. 18

[8] E. de Klerk, M. E.-Nagy, and R. Sotirov. On semidefinite programming bounds for graph bandwidth. *Optim. Methods Softw.*, 28(3):485–500, 2013. 5

[9] J.W. Demmel. *Applied numerical linear algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. 12

[10] J. Falkner, F. Rendl, and H. Wolkowicz. A computational study of graph partitioning. *Math. Programming*, 66(2, Ser. A):211–239, 1994. 3, 6, 8

[11] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In *Global optimization*, volume 84 of *Nonconvex Optim. Appl.*, pages 155–210. Springer, New York, 2006. 23

[12] S.W. Hadley, F. Rendl, and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Math. Oper. Res.*, 17(3):727–739, 1992. 3, 8

[13] W.W. Hager and J.T. Hungerford. A continuous quadratic programming formulation of the vertex separator problem. Report, University of Florida, Gainesville, 2013. 13

[14] A.J. Hoffman and H.W. Wielandt. The variation of the spectrum of a normal matrix. *Duke Mathematics*, 20:37–39, 1953. 6, 7

[15] R.A. Horn and C.R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original. 11

[16] R.H. Lewis. Yet another graph partitioning problem is NP-Hard. Report arXiv:1403.5544, [cs.CC], 2014. 3

[17] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991. 16

[18] R. Martí, V. Campos, and E. Piñana. A branch and bound algorithm for the matrix bandwidth minimization. *European J. Oper. Res.*, 186(2):513–528, 2008. 5

[19] Janez Povh and Franz Rendl. Approximating non-convex quadratic programs by semidefinite and copositive programming. In *KOI 2006—11th International Conference on Operational Research*, pages 35–45. Croatian Oper. Res. Soc., Zagreb, 2008. 13, 29

[20] 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. 3, 5, 6, 7, 9, 12, 29

[21] F. Rendl and H. Wolkowicz. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Math. Programming*, 53(1, Ser. A):63–78, 1992. 3, 6

[22] 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). 3, 6, 8, 9

[23] Alexander Schrijver. *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication. 5, 12

[24] H.D. Sherali and W.P. Adams. Computational advances using the reformulation-linearization technique (rlt) to solve discrete and continuous nonconvex problems. *Optima*, 49:1–6, 1996. 16

[25] E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34(2):250–256, 1986. 21

[26] E. Tardos. Strongly polynomial and combinatorial algorithms in optimization. In *Proceedings of the International Congress of Mathematicians, Vol. I, II (Kyoto, 1990)*, pages 1467–1478, Tokyo, 1991. Math. Soc. Japan. 21

[27] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95(2, Ser. B):189–217, 2003. Computational semidefinite and second order cone programming: the state of the art. 23

[28] 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. 3, 6, 16, 18, 19, 20, 29

[29] 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). 16, 22