DISCRETE APPLIED MATHEMATICS

ELSEVIER Discrete Applied Mathematics 96–97 (1999) 461–479

# Semidefinite programming relaxations for the graph partitioning problem ☆

Henry Wolkowicz [*,1], Qing Zhao [2,3]

*Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ont., Canada N2L 3G1*

Received 8 October 1996; received in revised form 29 June 1998; accepted 23 November 1998

**Abstract**

A new semidefinite programming, SDP, relaxation for the *general* graph partitioning problem, GP, is derived. The relaxation arises from the dual of the (homogenized) Lagrangian dual of an appropriate quadratic representation of GP. The quadratic representation includes a representation of the 0,1 constraints in GP. The special structure of the relaxation is exploited in order to project onto the *minimal face* of the cone of positive-semidefinite matrices which contains the feasible set. This guarantees that the Slater constraint qualification holds, which allows for a numerically stable primal–dual interior-point solution technique. A *gangster operator* is the key to providing an efficient representation of the constraints in the relaxation. An incomplete preconditioned conjugate gradient method is used for solving the large linear systems which arise when finding the Newton direction. Only dual feasibility is enforced, which results in the desired lower bounds, but avoids the expensive primal feasibility calculations. Numerical results illustrate the efficacy of the SDP relaxations. 

*MSC:* 90C25; 90C27; 15A48

*Keywords*: Graph partitioning; Lagrangian relaxations; Semidefinite programming relaxations

## 1. Introduction

In this paper we present a new semidefinite programming, SDP, relaxation for the general graph partitioning, GP, problem; i.e. the problem consists in partitioning the node set of a graph into $k$ disjoint subsets of given, though *not* necessarily equal, sizes

☆ This report is available by anonymous ftp at orion.uwaterloo.ca, in directory pub/henry/reports and also with URL: http://orion.uwaterloo.ca/˜hwolkowi/henry/reports/graphpart.ps.gz

* Corresponding author.

*E-mail addresses:* henry@orion.uwaterloo.ca (H. Wolkowicz), qing.zhao@delta-air.com (Q. Zhao)

[1] Research supported by Natural Sciences Engineering Research Council Canada.

[2] Research supported by Natural Sciences Engineering Research Council Canada.

[3] Currently at Delta Technology, Atlanta.


PII: S0166-218X(99)00102-X

so that the sum of the weights of the edges between the disjoint subsets is minimized. Our relaxation is obtained through the dual of the Lagrangian dual of a quadratic model of GP. In particular, we add a quadratic representation of the binary constraints to the quadratic model; and, we exploit the structure of a *gangster operator* in the SDP relaxation to enforce zeros. These additional constraints provide strengthened bounds for this NP-hard problem.

We further exploit the special structure of our SDP relaxation and explicitly find the minimal face of the cone of $n \times n$ positive-semidefinite matrices, $\mathscr{P}$, which contains the feasible set of the relaxation. We then consider the SDP in the span of this minimal face. This guarantees that the Slater constraint qualification (strict feasibility) holds. This, in turn, allows us to use a numerically stable primal–dual interior-point, p–d i-p, approach to solve the SDP. The Newton equation which arises in each iteration can be very large. We solve these large linear systems using an incomplete conjugate gradient method. At each iteration we obtain a lower bound, since we maintain dual feasibility. We disregard primal feasibility since that involves a very large linear system.

### 1.1. Background

GP can best be described as follows.

> Given: an undirected graph $G = (\mathscr{V}, \mathscr{E})$ having nodes $\mathscr{V}$ and edges $\mathscr{E}$ and a weight, $a_{ij}$, for each edge. We consider the problem of partitioning $\mathscr{V}$ into $k$ disjoint subsets $\mathscr{V}_1, \ldots, \mathscr{V}_k$ of given sizes $m_1 \geqslant \cdots \geqslant m_k$ in such a way that the sum of weights of edges that connect nodes in different subsets (cut edges) is minimized.

We use $a_{ij}$ for the weight for the edge between node $i$ and node $j$, where $a_{ij} = 0$ if there is no edge between node $i$ and node $j$. The symmetric matrix $A = \{a_{ij}\}$, with 0 on the diagonal, is the *adjacency matrix* of the graph. For a given partition of the graph into $k$ subsets, let $X = (x_{ij})$ be the $n \times k$ matrix ($n = \sum_i m_i$ is the cardinality of $\mathscr{V}$) defined by

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

Thus the $j$th column $X_{:j}$ is the indicator set for the $j$th subset. Such an $X$ can represent the partition. We let

$$\mathscr{F}_k = \{X \in \Re^{n \times k} \colon X \text{ represents a partition}\}.$$

For each such partition $X$,

$$\tfrac{1}{2}\operatorname{trace} X^{\mathrm{t}} A X = \tfrac{1}{2}\operatorname{trace} A X X^{\mathrm{t}}$$

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

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

where $e$ is the vector of ones. Note that for any partition matrix $X$, we have

$$\operatorname{trace} X^{\mathrm{t}} \operatorname{Diag}(Ae)X = e^{\mathrm{t}}Ae,$$

where Diag is the diagonal matrix formed from the vector. Therefore, the minimal weight of cut edges can be obtained by solving the graph partitioning problem in the trace formulation.

$$(GP) \quad \begin{array}{lll} w^*(E_{\mathrm{cut}}) := & \min & \frac{1}{2} \operatorname{trace} X^{\mathrm{t}}LX \\ & \text{s.t.} & X \in \mathscr{F}_k, \end{array}$$

where the matrix

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

is called the *Laplace matrix of the graph*.

The graph partitioning problem is well known to be NP-hard and therefore finding an optimal solution is likely very difficult. Yet this problem has many applications. One important application is VLSI design; see e.g. [15] for a survey of Integrated Circuit Layout.

One popular and very successful heuristic for finding "good" partitions was proposed by Kernighan and Lin [14] in 1970. (See also [9] for its application on netlist partitioning.) In the early 1970s Donath and Hoffman [7] provided an eigenvalue-based bound for GP. Several new strengthened eigenvalue-based bounds were presented by Rendl and Wolkowicz [18]; a computational study showed these bounds to be very good, see Falkner et al. [8]. In [1], Alizadeh introduced several semidefinite relaxations for various graph related problems. In particular, he showed that the Donath–Hoffman bound can be obtained as the dual of a semidefinite relaxation of GP. More recently, Anstreicher and Wolkowicz [2] show that the Donath–Hoffman bound can actually be obtained using the Lagrangian dual of an appropriate quadratically constrained problem. A semidefinite relaxation technique for the equal-partitioning problem, which included additional polyhedral constraints, has been successfully developed in [12], see also [13]. These last two papers contain excellent detailed descriptions and historical background of these various bounds; in addition, the detailed relationships between these bounds is given in [12]. We give some details on comparisons with our bound in Remark 2.1.

### *1.2. Outline*

The main result in this paper is the application of an incomplete conjugate gradient approach within a p–d i-p method that solves an SDP relaxation for the general (not necessarily equipartitions) GP problem.

A preliminary (unreduced) SDP relaxation is presented in Section 2. Therein it is noted that the standard Slater CQ fails. Also, the connection to the Donath–Hoffman bound is discussed, see Remark 2.1. The geometry of the relaxation is studied in Section 3. The minimal face of $\mathscr{P}$ that contains the feasible set is characterized. This characterization is used in Section 4 to project the problem onto the span of the minimal face. Moreover, redundant constraints are eliminated resulting in our final, very efficient SDP relaxation, (4.3), (4.4).

We then present numerical results in Section 5 and summarize in Section 6.

## 2. A preliminary SDP relaxation

We now follow the approach in [17,22] and derive our SDP relaxation using Lagrangian duality. (This is sometimes called the Shor relaxation, see e.g. [20].) This results in a relaxation with many redundant constraints and with no strict interior for the feasible set. We project the feasible set onto a face of the semidefinite cone and then identify the redundant constraints and obtain the final form of the relaxation. The reader may wish to skip the details and go straight to the final SDP relaxation, (4.3), (4.4).

In order to derive a semidefinite programming relaxation, we will formulate GP as a quadratically constrained quadratic programming problem. The SDP relaxation is then found from the dual of the Lagrangian dual of this program. We let $\circ$ denote the Hadamard (elementwise) product, $e$ is the vector of ones, and $\bar{m} = (m_1, \dots, m_k)^{\mathrm{t}}$. We first note that we can formulate GP as follows:

$$\begin{array}{rll} w^*(E_{\text{cut}}) = \min & \frac{1}{2}\operatorname{trace} X^{\mathrm{t}} L X & \\ \text{s.t.} & X \circ X = X & (0,1 \text{ constraints}), \\ & Xe = e & (\text{one set per node}), \\ & X^{\mathrm{t}} e = \bar{m} & (m_i \text{ nodes in set } i), \\ & X_{:i} \circ X_{:j} = 0, \ \forall i \neq j & (\text{Hadamard column orthogonality}). \end{array} \tag{2.1}$$

The last strong orthogonality constraint is redundant. However, redundant constraints do not have to be redundant in the Lagrangian relaxation.

An equivalent quadratically constrained quadratic problem is

$$\begin{array}{rl} w^*(E_{\text{cut}}) = \min & \frac{1}{2}\operatorname{trace} X^{\mathrm{t}} L X \\ \text{s.t.} & X \circ X = X, \\ & \|Xe - e\|^2 = 0, \\ & \|X^{\mathrm{t}} e - \bar{m}\|^2 = 0, \\ & X_{:i} \circ X_{:j} = 0 \ \forall i \neq j. \end{array} \tag{2.2}$$

**Remark 2.1.** We emphasize here that our relaxation is for general graph partitioning where the subsets of nodes do not have to be equal. It is hard to compare our results with others in the literature since most other tests are done on equipartitioning problems. For example, our numerical tests showed that our bounds could be better or worse than the classical Donath–Hoffman bounds.

However, there is a way to explicitly compare the bounds. A subset of the constraints in (2.2) are included in the following constraints:

$$X^{t}X = \mathrm{Diag}(m), \quad \mathrm{diag}(XX^{t}) = e. \tag{2.3}$$

If we had included these constraints in our relaxation, then we would have a provably stronger bound than the Donath–Hoffman bound, since the Lagrangian relaxation with only these two sets of bounds yields the Donath–Hoffman bound. (This is proved in [2].) The addition of the missing constraints is the subject of ongoing research.

To derive the semidefinite relaxation, we can now take the dual of the (homogenized) Lagrangian dual of this problem. (See [21,22] for the details of this approach applied to the quadratic assignment problem. We have to square the linear terms or they disappear in the Lagrangian relaxation.) A direct approach is based on the now well-known *lifting process* (e.g. [3,16,19]), i.e. we use the substitution (or linearization)

$$Y_X := \begin{pmatrix} 1 \\ \mathrm{vec}(X) \end{pmatrix} (1 \ \mathrm{vec}(X)^{t}),$$

where $\mathrm{vec}(X)$ is the vector formed from the columns of $X$ and $Y_X \succcurlyeq 0$, i.e. is positive semidefinite. For example, the objective function becomes

$$\min \tfrac{1}{2}\,\mathrm{trace}\, X^{t}LX = \mathrm{trace}\, L_A Y,$$

where $L_A$ is defined below in (2.5). We then remove the rank one restriction and replace $Y_X$ by a general symmetric matrix $Y$.

We get the following semidefinite relaxation:

$$\text{(RGP)} \qquad \begin{array}{rl} w^*(E_{\mathrm{cut}}) \geqslant \mu^* := \min & \mathrm{trace}\, L_A Y \\ \text{s.t.} & \mathrm{arrow}(Y) = e_0, \\ & \mathrm{trace}\, D_1 Y = 0, \\ & \mathrm{trace}\, D_2 Y = 0, \\ & \mathcal{G}_J(Y) = 0, \\ & Y_{00} = 1, \\ & Y \succcurlyeq 0, \end{array} \tag{2.4}$$

where:

$$L_A := \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} I \otimes L \end{bmatrix}, \tag{2.5}$$

the *arrow operator*, acting on the $(n^2+1)\times(n^2+1)$ matrix $Y$, is defined as

$$\mathrm{arrow}(Y) := \mathrm{diag}(Y) - (0, Y_{0,1:n^2})^{t}, \tag{2.6}$$

where $Y_{0,1:n^2}$ is the vector formed from the last $n^2$ components of the first, or 0, row of $Y$ and diag denotes the vector formed from the diagonal elements or the adjoint operator of Diag; the arrow constraint represents the 0,1 constraints by guaranteeing that the diagonal and 0th row (or column) are identical; $e_0$ is the first unit vector; the gangster operator $\mathscr{G}_J : \mathscr{S}_{n^2+1} \to \mathscr{S}_{n^2+1}$ shoots "holes" in a matrix, i.e. the $ij$ component is defined as

$$(\mathscr{G}_J(Y))_{ij} := \begin{cases} Y_{ij} & \text{if } (i,j) \text{ or } (j,i) \in J, \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

where the set

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

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

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

and

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

Since both $D_1$ and $D_2$ are positive semidefinite, the feasible set of problem (RGP) has no strictly feasible (positive definite) points. There can be numerical difficulties if we apply an interior-point method directly to a problem without interior. However, one can find a very simple structured matrix in the relative interior of the feasible set in order to project (and regularize) the problem into a smaller dimension. This we do in Section 3.

## 3. Geometry

In this section we study the geometrical structure of the feasible set, denoted $\mathscr{F}$, and of the convex cone $\mathscr{P}$ of the SDP relaxation (RGP). (More details on the classical results on $\mathscr{P}$ can be found in e.g. [4,5].)

A set $T$ is a cone (convex) if $T + T \subset T$ and $\lambda T \subset T,\ \forall \lambda \in \Re$. The cone $K \subset T$ is a *face* of the cone $T$, denoted $K \lhd T$, if

$$x, y \in T,\ x + y \in K \quad \Rightarrow \quad x, y \in K. \tag{3.1}$$

The faces of $\mathscr{P}$ have very special structure. Each face, $K \lhd \mathscr{P}$, is characterized by a unique subspace, $S \subset \Re^n$:

$$K = \{X \in \mathscr{P}\colon\ \mathscr{N}(X) \supset S\},$$

where $\mathscr{N}$ denotes null space. Moreover, the relative interior

$$\operatorname{relint}(K) = \{X \in \mathscr{P}\colon\ \mathscr{N}(X) = S\}.$$

The *complementary* (or conjugate) face of $K$ is $K^{\rm c} = K^{\perp} \cap \mathscr{P}$ and

$$K^{\rm c} = \{X \in \mathscr{P}\colon \mathscr{N}(X) \supset S^{\perp}\}. \tag{3.2}$$

Again,

$$\operatorname{relint}(K^{\rm c}) = \{X \in \mathscr{P}\colon \mathscr{N}(X) = S^{\perp}\}.$$

Note that a subset $K$ of a convex set $C$ is called a *face* if

$$x, y \in C,\ \alpha x + (1-\alpha) y \in K,\ 0 \leqslant \alpha \leqslant 1 \quad \Rightarrow \quad x, y \in K.$$

We now characterize the minimal face of $\mathscr{P}$ which contains $\mathscr{F}$. It is clear that the matrices

$$Y_X := \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} (1 \ \operatorname{vec}(X)^{\rm t}) \quad \text{for } X \text{ a partition}$$

are in $\mathscr{F}$. From the structure of the faces of $\mathscr{P}$, every matrix in the relative interior of a face has the same null space (and range space). Therefore, since these points $Y_X$ are rank one matrices, we see that they are contained in the set of extreme points of $\mathscr{F}$. We need only consider the intersection of faces of $\mathscr{P}$ which contain all of these extreme points $Y_X$. The following theorem characterizes the minimal face by finding a point in its relative interior, namely the *barycenter* point. This point has a very simple and elegant structure.

**Theorem 3.1.** *Let* $x = \operatorname{vec}(X)$. *Define the barycenter point*

$$\hat{Y} := \frac{m_1! \ldots m_k!}{n!} \sum_{\text{partitions } X} \begin{bmatrix} 1 & x^{\rm t} \\ x & xx^{\rm t} \end{bmatrix}. \tag{3.3}$$

*Then*:

1. *the barycenter is*

$$\hat{Y} = \begin{bmatrix} 1 & \frac{m_1}{n} e_n^{\rm t} & \cdots & \frac{m_k}{n} e_n^{\rm t} \\ \frac{m_1}{n} e_n & (\frac{m_1}{n} I_n + \frac{m_1(m_1-1)}{n(n-1)}(E_n - I_n)) & \cdots & (\frac{m_1 m_k}{n(n-1)})(E_n - I_n) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{m_k}{n} e_n & (\frac{m_1 m_k}{n(n-1)})(E_n - I_n) & \cdots & (\frac{m_k}{n} I_n + \frac{m_k(m_k-1)}{n(n-1)}(E_n - I_n)) \end{bmatrix},$$

2. *the rank of the barycenter*

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

3. *the rows of*

$$T := \begin{bmatrix} -m_1 & e_n^{\rm t} & 0 & \cdots & \cdots & 0 \\ -m_2 & 0 & e_n^{\rm t} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -m_k & 0 & \cdots & \cdots & 0 & e_n^{\rm t} \\ -e_n & I_n & I_n & \cdots & \cdots & I_n \end{bmatrix} \tag{3.4}$$

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

4. *the columns of*

$$\hat{V} := \begin{bmatrix} 1 & 0 \\ \frac{1}{n}\bar{m} \otimes e_n & V_k \otimes V_n \end{bmatrix} \tag{3.5}$$

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

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

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

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

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

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

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

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

1. Assume that $j = q$ and $i = p$. There are again $((n-1)!m_j)/(m_1!\dots m_k!)$ partitions to GP with $x_{(p-1)n+q} = 1$, i.e. this confirms the fact that the diagonal elements are equal to the elements of the 0th row.
2. Assume that the node indices $i = p$ while the set indices $j \neq q$. Since the same node cannot be in two different sets, this implies that the diagonal elements of the off-diagonal blocks of the matrices $Y_X$ are all 0.
3. Assume that the node indices $i \neq p$ while the set indices $j = q$. These are the off-diagonal elements of the diagonal blocks. Then there are $((n-2)!m_j(m_j-1))/(m_1!\dots m_k!)$ possible partitions. After dividing this by $n!/(m_1!\dots m_k!)$, and adding the diagonal parts, we get the blocks $(m_j/n)I_n + (m_j(m_j-1)/n(n-1))(E_n - I_n)$.
4. If both the node indices $i \neq p$ and the set indices $j \neq q$, then these are the off-diagonal elements of the off-diagonal blocks. There are $((n-2)!m_j m_q)/(m_1!\dots m_k!)$ possible partitions. After dividing appropriately, we get the expression in for $\hat{Y}$ in 1.

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

$$\hat{Y} = \begin{bmatrix} 1 & z^{\mathrm{t}} \\ z & W \end{bmatrix},$$

where $z=(1/n)\bar{m}\otimes e_n$. Then

$$\begin{bmatrix} 1 & 0 \\ -\frac{1}{n}\bar{m}\otimes e_n & I \end{bmatrix} \hat{Y} \begin{bmatrix} 1 & -\frac{1}{n}\bar{m}^{\mathrm{t}}\otimes e_n^{\mathrm{t}} \\ 0 & I \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & S \end{bmatrix}, \tag{3.6}$$

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

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

Direct verification shows that

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

The null space of $(n\,\mathrm{Diag}(\bar{m})-\bar{m}\bar{m}^{\mathrm{t}})$ and the null space of $(nI_n-E_n)$ are spanned by $e_k$ and $e_n$, respectively. Therefore, their range spaces are spanned by the columns of $V_k$ and $V_n$, respectively. Hence, the range space of $S$ is spanned by the columns of $V_k\otimes V_n$. This implies that $\mathrm{rank}(S)=(k-1)(n-1)$. This proves 2 and 3. Moreover, we have that the null space of $\hat{Y}$ is of dimension $k+n-1$. Since

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

and

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

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

**Remark 3.1.** The structure of the polytope of partitions has been well studied. The feasible set $\mathscr{F}$ is a relaxation of the polytope obtained by lifting the partition matrices into the higher-dimensional matrix space. Therefore the dimension of the minimal face and the structure of the null space can be studied from the known results of the polytope of partitions. [4]

## 4. The final semidefinite relaxation

From Theorem 3.1 we conclude that $Y\succcurlyeq 0$ is in the minimal face if and only if $Y=\hat{V}Z\hat{V}^{\mathrm{t}}$, for some $Z\succcurlyeq 0$. We can now substitute $\hat{V}Z\hat{V}^{\mathrm{t}}$ for $Y$ in the SDP relaxation (RGP). We get the following reduced SDP relaxation:

$$\begin{array}{ll} \min & \mathrm{trace}\,\hat{V}^{\mathrm{t}}L_A\hat{V}Z \\ \text{s.t.} & \mathrm{arrow}(\hat{V}Z\hat{V}^{\mathrm{t}})=0, \\ & \mathscr{G}_J(\hat{V}Z\hat{V}^{\mathrm{t}})=0, \\ & (\hat{V}Z\hat{V}^{\mathrm{t}})_{00}=1, \\ & Z\succcurlyeq 0. \end{array} \tag{4.1}$$

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

[4] Thanks to Levent Tuncel for pointing this out.

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

$$Z = \begin{bmatrix} Z_{00} & Z_{01} & \dots & Z_{0(k-1)} \\ Z_{10} & Z_{11} & \dots & Z_{1(k-1)} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{(k-1)0} & Z_{(k-1)1} & \dots & Z_{(k-1)(k-1)} \end{bmatrix},$$

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

$$Y = \begin{bmatrix} Y_{00} & Y_{01} & \dots & Y_{0k} \\ Y_{10} & Y_{11} & \dots & Y_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k0} & Y_{k1} & \dots & Y_{kk} \end{bmatrix},$$

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

(a)

$$Y_{00} = Z_{00},$$

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

*and*

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

(b)

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

(c)

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

*and*

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

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

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

The remaining results follow from direct verification. □

From Lemma 4.1, we conclude that the arrow operator is redundant if both the gangster constraint holds and $(\hat{V} Z \hat{V}^{t})_{00} = 1$. Now we will show that when we project the gangster operator onto its range, then there are no other redundant constraints. We do this by showing that the null space of the adjoint operator is 0.

**Lemma 4.2.** *Suppose that* $W \in \mathscr{S}_{n^2+1}$. *Then*

$$\hat{V}^{\mathrm{t}}\mathscr{G}(W)\hat{V} = 0 \;\Rightarrow\; \mathscr{G}(W) = 0.$$

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

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

where $Y_{ij}$ for $i,j \in \{1,\dots,k\}$ are $n \times n$ matrices. Therefore from

$$\hat{V}^{\mathrm{t}} Y \hat{V} = 0$$

we let

$$Z = (V \otimes V)^{\mathrm{t}} \begin{bmatrix} Y_{11} & \dots & Y_{1k} \\ \vdots & \ddots & \vdots \\ Y_{k1} & \dots & Y_{kk} \end{bmatrix} (V \otimes V),$$

then $Z = 0$. Note that

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

Therefore if we write the above matrix $Z$ as

$$\begin{bmatrix} Z_{11} & \dots & Z_{1k-1} \\ \vdots & \ddots & \vdots \\ Z_{k-11} & \dots & Z_{k-1k-1} \end{bmatrix}.$$

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

$$Z_{ij} = V^{\mathrm{t}}(Y_{ij} - Y_{kj} - Y_{ik} + Y_{kk})V = 0. \tag{4.2}$$

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

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

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

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

Then

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

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

Therefore, by eliminating the redundant constraints we can get a very simple projected relaxation. We let $\bar{J} = J \cup \{(0,0)\}$ and we add $\mathscr{G}_{\bar{J}}$ to the right-hand side in order to emphasize the restriction to the range of this operator.

$$\text{(PRGP)} \quad \begin{array}{ll} \min & \operatorname{trace}(\hat{V}^{t} L_A \hat{V})Z \\ \text{s.t.} & \mathscr{G}_{\bar{J}}(\hat{V} Z \hat{V}^{t}) = \mathscr{G}_{\bar{J}}(E_{00}), \\ & Z \succcurlyeq 0. \end{array} \tag{4.3}$$

Its dual problem is

$$\text{(DPRGP)} \quad \begin{array}{ll} \max & W_{00} \\ \text{s.t.} & \hat{V}^{t} \mathscr{G}_{\bar{J}}(W) \hat{V} \preccurlyeq \hat{V}^{t} L_A \hat{V}. \end{array} \tag{4.4}$$

The dimension of the range of the gangster operator $\mathscr{G}_{\bar{J}}(\cdot)$ is the cardinality of the set $\bar{J}$. When solving this pair of dual problems we can restrict the operator to this space. From the above lemma, this guarantees that the operator is onto. The dual has to be adjusted accordingly.

For p–d i-p methods, it is useful to have positive-definite feasible points for both the primal and dual feasible sets.

**Theorem 4.1.** *The* $((k-1)(n-1)+1) \times ((k-1)(n-1)+1)$ *matrix*

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

*where*

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

*is a positive-definite feasible point for the primal feasible set* (4.3).

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

The rest of the proof follows from showing

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

where $\hat{Y}$ is the barycenter, see Theorem 3.1 part 1. We see that

$$\begin{aligned}
\hat{V}\hat{X}\hat{V}^{\mathrm{t}} &= \begin{bmatrix} 1 & 0 \\ \frac{1}{n}\bar{m}\otimes e_n & V_k\otimes V_n \end{bmatrix} \hat{X} \begin{bmatrix} 1 & \frac{1}{n}\bar{m}^{\mathrm{t}}\otimes e_n^{\mathrm{t}} \\ 0 & V_k^{\mathrm{t}}\otimes V_n^{\mathrm{t}} \end{bmatrix} \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}\bar{m}\otimes e_n \end{pmatrix} (1, \tfrac{1}{n}\bar{m}^{\mathrm{t}}\otimes e_n^{\mathrm{t}}) \\
&\quad + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{n^2(n-1)}(V_k(n\,\mathrm{Diag}(\bar{m}_{k-1}) - \bar{m}_{k-1}\bar{m}_{k-1}^{\mathrm{t}})V_k^{\mathrm{t}})\otimes(V_n(nI_{n-1}-E_{n-1})V_n^{\mathrm{t}}) \end{bmatrix} \\
&= \begin{pmatrix} 1 \\ \frac{1}{n}\bar{m}\otimes e_n \end{pmatrix} (1, \tfrac{1}{n}\bar{m}^{\mathrm{t}}\otimes e_n^{\mathrm{t}}) \\
&\quad + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{n^2(n-1)}(n\,\mathrm{Diag}(\bar{m}) - \bar{m}\bar{m}^{\mathrm{t}})\otimes(nI_n - E_n) \end{bmatrix} \\
&= \hat{Y}
\end{aligned}$$

follows from

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

and

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

**Theorem 4.2.** *The matrix*

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

*is a strictly feasible point for the dual feasible set* (4.4), *if* $\alpha$ *is a sufficiently negative real scalar.*

**Proof.** Note that $\mathscr{G}_{\bar{J}}(\hat{W}) = \hat{W}$. We can write $\hat{V}^{\mathrm{t}}(\hat{W} - L_A)\hat{V}$ as

$$\hat{V}^{\mathrm{t}} \begin{bmatrix} 0 & 0 \\ 0 & I\otimes L \end{bmatrix} \hat{V} + \hat{V}^{\mathrm{t}} \begin{bmatrix} -\alpha & 0 \\ 0 & (I_k - E_k)\otimes I_n \end{bmatrix} \hat{V}.$$

Note that

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

We have for the first term

$$\begin{aligned}
\hat{V}^{\mathrm{t}} \begin{bmatrix} 0 & 0 \\ 0 & I\otimes L \end{bmatrix} \hat{V} &= \begin{bmatrix} 1 & \bar{m}^{\mathrm{t}}\otimes e^{\mathrm{t}}/n \\ 0 & V_k^{\mathrm{t}}\otimes V_n^{\mathrm{t}} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & I\otimes L \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \bar{m}\otimes e/n & V_k\otimes V_n \end{bmatrix} \\
&= \begin{bmatrix} 0 + (\bar{m}^{\mathrm{t}}\bar{m})\otimes(e^{\mathrm{t}}Le)/n^2 & (\bar{m}^{\mathrm{t}}V_k)\otimes(e^{\mathrm{t}}LV_n)/n \\ (V_k^{\mathrm{t}}\bar{m})\otimes(V_n^{\mathrm{t}}Le)/n & (V_k^{\mathrm{t}}V_k)\otimes(V_n^{\mathrm{t}}LV_n) \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 \\ 0 & (I_{k-1} + E_{k-1})\otimes(V_n^{\mathrm{t}}LV_n) \end{bmatrix}.
\end{aligned}$$

Since $(I_{k-1}+E_{k-1})$ is positive definite and $V_n^{\mathrm{t}} L_A V_n$ is positive semidefinite, we get $(I_{k-1}+E_{k-1})\otimes(V_n^{\mathrm{t}} L_A V_n)$ is positive semidefinite, i.e. $\hat{V}^{\mathrm{t}} L_A \hat{V}$ is positive semidefinite. Now for the second term, note that $e^{\mathrm{t}} V = 0$, and we have

$$\begin{aligned}
\hat{V}^{\mathrm{t}} \hat{W} \hat{V} &= \begin{bmatrix} 1 & \bar{m}^{\mathrm{t}} \otimes e^{\mathrm{t}}/n \\ 0 & V_k^{\mathrm{t}} \otimes V_n^{\mathrm{t}} \end{bmatrix} \begin{bmatrix} -\alpha & 0 \\ 0 & ((I_k - E_k)\otimes I_n \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \bar{m}\otimes e/n & V_k \otimes V_n \end{bmatrix} \\
&= \begin{bmatrix} -\alpha + \bar{m}^{\mathrm{t}}(I_k - E_k)\bar{m}/n & (\bar{m}^{\mathrm{t}}(I_k - E_k)V_k)\otimes(e^{\mathrm{t}} V_n)/n \\ (V_k^{\mathrm{t}}(I_k - E_k)\bar{m})\otimes(V_n^{\mathrm{t}} e)/n & (V_k^{\mathrm{t}}(I_k - E_k)V_k)\otimes(V_n^{\mathrm{t}} V_n) \end{bmatrix} \\
&= \begin{bmatrix} -\alpha + \bar{m}^{\mathrm{t}}(I_k - E_k)\bar{m}/n & 0 \\ 0 & (I_{k-1}+E_{k-1})\otimes(I_{n-1}+E_{n-1}) \end{bmatrix}.
\end{aligned}$$

Since both $I_{k-1}+E_{k-1}$ and $I_{n-1}+E_{n-1}$ are positive definite, we can see that when $-\alpha$ is large enough $\hat{V}^{\mathrm{t}} \hat{W} \hat{V}$ is negative definite. □

## 5. Numerical tests

The algorithm (a p–d i-p approach) we use to solve the SDP relaxation is very similar to the one in [21,22] for the quadratic assignment problem. Therefore, we only give a brief outline here. An incomplete conjugate gradient method is used to solve the large Newton equations that arise. After solving the relaxation, we obtain not only a lower bound for the graph partitioning problem but also an appropriate solution $\bar{Y}$ for the SDP relaxation. By re-shaping the diagonal of $\bar{Y}$, we can get an $n \times k$ matrix $\bar{Z}$ which satisfies all the feasible constraints except the 0–1 constraint for the original graph partitioning problem. By solving a network subproblem with $\bar{Z}$ as its adjacency matrix, we can find an upper bound for the graph partitioning problem. With this upper bound as an initial solution, we use Adaptive Simulated Annealing technique (or VFSR, see e.g. [10]) to generate a better upper bound. To measure how close our upper bound is to the optimal solution, we use the measure

$$\text{gap} := \frac{\text{upper bound} - \text{lower bound}}{\text{lower bound}}.$$

Our numerical results are based on random unweighted and weighted graphs. We include two instances (labelled #a, #b) for each case. First, eight unweighted graphs were randomly generated. Each edge was generated independent of other edges with probability 0.5. These graphs have vertices of 36, 60, 84 and 108, respectively. The number of partitions $k$ are 2, 3, 4. The size for each partition is randomly generated. Next, another eight weighted graphs were randomly generated. Each edge was generated independent of other edges. The weights are integer numbers between 0 and 10. These graphs have vertices of 36, 60, 84 and 108, respectively. The number of partitions $k$ are 2, 3, 4. The size for each partition is randomly generated. In Tables 1–6 the column under LB is the lower bound, the column under INIT is the initial upper bound and the column under BEST is the upper bound generated by the VFSR. The last column under GAP is for the gap.

Table 1
Bisection for unweighted graphs

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

Table 2
3-partition for unweighted graphs

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

Table 3
4-partition for unweighted graphs

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

Table 4
Bi-partition for weighted graphs

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

Table 5
3-partition for weighted graphs

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

Table 6
4-partition for weighted graphs

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

From the tables for weighted graphs (Tables 4–6), we observe that the gaps are less than 0.05. However, for unweighted graph (Tables 1–3), the gaps are mostly between 0.05 and 0.10. The initial upper bounds derived from the SDP solution are very good as we can see that the upper bound can hardly be improved by VFSR.

The results significantly improve those in [8] and so illustrate that our bound is better than both the D–H bound and the projected D–H bound. Moreover, the results are comparable to the results in [11,12], where we must emphasize that the results in [11,12] are for the very restrictive equipartition case and their bounds are obtained using additional polyhedral bounds (cuts) which can be added to our relaxation to further improve our bounds.

## 6. Conclusion

We have derived a semidefinite programming relaxation for the general (not restricted to equipartioning) graph partitioning problem. This relaxation includes many new equality constraints (such as the gangster operator) that make it stronger than previous relaxations which were based on equalities arising from a quadratic formulation of GP. We have not included inequality constraints, though these should strengthen the relaxation once they are included, as was seen in the relaxation used in [12].

Due to the additional constraints, our SDP relaxation is larger than the one presented in [1] or the ones used in [12]. This is because we had to use the lifting

$$Y_X := \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} \begin{pmatrix} 1 & \operatorname{vec}(X)^{\mathrm{t}} \end{pmatrix},$$

rather than the smaller lifting $Y_X := XX^{\mathrm{t}}$, which can be done in the equipartition case with fewer constraints.

We have applied a primal–dual interior-point method to solve the relaxation. We have used an incomplete conjugate gradient method to solve the large linear system resulting from the search direction equations. In addition, we have exploited the fact that we are looking at finding a lower bound, i.e. we do not have to close the duality gap but in fact, we just keep improving our lower bound at each iteration. This fact was a key in the lower bounds for the QAP in [22]. Since then it has also been exploited in [6] where problems of much larger size have been tackled. Therefore, this approach shows much promise for the future.

## Appendix A. Notation

| | |
|---|---|
| SDP | semidefinite programming problem |
| GP | graph partitioning problem |
| $\mathscr{P}_n$ or $\mathscr{P}$ | the cone of positive semidefinite matrices in $\mathscr{S}_n$ |
| Slater CQ | the Slater constraint qualification; strict feasibility |
| p–d i-p | primal–dual interior-point method |
| $G = (\mathscr{V}, \mathscr{E})$ | graph with node set $\mathscr{V}$ and edge set $\mathscr{E}$ |
| $\bar{m}$ | $(m_1, \ldots, m_k)^{\mathrm{t}}$ |
| cut edge | an edge connecting nodes in different subsets of a partition |
| $A = (a_{ij})$ | adjacency matrix of the graph |
| $X = (x_{ij})$ | partition matrix |
| $\mathscr{F}_k$ | set of partition matrices |
| $w(E_{\mathrm{cut}})$ | total weight of cut edges of the partition |
| $w^*(E_{\mathrm{cut}})$ | minimal total weight of cut edges over all partitions |
| Diag | the diagonal matrix formed from the vector |
| diag | the vector formed from the diagonal elements |
| $L$ | Laplace matrix of the graph |
| $B \circ C$ | the Hadamard product of $B$ and $C$ |
| $Y_X$ | partition matrix lifted into higher dimensional matrix space |
| $e$ | the vector of ones in the appropriate dimension |
| $A \otimes B$ | the Kronecker product of $A$ and $B$ |
| $\operatorname{vec}(X)$ | the vector formed from the columns of the matrix $A$ |
| $Y_{0,1:n^2}$ | the $n^2$ vector from the first row of $Y$ |

| | |
|---|---|
| arrow | the arrow operator $\mathrm{diag}(Y) - (0, (Y_{0,1:n^2})^{\mathrm{t}}$ |
| $\mathscr{G}_J$ | the gangster operator $\mathscr{G}_J$ shoots "holes" in a matrix |
| $\bar{J}$ | $\bar{J} = J \cup \{(0,0)\}$ |
| $K \lhd C$ | $K$ is a face of $C$ |
| relint | relative interior |
| $\mathscr{R}(B)$ | range space of $B$ |
| $\mathscr{N}(B)$ | null space of $B$ |
| $Q \preccurlyeq R$ | $R - Q$ is positive semidefinite |
| $\mathrm{diag}(A)$ | the vector formed from the diagonal of the matrix $A$ |
| $\mathrm{Diag}(v)$ | the diagonal matrix formed from the vector $v$ |
| $E_n$ | the matrix of ones in $\mathscr{S}_n$ |
| $E_{ij}$ | the $ij$ unit matrix in $\mathscr{S}_n$ |

## References


[1] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Optim. 5 (1995) 13–51.

[2] K. Anstreicher, H. Wolkowicz, On Lagrangian relaxation of quadratic matrix constraints, Research Report, corr 98-24, University of Waterloo, Waterloo, Ontario, 1998.

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

[4] G.P. Barker, The lattice of faces of a finite dimensional cone, Linear Algebra Appl. 7 (1973) 71–82.

[5] G.P. Barker, D. Carlson, Cones of diagonally dominant matrices, Pacific J. Math. 57 (1975) 15–32.

[6] S.J. Benson, Y. Ye, X. Zhang, Solving large-scale sparse semidefinite programs for combinatorial optimization, Technical Report, The University of Iowa, 1997.

[7] W.E. Donath, A.J. Hoffman, Lower bounds for the partitioning of graphs, IBM J. Res. Development 17 (1973) 420–425.

[8] J. Falkner, F. Rendl, H. Wolkowicz, A computational study of graph partitioning, Math. Programming 66(2) (1994) 211–240. URL: ftp://orion.uwaterloo.ca/pub/henry/reports/gpcomput.ps.gz.

[9] S.W. Hadley, B.L. Mark, A. Vannelli, An efficient eigenvector approach for finding netlist partitions, IEEE Trans. Comput. Aided Des. 11 (1992) 885–892.

[10] L. Ingber, Very fast simulated re-annealing, Math. Comput. Modeling 12 (1989) 967–973.

[11] S. Karisch, Nonlinear approaches for quadratic assignment and graph partition problems, Ph.D. Thesis, University of Graz, Graz, Austria, 1995.

[12] S.E. Karisch, F. Rendl, Semidefinite programming and graph equipartition, in: Topics in Semidefinite and Interior-Point Methods, The Fields Institute for Research in Mathematical Sciences, vol. 18, Communications Series, American Mathematical Society, Providence, RI, 1998. URL http://www.diku.dk/%7Ekarisch/rep302.ps.

[13] S.E. Karisch, F. Rendl, J. Clausen, Solving graph bisection problems with semidefinite programming, Technical Report Report DIKU-Tr-97/9, Department of Computer Science, University of Copenhagen, 1997.

[14] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Systems Tech. J. 49 (1970) 291–307.

[15] T. Lengauer, Combinatorial Algorithms for Integrated Circuit Layout, Wiley, Chicester, 1990.

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

[17] S. Poljak, F. Rendl, H. Wolkowicz, A recipe for semidefinite relaxation for (0,1)-quadratic programming, J. Global Optim. 7 (1995) 51–73.

[18] F. Rendl, H. Wolkowicz, A projection technique for partitioning the nodes of a graph, Ann. Oper. Res. 58 (1995) 155–180. URL: ftp://orion.uwaterloo.ca /pub/henry/reports/gp.ps.gz.

[19] H.D. Sherali, W.P. Adams, Computational advances using the reformulation-linearlization technique (rlt) to solve discrete and continuous nonconvex problems, Optima 49 (1996) 1–6.
[20] N.Z. Shor, Quadratic optimization problems, Soviet J. Comput. Systems Sci. 25 (6) (1997) 1–11.
[21] Q. Zhao, Semidefinite programming for assignment and partitioning problems, Ph.D. Thesis, University of Waterloo, 1996. URL: ftp://orion.uwaterloo.ca/pub/henry/software/qap.d/zhaophdthesis.ps.gz.
[22] Q. Zhao, S. Karisch, F. Rendl, H. Wolkowicz, Semidefinite programming relaxations for the quadratic assignment problem, J. Combin. Optim. 2.1, 1998. CORR 95-27, URL: ftp://orion.uwaterloo.ca/pub/henry/reports/qapsdp.ps.gz.