A Strictly Contractive Peaceman-Rachford Splitting Method for the Doubly Nonnegative Relaxation of the Minimum Cut Problem *

Xinxin Li† Ting Kei Pong‡ Hao Sun § Henry Wolkowicz ¶

Department of Combinatorics & Optimization
University of Waterloo, Canada

Tuesday 22^{nd} October, 2019

Abstract

The minimum cut problem, **MC**, and the special case of the vertex separator problem, consists in partitioning the set of nodes of a graph G into k subsets of given sizes in order to minimize the number of edges cut after removing the k-th set. Previous work on this topic uses eigenvalue, semidefinite programming, **SDP**, and doubly nonnegative, **DNN**, bounds, with the latter being strong but expensive. In this paper, we derive strengthened **SDP** and **DNN** relaxations, and propose a scalable algorithmic approach for efficiently evaluating both upper and lower bounds.

Our stronger relaxations are based on a new *gangster set*, and we demonstrate how *facial reduction*, **FR**, fits in well to allow for *regularized* relaxations. Moreover, the **FR** appears to be perfectly well suited for a *natural* splitting of variables and thus for the application of *splitting* methods. Here, we adopt the strictly contractive Peaceman-Rachford splitting method, **sPRSM**. We discuss how *useful* redundant constraints can be brought back to the subproblems involved to empirically accelerate the **sPRSM**. We also propose new strategies for obtaining lower bounds and upper bounds of the optimal value of **MC** from the iterates of the **sPRSM** to help the algorithm terminate early. Numerical experiments on random datasets and vertex separator problems comparing with other existing approaches demonstrate the efficiency and robustness of the proposed method.

Key Words: Semidefinite relaxation, doubly nonnegative relaxation, min-cut, graph partitioning, vertex separator, Peaceman-Rachford splitting method, facial reduction.
AMS Subject Classification: 05C70, 90C22, 90C25, 90C27, 90C59

---

*This paper is partially based on the Master's thesis of Hao Sun [26]. The authors Hao Sun and Henry Wolkowicz thank the Natural Sciences and Engineering Research Council of Canada for their support.

†School of Mathematics, Jilin University, Changchun, China. E-mail: xinxinli@jlu.edu.cn. This work was supported by the National Natural Science Foundation of China (No.11601183, No. 61872162) and Natural Science Foundation for Young Scientist of Jilin Province (No. 20180520212JH).

‡Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, The People's Republic of China, E-mail: tk.pong@polyu.edu.hk. Research was supported partly by Hong Kong Research Grants Council PolyU153004/18p.

§Email: h34sun@uwaterloo.ca

¶Email: hwolkowicz@uwaterloo.ca

Contents

List of Tables

List of Algorithms

1 Introduction

We present strengthened doubly nonnegative, both positive semidefinite and nonnegative elementwise, relaxations for the min-cut problem, **MC**, i.e., the problem of partitioning the set of nodes of a graph G into k subsets of given sizes in order to minimize the number of edges cut after removing the k-th set. Our relaxations are aimed at specifically applying splitting methods based on using the regularization technique facial reduction, **FR**, as well as employing new so-called *gangster constraints*. This results in strengthened upper and lower bounds for **MC**.

We consider an *undirected graph* $G = (\mathcal{V}, \mathcal{E})$ with vertex and edge sets $\mathcal{V}$, $\mathcal{E}$, respectively, and $|\mathcal{V}| = n$. We let $m = (m_1 \ m_2 \ \dots \ m_k)^T$, $\sum_{i=1}^k m_i = n$, denote a given partition of n into k sets. The special type of *minimum cut problem*, **MC**, we consider consists in partitioning the vertex set $\mathcal{V}$ into k subsets, with given sizes in m, in order to *minimize the cut* obtained after removing the k-th set, i.e., we minimize the number of edges connecting distinct sets other than those edges connected to the k-th set, see e.g., [21]. This problem arises for example 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 within a Cholesky factorization, e.g., [22]. The **MC** has further applications in computer program segmentation, solving symmetric systems of equations, microchip design and circuit board, floor planning and other layout problems [20]. In particular herein, we include consideration of the vertex separator problem, i.e., finding a vertex set whose removal splits the graph into *two* disconnected subsets, see e.g., [8, 22].

It is well known that **MC** is an NP-hard problem when $k \geq 3$, see e.g., [15,21]. Solution techniques rely on efficiently calculating lower and upper bounds. We refer the readers to [7,11,19,21,22] and the references therein for recent results for finding bounds and solving **MC**; and also to [22, Section 2] for a recent overview of existing relaxation techniques for solving **MC**. An important tool for finding lower bounds is the *semidefinite programming, **SDP***, relaxation of **MC**; this is included in [19]. Moreover, this relaxation uses *facial reduction* **FR** to guarantee strict feasibility and robustness for both the relaxation and its dual. However, these **SDP** problems are typically solved by interior point methods: these methods often do not scale well and cannot properly exploit sparsity. Moreover, while **SDP** lower bounds can be strengthened to yield better approximations to **MC** by adding extra nonnegativity and cutting plane constraints, the resulting optimization problems can be prohibitively expensive to solve for interior point solvers. Thus, in order to improve **MC** approximations, besides deriving tighter upper and lower bounds, one also needs to design efficient and scalable algorithms for computing these bounds.

1.1 Main Contributions

In this paper, we derive tighter (lower and upper) bounds and design efficient algorithms for their evaluation. The bounds are based on strengthened **SDP** and doubly nonnegative, **DNN**, relaxations within a **FR** framework. Moreover, we introduce a random weighted sampling of eigenvectors to strengthen the upper bounds.

Our stronger relaxations use a new *gangster* set; see Definition 2.4. This set can be larger than the one used in the literature, e.g., [19,28], when some of the set sizes $m_i = 1$. Then, as in [19], we apply **FR** to simplify these stronger **SDP** and **DNN** relaxations so that the facially reduced problems satisfy Robinson's regularity condition. In addition, we show that many of the constraints are redundant in the facially reduced problem, resulting in a greatly simplified relaxation.

Although many redundant constraints are removed, our final **DNN** relaxation is still very difficult to solve for interior point solvers. Here, we propose a scalable algorithmic approach. The key idea is that **FR** gives a natural way of reformulating the facially reduced **DNN** relaxation into a *separable* convex programming problem with linear coupling constraints. This sets the stage for an application of *splitting methods* such as *alternating direction method of multipliers,* ***ADMM*** [4]. These methods typically involve updating the multiplier(s) and solving several subproblems every iteration. Their efficiency depends highly on the simplicity of the subproblems, and they can take a lot of iterations to obtain high accuracy solutions.

Herein we employ a particular variant of **ADMM**, the *strictly contractive Peaceman-Rachford splitting method,* ***sPRSM***, [12,13]. This method involves two subproblems and two updates of the multiplier at every iteration. While a direct application of this method can be slow (i.e., takes a lot of iterations), we introduce *two* key ingredients for empirical acceleration. First, instead of just using the natural splitting induced by **FR**, as in the recent work [18], we bring back some provably redundant constraints that are *not* redundant for the subproblems as long as the constraint does not significantly increase the computational cost. Second, we derive new strategies for obtaining lower bounds and upper bounds of the true optimal value of **MC**. This helps with early termination of **sPRSM** when the two bounds agree. Specifically, we compute a lower bound by looking at the Fenchel dual. Moreover, we mimic the now classical Goeman-Williamson's approach for MAXCUT and use a random weighted sampling of eigenvectors of an iterate of the **sPRSM** before projecting it onto the set of partition matrices for computing an upper bound.

In the numerical experiments, we illustrate the efficiency of our proposed algorithmic approach (based on the strengthened **DNN** relaxation model) by comparing with the **DNN** relaxation model in [19], as well as the $\mathbf{SDP}_4$ model in [22]. Our experiments show that our approach takes less computational time and the bounds obtained are typically tighter.

1.1.1 Outline

In Section 2 we discuss properties of our new gangster sets and our facially reduced **SDP** and **DNN** relaxations. Our algorithmic **sPRSM** approach is presented in Section 3. We discuss the usefulness of redundant constraints and include details of the subproblems of **sPRSM**. And, we describe methods for obtaining both lower and upper bounds from possibly inaccurate solutions of the **sPRSM**. Our numerical results are presented in Section 4. Concluding remarks are given in Section 5.

1.2 Preliminaries

Let A be the adjacency matrix of our graph, $G = (\mathcal{V}, \mathcal{E})$. Let e be the all ones vector, E be the square matrix of all ones and I be the identity matrix, all of appropriate sizes.[1] We set

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

where $\mathbb{S}^k$ is the space of real symmetric $k \times k$ matrices equipped with the *trace inner product*, $\langle S, T \rangle = \operatorname{trace} ST$, and the corresponding *Fröbenius norm,* $\|S\|_F$. We use $\|S\| = \|S\|_F$, when the meaning is clear.

Let $m = (m_1, \ldots, m_k)^T \in \mathbb{Z}^k_+, k > 2$, and let $n = |\mathcal{V}| = m^T e$. Let $S = \{S_1, S_2, \ldots, S_k\}$ be a partition of the vertex set with cardinalities $|S_i| = m_i > 0$, $i = 1, \ldots, k$, i.e., the sets are nonempty, pairwise disjoint, and the union is S. In addition, we let $M = \operatorname{Diag}(m)$ denote the diagonal matrix formed from the vector m. More generally, for a vector $x \in \mathbb{R}^j$, we define $\operatorname{Diag} : \mathbb{R}^j \to \mathbb{S}^j$ to be the linear transformation that maps x to the diagonal matrix whose diagonal is x; we denote its adjoint linear transformation by diag, i.e, $\operatorname{diag} := \operatorname{Diag}^*$. Next, we define the set of edges between two sets of nodes by

$$\delta(S_i, S_j) := \{uv \in \mathcal{E} : u \in S_i, v \in S_j\}.$$

The *cut of a partition S, $\delta(S)$*, is then defined as the union of all edges cut by the first $k-1$ sets of the partition, i.e.,

$$\delta(S) := \cup \{\delta(S_i, S_j) \,:\, 1 \le i < j \le k-1\}.$$

Our objective is to minimize the cardinality of the cut, i.e., $|\delta(S)|$. In [21], it is shown that $|\delta(S)|$ can be represented in terms of a quadratic form of the partition matrix X. This quadratic form for the **MC** problem in the trace formulation is

$$\begin{array}{rcl} \operatorname{cut}(m) = & \min & \frac{1}{2} \operatorname{trace} AXBX^T \\ & \text{s.t.} & X \in \mathcal{M}_m, \end{array} \tag{1.1}$$

where the set of *partition matrices, $\mathcal{M}_m$* is defined by

$$\mathcal{M}_m = \left\{ X \in \mathbb{R}^{n \times k} : Xe = e,\ X^T e = m, X_{ij} \in \{0, 1\} \right\},$$

i.e., column j of a partition matrix X is the *indicator vector* for set S_j. We let $x = \operatorname{vec}(X) \in \mathbb{R}^{nk}$ denote the columnwise vectorization of the matrix X. The inverse and *adjoint linear transformation* $\operatorname{Mat} : \mathbb{R}^{nk} \to \mathbb{R}^{n \times k}$ is

$$X = \operatorname{Mat}(x) = \operatorname{vec}^*(x) = \operatorname{vec}^{-1}(x).$$

2 SDP and DNN relaxations of MC

In this section, we strengthen the facially reduced **SDP** relaxation presented in [19] and present our strengthened **DNN** relaxation to be used with our **sPRSM** approach below in Section 3. One way to derive an **SDP** relaxation for (1.1) is to start by considering a Lagrangian relaxation of a quadratic-quadratic model of **MC**. Taking the dual of the dual of this Lagrangian relaxation then

[1] We will also use subscripts to specify the dimension whenever necessary, i.e., for a positive integer j, e_j is the j-dimensional vector of all ones, $E_j = e_j e_j^T$ and I_j is the $j \times j$ identity matrix.

gives the **SDP** relaxation for (1.1); see also [28,30] for the development for other hard combinatorial problems. Alternatively, we can obtain the *same* **SDP** relaxation directly using the well-known *lifting process*, e.g., [2,16,25,28,30].

2.1 Quadratic-quadratic models

In our approach, we start with the following two equivalent quadratically constrained quadratic problems to (1.1):

$$\begin{array}{rll} \operatorname{cut}(m) = & \min & \frac{1}{2}\operatorname{trace} AXBX^T \\ & \text{s.t.} & X \circ X = X \\ & & \|Xe-e\|^2 = 0 \\ & & \|X^Te-m\|^2 = 0 \\ & & X_{:i} \circ X_{:j} = 0,\ \forall i \neq j \\ & & X^TX - M = 0 \\ & & \operatorname{diag}(XX^T) - e = 0 \end{array} \quad \begin{array}{rll} = & \min & \frac{1}{2}\operatorname{trace} AXBX^T \\ & \text{s.t.} & X \circ X = x_0X \\ & & \|Xe-x_0e\|^2 = 0 \\ & & \|X^Te-x_0m\|^2 = 0 \\ & & X_{:i} \circ X_{:j} = 0,\ \forall i \neq j \\ & & X^TX - M = 0 \\ & & \operatorname{diag}(XX^T) - e = 0 \\ & & x_0^2 = 1. \end{array} \tag{2.1}$$

The equivalence of the constraint set in the first optimization problem in (2.1) to $\mathcal{M}_m$ can be found in [29]. Here $u \circ v$ denotes the Hadamard (elementwise) product of the two vectors u, v. Note that we add x_0 and the constraint $x_0^2 = 1$ to *homogenize* the linear terms. 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 the second optimization problem in (2.1) using the matrix lifting

$$Y := \begin{pmatrix} x_0 \\ x \end{pmatrix} \begin{pmatrix} x_0 & x^T \end{pmatrix}, \quad x = \operatorname{vec}(X). \tag{2.2}$$

Then $Y \in \mathbb{S}_+^{nk+1}$ and is rank-one. The rows and columns of Y are indexed from 0 to nk. Note that Y in (2.2) can be blocked appropriately as

$$Y = \begin{bmatrix} Y_{00} & Y_{1:nk\,0}^T \\ Y_{1:nk\,0} & \overline{Y} \end{bmatrix}, \quad Y_{1:nk\,0} = \begin{bmatrix} Y_{(10)} \\ Y_{(20)} \\ \vdots \\ Y_{(k0)} \end{bmatrix}, \quad \overline{Y} = \begin{bmatrix} \overline{Y}_{(11)} & \overline{Y}_{(12)} & \cdots & \overline{Y}_{(1k)} \\ \overline{Y}_{(21)} & \overline{Y}_{(22)} & \cdots & \overline{Y}_{(2k)} \\ \vdots & \ddots & \ddots & \vdots \\ \overline{Y}_{(k1)} & \ddots & \ddots & \overline{Y}_{(kk)} \end{bmatrix}, \tag{2.3}$$

with

$$\overline{Y}_{(ij)} \in \mathbb{R}^{n\times n},\ \forall i \neq 0, \forall j \neq 0,\ \text{and } Y_{(j0)} \in \mathbb{R}^n,\ \forall j = 1,\ldots,k.$$

With the matrix lifting for Y, we can rewrite the objective function in (2.1) in linearized form as

$$\frac{1}{2}\operatorname{trace} AXBX^T = \frac{1}{2}\operatorname{trace} L_AY, \tag{2.4}$$

where

$$L_A := \begin{bmatrix} 0 & 0 \\ 0 & B \otimes A \end{bmatrix}.$$

We next recall how to obtain linearized formulations for the constraints in the second optimization problem in (2.1), i.e., all the quadratic terms in (2.1) are linearized with the rank-one positive semidefinite matrix Y in (2.2). Therefore, we obtain an equivalent rank-one **SDP** model.

2.2 SDP and DNN constraints

2.2.1 The arrow constraint

It follows from the first constraint in the second optimization problem in (2.1), $x_0^2 = 1$ and (2.2) that the diagonal equals the first column (and row) and that $Y_{00} = 1$, i.e.,

$$Y \in \{Y \in \mathbb{S}^{nk+1} : Y_{00} = 1, \, \operatorname{diag}(Y) = Y_{:0}\}.$$

The above set is further clarified by using the linear mapping *arrow*: $\mathbb{S}^{nk+1} \to \mathbb{R}^{nk+1}$, and the corresponding constraint

$$\operatorname{arrow}(Y) := \operatorname{diag}(Y) - \begin{bmatrix} 0 \\ Y_{1:nk\,0} \end{bmatrix} = e_0, \tag{2.5}$$

where e_0 is the first (0-th) unit vector. This constraint is redundant in the final **SDP** relaxation (see Theorem 2.13 below).

2.2.2 DNN, doubly nonnegative

From the matrix lifting in (2.2), we obtain $Y \succeq 0$. Then the arrow constraint yields nonnegativity for the first row (and column) of Y. Now from the first and last constraints in the second optimization problem in (2.1), and relaxing the $0,1$ property of $x_0 X \in \mathcal{M}_m$ to $0 \le x_0 X \le 1$, we obtain the following constraints

$$Y \in \mathbf{DNN} \cap \{Y \in \mathbb{S}^{nk+1} \, : \, 0 \le Y \le 1\}, \tag{2.6}$$

where, by abuse of notation, **DNN** also stands for the doubly nonnegative cone, i.e., the intersection of the positive semidefinite cone and the nonnegative orthant.

2.2.3 Trace constraints

Using (2.2), the second and third constraints in the second optimization problem in (2.1) along with $x_0^2 = 1$ yields

$$\begin{aligned} \operatorname{trace} D_1 Y = 0, \qquad & D_1 := \begin{bmatrix} n & -e_k^T \otimes e_n^T \\ -e_k \otimes e_n & (e_k e_k^T) \otimes I_n \end{bmatrix}, \\ \operatorname{trace} D_2 Y = 0, \qquad & D_2 := \begin{bmatrix} m^T m & -m^T \otimes e_n^T \\ -m \otimes e_n & I_k \otimes (e_n e_n^T) \end{bmatrix}, \end{aligned} \tag{2.7}$$

where e_j is the vector of ones of dimension j. Here $D_i \succeq 0, i = 1, 2$. The nullspaces of these matrices yield the facial reduction, as we will discuss in Section 2.3 below. The detailed derivation can be found in e.g., [10]. These two constraints are redundant in the **SDP** relaxation after the **FR**; see Theorem 2.13 below.

2.2.4 Block: trace, diagonal and off-diagonal

We now consider the fifth and the sixth constraints in (2.1). We define the following linear transformations.

Definition 2.1. *Let $Y \in \mathbb{S}^{nk+1}$ be blocked as in (2.3). Define the linear transformation $\mathcal{D}_t : \mathbb{S}^{nk+1} \to \mathbb{S}^k$ so that $(\mathcal{D}_t(Y))_{ij}$ is the trace of the block $\overline{Y}_{(ij)}$, i.e.,*

$$\mathcal{D}_t(Y) := \left(\operatorname{trace} \overline{Y}_{(ij)}\right) \in \mathbb{S}^k;$$

define the linear transformation $\mathcal{D}_d : \mathbb{S}^{nk+1} \to \mathbb{R}^n$ as the sum of diagonals in each block $\overline{Y}_{(ii)}$, i.e.,

$$\mathcal{D}_d(Y) := \sum_{i=1}^{k} \operatorname{diag} \overline{Y}_{(ii)} \in \mathbb{R}^n;$$

define the linear transformation $\mathcal{D}_o : \mathbb{S}^{nk+1} \to \mathbb{S}^k$ so that $(\mathcal{D}_o(Y))_{ij}$ is the sum of off-diagonal entries in the block $\overline{Y}_{(ij)}$, i.e.,

$$\mathcal{D}_o(Y) := \left(\sum_{s \neq t} \left(\overline{Y}_{(ij)}\right)_{st} \right) \in \mathbb{S}^k.$$

We have the following results for the transformations $\mathcal{D}_t, \mathcal{D}_d$, and $\mathcal{D}_o$.

Proposition 2.2. *Let Y be defined as in (2.2) with X and x_0 satisfying the constraints in the second optimization problem in (2.1). Let $\widehat{M} := mm^T - M$. Then the following holds:*

$$\mathcal{D}_t(Y) = M; \quad \mathcal{D}_d(Y) = e_n; \quad \mathcal{D}_o(Y) = \widehat{M}. \tag{2.8}$$

Proof. For any feasible Y blocked as in (2.3), along with the fifth, sixth and seventh constraints in (2.1), we have the corresponding block trace and block diagonal constraints:

$$D_t(Y) = \left(\operatorname{trace} \overline{Y}_{(ij)}\right) = \left(\operatorname{trace} X_{:i} X_{:j}^T\right) = \left(\operatorname{trace} X_{:j}^T X_{:i}\right) = \left(X_{:j}^T X_{:i}\right) = X^T X = M;$$
$$D_d(Y) = \sum_{i=1}^{k} \operatorname{diag} \overline{Y}_{(ii)} = \sum_{i=1}^{k} \operatorname{diag}(X_{:i} X_{:i}^T) = \operatorname{diag}(\sum_{i=1}^{k} X_{:i} X_{:i}^T) = \operatorname{diag}(XX^T) = e.$$

These prove the first two equations in (2.8). Next, note that

$$\mathcal{D}_o(Y) = \left(\textstyle\sum_{s \neq t} \left(\overline{Y}_{(ij)}\right)_{st}\right) = \left(e^T \overline{Y}_{(ij)} e\right) - \left(\operatorname{trace} \overline{Y}_{(ij)}\right).$$

Using this together with the third and the last constraints in (2.1), we have

$$\left(e^T \overline{Y}_{(ij)} e\right) = \left(e^T \left(X_{:i} X_{:j}^T\right) e\right) = \left(m_i x_0 m_j x_0\right) = mm^T.$$

It then follows from the above two equations and the first equation in (2.8) that

$$\mathcal{D}_o(Y) = mm^T - M.$$

□

Corollary 2.3. *Let Y be defined as in (2.2) with X and x_0 satisfying the constraints in the second optimization problem in (2.1). Partition Y in blocks as in (2.3). Then we have*

$$\operatorname{trace} Y = n+1 \tag{2.9}$$

and

$$e^T Y_{(i\,0)} = m_i,\; i=1,\ldots,k. \tag{2.10}$$

Moreover, the objective value in (2.4) satisfies

$$\frac{1}{2}\operatorname{trace}(L_A+\alpha I)Y = \frac{1}{2}\operatorname{trace} L_A Y + \frac{\alpha}{2}(n+1),\; \forall \alpha\in\mathbb{R}. \tag{2.11}$$

Proof. The first equation (2.9) follows from $\mathcal{D}_t(Y)=M$ in (2.8), and the facts that $e^Tm=n$ and $Y_{00}=1$. The second equation (2.10) can be obtained by combining $\mathcal{D}_t(Y)=M$ and the arrow constraint (2.5). The last equation follows immediately from (2.9). □

All the constraints in (2.8) are redundant in the final **SDP** relaxation; see Theorem 2.13 below.

2.2.5 Gangster constraints

We now obtain constraints on the individual blocks in the submatrix $\overline{Y}$, based on the fourth constraint in (2.1). These constraints typically result in elements of Y being set to 0.[2] We let $\mathcal{G}_\Omega$ represent the *coordinate projection map* on $\mathbb{S}^{nk+1}$ that chooses the elements in the index set Ω, i.e,

$$\mathcal{G}_\Omega(Y) \;=\; \left(Y_{ij}\right)_{ij\in\Omega}\; (\in \mathbb{R}^{|\Omega|}),\quad \Omega\subseteq\Delta_{0:nk} := \{ij : 0\le i\le j\le nk\}.$$

By abuse of notation, we assume that the (gangster) indices are restricted to the upper triangular indices $\Delta_{0:nk}$, even when not specified so. We denote the complement of Ω in $\Delta_{0:nk}$ by Ω^c. The adjoint of $\mathcal{G}_\Omega$, denoted by $\mathcal{G}^*_\Omega : \mathbb{R}^{|\Omega|}\to\mathbb{S}^{nk+1}$, is given by

$$(\mathcal{G}^*_\Omega(w))_{ij} \;=\; \begin{cases} \frac{1}{2}w_{ij} & \text{if } \; i\neq j \;\text{ and }\; ij \text{ or } ji \;\in\Omega,\\ w_{ii} & \text{if } \; ii\; \in\Omega,\\ 0 & \text{otherwise.}\end{cases}$$

We now define the following index sets, including the gangster index set.

Definition 2.4 (Restricted gangster set). *Let $\mathcal{K}:=\{1,\ldots,k\}$, $\mathcal{I}:=\{i\in\mathcal{K} : m_i=1\}$, and the complement $\mathcal{I}^c := \mathcal{K}\backslash\mathcal{I}$. Define $m_{\text{one}}\in\mathbb{R}^k$ by*

$$(m_{one})_i = \begin{cases} 1 & \text{if } i\in\mathcal{I},\\ 0 & \text{if } i\in\mathcal{I}^c.\end{cases}$$

Define $J_0\subseteq\Delta_{0:nk}$ to be the set of (gangster) indices corresponding to the ones in $(E_k-I_k)\otimes I_n + \operatorname{Diag}(m_{one})\otimes(E_n-I_n)$, i.e.,

$$J_0 := \Delta_{0:nk}\cap(\Theta_o\cup\Theta_{\mathcal{I}}), \tag{2.12}$$

where

$$\Theta_o := \{\textit{all diagonal positions of all off-diagonal blocks}\},$$
$$\Theta_{\mathcal{I}} := \{\textit{all off-diagonal positions of the ith diagonal blocks if } m_i=1\}.$$

[2]The name gangster refers to *shooting holes* in the matrix, a term coined originally by Philippe Toint.

Fix a $j_0 \in \mathcal{I}^c$. Define the gangster subsets, $J_i, i = 1, 2, 3$, by

$$
\begin{aligned}
J_1 &:= \textit{all diagonal positions of the } (i,k) \textit{ (and } (k,i)\textit{) blocks, } \forall i \in \mathcal{I}\backslash\{k\};\\
J_2 &:= \textit{all diagonal positions of the } (j_0,k) \textit{ (and } (k,j_0)\textit{) blocks;}\\
J_3 &:= \textit{all diagonal positions of the } (k-2,k-1) \textit{ (and } (k-1,k-2)\textit{) blocks.}
\end{aligned}
$$

Then we define the restricted gangster set, $J_{\mathcal{I}}$, as follows:

$$
(\Delta_{0:nk} \supseteq)\ J_{\mathcal{I}} = \begin{cases} J_0, & \text{if } \mathcal{I} = \emptyset \\ J_0\backslash J_1, & \text{if } k \notin \mathcal{I} \neq \emptyset \\ J_0\backslash (J_1 \cup J_2), & \text{if } k \in \mathcal{I} \neq \mathcal{K} \\ J_0\backslash (J_1 \cup J_3), & \text{if } \mathcal{I} = \mathcal{K}. \end{cases} \tag{2.14}
$$

We now have the following results concerning the restricted gangster set $J_{\mathcal{I}}$.

Proposition 2.5. *Let Y be defined as in (2.2) with X and x_0 satisfying the constraints in the second optimization problem in (2.1). Given the gangster set $J_0 \subseteq \Delta_{0:nk}$, the index set $\mathcal{I}$ and the restricted gangster set $J_{\mathcal{I}} \subseteq \Delta_{0:nk}$ as defined in Definition 2.4, the following gangster constraint and restricted gangster constraint on Y hold:*

$$
\mathcal{G}_{J_0}(Y) = 0 \quad \text{and} \quad \mathcal{G}_{J_{\mathcal{I}}}(Y) = 0. \tag{2.15}
$$

Proof. Because of the matrix lifting in (2.2) and the fourth constraints in (2.1), i.e., $X_{:i} \circ X_{:j} = 0, \forall\, i \neq j$, we conclude that all diagonal positions of all off-diagonal blocks of Y are zero.

Next, note that for any $i \in \mathcal{I}$, we have $m_i = 1$. From $\mathcal{D}_o(Y) = \widehat{M}$ in (2.8) we have

$$
(\mathcal{D}_o(Y))_{ii} = \left(\sum_{s \neq t} \left(\overline{Y}_{(ii)} \right)_{st} \right) = (mm^T - M)_{ii} = m_i(m_i - 1) = 0.
$$

It follows from the above equation and $Y \geq 0$ that the off-diagonal elements of $\overline{Y}_{(ii)}$ are zero. As a result, all diagonal positions of all off-diagonal blocks and all off-diagonal positions of the i-th diagonal blocks $\forall i \in \mathcal{I}$ are zero, i.e., $\mathcal{G}_{J_0}(Y) = 0$. Since $J_{\mathcal{I}} \subseteq J_0$, we conclude $\mathcal{G}_{J_{\mathcal{I}}}(Y) = 0$. □

Remark 2.6.

1. *We see that if $m_i = 1, \forall i$, then necessarily all the diagonal elements of all off-diagonal blocks and all the off-diagonal elements of all diagonal blocks are zero. This is precisely the case for the quadratic assignment problem,* ***QAP****, e.g., [18, 30].*

2. *Our definition of the gangster mapping differs from that in [19]. Specifically, we use the coordinate projection rather than an operator on the matrix space. Moreover, note that the gangster set J_0 is larger than the one used in [19].*

3. *The restricted gangster set $J_{\mathcal{I}}$ is obtained from J_0 by removing some indices. We will see later in Remark 2.12 that $J_{\mathcal{I}}$ is in some sense the "largest effective subset" in J_0.*

2.3 SDP relaxation

We now summarize the results on our **SDP** relaxation of (1.1) without including the nonnegativity box constraints. This strengthens the relaxation in [19, 28] in the case where some of the set sizes $m_i = 1$, since we are using the larger gangster set J_0.

We use the objective function (2.4) and constraints (2.5), (2.6), (2.7), (2.8) and (2.15), and ignore the hard rank-one constraint, the nonnegativity constraint and the upper bound (by one) constraint. We obtain our **SDP** relaxation:

$$\begin{array}{rll} \operatorname{cut}(m) \geq p^*_{\mathbf{SDP}} := & \min & \frac{1}{2} \operatorname{trace} L_A Y \\ & \text{s.t.} & \operatorname{arrow}(Y) = e_0 \\ & & \operatorname{trace} D_1 Y = 0, \ \operatorname{trace} D_2 Y = 0 \\ & & \mathcal{G}_{J_0}(Y) = 0, \ Y_{00} = 1 \\ & & \mathcal{D}_t(Y) = M, \ \mathcal{D}_d(Y) = e, \ \mathcal{D}_o(Y) = \widehat{M} \\ & & Y \succeq 0. \end{array} \tag{2.16}$$

From Section 2.2.3 we have that both D_1 and D_2 are positive semidefinite. Therefore the constraints $\operatorname{trace} D_i Y = 0, i = 1, 2$, imply that the feasible set of (2.16) has no strictly feasible (positive definite) point $Y \succ 0$, i.e., the *(generalized) Slater condition*, *strict feasibility*, fails for the **SDP** relaxation (2.16). Serious numerical difficulties can arise when algorithms such as interior-point methods or alternating projection methods are applied to a problem where the *Slater condition*, fails, e.g., [9, 10]. Nonetheless, as noted in [19, 28], we can find a simple matrix in the relative interior of the feasible set and use its structure to project (and regularize) the problem into a smaller dimension. This is achieved 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, **FR***, [3, 6, 10].

Such matrices V are discussed in [19, 28]. Let $V_j \in \mathbb{R}^{j \times (j-1)}$ have full column rank with $V_j^T e = 0$. To be specific, we set

$$V_j := \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ -1 & \dots & \dots & -1 & -1 \end{bmatrix}_{j \times (j-1)} . \tag{2.17}$$

Denote

$$y = \frac{1}{n}(m \otimes e_n), \tag{2.18}$$

and let

$$\widetilde{V} := \begin{bmatrix} 1 & 0 \\ y & V_k \otimes V_n \end{bmatrix} \in \mathbb{R}^{(nk+1) \times ((k-1)(n-1)+1)}. \tag{2.19}$$

Notice that the feasible set of (2.16) must be contained in the following face

$$F = \widetilde{V} \mathbb{S}_+^{(k-1)(n-1)+1} \widetilde{V}^T. \tag{2.20}$$

We can thus *facially reduce* (2.16) using the substitution

$$Y = \widetilde{V} R \widetilde{V}^T \in \mathbb{S}_+^{nk+1}, \quad R \in \mathbb{S}_+^{(k-1)(n-1)+1}.$$

The facially reduced **SDP** is then given by

$$\begin{array}{rll} \mathrm{cut}(m) \geq p^*_{\mathbf{SDP}} & = \min & \frac{1}{2}\,\mathrm{trace}\, \widetilde{V}^T L_A \widetilde{V} R \\ & \text{s.t.} & \mathrm{arrow}(\widetilde{V} R \widetilde{V}^T) = e_0 \\ & & \mathcal{G}_{\widehat{J}_0}(\widetilde{V} R \widetilde{V}^T) = \mathcal{G}_{\widehat{J}_0}(e_0 e_0^T) \\ & & \mathcal{D}_t(\widetilde{V} R \widetilde{V}^T) = M,\, \mathcal{D}_d(\widetilde{V} R \widetilde{V}^T) = e,\, \mathcal{D}_o(\widetilde{V} R \widetilde{V}^T) = \widehat{M} \\ & & R \succeq 0, \end{array} \tag{2.21}$$

where we let $\widehat{J}_0 := J_0 \cup (00)$, J_0 is defined in (2.12).

It is not clear whether or not (2.21) satisfies a proper regularity condition. Regarding this concern, the gangster constraint in (2.21) plays a crucial role. In Section 2.3.1, we study further properties of the gangster set J_0 and the restricted gangster set $J_{\mathcal{I}}$ defined in Definition 2.4. Then in Section 2.3.2, we present our simplified facially reduced **SDP** relaxation (2.49) (which uses $J_{\mathcal{I}}$ in place of J_0) and establish some desirable regularity conditions. Specifically, we show that the *Robinson regularity*[3] holds for (2.49). This implies in particular that F in (2.20) is the smallest face of the positive semidefinite cone containing the feasible set of (2.16), and the range of $\widetilde{V}$ is indeed equal to the range of (any) $\widehat{Y} \in \mathrm{relint}\, F$.

2.3.1 Gangster sets $J_{\mathcal{I}}$ and J_0

Recall that $J_{\mathcal{I}}$ is obtained from J_0 by removing certain indices. We show here that, together with the facial structure defined by $\widetilde{V} \cdot \widetilde{V}^T$, the gangster constraint defined using $J_{\mathcal{I}}$ is as strong as that defined using J_0, and the corresponding linear map is onto.

Lemma 2.7. *Suppose $Z \in \mathbb{S}^n$. If Z is a diagonal matrix or a matrix with diagonal equal to zero, then*

$$V_n^T Z V_n = 0 \implies Z = 0,$$

where V_n is defined in (2.17).

Proof. We consider two cases.

Case 1: Let $Z = \mathrm{Diag}(a) \in \mathbb{S}^n$. Then

$$V_n^T Z V_n = \begin{bmatrix} a_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & a_{n-1} \end{bmatrix} + a_n E = 0 \implies a = 0 \implies Z = 0.$$

Case 2: Let $Z \in \mathbb{S}^n$ with $\mathrm{diag}(Z) = 0$. We can then write

$$Z = \begin{bmatrix} C & b \\ b^T & 0 \end{bmatrix}$$

for some $C \in \mathbb{S}^{n-1}$ with $\mathrm{diag}(C) = 0$ and some $b \in \mathbb{R}^{n-1}$. Then

$$V_n^T Z V_n = C - eb^T - be^T = 0 \implies b = 0,\ C = 0 \implies Z = 0.$$

[3]Strict feasibility holds and the linear constraints are onto, [23].

□

We prove in the following Proposition 2.8 the onto property of the linear map defining the restricted gangster constraints, i.e., the constraint $\mathcal{G}_{J_{\mathcal{I}}}(\widetilde{V}R\widetilde{V}^T) = 0$. A related result for the general graph partitioning problem but with another gangster set is given in [28, 29]. The basic idea is to show that the null space of its adjoint $\widetilde{V}^T\mathcal{G}^*_{J_{\mathcal{I}}}(\cdot)\widetilde{V}$ is zero.

Proposition 2.8. *For all $w \in \mathbb{R}^{|J_{\mathcal{I}}|}$, we have*

$$\widetilde{V}^T\mathcal{G}^*_{J_{\mathcal{I}}}(w)\widetilde{V} = 0 \implies w = 0,$$

where $\widetilde{V}$ is defined in (2.19) and $J_{\mathcal{I}}$ is defined in (2.14).

Proof. Let $Y = \mathcal{G}^*_{J_{\mathcal{I}}}(w) \in \mathbb{S}^{nk+1}$. Then we immediately have $\widetilde{V}^TY\widetilde{V} = 0$. On the other hand, using the definition of $\mathcal{G}^*_{J_{\mathcal{I}}}$, we see that the symmetric matrix Y can be written as

$$Y = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \overline{Y}_{(11)} & \dots & \overline{Y}_{(1k)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \overline{Y}_{(k1)} & \dots & \overline{Y}_{(kk)} \end{bmatrix},$$

where $\overline{Y}_{(ij)}$, $i, j \in \mathcal{K}$ are $n \times n$ matrices, and $\overline{Y}_{(ij)}$ is diagonal whenever $i \neq j$. Let

$$Z := (V_k \otimes V_n)^T \begin{bmatrix} \overline{Y}_{(11)} & \dots & \overline{Y}_{(1k)} \\ \vdots & \ddots & \vdots \\ \overline{Y}_{(k1)} & \dots & \overline{Y}_{(kk)} \end{bmatrix} (V_k \otimes V_n). \tag{2.22}$$

It follows from $\widetilde{V}^TY\widetilde{V} = 0$ that $Z = 0$. Note that

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

Therefore, if we write the above matrix Z in (2.22) as

$$\begin{bmatrix} Z_{(1\,1)} & \dots & Z_{(1\,k-1)} \\ \vdots & \ddots & \vdots \\ Z_{(k-1\,1)} & \dots & Z_{(k-1\,k-1)} \end{bmatrix},$$

we have

$$Z_{(ij)} = V_n^T \left(\overline{Y}_{(ij)} - \overline{Y}_{(kj)} - \overline{Y}_{(ik)} + \overline{Y}_{(kk)}\right) V_n = 0, \; \forall\, i, j \in \{1, \dots, k-1\}. \tag{2.23}$$

Furthermore, using the fact that $\overline{Y}_{(ij)}$ is diagonal whenever $i \neq j$, we have

$$Z_{(ii)} = V_n^T \left(\overline{Y}_{(ii)} - 2\overline{Y}_{(ik)} + \overline{Y}_{(kk)}\right) V_n = 0, \; \forall\, i \in \{1, \dots, k-1\}. \tag{2.24}$$

It follows from (2.23) and (2.24) that

$$V_n^T\left(2\overline{Y}_{(ij)}-\overline{Y}_{(ii)}-\overline{Y}_{(jj)}\right)V_n=0,\ \forall\, i,j\ \in\{1,\dots,k-1\}. \tag{2.25}$$

We now claim that

$$\overline{Y}_{(ii)}=0,\ \forall\, i\in\{1,\dots,k\}, \tag{2.26}$$

holds under the different choices of $\mathcal{I}$ in $J_{\mathcal{I}}$ given in (2.14).

- If $\mathcal{I}=\emptyset$, by (2.14), we have $J_{\mathcal{I}}=J_0$, i.e., (2.26) holds.
- If $k\notin\mathcal{I}\neq\emptyset$, then by (2.14), we have $J_{\mathcal{I}}=J_0\backslash J_1$, i.e., the following equalities hold:

$$\overline{Y}_{(kk)}=0 \tag{2.27}$$

$$\overline{Y}_{(ik)}=\overline{Y}_{(ki)}=0,\qquad \forall\, i\in\mathcal{I} \tag{2.28}$$

$$\overline{Y}_{(ii)}=0,\qquad \forall\, i\in\{1,\dots,k-1\}\backslash\mathcal{I}. \tag{2.29}$$

From (2.27), (2.28) and (2.24) we get $V_n^T\overline{Y}_{(ii)}V_n=0,\ \forall\, i\in\mathcal{I}$. Notice that $\overline{Y}_{(ii)}$ is a symmetric matrix with zeros on the diagonal, by Lemma 2.7, we get $\overline{Y}_{(ii)}=0,\ \forall\, i\in\mathcal{I}$. This, together with (2.27) and (2.29), yields (2.26).

- If $k\in\mathcal{I}\neq\mathcal{K}$, then $\mathcal{I}^c\neq\emptyset$. By (2.14), we have $J_{\mathcal{I}}=J_0\backslash(J_1\cup J_2)$, i.e

$$\overline{Y}_{(ii)}=0,\qquad \forall\, i\in\mathcal{I}^c \tag{2.30}$$

$$\overline{Y}_{(kj_0)}=\overline{Y}_{(j_0k)}=0,\qquad \text{for the } j_0\in\mathcal{I}^c \tag{2.31}$$

$$\overline{Y}_{(ki)}=\overline{Y}_{(ik)}=0,\qquad \forall\, i\in\mathcal{I}\backslash\{k\}. \tag{2.32}$$

It follows from (2.30), (2.31), (2.24) and Lemma 2.7 that

$$\overline{Y}_{(kk)}=0. \tag{2.33}$$

In view of (2.32), (2.33), (2.24) and Lemma 2.7, we have $\overline{Y}_{(ii)}=0,\ \forall\, i\in\mathcal{I}\backslash\{k\}$. This, together with (2.30) and (2.33), yields (2.26).

- If $\mathcal{I}=\mathcal{K}$, then by (2.14), we have $J_{\mathcal{I}}=J_0\backslash(J_1\cup J_3)$, i.e.,

$$\begin{aligned}&\overline{Y}_{(k-1,k-2)}=\overline{Y}_{(k-2,k-1)}=0,\\ &\overline{Y}_{(ki)}=\overline{Y}_{(ik)}=0,\qquad \forall\, i\in\{1,\dots,k-1\}.\end{aligned} \tag{2.34}$$

With $i=k-1, j=k-2$ in (2.23), by (2.34) and Lemma 2.7, we have $\overline{Y}_{(kk)}=0$. This together with (2.34), (2.24) and Lemma 2.7 yields (2.26).

In summary, the claim (2.26) holds. Combining (2.26) and (2.24), we get

$$V_n^T \overline{Y}_{(ki)} V_n = V_n^T \overline{Y}_{(ik)} V_n = 0 \quad \forall i \in \{1, \ldots, k-1\}. \tag{2.35}$$

In addition, it follows from (2.26) and (2.25) that

$$V_n^T \overline{Y}_{(ij)} V_n = 0 \quad \forall i, j \in \{1, \ldots, k-1\}. \tag{2.36}$$

Combining (2.35), (2.36) and (2.26), we have

$$V_n^T \overline{Y}_{(ij)} V_n = 0 \quad \forall i, j \in \{1, \ldots, k\}.$$

Since $\overline{Y}_{(ij)}$ is either a diagonal matrix or a matrix with diagonal equal to zeros, by Lemma 2.7 we have $\overline{Y}_{(ij)} = 0$, for all $i, j \in \{1, \ldots, k\}$. Therefore, $Y = 0$. Thus, it follows that $w = 0$. $\square$

We now extend the results in Proposition 2.8 to show that the operator $\mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widetilde{V} \cdot \widetilde{V}^T)$ is onto when considered as a linear transformation mapping into $\mathbb{R}^{|J_{\mathcal{I}}|+1}$, where $\widehat{J}_{\mathcal{I}} := J_{\mathcal{I}} \cup \{00\}$ with $J_{\mathcal{I}}$ defined in (2.14).

Theorem 2.9. *For all $w \in \mathbb{R}^{|J_{\mathcal{I}}|+1}$, it holds that*

$$\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) \widetilde{V} = 0 \implies w = 0,$$

where $\widetilde{V}$ is defined in (2.19) and $\widehat{J}_{\mathcal{I}} := J_{\mathcal{I}} \cup \{00\}$ with $J_{\mathcal{I}}$ defined in (2.14). This means the operator $\mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widetilde{V} \cdot \widetilde{V}^T)$ is onto when considered as a linear transformation mapping into $\mathbb{R}^{|J_{\mathcal{I}}|+1}$.

Proof. For $w \in \mathbb{R}^{|J_{\mathcal{I}}|+1}$, write $w = \begin{bmatrix} w_{00} & \breve{w}^T \end{bmatrix}^T$, where $\breve{w} \in \mathbb{R}^{|J_{\mathcal{I}}|}$. Then we have

$$\mathcal{G}^*_{J_{\mathcal{I}}}(\breve{w}) = \begin{bmatrix} 0 & 0 \\ 0 & \overline{W} \end{bmatrix} \quad \text{and} \quad \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) = \begin{bmatrix} w_{00} & 0 \\ 0 & \overline{W} \end{bmatrix}$$

for some $\overline{W} \in \mathbb{S}^{nk}$. A direct computation using the definition of $\widetilde{V}$ yields

$$\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) \widetilde{V} = \begin{bmatrix} w_{00} + y^T \overline{W} y & y^T \overline{W}(V_k \otimes V_n) \\ (V_k^T \otimes V_n^T)\overline{W} y & (V_k^T \otimes V_n^T)\overline{W}(V_k \otimes V_n) \end{bmatrix}, \tag{2.37}$$

$$\widetilde{V}^T \mathcal{G}^*_{J_{\mathcal{I}}}(\breve{w}) \widetilde{V} = \begin{bmatrix} y^T \overline{W} y & y^T \overline{W}(V_k \otimes V_n) \\ (V_k^T \otimes V_n^T)\overline{W} y & (V_k^T \otimes V_n^T)\overline{W}(V_k \otimes V_n) \end{bmatrix}. \tag{2.38}$$

Now, assume that $\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) \widetilde{V} = 0$. Then we see from (2.37) that $(V_k^T \otimes V_n^T)\overline{W}(V_k \otimes V_n) = 0$. Following the same argument as in the proof of Proposition 2.8 (start from (2.22) and use $\overline{W}$ in place of $\overline{Y}$ there), we conclude that $\overline{W} = 0$. Combining this with (2.37) and the assumption $\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) \widetilde{V} = 0$ gives

$$\begin{bmatrix} w_{00} & 0 \\ 0 & 0 \end{bmatrix} = \widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) \widetilde{V} = 0,$$

showing that $w_{00} = 0$. On the other hand, we can deduce from (2.38) and the fact $\overline{W} = 0$ that

$$\widetilde{V}^T \mathcal{G}^*_{J_{\mathcal{I}}}(\breve{w}) \widetilde{V} = 0.$$

This implies $\breve{w} = 0$, according to Proposition 2.8. Consequently, $w = \begin{bmatrix} w_{00} & \breve{w}^T \end{bmatrix}^T = 0$. This completes the proof. $\square$

We next show in Theorem 2.11 below that the nullspaces of $\mathcal{G}_{J_\mathcal{I}}(\widetilde{V}\cdot\widetilde{V}^T)$ and $\mathcal{G}_{J_0}(\widetilde{V}\cdot\widetilde{V}^T)$ are the same. Since the restricted gangster set $J_\mathcal{I}$ is obtained by removing indices in J_0 and the linear map $\mathcal{G}_{J_\mathcal{I}}(\widetilde{V}\cdot\widetilde{V}^T)$ is onto according to Proposition 2.8, this suggests that we have removed *just the right number of indices* from J_0. Before presenting Theorem 2.11, we first recall the following result from [28, Lemma 4.1] that is used in our analysis below.

Lemma 2.10 ([28, Lemma 4.1])**.** *Let $R \in \mathbb{S}^{(n-1)(k-1)+1}$ be given, $\widetilde{V}$ be as in (2.19), and let*

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

Then the block notation of (2.3) yields

$$m_i Y_{(j0)}^T = e^T \overline{Y}_{(ij)}, \quad \forall\, i,j \in \{1,\ldots,k\}, \tag{2.39}$$

and

$$\sum_{i=1}^{k} \operatorname{diag}(\overline{Y}_{(ij)}) = Y_{(j0)}, \quad \forall\, j \in \{1,\ldots,k\}. \tag{2.40}$$

□

Theorem 2.11. *Let $Y = \widetilde{V} R \widetilde{V}^T$ for some $R \in \mathbb{S}^{(n-1)(k-1)+1}$ with $\widetilde{V}$ defined in (2.19). Then*

$$\mathcal{G}_{J_\mathcal{I}}(Y) = 0 \Longleftrightarrow \mathcal{G}_{J_0}(Y) = 0, \tag{2.41}$$

where J_0 is defined in (2.12) and $J_\mathcal{I}$ is defined in (2.14).

Proof. The alleged equivalence (2.41) is trivially true if $\mathcal{I} = \emptyset$, because $J_\mathcal{I} = J_0$ in this case. Thus, we assume $\mathcal{I} \neq \emptyset$ from now on.

Since $J_\mathcal{I} \subseteq J_0$, we trivially have $\mathcal{G}_{J_0}(Y) = 0 \implies \mathcal{G}_{J_\mathcal{I}}(Y) = 0$. Hence, to establish (2.41), it remains to prove the converse implication, i.e., to show that

$$\mathcal{G}_{J_\mathcal{I}}(Y) = 0 \Longrightarrow \mathcal{G}_{J_0}(Y) = 0 \tag{2.42}$$

In view of the definition of $J_\mathcal{I}$, to prove (2.42), it amounts to proving the following three implications:

$$\begin{cases} \mathcal{G}_{J_0\backslash J_1}(Y) = 0 \implies \mathcal{G}_{J_1}(Y) = 0 & \text{if } k \notin \mathcal{I} \neq \emptyset; \\ \mathcal{G}_{J_0\backslash (J_1\cup J_2)}(Y) = 0 \implies \mathcal{G}_{J_1}(Y) = 0, \mathcal{G}_{J_2}(Y) = 0 & \text{if } k \in \mathcal{I} \neq \mathcal{K}; \\ \mathcal{G}_{J_0\backslash (J_1\cup J_3)}(Y) = 0 \implies \mathcal{G}_{J_1}(Y) = 0, \mathcal{G}_{J_3}(Y) = 0 & \text{if } \mathcal{I} = \mathcal{K}. \end{cases} \tag{2.43}$$

To prove these implications, we write Y in the block matrix form (2.3). Since $m_i = 1$, $\forall\, i \in \mathcal{I}$, from (2.39), we obtain $Y_{(i0)}^T = e^T \overline{Y}_{(ii)}$, $\forall\, i \in \mathcal{I}$. This, together with $\mathcal{G}_{J_0\backslash(J_1\cup J_2\cup J_3)}(Y) = 0$, yields that

$$Y_{(i0)} = \operatorname{diag}(\overline{Y}_{(ii)}), \quad \forall\, i \in \mathcal{I}. \tag{2.44}$$

We can now prove the first assertion in (2.43). Using (2.40) and $\mathcal{G}_{J_0\backslash J_1}(Y) = 0$, we have

$$Y_{(j0)} = \operatorname{diag}(\overline{Y}_{(jj)}) + \operatorname{diag}(\overline{Y}_{(kj)}), \quad \forall\, j \in \mathcal{I}\backslash\{k\}.$$

Combining this with (2.44) and the symmetry of Y, we see that

$$\operatorname{diag}(\overline{Y}_{(jk)}) = \operatorname{diag}(\overline{Y}_{(kj)}) = 0, \quad \forall\, j \in \mathcal{I}\backslash\{k\}, \tag{2.45}$$

i.e., $\mathcal{G}_{J_1}(Y) = 0$.

Next, we prove the second assertion in (2.43). The reasoning for $\mathcal{G}_{J_1}(Y) = 0$ is the same as in the previous case. In addition, from $\mathcal{G}_{J_0\backslash(J_1\cup J_2)}(Y) = 0$, (2.45) and (2.40), we have

$$Y_{(k0)} = \operatorname{diag}(\overline{Y}_{(j_0\, k)}) + \operatorname{diag}(\overline{Y}_{(kk)}).$$

Since $k \in \mathcal{I}$, from (2.44), we have

$$Y_{(k0)} = \operatorname{diag}(\overline{Y}_{(kk)}).$$

In view of the above two equations and the symmetry of Y, we obtain

$$\operatorname{diag}(\overline{Y}_{(k\, j_0)}) = \operatorname{diag}(\overline{Y}_{(j_0\, k)}) = 0,$$

i.e., $\mathcal{G}_{J_2}(Y) = 0$.

Finally, we prove the third assertion in (2.43). It follows from (2.40) and $\mathcal{G}_{J_0\backslash(J_1\cup J_3)}(Y) = 0$ that

$$Y_{(j0)} = \operatorname{diag}(\overline{Y}_{(jj)}) + \operatorname{diag}(\overline{Y}_{(kj)}), \quad \forall\, j \in \mathcal{I}\backslash\{k-2, k-1, k\}.$$

Together with (2.44) and the symmetry of Y, we have

$$\operatorname{diag}(\overline{Y}_{(jk)}) = \operatorname{diag}(\overline{Y}_{(kj)}) = 0, \quad \forall\, j \in \mathcal{I}\backslash\{k-2, k-1, k\}. \tag{2.46}$$

Combining this with (2.40) and $\mathcal{G}_{J_0\backslash(J_1\cup J_3)}(Y) = 0$ gives

$$\begin{cases} \operatorname{diag}(\overline{Y}_{(k-2\, k-2)}) + \operatorname{diag}(\overline{Y}_{(k-1\, k-2)}) + \operatorname{diag}(\overline{Y}_{(k\, k-2)}) = Y_{(k-2\, 0)} \\ \operatorname{diag}(\overline{Y}_{(k-2\, k-1)}) + \operatorname{diag}(\overline{Y}_{(k-1\, k-1)}) + \operatorname{diag}(\overline{Y}_{(k\, k-1)}) = Y_{(k-1\, 0)} \\ \operatorname{diag}(\overline{Y}_{(k-2\, k)}) + \operatorname{diag}(\overline{Y}_{(k-1\, k)}) + \operatorname{diag}(\overline{Y}_{(k\, k)}) = Y_{(k\, 0)} \end{cases}$$

Using this together with (2.44) and the symmetry of Y, we obtain

$$\begin{cases} \operatorname{diag}(\overline{Y}_{(k-2\, k-1)}) + \operatorname{diag}(\overline{Y}_{(k-2\, k)}) = 0 \\ \operatorname{diag}(\overline{Y}_{(k-2\, k-1)}) + \operatorname{diag}(\overline{Y}_{(k-1\, k)}) = 0 \\ \operatorname{diag}(\overline{Y}_{(k-2\, k)}) + \operatorname{diag}(\overline{Y}_{(k-1\, k)}) = 0 \end{cases}$$

Therefore, we have

$$\operatorname{diag}(\overline{Y}_{(k-2\, k)}) = \operatorname{diag}(\overline{Y}_{(k-1\, k)}) = \operatorname{diag}(\overline{Y}_{(k-2\, k-1)}) = \operatorname{diag}(\overline{Y}_{(k-1\, k-2)}) = 0,$$

which together with (2.46) yields that $\mathcal{G}_{J_1}(Y) = 0$ and $\mathcal{G}_{J_3}(Y) = 0$. □

Remark 2.12. *Combining Theorem 2.11 with Proposition 2.8, we see that the linear map $\mathcal{G}_{J_0}(\widetilde{V}\cdot\widetilde{V}^T)$ is* not *onto but $\mathcal{G}_{J_{\mathcal{I}}}(\widetilde{V}\cdot\widetilde{V}^T)$ is, and the two linear maps have the same nullspace. Thus, in some sense, the restricted gangster set $J_{\mathcal{I}}$ is the "largest effective subset" of J_0: no redundant indices in $J_{\mathcal{I}}$.*

2.3.2 Facially reduced SDP relaxation

We are now ready to present our facially reduced SDP relaxation. In Theorem 2.13 below, we show that the facial reduction in combination with the restricted gangster constraints essentially makes the rest of the constraints in (2.21) redundant, and that the Robinson regularity holds.

Similar to [28, Theorem 4.1], to study primal strict feasibility, we make use of the barycenter of the rank-1 matrices of the lifting (see [28, Equation (3.3)]), defined as

$$\widehat{Y} := \frac{m_1! \dots m_k!}{n!} \sum_{\mathrm{Mat}(x) \in \mathcal{M}_m} \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix}.$$

Recall from [28, Theorem 3.1] that the above barycenter can be written as

$$\widehat{Y} = \begin{bmatrix} 1 & \frac{m_1}{n} e_n^T & \dots & \frac{m_k}{n} e_n^T \\ \frac{m_1}{n} e_n & \left(\frac{m_1}{n} I_n + \frac{m_1(m_1-1)}{n(n-1)} (E_n - I_n) \right) & \dots & \left(\frac{m_1 m_k}{n(n-1)} \right) (E_n - I_n) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{m_k}{n} e_n & \left(\frac{m_1 m_k}{n(n-1)} \right) (E_n - I_n) & \dots & \left(\frac{m_k}{n} I_n + \frac{m_k(m_k-1)}{n(n-1)} (E_n - I_n) \right) \end{bmatrix}. \tag{2.47}$$

On the other hand, to analyze dual strict feasibility, we define the following matrices

$$\widetilde{W} := \beta \begin{bmatrix} \alpha & 0 \\ 0 & 2Q_{\mathcal{I}} \end{bmatrix} \quad \text{and} \quad Q_{\mathcal{I}} := T_{\mathcal{I}} \otimes I_n + S_{\mathcal{I}} \otimes (E_n - I_n), \tag{2.48}$$

with $\alpha < 0 < \beta$ and

$$(T_{\mathcal{I}}, S_{\mathcal{I}}) = \begin{cases} (E_k - I_k, 0) & \text{if } \mathcal{I} = \emptyset, \\ (E_k - I_k - \widehat{M}_{\mathrm{one}}, e^T m_{\mathrm{one}} M_{\mathrm{one}}) & \text{if } k \notin \mathcal{I} \neq \emptyset, \\ (E_k - I_k - \widehat{E}, M_{\mathrm{one}}) & \text{if } k \in \mathcal{I} \neq \mathcal{K}, \\ (0, I_k) & \text{if } \mathcal{I} = \mathcal{K}, \end{cases}$$

where m_{one}, $\mathcal{I}$ and $\mathcal{K}$ are defined in Definition 2.4, $\widehat{E} = \begin{bmatrix} 0 & e_{k-1} \\ e_{k-1}^T & 0 \end{bmatrix} \in \mathbb{S}^k$, $M_{one} = \mathrm{Diag}(m_{\mathrm{one}})$, and $\widehat{M}_{one} = \begin{bmatrix} 0 & \hat{m}_{\mathrm{one}} \\ \hat{m}_{\mathrm{one}}^T & 0 \end{bmatrix} \in \mathbb{S}^k$ with $\hat{m}_{one} \in \mathbb{R}^{k-1}$ being the vector that contains the first $k-1$ entries of m_{one}.

Theorem 2.13. *The following holds:*

1. *The facially reduced* ***SDP*** (2.21) *is equivalent to the single equality constrained problem*

$$\begin{array}{rl} \mathrm{cut}(m) \geq p^*_{\boldsymbol{SDP}} = & \min \ \frac{1}{2} \operatorname{trace} \left(\widetilde{V}^T L_A \widetilde{V} \right) R \\ & \text{s.t. } \ \mathcal{G}_{\hat{J}_{\mathcal{I}}}(\widetilde{V} R \widetilde{V}^T) = \mathcal{G}_{\hat{J}_{\mathcal{I}}}(e_0 e_0^T) \\ & \quad\ \ R \succeq 0. \end{array} \tag{2.49}$$

2. *The primal model* (2.49) *satisfies strict feasibility, with (generalized) Slater point*

$$\widetilde{R} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)} (n \,\mathrm{Diag}(\hat{m}_{k-1}) - \hat{m}_{k-1} \hat{m}_{k-1}^T) \otimes (n I_{n-1} - E_{n-1}) \end{array} \right] \in \mathbb{S}^{(k-1)(n-1)+1}_{++}, \tag{2.50}$$

where $\hat{m}_{k-1} = (m_1, \ldots, m_{k-1})^T \in \mathbb{Z}_+^{k-1}$. Moreover, it holds that $\widetilde{V}\widetilde{R}\widetilde{V}^T = \widehat{Y}$, where $\widehat{Y}$ is given in (2.47). Furthermore, the Robinson regularity holds for (2.49).

3. *The dual problem of (2.49) is*

$$\begin{aligned} &\max\ \tfrac{1}{2} w_{00} \\ &\text{s.t. } \widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w)\widetilde{V} \preceq \widetilde{V}^T L_A \widetilde{V}. \end{aligned} \tag{2.51}$$

Moreover, with $\widetilde{W}$ defined as in (2.48), the point $\tilde{w}_{\mathcal{I}} := \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widetilde{W})$ is strictly feasible for (2.51) for all sufficiently positive β and sufficiently negative α.

Proof. Item 1: It suffices to show that any R feasible for (2.49) is also feasible for (2.16). To this end, let R be feasible for (2.49) and let $Y := \widetilde{V}R\widetilde{V}^T$. Therefore, it holds that $\mathcal{G}_{J_{\mathcal{I}}}(Y) = 0$, where $J_{\mathcal{I}}$ is defined in (2.14). According to Theorem 2.11, we have $\mathcal{G}_{J_0}(Y) = 0$, where J_0 is defined in (2.12). Hence, all the diagonal elements of off-diagonal blocks of $\overline{Y}$ (see the block structure in (2.3)) are zero. This together with $Y_{00} = 1$ and $R \succeq 0$ shows that $Y = \widetilde{V}R\widetilde{V}^T$ satisfies all the constraints in (2.21) except for

$$\mathcal{D}_o(Y) = \widehat{M}, \tag{2.52}$$

as shown in [19, Theorem 5.1]. Therefore, it remains to show that (2.52) is also redundant in the facially reduced **SDP** (2.21), i.e., to show that Y satisfies (2.52).

Let D_2 be as defined in (2.7). Since $R \succeq 0$ and $\mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widetilde{V}R\widetilde{V}^T) = \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(e_0 e_0^T)$, we have $Y \succeq 0$, $Y_{00} = 1$, and $\operatorname{trace} D_2 Y = 0$. Let $v_1 := Y_{0:kn\,0}$. Then we have

$$\begin{aligned} Y - v_1 v_1^T &= \begin{bmatrix} 1 & Y_{1:nk\,0}^T \\ Y_{1:nk\,0} & \overline{Y} \end{bmatrix} - \begin{bmatrix} 1 \\ Y_{1:nk\,0} \end{bmatrix} \begin{bmatrix} 1 \\ Y_{1:nk\,0} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \overline{Y} - Y_{1:nk\,0} Y_{1:nk\,0}^T \end{bmatrix}. \end{aligned} \tag{2.53}$$

Note that $\overline{Y} - Y_{1:nk\,0}Y_{1:nk\,0}^T$ is the Schur complement of Y_{00} in Y and $Y \succeq 0$. Hence, it holds that $\overline{Y} - Y_{1:nk\,0}Y_{1:nk\,0}^T \succeq 0$. Consequently, we deduce from (2.53) that $Y \succeq v_1 v_1^T$.

Let $X = \operatorname{Mat}(Y_{1:kn\,0})$. Since

$$\operatorname{trace} D_2 Y = 0, \quad D_2 = \begin{bmatrix} m^T m & -m^T \otimes e_n^T \\ -m \otimes e_n & I_k \otimes (e_n e_n^T) \end{bmatrix} = \begin{bmatrix} -m^T \\ I_k \otimes e_n \end{bmatrix} \begin{bmatrix} -m^T \\ I_k \otimes e_n \end{bmatrix}^T \succeq 0, \quad \text{and } Y \succeq v_1 v_1^T,$$

we see that

$$0 = \operatorname{trace}(D_2 Y) \geq \operatorname{trace}(D_2 v_1 v_1^T) = \|X^T e - m\|^2 \quad \text{and} \quad Y \begin{bmatrix} -m^T \\ I_k \otimes e_n \end{bmatrix} = 0. \tag{2.54}$$

Using the second relation in (2.54) together with the block partition of Y in (2.3), we have

$$-Y_{1:nk\,0} m^T + \overline{Y}(I_k \otimes e_n) = 0.$$

Multiplying the above relation on the left by $I_k \otimes e_n^T$, we obtain further that

$$-(I_k \otimes e_n^T) Y_{1:nk\,0} m^T + (I_k \otimes e_n^T)\overline{Y}(I_k \otimes e_n) = 0. \tag{2.55}$$

Next, recall from the first relation in (2.54) that $(I_k \otimes e_n^T)Y_{1:nk\,0} = X^T e_n = m$. Moreover, a direct computation shows that $(I_k \otimes e_n^T)\overline{Y}(I_k \otimes e_n) = \left(e_n^T \overline{Y}_{(ij)} e_n\right)$. Combining these with (2.55) yields

$$\left(e_n^T \overline{Y}_{(ij)} e_n\right) = mm^T.$$

Finally, recall that $\mathcal{D}_t(Y) = \mathcal{D}_t(\widetilde{V} R \widetilde{V}^T) = M$ in (2.21) can be inferred from the constraints in (2.49), thanks to Theorem 2.11 and [19, Theorem 5.1]. Therefore, it holds that

$$\mathcal{D}_o(Y) = \left(\sum_{s \neq t} \left(\overline{Y}_{(ij)}\right)_{st}\right) = \left(e_n^T \overline{Y}_{(ij)} e_n\right) - \mathcal{D}_t(Y) = mm^T - M = \widehat{M}.$$

Item 2: Recall from [28, Theorem 4.1] that $\widetilde{R} \succ 0$. Moreover, in the proof of [28, Theorem 4.1], it is shown that $\widetilde{V}\widetilde{R}\widetilde{V}^T = \widehat{Y}$. Furthermore, following the block structure of $\overline{Y}$ described in (2.3), the barycenter $\widehat{Y}$ in (2.47) is zero along the diagonal of each off-diagonal blocks as well as at all off-diagonal positions of the ith diagonal block if $m_i = 1$. Thus, it holds that $\mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widehat{Y}) = \mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(e_0 e_0^T)$. This together with $\widetilde{R} \succ 0$ and $\widetilde{V}\widetilde{R}\widetilde{V}^T = \widehat{Y}$ proves the strict feasibility of $\widetilde{R}$ for (2.49). The Robinson regularity holds in view of the strict feasibility of $\widetilde{R}$ and Theorem 2.9.

Item 3: It is standard to show that the dual problem of (2.49) is given by (2.51). We now prove the claim concerning strict feasibility.

With the y in (2.18), the $\widetilde{V}$ in (2.19), the definitions of $\widetilde{W}$ and $\widetilde{w}_{\mathcal{I}}$, and the definition of $J_{\mathcal{I}}$ in Definition 2.4, we can compute that

$$\begin{aligned}\widetilde{V}^T \mathcal{G}^*_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}})\widetilde{V} &= \beta \begin{bmatrix} 1 & y^T \\ 0 & V_k^T \otimes V_n^T \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & Q_{\mathcal{I}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ y & V_k \otimes V_n \end{bmatrix} \\ &= \beta \begin{bmatrix} \alpha + y^T Q_{\mathcal{I}} y & y^T Q_{\mathcal{I}}(V_k \otimes V_n) \\ (V_k^T \otimes V_n^T) Q_{\mathcal{I}} y & (V_k^T \otimes V_n^T) Q_{\mathcal{I}} (V_k \otimes V_n) \end{bmatrix}.\end{aligned} \tag{2.56}$$

Now, recall the following relations, which are immediate consequences of the definition of V_j:

$$V_j^T = \begin{bmatrix} I_{j-1} & -e_{j-1} \end{bmatrix}, \quad V_j^T E_j = V_j^T e_j e_j^T = 0, \quad \text{and} \quad V_j^T V_j = E_{j-1} + I_{j-1}.$$

Then we have

$$\begin{aligned}(V_k^T \otimes V_n^T) Q_{\mathcal{I}} y &= (V_k^T \otimes V_n^T)(T_{\mathcal{I}} \otimes I_n + S_{\mathcal{I}} \otimes (E_n - I_n))y \\ &= (V_k^T T_{\mathcal{I}} \otimes V_n^T + V_k^T S_{\mathcal{I}} \otimes V_n^T (E_n - I_n))y \\ &= (V_k^T T_{\mathcal{I}} \otimes V_n^T - V_k^T S_{\mathcal{I}} \otimes V_n^T)y \\ &= \frac{1}{n}(V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) \otimes V_n^T)(m \otimes e_n) \\ &= \frac{1}{n}(V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}})m) \otimes V_n^T e_n = 0\end{aligned}$$

and

$$\begin{aligned}(V_k^T \otimes V_n^T) Q_{\mathcal{I}} (V_k \otimes V_n) &= (V_k^T \otimes V_n^T)(T_{\mathcal{I}} \otimes I_n + S_{\mathcal{I}} \otimes (E_n - I_n))(V_k \otimes V_n) \\ &= V_k^T T_{\mathcal{I}} V_k \otimes V_n^T V_n + V_k^T S_{\mathcal{I}} V_k \otimes V_n^T (E_n - I_n) V_n\end{aligned}$$

$$
\begin{aligned}
&= V_k^T T_{\mathcal{I}} V_k \otimes V_n^T V_n - V_k^T S_{\mathcal{I}} V_k \otimes V_n^T V_n \\
&= V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k \otimes V_n^T V_n \\
&= V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k \otimes (I_{n-1} + E_{n-1}).
\end{aligned}
$$

Combining the above two displays with (2.56), we obtain

$$
\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}) \widetilde{V} = \beta \begin{bmatrix} \alpha + y^T Q_{\mathcal{I}} y & 0 \\ 0 & V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k \otimes (I_{n-1} + E_{n-1}) \end{bmatrix}. \tag{2.57}
$$

We next show that $V_k(T_{\mathcal{I}} - S_{\mathcal{I}})V_k \prec 0$ in each of the four cases in the definition of $J_{\mathcal{I}}$.

- If $\mathcal{I} = \emptyset$, then $V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k = V_k^T (E_k - I_k) V_k = -V_k^T V_k = -(I_{k-1} + E_{k-1}) \prec 0$,
- If $k \notin \mathcal{I} \neq \emptyset$, then we have

$$
\begin{aligned}
V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k &= V_k^T (E_k - I_k - \widehat{M}_{\text{one}} - e^T m_{\text{one}} M_{\text{one}}) V_k \\
&= -I_{k-1} - E_{k-1} - V_k^T (\widehat{M}_{\text{one}} + e^T m_{\text{one}} M_{\text{one}}) V_k \\
&\preceq -I_{k-1} - E_{k-1} - V_k^T (\widehat{M}_{\text{one}} + m_{\text{one}} m_{\text{one}}^T) V_k \\
&= -I_{k-1} - E_{k-1} - V_k^T \left(\begin{bmatrix} 0 & \hat{m}_{\text{one}} \\ \hat{m}_{\text{one}}^T & 0 \end{bmatrix} + \begin{bmatrix} \hat{m}_{\text{one}} \\ 0 \end{bmatrix} \begin{bmatrix} \hat{m}_{\text{one}}^T & 0 \end{bmatrix} \right) V_k \\
&= -I_{k-1} - E_{k-1} - \begin{bmatrix} I_{k-1} & -e_{k-1} \end{bmatrix} \begin{bmatrix} \hat{m}_{\text{one}} \hat{m}_{\text{one}}^T & \hat{m}_{\text{one}} \\ \hat{m}_{\text{one}}^T & 0 \end{bmatrix} \begin{bmatrix} I_{k-1} \\ -e_{k-1}^T \end{bmatrix} \\
&= -I_{k-1} - E_{k-1} - (\hat{m}_{\text{one}} \hat{m}_{\text{one}}^T - e_{k-1} \hat{m}_{\text{one}}^T - \hat{m}_{\text{one}} e_{k-1}^T) \\
&= -I_{k-1} - (e_{k-1} - \hat{m}_{\text{one}})(e_{k-1} - \hat{m}_{\text{one}})^T \\
&\preceq -I_{k-1} \prec 0,
\end{aligned}
$$

  where the first "$\preceq$" follows from the observation that $e^T m_{\text{one}} M_{\text{one}} \succeq m_{\text{one}} m_{\text{one}}^T$.
- If $k \in \mathcal{I} \neq \emptyset$, then we have

$$
\begin{aligned}
V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k &= V_k^T (E_k - I_k - \widehat{E} - M_{\text{one}}) V_k \\
&= -I_{k-1} - E_{k-1} - V_k^T (\widehat{E} + M_{\text{one}}) V_k \\
&= -I_{k-1} - E_{k-1} - \begin{bmatrix} I_{k-1} & -e_{k-1} \end{bmatrix} \begin{bmatrix} \operatorname{Diag}(\hat{m}_{\text{one}}) & e \\ e^T & 1 \end{bmatrix} \begin{bmatrix} I_{k-1} \\ -e^T \end{bmatrix} \\
&= -I_{k-1} - E_{k-1} - (\operatorname{Diag}(\hat{m}_{\text{one}}) - E_{k-1}) \\
&= -I_{k-1} - \operatorname{Diag}(\hat{m}_{\text{one}}) \prec 0
\end{aligned}
$$

- If $\mathcal{I} = \mathcal{K}$, then we have $V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k = V_k^T (-I_k) V_k = -(E_{k-1} + I_{k-1}) \prec 0$.

In summary, we have $V_k(T_{\mathcal{I}} - S_{\mathcal{I}})V_k \prec 0$, which together with $I_{n-1} + E_{n-1} \succ 0$ yields that $V_k^T (T_{\mathcal{I}} - S_{\mathcal{I}}) V_k \otimes (I_{n-1} + E_{n-1}) \prec 0$ in (2.57). Therefore, with $\alpha \ll 0 \ll \beta$, we have $\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}) \widetilde{V} \preceq \widetilde{V}^T L_A \widetilde{V}$, i.e., $\widetilde{w}_{\mathcal{I}}$ is strictly feasible for (2.51). □

We emphasize that (2.49) is a **SDP** relaxation of model (1.1). It uses facial reduction to guarantee strict feasibility for both the relaxation and its dual. The Robinson regularity condition holds and thus we obtain robustness. In addition, facial reduction greatly simplifies the constraints by making many of them redundant.

2.4 DNN relaxation

For our **DNN** relaxation and algorithm in Section 3, below, we need the following orthogonal matrix, $\widehat{V}$.

Assumption 2.14. *Without loss of generality, by using a QR or SVD factorization on $\widetilde{V}$ in (2.19), or some other special construction if needed, we assume that the columns of $\widehat{V}$ form an orthonormal basis for the range of $\widetilde{V}$. One such choice of $\widehat{V}$ is*

$$\widehat{V} = \begin{bmatrix} s & 0 \\ sy & \widehat{V}_k \otimes \widehat{V}_n \end{bmatrix}, \tag{2.58}$$

where $s := \sqrt{\frac{n}{n+\|m\|^2}}$ with $\|m\|$ denoting the ℓ_2 norm of m; and $\widehat{V}_j$ is a matrix with orthonormal columns that satisfies $\widehat{V}_j^T e_j = 0$.

Since the range of $\widehat{V}$ is the same as the range of $\widetilde{V}$, we obtain the same minimal face

$$\widehat{V}\mathbb{S}_+^{(k-1)(n-1)+1}\widehat{V}^T = \widetilde{V}\mathbb{S}_+^{(k-1)(n-1)+1}\widetilde{V}^T.$$

Using $\widehat{V}$ in place of $\widetilde{V}$, the simplified facially reduced **SDP** (2.49) can be equivalently written as

$$\begin{array}{rll} \operatorname{cut}(m) \geq p^*_{\mathbf{SDP}} & = \min & \frac{1}{2}\operatorname{trace}\left(\widehat{V}^T L_A \widehat{V}\right) R \\ & \text{s.t.} & \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widehat{V}R\widehat{V}^T) = \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(e_0 e_0^T) \\ & & R \succeq 0. \end{array} \tag{2.59}$$

The dual problem of (2.59) is

$$\begin{array}{rl} \max & \frac{1}{2} w_{00} \\ \text{s.t.} & \widehat{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w)\widehat{V} \preceq \widehat{V}^T L_A \widehat{V}. \end{array} \tag{2.60}$$

The **SDP** relaxation (2.59) can be further strengthened by adding additional constraints. With the additional nonnegativity box constraint $0 \leq (\widehat{V}R\widehat{V}^T)_{\widehat{J}_0^c} \leq 1$, where $\widehat{J}_0^c$ is the complement of $\widehat{J}_0$, we obtain the following doubly nonnegative, **DNN**, relaxation,

$$\begin{array}{rll} \operatorname{cut}(m) \geq p^*_{\mathbf{DNN}} & = \min & \frac{1}{2}\operatorname{trace}\left(\widehat{V}^T L_A \widehat{V}\right) R \\ & \text{s.t.} & \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(\widehat{V}R\widehat{V}^T) = \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(e_0 e_0^T) \\ & & R \succeq 0 \\ & & 0 \leq \left(\widehat{V}R\widehat{V}^T\right)_{\widehat{J}_0^c} \leq 1. \end{array} \tag{2.61}$$

Note that the term **DNN** refers to the two nonnegative cones in the constraints of (2.61), i.e., the positive semidefinite cone and the nonnegative cone.

The following Theorem 2.15 shows that the Slater point $\widetilde{w}_{\mathcal{I}}$ for (2.51) found in Theorem 2.13 is still strictly feasible for (2.60). Moreover, starting from the generalized Slater point $\widetilde{R}$ in (2.50) for (2.49), one can construct a generalized Slater point for both (2.59) and (2.61): the fact that (2.61) has a generalized Slater point will be important for our algorithmic development later.

Theorem 2.15. *The strictly feasible point $\widetilde{w}_{\mathcal{I}}$ for (2.51) found in Theorem 2.13 is strictly feasible for (2.60). Moreover, define*

$$\widehat{R} := \widehat{V}^{\dagger}\widetilde{V}\widetilde{R}\widetilde{V}^T(\widehat{V}^{\dagger})^T, \tag{2.62}$$

where $\widetilde{R}$ is defined in (2.50), $\widehat{V}^{\dagger}$ is the pseudoinverse of $\widehat{V}$, and $\widetilde{V}$ and $\widehat{V}$ are given in (2.19) and (2.58), respectively. Then it holds that $\widehat{R}$ is strictly feasible for both (2.59) and (2.61), and $\widehat{V}\widehat{R}\widehat{V}^T = \widehat{Y}$, where $\widehat{Y}$ is defined in (2.47).

Proof. 1. Note that $\mathrm{Range}(\widehat{V}) = \mathrm{Range}(\widetilde{V})$ by construction. This implies that $\widehat{V}\widehat{V}^{\dagger}\widetilde{V} = \widetilde{V}$. Thus, we have

$$\widetilde{V}^T(\widehat{V}^T)^{\dagger}\widehat{V}^T(L_A - \mathcal{G}^*_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}))\widehat{V}\widehat{V}^{\dagger}\widetilde{V} = \widetilde{V}^T(L_A - \mathcal{G}^*_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}))\widetilde{V} \succ 0,$$

where the positive definiteness follows from the fact that $\widetilde{w}_{\mathcal{I}}$ is strictly feasible for (2.51). Since $(\widehat{V}^{\dagger}\widetilde{V})^T = \widetilde{V}^T(\widehat{V}^T)^{\dagger}$ is a square matrix, we conclude from the above display that the matrix $\widetilde{V}^T(\widehat{V}^T)^{\dagger}$ is nonsingular. Thus, we deduce further that

$$\widehat{V}^T(L_A - \mathcal{G}^*_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}))\widehat{V} = [\widetilde{V}^T(\widehat{V}^T)^{\dagger}]^{-1}\widetilde{V}^T(L_A - \mathcal{G}^*_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widetilde{w}_{\mathcal{I}}))\widetilde{V}[\widehat{V}^{\dagger}\widetilde{V}]^{-1} \succ 0,$$

i.e., $\widetilde{w}_{\mathcal{I}}$ is strictly feasible for (2.60).

2. The positive definiteness of $\widehat{R}$ follows immediately from the fact that $\widetilde{R} \succ 0$ (see Theorem 2.13 Item 2) and the nonsingularity of $\widetilde{V}^T(\widehat{V}^T)^{\dagger}$ just established. In addition, since $\mathrm{Range}(\widehat{V}) = \mathrm{Range}(\widetilde{V})$, we have $\widehat{V}\widehat{V}^{\dagger}\widetilde{V} = \widetilde{V}$. Using this and the definition of $\widehat{R}$, we see further that

$$\widehat{V}\widehat{R}\widehat{V}^T = \widehat{V}\widehat{V}^{\dagger}\widetilde{V}\widetilde{R}\widetilde{V}^T(\widehat{V}^{\dagger})^T\widehat{V}^T = \widetilde{V}\widetilde{R}\widetilde{V}^T = \widehat{Y},$$

where the last equality follows from Theorem 2.13 Item 2. Then we obtain immediately that $\mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widehat{V}\widehat{R}\widehat{V}^T) = \mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(\widehat{Y}) = 0$. Consequently, $\widehat{R}$ is strictly feasible for (2.59).

Finally, notice that entries of $\widehat{Y}$ in $\widehat{\mathcal{J}}_0^c$ are strictly positive and strictly less than 1. Hence, we also have $0 < \left(\widehat{V}\widehat{R}\widehat{V}^T\right)_{\widehat{\mathcal{J}}_0^c} < 1$. Thus, we have shown that $\widehat{R}$ is strictly feasible for (2.61) and $\widehat{V}\widehat{R}\widehat{V}^T = \widehat{Y}$.

□

The **DNN** problem (2.61) is extremely difficult for interior point methods, especially when the dimension is large. Motivated by the recent success in the application of splitting methods to quadratic assignment problems in [18], we adopt a similar approach here. We first introduce a new variable and add the constraint $Y = \widehat{V}R\widehat{V}^T$ to (2.61). By doing so, we essentially double the number of variables and transform the original problem (2.61) to the following equivalent model,

$$\begin{array}{rll} p^*_{\mathbf{DNN}} = & \min & \frac{1}{2}\,\mathrm{trace}\, L_A Y \\ & \text{s.t.} & Y = \widehat{V}R\widehat{V}^T \\ & & \mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(Y) = \mathcal{G}_{\widehat{\mathcal{J}}_{\mathcal{I}}}(e_0e_0^T) \\ & & R \succeq 0 \\ & & 0 \le \mathcal{G}_{\widehat{\mathcal{J}}_0^c}(Y) \le 1. \end{array} \tag{2.63}$$

This is a separable convex programming problem with linear coupling constraints from the facial reduction. One can then apply first order splitting methods, which allows us to take advantage of the two variables and the two cones to obtain two separate subproblems. We will discuss one such method in Section 3 below and discuss how the corresponding subproblems can be solved efficiently (by giving a closed form solution).

In passing, we would like to emphasize that the problem (2.63) is stable in that it has no redundant equality constraints, even though we added an extra linear constraint and a new variable Y. In detail, let $\mathcal{T} : \mathbb{S}^{nk+1} \times \mathbb{S}^{(n-1)(k-1)+1} \to \mathbb{S}^{nk+1} \times \mathbb{R}^{|J_{\mathcal{I}}|+1}$ be the linear operator defined as

$$\mathcal{T}(Y,R) = \begin{bmatrix} Y - \widehat{V} R \widehat{V}^T \\ \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(Y) \end{bmatrix}, \tag{2.64}$$

where $\widehat{V}$ is defined in (2.58). We show in Proposition 2.16 below, that the operator $\mathcal{T}$ is an *onto* linear transformation.

Proposition 2.16. *1. Suppose that $\mathcal{T}$ is given in (2.64) and $(W,w) \in \mathbb{S}^{nk+1} \times \mathbb{R}^{|J_{\mathcal{I}}|+1}$. Then*

$$\mathcal{T}^*(W,w) = 0 \implies (W,w) = 0.$$

2. Primal (generalized) Slater points of model (2.63) are given by $\widehat{R}$ in (2.62) and $\widehat{Y}$ in (2.47).

Proof. 1. Algebraic manipulation of $\mathcal{T}^*(W,w) = 0$ yields the following two equations,

$$W + \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w) = 0 \quad \text{and} \quad \widehat{V}^T W \widehat{V} = 0. \tag{2.65}$$

Combining the above two equations, we have $\widehat{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w)\widehat{V} = 0$. This implies that

$$\widetilde{V}^T(\widehat{V}^T)^\dagger \widehat{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w)\widehat{V}\widehat{V}^\dagger \widetilde{V} = 0.$$

Next, recall that $\text{Range}(\widehat{V}) = \text{Range}(\widetilde{V})$ by construction. Thus, we have $\widehat{V}\widehat{V}^\dagger\widetilde{V} = \widetilde{V}$. Combining this with the above display yields $\widetilde{V}^T \mathcal{G}^*_{\widehat{J}_{\mathcal{I}}}(w)\widetilde{V} = 0$. Then we deduce from Theorem 2.9 that $w = 0$. This together with the first relation in (2.65) gives $W = 0$ and completes the proof.

2. This follows immediately from Theorem 2.15.

□

3 sPRSM for DNN relaxation

In this section, we adapt the P-R splitting method [12] for solving our **DNN** relaxation (2.63). In essence, we separate the semidefinite cone constraints from the polyhedral constraints and obtain two subproblems. However, we also add back some provably redundant constraints. This is because these constraints are *not* redundant when the subproblems are considered as *independent optimization problems*. We take advantage of this and bring a constraint back if it does not increase the computational cost excessively. We denote this new method by ***FRSMR***.

3.1 FRSMR, A facially reduced splitting method with redundancies

Let $L_s := \frac{1}{2}L_A$. We can rewrite (2.63) trivially as

$$\begin{array}{rl} p^*_{\mathbf{DNN}} = & \min \ \operatorname{trace} L_s Y + \mathbb{1}_{\mathcal{Y}_o}(Y) + \mathbb{1}_{\mathcal{R}_o}(R) \\ & \text{s.t.} \ \ Y = \widehat{V} R \widehat{V}^T. \end{array} \tag{3.1}$$

where we use the *indicator function*, $\mathbb{1}_{\mathcal{S}}(S)$, that takes the value 0 on the set $\mathcal{S}$ and ∞ outside of $\mathcal{S}$, and the two constraint sets in (3.1) are

$$\mathcal{R}_o := \mathbb{S}_+^{(k-1)(n-1)+1}, \quad \mathcal{Y}_o := \{Y \in \mathbb{S}^{nk+1} : \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(Y) = \mathcal{G}_{\widehat{J}_{\mathcal{I}}}(e_0 e_0^T), 0 \le \mathcal{G}_{\widehat{J}_0^c}(Y) \le 1\}. \tag{3.2}$$

While this trivial decomposition is intuitive, a *splitting* method might benefit by operating on *tighter* constraint sets in the variables R and Y. Here, we shrink the sets in (3.2) by adding the following redundant constraints to (3.1):

1. $\operatorname{trace} R = n+1$. Note that this is a redundant constraint in (3.1) because for any (R, Y) feasible for (2.63), we have

   $$\operatorname{trace} R = \operatorname{trace} \widehat{V} R \widehat{V}^T = \operatorname{trace} Y = n+1,$$

   where the last equality follows from the (redundant) constraint $\mathcal{D}_t(Y) = M$ (see Theorem 2.13 Item 1).

2. $\mathcal{D}_o(Y) = \widehat{M}$, whose redundancy follows from Theorem 2.13 Item 1.

3. $\mathcal{G}_{\widehat{J}_0 \setminus \widehat{J}_{\mathcal{I}}}(Y) = \mathcal{G}_{\widehat{J}_0 \setminus \widehat{J}_{\mathcal{I}}}(e_0 e_0^T)$, whose redundancy follows from Theorem 2.11.

4. $e^T Y_{(i0)} = m_i$ for $i = 1, \ldots, k$. This is redundant because any feasible (R, Y) for (2.63) satisfies $\mathcal{D}_t(Y) = M$ and the arrow constraint, thanks to Theorem 2.13 Item 1.

We thus arrive at the following equivalent problem of (3.1):

$$\begin{array}{rl} p^*_{\mathbf{DNN}} = & \min \ \operatorname{trace} L_s Y + \mathbb{1}_{\mathcal{Y}}(Y) + \mathbb{1}_{\mathcal{R}}(R) \\ & \text{s.t.} \ \ Y = \widehat{V} R \widehat{V}^T, \end{array} \tag{3.3}$$

where

$$\begin{array}{rcl} \mathcal{R} & := & \left\{ R \in \mathbb{S}_+^{(k-1)(n-1)+1} : \ \operatorname{trace} R = n+1 \right\}; \\ \mathcal{Y} & := & \{ Y \in \mathbb{S}^{nk+1} : \ \mathcal{G}_{\widehat{J}_0}(Y) = \mathcal{G}_{\widehat{J}_0}(e_0 e_0^T), 0 \le \mathcal{G}_{\widehat{J}_0^c}(Y) \le 1, \\ & & \qquad \mathcal{D}_o(Y) = \widehat{M}, e^T Y_{(i0)} = m_i, i = 1, \ldots, k \}. \end{array}$$

Notice that the sets $\mathcal{R}$ and $\mathcal{Y}$ are much smaller than $\mathcal{R}_o$ and $\mathcal{Y}_o$, respectively. This property may help bring the Y and R iterates closer to the optimal solution set more quickly when a splitting method is applied. In addition, as we shall see later in Section 3.1.1 and Section 3.1.2, these redundant constraints do not significantly increase the computational cost.

We now describe our splitting method for solving (3.3) (which is equivalent to solving (2.63)). We start by writing down the augmented Lagrangian function for (3.3):

$$\mathcal{L}_\beta(R, Y, Z) \ = \ f_{\mathcal{R}}(R) + g_{\mathcal{Y}}(Y) + \langle Z, Y - \widehat{V} R \widehat{V}^T \rangle + \frac{\beta}{2} \left\| Y - \widehat{V} R \widehat{V}^T \right\|^2.$$

where $\beta > 0$ is a penalty parameter for the quadratic penalty term, and $f_{\mathcal{R}}(R)$ and $g_{\mathcal{Y}}(Y)$ are defined respectively as

$$f_{\mathcal{R}}(R) = \mathbb{1}_{\mathcal{R}}(R), \quad g_{\mathcal{Y}}(Y) = \operatorname{trace} L_s Y + \mathbb{1}_{\mathcal{Y}}(Y).$$

Our main Algorithm 3.1 for solving (3.3), which is a standard application of the *strictly contractive Peaceman-Rachford splitting method,* ***sPRSM*** [12] to (3.3), can now be summarized as follows: alternate minimization of $\mathcal{L}_\beta$ in the variables Y and R interlaced by an update of the Z variable. In particular, we update the dual variable Z both after the R-update *and* the Y-update. We need to point out that the R-update and the Y-update in (3.4) are well defined, i.e., the subproblems involved have unique solutions. This is because both constraint sets are closed convex and both objective functions (i.e., the quadratic functions) are strongly convex. (Recall that $\widehat{V}^T\widehat{V} = I$.)

Algorithm 3.1: FRSMR for **DNN** relaxation

Step 1. Pick any $Y^0, Z^0 \in \mathbb{S}^{nk+1}$. Fix $\beta > 0$ and $\gamma \in (0,1)$. Set $t = 0$.

Step 2. For each $t = 0, 1, \ldots,$ update

$$\begin{array}{rcl} R^{t+1} & = & \underset{R \in \mathcal{R}}{\arg\min}\, \mathcal{L}_\beta(R, Y^t, Z^t) = \underset{R}{\arg\min}\, f_{\mathcal{R}}(R) - \langle Z^t, \widehat{V} R \widehat{V}^T \rangle + \frac{\beta}{2}\left\| Y^t - \widehat{V} R \widehat{V}^T \right\|^2, \\ Z^{t+\frac{1}{2}} & = & Z^t + \gamma\beta(Y^t - \widehat{V} R^{t+1} \widehat{V}^T), \\ Y^{t+1} & = & \underset{Y \in \mathcal{Y}}{\arg\min}\, \mathcal{L}_\beta(R^{t+1}, Y, Z^{t+\frac{1}{2}}) = \underset{Y}{\arg\min}\, g_{\mathcal{Y}}(Y) + \langle Z^{t+\frac{1}{2}}, Y \rangle + \frac{\beta}{2}\left\| Y - \widehat{V} R^{t+1} \widehat{V}^T \right\|^2, \\ Z^{t+1} & = & Z^{t+\frac{1}{2}} + \gamma\beta(Y^{t+1} - \widehat{V} R^{t+1} \widehat{V}^T). \end{array} \tag{3.4}$$

We next discuss convergence of the sequence generated by Algorithm 3.1. Recall from Proposition 2.16 that (2.63) has primal generalized Slater points. Consequently, (Y^*, R^*) solves (3.3) if and only if there exists Z^* so that the following first order optimality condition holds:

$$\begin{array}{rcl} 0 & \in & -\widehat{V}^T Z^* \widehat{V} + \mathcal{N}_{\mathcal{R}}(R^*), \\ 0 & \in & L_s + Z^* + \mathcal{N}_{\mathcal{Y}}(Y^*), \\ Y^* & = & \widehat{V} R^* \widehat{V}^T, \end{array} \tag{3.5}$$

where $\mathcal{N}_S(x)$ denotes the normal cone of S at x. The following Theorem 3.1 states that the sequence generated by Algorithm 3.1 converges to a point satisfying (3.5). Its proof can be found in [12].

Theorem 3.1. *Let $\{R^t\}, \{Y^t\}, \{Z^t\}$ be the sequences generated by Algorithm 3.1. Then $\{(R^t, Y^t)\}$ converges to an optimal solution (R^*, Y^*) of (3.3), and $\{Z^t\}$ converges to some Z^* so that (R^*, Y^*, Z^*) satisfies (3.5).*

In Algorithm 3.1, the explicit Z-update in (3.4) is simple and easy. We now show that we have explicit expressions for the R- and Y-updates too.

3.1.1 R-subproblem

Recall that Assumption 2.14 guarantees that $\widehat{V}$ is normalized so that $\widehat{V}^T\widehat{V} = I$. Then the R-subproblem can be explicitly solved by projecting onto the set $\mathcal{R}$

$$\begin{aligned}
R^{t+1} &= \arg\min_{R\in\mathcal{R}} -\langle Z^t, \widehat{V}R\widehat{V}^T\rangle + \tfrac{\beta}{2}\left\|Y^t - \widehat{V}R\widehat{V}^T\right\|^2 \\
&= \arg\min_{R\in\mathcal{R}} \tfrac{\beta}{2}\left\|Y^t - \widehat{V}R\widehat{V}^T + \tfrac{1}{\beta}Z^t\right\|^2 \\
&= \arg\min_{R\in\mathcal{R}} \tfrac{\beta}{2}\left\|R - \widehat{V}^T(Y^t + \tfrac{1}{\beta}Z^t)\widehat{V}\right\|^2 \\
&= \mathcal{P}_{\mathcal{R}}(\widehat{V}^T(Y^t + \tfrac{1}{\beta}Z^t)\widehat{V}),
\end{aligned}$$

where $\mathcal{P}_{\mathcal{R}}$ denotes the projection (nearest point) onto the intersection of the positive semidefinite cone $\mathbb{S}_+^{(k-1)(n-1)+1}$ and the hyperplane $\{R \in \mathbb{S}^{(k-1)(n-1)+1} : \operatorname{trace} R = n+1\}$. For any symmetric matrix $W \in \mathbb{S}^{(n-1)(k-1)+1}$, we have

$$\mathcal{P}_{\mathcal{R}}(W) = U\operatorname{Diag}(\mathcal{P}_{\bar{\Lambda}}(\operatorname{diag}(\Lambda)))U^T,$$

where (U,Λ) contains the eigenpairs of W and $\mathcal{P}_{\bar{\Lambda}}$ denotes the projection of the vector of eigenvalues, i.e., $\operatorname{diag}(\Lambda)$, onto the simplex $\bar{\Lambda} = \{\lambda \in \mathbb{R}_+^{(k-1)(n-1)+1} : \lambda^T e = n+1\}$. Projection onto simplices can be performed efficiently via some standard root-finding strategies; see, for example, [5, 27].

3.1.2 Y-subproblem

The Y-subproblem involves projection onto the polyhedral set $\mathcal{Y}$, i.e.,

$$\begin{aligned}
Y^{t+1} &= \arg\min_{Y\in\mathcal{Y}} \langle L_s, Y\rangle + \langle Z^{t+\frac{1}{2}}, Y - \widehat{V}R^{t+1}\widehat{V}^T\rangle + \tfrac{\beta}{2}\left\|Y - \widehat{V}R^{t+1}\widehat{V}^T\right\|^2 \\
&= \arg\min_{Y\in\mathcal{Y}} \tfrac{\beta}{2}\left\|Y - \widehat{V}R^{t+1}\widehat{V}^T + \tfrac{1}{\beta}(L_s + Z^{t+\frac{1}{2}})\right\|^2.
\end{aligned} \tag{3.6}$$

To present a closed form solution for the update, we let $\Upsilon := \widehat{V}R^{t+1}\widehat{V}^T - \frac{1}{\beta}(L_s + Z^{t+\frac{1}{2}})$ and assume that Υ is blocked as in (2.3). We now partition the set of indices of J_0^c into the following three disjoint sets:

- ζ_r: it includes the 0-th row of Υ except for the 00-element.
- $\zeta_o(\subseteq J_0^c)$: it includes all off-diagonal elements of the blocks in Υ whenever these off-diagonal elements belong to J_0^c.
- ζ_d: it includes the diagonal of Υ except for the 00-element.

We also define the following subsets:

$$\begin{aligned}
\mathcal{Y}_g &:= \{Y \in \mathbb{S}^{nk+1} : \mathcal{G}_{\widehat{J}_0}(Y) = \mathcal{G}_{\widehat{J}_0}(e_0e_0^T)\}; \\
\mathcal{Y}_r &:= \{Y \in \mathbb{S}^{nk+1} : 0 \le \mathcal{G}_{\zeta_r}(Y) \le 1, e^TY_{(i0)} = m_i, i = 1,\ldots,k\}; \\
\mathcal{Y}_o &:= \{Y \in \mathbb{S}^{nk+1} : 0 \le \mathcal{G}_{\zeta_o}(Y) \le 1, \mathcal{D}_o(Y) = \widehat{M}\}; \\
\mathcal{Y}_d &:= \{Y \in \mathbb{S}^{nk+1} : 0 \le \mathcal{G}_{\zeta_d}(Y) \le 1\}.
\end{aligned}$$

Note that $\mathcal{Y} = \mathcal{Y}_g \cap \mathcal{Y}_d \cap \mathcal{Y}_r \cap \mathcal{Y}_o$. The next iterate Y^{t+1} can now be computed as follows:

$$(Y^{t+1})_{ij} = \begin{cases} 1 & \text{if } i = j = 0, \\ 0 & \text{if } ij \in J_0, \\ (\mathcal{P}_{\mathcal{Y}_r}(\Upsilon))_{ij} & \text{if } ij \in \zeta_r, \\ (\mathcal{P}_{\mathcal{Y}_o}(\Upsilon))_{ij} & \text{if } ij \in \zeta_o, \\ \min(1, \max(\Upsilon_{ij}, 0)) & \text{if } ij \in \zeta_d, \end{cases}$$

where $\mathcal{P}_{\mathcal{Y}_r}$ and $\mathcal{P}_{\mathcal{Y}_o}$ denote the orthogonal projection onto the $\mathcal{Y}_r$ and $\mathcal{Y}_o$ respectively. Both $\mathcal{Y}_r$ and $\mathcal{Y}_o$ are intersections of a hyperplane and a box. The projection can be obtained efficiently via standard root-finding algorithms; see, for example, [14, 17].

Denote the inexact approximate solution from **FRSMR** by $(R^{\text{out}}, Y^{\text{out}}, Z^{\text{out}})$. In the following two subsections, we illustrate how we compute the lower and upper bounds with the obtained Z^{out} and Y^{out} , respectively.

3.2 Lower bound from inaccurate relaxation

Since (3.3) is a relaxation of **MC**, we conclude that exact solutions provide a lower bound for the original **MC**. However, the problem size of (3.3) can be extremely large, and it could be very expensive to obtain highly accurate solutions. In the following, we provide an inexpensive way to get a valid lower bound from the output of our algorithm even when the solution is only obtained to a moderate accuracy. Our approach is based on the following function

$$g(Z) := \min_{Y \in \widetilde{\mathcal{Y}}} \langle L_s + Z, Y \rangle - (n+1)\lambda_{\max}(\widehat{V}^T Z \widehat{V}), \tag{3.7}$$

where $\lambda_{\max}(\widehat{V}^T Z \widehat{V})$ denotes the largest eigenvalue of $\widehat{V}^T Z \widehat{V}$ and the constraint set

$$\begin{aligned} \widetilde{\mathcal{Y}} := \{Y \in \mathbb{S}^{nk+1} : \mathcal{G}_{\widehat{J}_0}(Y) = \mathcal{G}_{\widehat{J}_0}(e_0 e_0^T),\ 0 \le \mathcal{G}_{\widehat{J}_0^c}(Y) \le 1, \\ \mathcal{D}_o(Y) = \widehat{M},\ \mathcal{D}_t(Y) = M,\ e^T Y_{(i0)} = m_i, i = 1, \ldots, k\}. \end{aligned}$$

In the following Theorem 3.2, we show that $\max_Z g(Z)$ is indeed a Fenchel dual problem of (3.3). Since the Fenchel dual problem is an unconstrained maximization problem, evaluating g in (3.7) at the t-th iterate Z^t returned by Algorithm 3.1 always yields a lower bound for p^*_{DNN}.[4]

Theorem 3.2. *Consider the problem*

$$d^*_Z := \max_Z g(Z), \tag{3.8}$$

where g is defined in (3.7). Then (3.8) is a concave maximization problem and strong duality holds between (3.3) and (3.8), i.e.,

$$d^*_Z = p^*_{\mathbf{DNN}}, \ \text{ and } d^*_Z \text{ is attained.}$$

Proof. We derive (3.8) as a Fenchel dual problem of (3.3) by finding a best lower bound as follows.

$$p^*_{\mathbf{DNN}} = \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \max_Z \left\{ \langle L_s, Y \rangle + \left\langle Z, Y - \widehat{V} R \widehat{V}^T \right\rangle \right\}$$

[4]This strengthens [18, Lemma 3.2].

$$= \min_{R\in\mathcal{R},Y\in\widetilde{\mathcal{Y}}} \max_Z \left\{ \langle L_s, Y\rangle + \left\langle Z, Y - \widehat{V}R\widehat{V}^T \right\rangle \right\} \tag{3.9a}$$

$$= \max_Z \min_{R\in\mathcal{R},Y\in\widetilde{\mathcal{Y}}} \left\{ \langle L_s, Y\rangle + \left\langle Z, Y - \widehat{V}R\widehat{V}^T \right\rangle \right\} \tag{3.9b}$$

$$= \max_Z \left\{ \min_{Y\in\widetilde{\mathcal{Y}}} \{\langle L_s, Y\rangle + \langle Z, Y\rangle\} + \min_{R\in\mathcal{R}} \langle Z, -\widehat{V}R\widehat{V}^T\rangle \right\}$$

$$= \max_Z \left\{ \min_{Y\in\widetilde{\mathcal{Y}}} \{\langle L_s, Y\rangle + \langle Z, Y\rangle\} + \min_{R\in\mathcal{R}} \langle \widehat{V}^T Z\widehat{V}, -R\rangle \right\}$$

$$= \max_Z \left\{ \min_{Y\in\widetilde{\mathcal{Y}}} \langle L_s + Z, Y\rangle - (n+1)\lambda_{\max}(\widehat{V}^T Z\widehat{V}) \right\} = d_Z^*, \tag{3.9c}$$

where:

1. (3.9a) follows from the redundancy of the constraint $\mathcal{D}_t(Y) = M$ as guaranteed by Theorem 2.13; [5]

2. (3.9b) follows from [24, Corollary 28.2.2], [24, Theorem 28.4] and the fact that (3.3) has generalized Slater points (see Proposition 2.16). [6]

3. (3.9c) follows from the definition of $\mathcal{R}$ and the Rayleigh Principle.

The concavity of g is clear, and we see from [24, Corollary 28.2.2] and [24, Corollary 28.4.1] that the dual value d_Z^* is attained. □

3.3 Upper bound from a feasible solution

We now move from lower bounds to finding upper bounds for cut(m). Given an output Y^{out} from our algorithm **FRSMR**, the procedures for computing upper bounds are:

1. We extract a column vector v from Y^{out} in one of the following three ways:[7]
   (a) use column 0 of Y^{out};
   (b) use the eigenvector corresponding to the largest eigenvalue of Y^{out};
   (c) sum of random weighted-eigenvalue eigenvectors of Y^{out}, i.e.,

$$v = \sum_{i=1}^{r} w_i \lambda_i v_i,$$

   where $\lambda_1 \geq \cdots \geq \lambda_r > 0$, are the ordered eigenvalues of Y^{out} with eigenpairs (λ_i, v_i), and $1 \geq w_1 \geq \ldots \geq w_r > 0$ are random ordered weights. The r here is the *numerical rank* of Y^{out}. [8]

[5] Note that the inner maximization forces $Y = \widehat{V}R\widehat{V}^T$.

[6] Note that the Lagrangian is linear in R, Y and linear in Z. Moreover, both constraint sets $\mathcal{R}, \mathcal{Y}$ are convex and compact. Therefore, the result also follows from the classical Von Neumann-Fan minmax theorem.

[7] Note that if Y^{out} is rank-1 and feasible, then the first two methods in Item 1a and Item 1b yield exact solutions to **MC**. This motivates the use of eigenvector information.

[8] MATLAB: $r = \min(\text{sum}(\lambda/(n+1) > 0.1) + 1, n+1)$;

2. For each vector v obtained in Step 1, we extract its last nk elements as a subvector v° and set $X^\circ = \mathrm{mat}(v^\circ)$.

3. For each X° obtained, we find the nearest partition matrix X^* to it. (See Proposition 3.4, below.)

4. For each X^* obtained, an upper bound of **MC** is found as $\frac{1}{2}\,\mathrm{trace}(AX^*BX^{*T})$. We save the best (smallest) upper bound obtained and the corresponding X^*. (We repeat the random choice in Item 1c $\lceil \log(n) \rceil$ times.)

Remark 3.3. *1. First of all, the projection in Item 3 can be done efficiently using linear programming. (Actually in strongly polynomial time if one uses something like the classical Hungarian algorithm.) This is similar to what is done in [18, 19, 30].*

2. In [18], we adopt a similar procedure for calculating upper bound, but only generate the column vector v from Y^{out} using the first two ways in item 1, i.e., Item 1a and Item 1b. In Figure 1, we compare the method in [18] with the above proposed procedure for calculating the upper bound. It demonstrates that Item 1c in our proposed procedure contributes greatly to the upper bound.

Proposition 3.4 ([19, Theorem 6.1])**.** *Let $X^\circ \in \mathbb{R}^{n\times k}$. Then the nearest partition matrix $X^* \in \mathcal{M}_m$ to X° can be found by solving the transportation type linear program*

$$\begin{array}{rl} X^* \in \arg\min & -\mathrm{trace}\, X^{\circ T} X \\ \text{s.t.} & Xe = e \\ & X^T e = m \\ & X \geq 0. \end{array} \tag{3.10}$$

Note that we get an exact solution if $\mathrm{rank}(Y^{\mathrm{out}}) = 1$ and $Y^{\mathrm{out}} = \widehat{V} R^{\mathrm{out}} \widehat{V}^T$. Proposition 3.5 below suggests that the methods described in Item 1a and Item 1b above likely yield reasonable approximate partition matrices. Recall that

$$\mathrm{conv}\,\mathcal{M}_m = \{X \in \mathbb{R}^{n\times k} : Xe = e, X^T e = m, X \geq 0\}.$$

Proposition 3.5 ([19, Proposition 5.2])**.** *Let Y be feasible for (2.63). Let $v_1 = Y_{1:nk\,0}$, and let $\begin{bmatrix} v_0 & v_2^T \end{bmatrix}^T$ denote a unit eigenvector of Y corresponding to the largest eigenvalue. Then $v_0 \neq 0$, and both*

$$X_1^\circ := \mathrm{Mat}(v_1),\; X_2^\circ := \mathrm{Mat}({v_0}^{-1} v_2) \in \mathrm{conv}\,\mathcal{M}_m.$$

However, in general Y^{out} is not an exact solution of the **DNN** relaxation. Then Item 1c plays an important role in generating many vectors v for finding an upper bound. We see this in Section 4.3.3 below. In fact, this allows us to stop the algorithm with much fewer iterations when we see that both the upper and lower bounds are not improving.

4 Numerical experiments

In this section we apply the proposed **FRSMR** method in Algorithm 3.1 to solve the **DNN** relaxation in (3.3). All the tests are performed using Matlab R2017a on a ThinkPad X1 with an Intel CPU (2.5GHz) and 8GB RAM running Windows 10.

4.1 Classes of problems and parameters

We consider three classes of problems, see Sections 4.3.1 to 4.3.3. We outline them here:

(a) (random structured graphs, Section 4.3.1.) We compare with the **DNN** relaxation in [19].[9] The latter relaxation is solved using an interior point approach with Mosek version 8.0.0.60. [1]. See Table 4.2.

(b) (partially random graphs with various sizes, Section 4.3.2.) There are four kinds of random graphs, classified by the number of 1's, $|\mathcal{I}|$, in the vector m. In particular, in the three cases where $\mathcal{I} \neq \emptyset$, we *almost always* obtain a zero gap and thus the optimal solution. See Tables 4.3 to 4.6.

(c) (vertex separator instances, Section 4.3.3.) We compare with the bounds obtained by solving the relaxation $\mathbf{SDP}_4$ in [22]. In addition, we include comparisons on the upper bounds on the size of the vertex separator. See Table 4.7.

4.2 Parameters, initialization, stopping criteria

In our implementation, we first shift the objective to obtain positive definiteness.

$$L \leftarrow L + \alpha I, \quad \alpha = 0.1 + \max\{0, -\lambda_{\min}(L)\}.$$

This does not change the optimum Y^* but it changes the dual Z and promotes $Z \preceq 0$, as can be seen from the expression for the Y-subproblem in (3.6). This in turn promotes a better lower bound from (3.9c).

We now specify the parameters used in **FRSMR** in Sections 4.3.1 to 4.3.3.

1. The penalty and step parameters are, respectively,

$$\beta = \frac{3k}{n}, \qquad \gamma = 0.9.$$

2. We terminate once one of the following Items 2a to 2c holds:

   (a) the number of iterations reaches 10000;

   (b) the relative gap, rel-gap, is either zero[10] or does not change in $\max\{5, \lceil n/10 \rceil\}$ consecutive iterations,

$$\text{rel-gap} = \frac{(\text{best upper bound} - \text{ best lower bound})}{(\text{best upper bound} + \text{best lower bound} + 1)/2}; \tag{4.1}$$

   (c)

$$\max\left\{\left\|Y^{t+1} - \widehat{V}R^{t+1}\widehat{V}^T\right\|, \left\|Y^{t+1} - Y^t\right\|\right\} < 10^{-12}; \tag{4.2}$$

   This criterion (4.2) is the same as that suggested in [13, Remark 2.3].

[9] The **DNN** relaxation in [19] imposes the additional nonnegativity constraints $\widehat{V}Z\widehat{V}^T \geq 0$ onto their $\mathbf{SDP}_{\text{final}}$ relaxation.

[10] Note that our data are integral and we round up the lower bound, therefore the gap is integer valued. Thus, finding a zero duality gap is reasonable. Moreover, the lower bounds are nonnegative.

3. We calculate: the lower bound and the upper bound every 100th iteration, using Theorem 3.2 (to compute a lower bound as $\lceil g(Z^t)\rceil$) and the procedures in Section 3.3. In the computation of the upper bound, we sample the random weight vector $\lceil \log(n)\rceil$ times. The linear program (3.10) involved in the computation of the upper bound is solved with Mosek using their function 'mosekopt' and the dual-simplex method.

4. The data terminology in our Tables are described in Table 4.1.

Table 4.1: Data terminology.

imax	the maximum size of each set
k	the number of sets
n	the number of nodes, i.e., the sum of the sizes of the sets
p	the density of the graph, i.e., $2\|E\|/(\|V\|(\|V\|-1))$
$l = e^T m_{\text{one}}$	the number of 1's in m
Iters	the number of iterations
Time	CPU time in seconds
Bounds	best lower and upper bounds and relative gap
Residuals	*final* values $\left\Vert Y^{t+1} - \widehat{V}R^{t+1}\widehat{V}^T\right\Vert$ ($\cong \Delta Z$); $\left\Vert Y^{t+1} - Y^t\right\Vert$ ($\cong \Delta Y$)

5. In Section 4.3.3 we consider the special class of vertex separator problems.

   (a) The penalty and step parameters in **FRSMR** are, respectively,

   $$\beta = 0.001, \qquad \gamma = 0.9.$$

   (b) The stopping criterion is set as the same as in Sections 4.3.1 and 4.3.2.

   (c) We calculate the lower bound every 100-th iteration using Theorem 3.2. We compute the upper bound every iteration using the procedures in Section 3.3. Other settings in the computation of the upper bound are the same as in Sections 4.3.1 and 4.3.2.

4.3 Three classes of problems

4.3.1 Random structured graphs

The structured graphs are generated as in [19, Sect. 7.1]. That is, we first generate k disjoint cliques of sizes $m_1, \ldots, m_k$, randomly chosen from $\{2, ..., \mathsf{imax}\}$. We then join the first $k-1$ cliques to every node of the k-th clique, and add u_0 edges between the first $k-1$ cliques, chosen uniformly at random from the complement graph. In our experiments below, we set $u_0 = \lfloor e_c d\rfloor$, where e_c is the number of edges in the complement graph and d is the density (percentage of edges in the complement graph to be added). By construction, $u_0 \geq \text{cut}(m)$.

We use small instances with $k = 4, 5$, $d = 10\%$ and imax $= 6, 8$. We compare our approach with the DNN relaxation model in [19] solved by Mosek [1]. The results in Table 4.2 illustrate the improvement in solution time.

Table 4.2: Comparison results for small structured graphs with DNN relaxation model in [19].

Data				Lower bounds		Upper bounds		Rel-gap		Time (cpu)	
n	k	$\|E\|$	u_0	**FRSMR**	Mosek	**FRSMR**	Mosek	**FRSMR**	Mosek	**FRSMR**	Mosek
20	4	136	6	6	6	6	6	0.00	0.00	0.14	5.41
25	4	222	8	8	8	8	8	0.00	0.00	0.22	10.24
25	5	170	14	14	14	14	14	0.00	0.00	0.27	30.36
31	5	265	22	22	22	22	22	0.00	0.00	1.15	126.11

4.3.2 (Partially) random graphs with various sizes

We test four groups of random graphs corresponding to different values of $\mathcal{I}$:

1. ($\mathcal{I} = \emptyset$) vector m is generated by choosing k integers randomly from $\{2, ..., \text{ imax}\}$;
2. ($k \notin \mathcal{I} \neq \emptyset$) after generating m as in Item 1 above, we randomly select elements from $\{m_1, m_2, \dots, m_{k-1}\}$ and set them to be 1;
3. ($k \in \mathcal{I} \neq \mathcal{K}$) after generating m as in Item 1 above, we set $m_k = 1$ and randomly select no more than $k-2$ elements from $\{m_1, m_2, \dots, m_{k-1}\}$ and set them to be 1;
4. ($\mathcal{I} = \mathcal{K}$) simply set imax $= 1$ and set all the elements of m to be 1.

Then, as $n = m^T e$ is the total number of nodes in the simple, undirected graph, we randomly generate an adjacency matrix A of a graph on n nodes with density $=$ densityA, and construct the Laplacian matrix.[11]

In Tables 4.3 to 4.6, we consider the four groups of random graphs in Items 1 to 4, above. In each group of random graphs, we generate m and A by choosing k and imax as given in the tables with various values for densityA; the density p of the graphs is also reported.

From Table 4.3, i.e, in the case of $\mathcal{I} = \emptyset$, we can see that the **FRSMR** in general takes a reasonably short time to converge. Moreover, in most instances, the rel-gap is very small; sometimes we even obtain a zero gap and hence the instance is solved to optimality. **FRSMR** appears to perform better in the cases when $\mathcal{I} \neq \emptyset$. The corresponding results are shown in Tables 4.4 to 4.6. We can see that in most instances, the rel-gap is zero and the problem is solved exactly. Moreover, the CPU times taken are reasonably small.

4.3.3 Vertex separator problem

We now test some vertex separator problems from https://sites.google.com/site/sotirovr/the-vertex-separator. We compare against the bounds obtained from the model $\mathbf{SDP}_4$ in [22]. In each instance, the m has the special structure that $k = 3$, $|m_1 - m_2| \leq 1$ and $\text{cut}(m) > 0$. In this case, by solving **MC**, one can separate the nodes of the graph into S_1, S_2 and S_3 so that the number of edges between S_1 and S_2 is minimized. If $\text{cut}(m) = 0$, for some $m = (m_1, m_2, m_3)^T$, then we say that S_3 separates S_1 and S_2, and S_3 is called a *vertex separator*. If $\text{cut}(m) > 0$, on the other hand, it means that no separator S_3 for the cardinalities specified in m exists. However, we can experiment with different choices of m, i.e, transferring nodes from S_1 and S_2 to S_3, in the hope of

[11] MATLAB: A = abs(sprandsym(sum(m),densityA))> 0; A = A - diag(diag(A));

Table 4.3: Results for random graphs with $\mathcal{I} = \emptyset$.

Specifications					Iters	Time (cpu)	Bounds			Residuals	
imax	k	n	p	l			lower	upper	rel-gap	primal	dual
4	5	17	0.43	0	500	0.94	16	17	0.06	9.51e-04	1.01e-04
4	5	17	0.32	0	100	0.19	10	10	0.00	1.93e-02	1.75e-02
5	6	23	0.35	0	500	1.75	37	42	0.13	1.81e-03	1.92e-04
5	6	23	0.30	0	600	1.92	30	34	0.12	1.07e-03	1.68e-04
6	7	30	0.28	0	900	5.99	42	48	0.13	1.65e-03	1.28e-04
6	7	30	0.22	0	600	4.14	31	40	0.25	3.24e-03	3.88e-04
7	8	37	0.18	0	700	9.03	32	38	0.17	6.29e-03	1.56e-03
7	8	37	0.14	0	700	9.13	18	22	0.20	5.22e-03	1.18e-03
8	9	49	0.10	0	1200	47.09	14	19	0.29	5.68e-03	8.18e-04
8	9	49	0.05	0	1000	45.52	0	6	1.71	1.31e-04	1.83e-04

Table 4.4: Results for random graphs with $k \notin \mathcal{I} \neq \emptyset$.

Specifications					Iters	Time (cpu)	Bounds			Residuals	
imax	k	n	p	l			lower	upper	rel-gap	primal	dual
4	5	14	0.37	1	100	0.17	6	6	0.00	1.59e-02	1.26e-02
4	5	14	0.37	1	100	0.17	5	5	0.00	2.88e-02	4.62e-02
5	6	16	0.35	2	400	0.92	11	11	0.00	1.70e-03	4.32e-04
5	6	16	0.32	2	100	0.24	11	11	0.00	2.81e-02	3.22e-02
6	7	19	0.27	4	500	1.79	8	9	0.11	2.73e-03	3.29e-04
6	7	19	0.22	4	500	1.76	4	5	0.20	1.75e-03	4.32e-04
7	8	12	0.20	7	100	0.21	0	0	0.00	1.20e-02	1.54e-02
7	8	12	0.17	7	100	0.21	0	0	0.00	2.19e-02	1.97e-02
8	9	16	0.12	8	100	0.38	0	0	0.00	4.78e-02	6.50e-02
8	9	16	0.06	8	100	0.38	0	0	0.00	3.06e-02	3.10e-02

eventually producing a separator. In this way, we can obtain an upper bound of the cardinality of a vertex separator. Here, we follow the approach described in [22, Section 8] to derive an upper bound of the cardinality of a vertex separator, using solutions obtained from **FRSMR**.

In Table 4.7, we compare the lower and upper bounds for cut(m) obtained from (3.3) and from the model $\mathbf{SDP}_4$ in [22]. We also report the upper bound of the cardinality of vertex separator obtained for each instance. The (upper and lower) bounds for $\mathbf{SDP}_4$ are obtained directly from [22, Table 3].[12] From Table 4.7, we can see that the **MC** upper bounds from the model (3.3) are very competitive with those obtained from the model $\mathbf{SDP}_4$. For most instances, the upper bounds are equal except for two instances, "grid3dt(5)" and "grid3dt(7)"; as for the comparison of upper bounds for vertex separator, still most upper bounds are equal, except for "can-144","gridt(15)"," gridt(5)","gridt(6)" and "gridt(7)".

Figure 1 shows the comparison of the upper bound using Section 3.3 (new upper bound derived via all three items there) and the method in [18] that only uses the Item 1a and Item 1b. It demonstrates that our new strategy can produce much better upper bound than the method that uses only the Item 1a and Item 1b.

[12]These results use extra cutting planes, and therefore they obtain stronger lower bounds on cut(m).

Table 4.5: Results for random graphs with $k \in \mathcal{I} \neq \mathcal{K}$.

Specifications					Iters	Time (cpu)	Bounds			Residuals	
imax	k	n	p	l			lower	upper	rel-gap	primal	dual
4	5	12	0.45	2	100	0.16	11	11	0.00	1.41e-03	2.03e-03
4	5	12	0.39	2	100	0.14	9	9	0.00	1.08e-02	1.38e-02
5	6	15	0.33	3	100	0.21	13	13	0.00	2.43e-02	3.80e-02
5	6	15	0.29	3	100	0.21	10	10	0.00	3.12e-02	5.09e-02
6	7	18	0.27	4	100	0.37	13	13	0.00	8.97e-02	1.03e-01
6	7	18	0.22	4	300	0.95	10	10	0.00	3.82e-03	2.76e-03
7	8	13	0.21	7	100	0.23	5	5	0.00	7.67e-03	8.75e-03
7	8	13	0.18	7	100	0.23	4	4	0.00	1.56e-02	1.94e-02
8	9	16	0.11	8	100	0.47	2	2	0.00	5.51e-02	1.04e-01
8	9	16	0.06	8	100	0.49	0	0	0.00	1.30e-02	1.47e-02

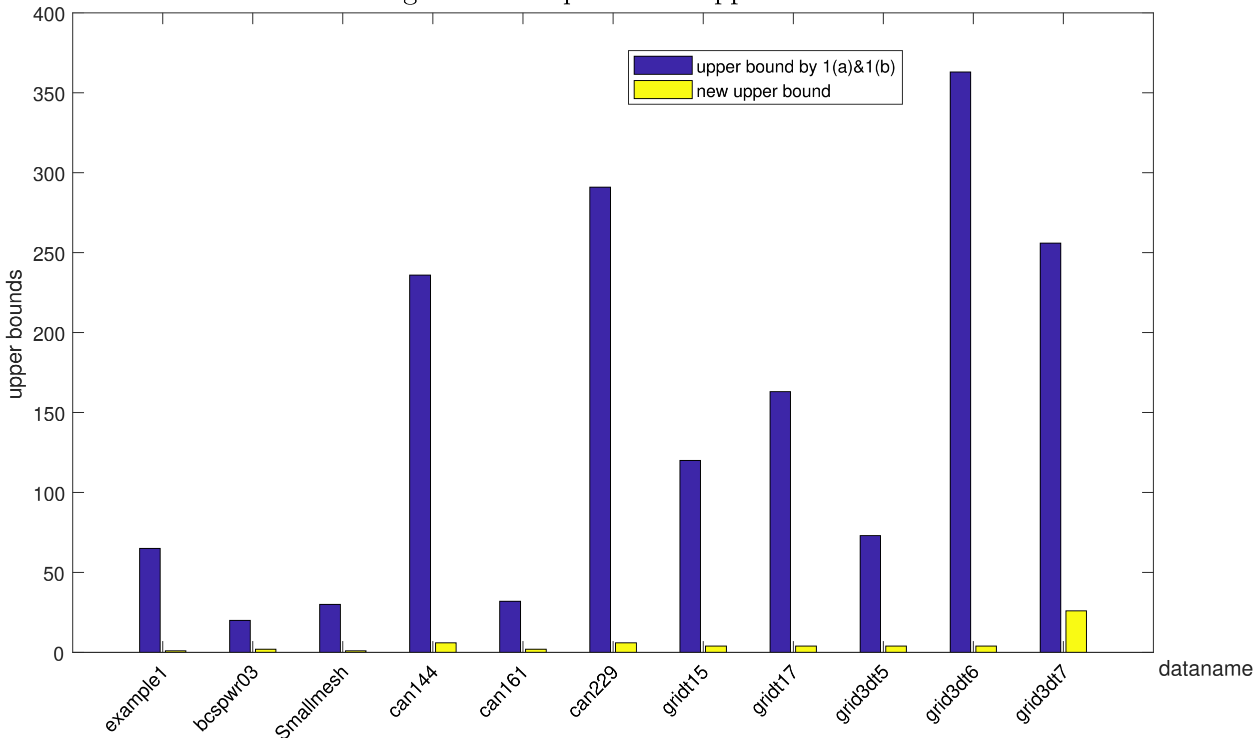
Table 4.6: Results for random graphs with $\mathcal{I} = \mathcal{K}$.

Specifications					Iters	Time (cpu)	Bounds			Residuals	
imax	k	n	p	l			lower	upper	rel-gap	primal	dual
1	8	8	0.64	8	100	0.17	12	12	0.00	4.22e-04	6.08e-04
1	10	10	0.69	10	100	0.26	23	23	0.00	9.94e-03	1.26e-02
1	12	12	0.47	12	100	0.39	23	23	0.00	1.86e-02	3.32e-02
1	14	14	0.46	14	100	0.66	33	33	0.00	6.37e-02	8.99e-02
1	16	16	0.44	16	100	1.04	43	43	0.00	1.69e-01	2.49e-01
1	18	18	0.39	18	200	3.71	48	48	0.00	1.45e-02	2.22e-02
1	20	20	0.29	20	200	7.31	47	47	0.00	3.75e-02	4.04e-02
1	22	22	0.25	22	200	11.24	47	47	0.00	1.39e-01	1.58e-01
1	24	24	0.13	24	200	16.41	31	31	0.00	1.06e-01	1.13e-01
1	26	26	0.05	26	200	23.75	10	10	0.00	1.19e-01	8.14e-02

Table 4.7: Comparisons on the bounds for **MC** and bounds for the cardinality of separators.

Name	n	$\|E\|$	m_1	m_2	m_3	lower	upper	lower	upper	lower	upper	upper
						MC by $\mathbf{SDP}_4$		**MC** by (3.3)		Separator by $\mathbf{SDP}_4$		Separator by (3.3)
Example 1	93	470	42	41	10	0.07	1	0	1	11	11	11
bcspwr03	118	179	58	57	3	0.56	1	0	2	4	5	5
Smallmesh	136	354	65	66	5	0.13	1	0	1	6	6	6
can-144	144	576	70	70	4	0.90	6	0	6	5	6	8
can-161	161	608	73	72	16	0.31	2	0	2	17	18	18
can-229	229	774	107	107	15	0.40	6	0	6	16	19	19
gridt(15)	120	315	56	56	8	0.29	4	0	4	9	11	12
gridt(17)	153	408	72	72	9	0.17	4	0	4	10	13	13
grid3dt(5)	125	604	54	53	18	0.54	2	0	4	19	19	22
grid3dt(6)	216	1115	95	95	26	0.28	4	0	4	27	30	31
grid3dt(7)	343	1854	159	158	26	0.60	22	0	27	27	37	44

Figure 1: Comparison of upper bounds



5 Conclusion

In this paper we introduced new methods for finding strengthened lower and upper bounds for the **MC** problem. **SDP** relaxations provide strong bounds that are further strengthened by nonnegativity constraints, i.e., by using the **DNN** relaxation. However, in general solving the **DNN** relaxation by interior-point methods is extremely expensive.

Strict feasibility fails for the **SDP** relaxation of **MC**, but **FR** can be used to regularize the problem and simultaneously make all but the gangster constraint redundant. The **FR** appears to provide a natural splitting for the variables $Y = \widehat{V} R \widehat{V}^T$, where Y, R are restricted to the polyhedral and cone constraints, respectively. We exploit this within a **sPRSM** framework.

We bring back previously redundant constraints to strengthen the two subproblems in Y, R. In addition, we periodically find lower and upper bound estimates in order to stop the algorithm early, i.e., with low accuracy.

Our numerical experiments show that our approach for solving **MC** improves on the existing approaches in [19, 22].

Index

References

[1] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2017. 31, 32

[2] 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. 6

[3] 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. 11

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Machine Learning*, 3(1):1–122, 2011. 4

[5] Y. Chen and X. Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011. 27

[6] 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. 11

[7] D. Cornaz, Y. Magnouche, A.R. Mahjoub, and S. Martin. The multi-terminal vertex separator problem: polyhedral analysis and branch-and-cut. *Discrete Appl. Math.*, 256:11–37, 2019. 3

[8] M. Didi Biha and M.-J. Meurs. An exact algorithm for solving the vertex separator problem. *J. Global Optim.*, 49(3):425–434, 2011. 3

[9] D. Drusvyatskiy, G. Li, and H. Wolkowicz. A note on alternating projections for ill-posed semidefinite feasibility problems. *Math. Program.*, 162(1-2, Ser. A):537–548, 2017. 11

[10] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. 7, 11

[11] W.W. Hager, J.T. Hungerford, and I. Safro. A multilevel bilinear programming algorithm for the vertex separator problem. *Comput. Optim. Appl.*, 69(1):189–223, 2018. 3

[12] B. He, H. Liu, Z. Wang, and X. Yuan. A strictly contractive Peaceman-Rachford splitting method for convex programming. *SIAM J. Optim.*, 24(3):1011–1040, 2014. 4, 24, 26

[13] Bingsheng He, Feng Ma, and Xiaoming Yuan. Convergence study on the symmetric version of ADMM with larger step sizes. *SIAM J. Imaging Sci.*, 9(3):1467–1501, 2016. 4, 31

[14] Carlos Lara, Juan J Flores, and Felix Calderon. On the hyperbox–hyperplane intersection problem. *INFOCOMP Journal of Computer Science*, 8(4):21–27, 2009. 28

[15] R.H. Lewis. Yet another graph partitioning problem is NP-hard. Technical report, 2014. 3

[16] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991. 6

[17] N. Maculan, C.P. Santiago, E.M. Macambira, and M.H.C. Jardim. An $O(n)$ algorithm for projecting a vector on the intersection of a hyperplane and a box in $\mathbb{R}^n$. *J. Optim. Theory Appl.*, 117(3):553–574, 2003. 28

[18] D.E. Oliveira, H. Wolkowicz, and Y. Xu. ADMM for the SDP relaxation of the QAP. *Math. Program. Comput.*, 10(4):631–658, 2018. 4, 10, 23, 28, 30, 34

[19] 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. 2, 3, 4, 5, 10, 11, 19, 20, 30, 31, 32, 33, 36

[20] 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. 3

[21] 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

[22] F. Rendl and R. Sotirov. The min-cut and vertex separator problem. *Comput. Optim. Appl.*, 69(1):159–187, 2018. 3, 4, 31, 33, 34, 36

[23] S. M. Robinson. Regularity and stability for convex multivalued functions. *Math. Oper. Res.*, 1:130–143, 1976. 12

[24] R.T. Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks. 29

[25] 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. 6

[26] H. Sun. ADMM for SDP relaxation of GP. Master's thesis, University of Waterloo, 2016. 1

[27] E. van den Berg and M.P. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008/09. 27

[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. 4, 6, 11, 13, 16, 18, 20

[29] Q. Zhao. *Semidefinite Programming for Assignment and Partitioning Problems*. PhD thesis, University of Waterloo, 1996. 6, 13

[30] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. volume 2, pages 71–109. 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Toronto, ON, 1996). 6, 10, 30