Graph bisection revisited

R. Sotirov *

Abstract

The graph bisection problem is the problem of partitioning the vertex set of a graph into two sets of given sizes such that the sum of weights of edges joining these two sets is optimized. We present a semidefinite programming relaxation for the graph bisection problem with a matrix variable of order n - the number of vertices of the graph - that is equivalent to the currently strongest semidefinite programming relaxation obtained by using vector lifting. The reduction in the size of the matrix variable enables us to impose additional valid inequalities to the relaxation in order to further strengthen it. The numerical results confirm that our simplified and strengthened semidefinite relaxation provides the currently strongest bound for the graph bisection problem in reasonable time.

Keywords: graph bisection, graph partition, semidefinite programming, boolean quadric polytope

1 Introduction

The graph bisection problem (GBP) is the problem of dividing the vertices of a graph into two sets of specified sizes such that the total weight of edges joining different sets is optimized. The GBP is an NP-hard combinatorial optimization problem, see [9]. It has many applications such as VLSI design [16], parallel computing [1, 13, 25], network partitioning [8, 24], and floor planing [2]. Graph partitioning also plays a role in machine learning (see e.g., [17]) and data analysis (see e.g., [21]).

There are several SDP relaxations for the GBP with matrix variables of different orders. In particular, there are relaxations whose matrices have orders n, $2n$, and $2n+1$, where n is the order of the graph. An SDP relaxation with a matrix variable of order n is introduced by Karisch, Rendl, and Clausen [15]. The same relaxation is used by Feige and Langberg [6], and Han, Ye, and Zhang [11] to derive approximation algorithms for the GBP. Another SDP relaxation with a matrix variable of order n that is derived from an SDP relaxation for the more general graph partition problem is introduced in [27]. In [27] it is also proven that the above mentioned SDP relaxations of order n are equivalent.

Wolkowicz and Zhao [29] derived an SDP relaxation with a matrix variable of order $2n+1$. This SDP relaxation with additional nonnegativity constraints dominates the SDP relaxations with matrix variables of order n, see [5, 27].

The GBP can be seen as a special case of the quadratic assignment problem (QAP). De Klerk, Pasechnik, Sotirov, and Dobre [4] exploited this to derive an SDP relaxation for the GBP from an SDP relaxation for the QAP, which however reduces to a much smaller

*Department of Econometrics and OR, Tilburg University, Warandelaan 2, 5000 LE, The Netherlands. email: r.sotirov@uvt.nl

semidefinite program than the original QAP relaxation (see also [5]). In particular that relaxation contains matrix variables of orders n and $2n$. In [27], it is proven that the QAP-based SDP relaxation for the GBP is equivalent to the strongest SDP relaxation, that is the SDP relaxation with nonnegativity constraints from [29].

For specific families of (symmetric) graphs, De Klerk et al. [4] improved the QAP-based SDP relaxation for the GBP by adding a constraint that fixes one vertex of the graph. Finally, in [3] the SDP relaxation for the GBP from [4] was further strengthened by adding two constraints that correspond to assigning two vertices of the graph to different parts of the partition. Both fixing-based strengthening perform well on highly symmetric graphs.

In this paper, we present an SDP relaxation for the bisection problem whose matrix variable is of order n. Our relaxation is equivalent to the strongest SDP relaxation for the GBP, that is the strongest Wolkowicz and Zhao relaxation from [29]. The new SDP relaxation exploits the fact that the matrix variables corresponding to the two parts in the bisection are related. Further, we consider adding the facet defining inequalities of the boolean quadric polytope to our relaxation. We also show that a large subset of the facet defining inequalities are redundant in the relaxation from [29]. The strengthened SDP bound outperforms all previously considered SDP bounds, including those tailored for highly symmetric graphs.

The paper is structured as follows. In Section 2 we provide an integer programming formulation of the problem, and in Section 3 an overview is given of the known SDP relaxations for the graph bisection problem. In Section 4 we present our SDP relaxation and prove that it is equivalent to the strongest SDP relaxation from [29]. We further suggest how to improve our relaxation. Finally, in Section 5 we present numerical results.

2 The graph bisection problem

In this section we formulate the minimum graph bisection problem as an integer optimization problem. Let $G = (V, E)$ be an undirected graph with vertex set V, where $|V| = n$ and edge set E. The goal is to find a partition of the vertex set into two disjoint subsets S_1 and S_2 of specified sizes $m_1 \geq m_2 > 0$, $m_1 + m_2 = n$ such that the sum of weights of edges joining S_1 and S_2 is minimized. If $m_1 = m_2$ then one refers to the associated problem as the graph equipartition problem. We consider here only the case that $m_1 > m_2$. For detailed analysis of the SDP relaxations for the graph equipartition problem, see [26].

Let us denote by A the adjacency matrix of G. For a given partition of the graph G into two subsets, let $Z = (z_{i,j})$ be the $n \times 2$ matrix defined by

$$z_{i,j} := \begin{cases} 1 & \text{if } i \in S_j \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, n, \; j = 1, 2.$$

The jth column of Z is the characteristic vector of S_j. The cut of the partition, which is the sum of weights of edges joining different sets, is equal to:

$$\frac{1}{2}\mathrm{tr}(A(J - ZZ^{\mathrm{T}})) = \frac{1}{2}\mathrm{tr}(LZZ^{\mathrm{T}}),$$

where $L = \mathrm{Diag}(Ae) - A$ is the Laplacian matrix of the graph, and J (resp. e) the all-ones matrix (resp. vector). Therefore, the minimum GBP problem can be formulated as follows

$$\min \left\{ \frac{1}{2}\mathrm{tr}(LZZ^{\mathrm{T}}) : Ze = e, \ Z^{\mathrm{T}}e = m, \ z_{i,j} \in \{0,1\}, \ \ \forall i,j \right\}, \tag{1}$$

where $m = (m_1, m_2)^{\mathrm{T}}$.

3 Overview of SDP relaxations

In this section we provide an overview of existing SDP relaxations for the GBP. The following SDP relaxation is derived in [27]

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathrm{tr}(LX) \\ \text{s.t.} \quad & \mathrm{diag}(X) = e, \ \mathrm{tr}(JX) = m_1^2 + m_2^2 \\ & 2X - J \succeq 0, \ X \in \mathcal{S}_n, \end{aligned} \tag{2}$$

where the 'diag' operator maps an $n \times n$ matrix to the n-vector given by its diagonal, and $\mathcal{S}_n$ denotes the space of $n \times n$ symmetric matrices. Nonnegativity constraints on the matrix variable in (2) are redundant. This follows from $\mathrm{diag}(X) = e$ and $2X - J \succeq 0$, see [3] for details. The SDP relaxation (2) is equivalent to the SDP relaxation with a matrix variable of order n from [15].

The following SDP relaxation for the GBP is derived in [29]:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathrm{tr}(L(Y_{11} + Y_{22})) \\ \text{s.t.} \quad & \mathrm{tr}(Y_{ii}) = m_i, \ \mathrm{tr}(JY_{ii}) = m_i^2, \ \ i = 1,2 \\ & \mathrm{diag}(Y_{12}) = 0, \ \mathrm{tr}J(Y_{12} + Y_{12}^{\mathrm{T}}) = 2m_1m_2 \\ & Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{12}^{\mathrm{T}} & Y_{22} \end{pmatrix} \ \ y = \mathrm{diag}(Y), \ \ Y - yy^{\mathrm{T}} \succeq 0, \ \ Y \geq 0, \end{aligned} \tag{3}$$

where $Y \in \mathcal{S}_{2n}$. From now on, we assume that matrices of order $2n$ have the block structure as given above.

Although the nonnegativity constraints were not included in the relaxation from [29], the authors mentioned that it would be worth adding them. The SDP relaxation (3) does not have strictly feasible solutions. From a computational point of view, this is an indication that the model may be difficult to solve directly as it is. Therefore, Wolkowicz and Zhao [29] derive the Slater feasible version of the relaxation whose matrix variable is of order n. However, that model includes multiplications with projection matrices of size $(2n + 1) \times n$. The above relaxation can be further strengthened by adding the following inequalities

$$0 \leq y_{i,j} \leq y_{i,i} \tag{4}$$

$$y_{i,i} + y_{j,j} \leq 1 + y_{i,j} \tag{5}$$

$$y_{i,k} + y_{j,k} \leq y_{k,k} + y_{i,j} \tag{6}$$

$$y_{i,i} + y_{j,j} + y_{k,k} \leq y_{i,j} + y_{i,k} + y_{j,k} + 1, \tag{7}$$

where $Y = (y_{i,j})$ and $1 \leq i,j,k \leq 2n$, $i \neq j$, $i \neq k$, $j \neq k$. The inequalities (4)–(7) are facet defining inequalities of the boolean quadric polytope (BQP), see [20].

Wolkowicz and Zhao [29] prove that for a matrix Y that is feasible for the SDP relaxation (3) the following is satisfied:

$$Y_{11} + Y_{12} = y_1 e^{\mathrm{T}}, \quad Y_{12}^{\mathrm{T}} + Y_{22} = y_2 e^{\mathrm{T}}, \quad y_1 + y_2 = e, \quad Y_{ii} e = m_i y_i \ (i = 1, 2), \tag{8}$$

where $y_i = \mathrm{diag}(Y_{ii})$ $(i = 1, 2)$. From here it follows that for given Y_{11} and y_1 the above equations uniquely determine Y_{12}, Y_{22} and y_2. We will exploit this to derive the simplified SDP relaxation in the following section.

Extensive numerical results in [27] show that (3) provides the strongest SDP relaxation for the GBP. To the best of our knowledge, we are not aware of numerical test that involve the SDP relaxation (3) and the inequalities (4)–(7).

Finally, we prove that the optimal value of the SDP relaxation (3) is at least that of the relaxation (2).

Proposition 1. *Let $m_1 > m_2$ and $m_1 + m_2 = n$. Then the SDP relaxation* (3) *dominates the SDP relaxation* (2).

Proof. Let Y_{ij} and y_i $(i, j = 1, 2)$ be feasible for (3), and set $X = Y_{11} + Y_{22}$. Now, $\mathrm{tr}(JX) = m_1^2 + m_2^2$ and $\mathrm{diag}(X) = e$ follow from feasibility of Y_{ii} $(i = 1, 2)$ and (8). The SDP constraint follows from summing $\begin{pmatrix} Y_{ii} & y_i \\ y_i^{\mathrm{T}} & 1 \end{pmatrix} \succeq 0$, $i = 1, 2$. □

The similar result is proven in [5]. In particular, it was proven that the QAP-based SDP relaxation for the GBP dominates the SDP relaxation from [15]. However, the QAP-based SDP relaxation for the GBP is equivalent to (3), and the relaxation from [15] to (2), see [27].

4 A simplified SDP relaxation

In this section we derive an SDP relaxation for the GBP with a matrix variable of order n, and prove that it is equivalent to the best known SDP relaxation for general graphs that is derived in [29]. To derive the relaxation we exploit the fact that the variables associated to the two sets in the bisection are related. Namely, variables coming from the assignment to the second set are redundant in the assignment constraints. It is surprising that this observation was not earlier exploited in the context of the GBP. However, a similar idea was used in [22] to derive an SDP relaxation for the vertex separator problem.

Our observation lead us to the following SDP relaxation:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathrm{tr}(L(2X + J - xe^{\mathrm{T}} - ex^{\mathrm{T}})) \\ \text{s.t.} \quad & x^{\mathrm{T}} e = m_1, \quad \mathrm{tr}(JX) = m_1^2, \quad Xe = m_1 x \\ & X \geq 0, \quad xe^{\mathrm{T}} - X \geq 0, \quad J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}} \geq 0 \\ & X \succeq 0, \quad \mathrm{diag}(X) = x, \quad X \in \mathcal{S}_n. \end{aligned} \tag{9}$$

All equality constraints in (9) are related to the variables associated to the set S_1. The constraints $X \geq 0$ ensure that the matrix variable corresponding to S_1 is nonnegative, while constraints $xe^{\mathrm{T}} - X \geq 0$, $J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}} \geq 0$ do the same for the slack matrix variables.

One may wish to replace the semidefinite constraint $X \succeq 0$ from (9) with the in general stronger constraint $X - xx^{\mathrm{T}} \succeq 0$. However, from the following result it follows that in our case those two semidefinite constraints are equivalent.

Proposition 2. ([10], Proposition 7) *Let X be a symmetric matrix of order n such that $c \cdot \mathrm{diag}(X) = Xe$ for some $c \in \mathbf{R}$, and*

$$\bar{X} = \begin{pmatrix} 1 & \mathrm{diag}(X)^{\mathrm{T}} \\ \mathrm{diag}(X) & X \end{pmatrix}.$$

Then the following are equivalent:

(i) $\bar{X}$ *is positive semidefinite,*
(ii) X *is positive semidefinite and* $\mathrm{tr}(JX) \geq (\mathrm{tr}X)^2$.

The equivalence of the two SDP constraints follows from the fact that for a feasible X for (9) one has $\mathrm{tr}(JX) = (\mathrm{tr}X)^2 = m_1^2$ and $Xe = m_1 \mathrm{diag}(X)$. We prove now our main result.

Theorem 3. *Let $m_1+m_2 = n$, $m_1 > m_2$. The SDP relaxations* (3) *and* (9) *are equivalent.*

Proof. Let X be feasible for (9) and $x = \mathrm{diag}(X)$. We construct a feasible Y, $y = \mathrm{diag}(Y)$ for (3) in the following way. Define $y_1 := x$, $y_2 := e - x$, $y^{\mathrm{T}} := (y_1^{\mathrm{T}}, y_2^{\mathrm{T}})$, matrices

$$Y_{11} := X, \quad Y_{22} := J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}}, \;\; Y_{12} := xe^{\mathrm{T}} - X,$$

and collect all blocks into the matrix

$$Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{12}^{\mathrm{T}} & Y_{22} \end{pmatrix} = \begin{pmatrix} X & xe^{\mathrm{T}} - X \\ ex^{\mathrm{T}} - X & J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}} \end{pmatrix}.$$

Now, we first prove that

$$\begin{pmatrix} X & xe^{\mathrm{T}} - X \\ ex^{\mathrm{T}} - X & J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}} \end{pmatrix} - \begin{pmatrix} xx^{\mathrm{T}} & x(e-x)^{\mathrm{T}} \\ (e-x)x^{\mathrm{T}} & (e-x)(e-x)^{\mathrm{T}} \end{pmatrix} \succeq 0.$$

To show this, we rewrite the left hand side of the matrix inequality above as it follows

$$\begin{pmatrix} X - xx^{\mathrm{T}} & xx^{\mathrm{T}} - X \\ xx^{\mathrm{T}} - X & X - xx^{\mathrm{T}} \end{pmatrix}.$$

Now, for arbitrary vectors $z_1, z_2 \in \mathbf{R}^n$ we have

$$(z_1^{\mathrm{T}}, z_2^{\mathrm{T}}) \begin{pmatrix} X - xx^{\mathrm{T}} & xx^{\mathrm{T}} - X \\ xx^{\mathrm{T}} - X & X - xx^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = (z_1 - z_2)^{\mathrm{T}}(X - xx^{\mathrm{T}})(z_1 - z_2) \geq 0,$$

from where it follows the claim. Let us now verify $\mathrm{tr}(JY_{22}) = m_2^2$. Namely,

$$\mathrm{tr}(JY_{22}) = \mathrm{tr}(J(J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}})) = n^2 + m_1^2 - 2nm_1 = m_2^2.$$

Similarly, the remaining constraints from (3) can be verified.

Conversely, let Y be feasible for (3). We set $X = Y_{11}$ and $x = \mathrm{diag}(Y_{11})$. Since every feasible matrix $Y \in \mathcal{S}_{2n}$ for (3) satisfies also (8), feasibility of X follows by direct verification. Finally, it is not difficult to check that the objectives coincide for any pair of feasible solutions (Y, X). □

Note that the result from the previous theorem is also valid when $m_1 = m_2$. However, it was proven in [26] that all known vector and matrix lifting based SDP relaxations for the k-equipartition problem ($k \geq 2$) are equivalent.

It is not difficult to verify that the SDP relaxation (9) has a strictly feasible point. In deed, the following matrix is feasible for (9) and is positive definite:

$$\hat{X} = \frac{m_1}{n} I + \frac{m_1(m_1-1)}{n(n-1)} (J - I),$$

where I is the identity matrix. Note that $\hat{X}$ has two distinct eigenvalues i.e., $\frac{m_1(n-m_1)}{n(n-1)}$ with multiplicity $n-1$, and $\frac{m_1^2}{n}$ with multiplicity one.

The following result is a direct consequence of Theorem 3.

Corollary 4. *The SDP relaxation* (3) *without nonnegativity constraints is equivalent to the SDP relaxation* (9) *without nonnegativity constraints, i.e.,*

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathrm{tr}(L(2X + J - xe^{\mathrm{T}} - ex^{\mathrm{T}})) \\ \text{s.t.} \quad & x^{\mathrm{T}}e = m_1, \quad \mathrm{tr}(JX) = m_1^2, \quad Xe = m_1 x \\ & X \succeq 0, \quad \mathrm{diag}(X) = x, \quad X \in \mathcal{S}_n. \end{aligned}$$

In order to improve the SDP relaxation (9) we can add the facet defining inequalities of the boolean quadric polytope, see [20]. We first note that the inequality constraints $X \geq 0$, $xe^{\mathrm{T}} - X \geq 0$, and $J + X - xe^{\mathrm{T}} - ex^{\mathrm{T}} \geq 0$ from the SDP relaxation (9) are exactly the following BQP constraints

$$0 \leq x_{i,j} \leq x_{i,i}, \quad x_{i,i} + x_{j,j} \leq 1 + x_{i,j}, \quad 1 \leq i,j \leq n, \ i \neq j.$$

Note also that the SDP relaxation from Corollary 4 differs from the SDP relaxation (9) exactly for those constraints. Thus, in order to strengthen the SDP relaxation (9) one can add the following BQP constraints:

$$x_{i,k} + x_{j,k} \leq x_{k,k} + x_{i,j}, \quad x_{i,i} + x_{j,j} + x_{k,k} \leq x_{i,j} + x_{i,k} + x_{j,k} + 1, \tag{10}$$

for $1 \leq i,j,k \leq n$, $i \neq j$, $i \neq k$, $j \neq k$.

Let us now show that the bound obtained by solving the SDP relaxation (3) with additional BQP constraints (4)–(7) is equal to the bound obtained by solving (9) with additional constraints (10).

We first prove that (4)–(5) are redundant for feasible matrices from (3).

Lemma 5. *Let* $Y = (y_{i,j}) \in \mathcal{S}_{2n}$ *be feasible for* (3)*. Then, the following BQP inequalities are satisfied:*

$$0 \leq y_{i,j} \leq y_{i,i}, \quad y_{i,i} + y_{j,j} \leq 1 + y_{i,j},$$

for $1 \leq i,j \leq 2n$, $i \neq j$.

Proof. The constraints $0 \leq y_{i,j} \leq y_{i,i}$ $(1 \leq i,j \leq 2n)$ follow trivially from (8) and $Y \geq 0$. To show that the following inequalities

$$y_{i,i} + y_{j,j} \leq 1 + y_{i,j}, \quad 1 \leq i,j \leq n, \ i \neq j,$$

are satisfied, it is instructive to look at (8). From (8) we have that $Y_{22} = J + Y_{11} - y_1 e^{\mathrm{T}} - e y_1^{\mathrm{T}}$. Now, from $Y_{22} \geq 0$ it follows that $J + Y_{11} \geq y_1 e^{\mathrm{T}} + e y_1^{\mathrm{T}}$, from where it is clear that the above constraints are satisfied. To verify the inequalities

$$y_{i,i} + y_{j,j} \leq 1 + y_{i,j}, \quad 1 \leq i \leq n, \quad n+1 \leq j \leq 2n,$$

we rewrite $y_{j,j}$ $(n+1 \leq j \leq 2n)$ in the terms of the elements from the first block by using $Y_{22} = J + Y_{11} - y_1 e^{\mathrm{T}} - e y_1^{\mathrm{T}}$. Similarly, we rewrite $y_{i,j}$ $(1 \leq i \leq n,\ n+1 \leq j \leq 2n)$ using $Y_{12} = y_1 e^{\mathrm{T}} - Y_{11}$. Now, the above inequalities reduce to the redundant constraints $y_{k,k} \geq y_{k,i}$ $(1 \leq i, k \leq n)$.

Similarly, one can show that $y_{i,i} + y_{j,j} \leq 1 + y_{i,j}$ for $n+1 \leq i, j \leq 2n$, and $n+1 \leq i \leq 2n$, $1 \leq j \leq n$. □

In the previous lemma we exploit the fact that a feasible matrix for (3) satisfies (8). By using the same equations, we can show that for a feasible Y from (3) many of the constraints (6)–(7) are equivalent. In particular, let $Y = (y_{i,j}) \in \mathcal{S}_{2n}$ be feasible for (3). Then the QBP constraints (6)–(7) for the elements in the blocks Y_{12}, Y_{12}^{T} and Y_{22} can be reformulated into the QBP constraints (6)–(7) for the elements in the block Y_{11}. For example, to reformulate the inequalities

$$y_{i,k} + y_{j,k} \leq y_{k,k} + y_{i,j}, \quad 1 \leq i \leq n,\ n+1 \leq j, k \leq 2n,\ j \neq k,$$

we exploit $Y_{22} = J + Y_{11} - y_1 e^{\mathrm{T}} - e y_1^{\mathrm{T}}$ to rewrite $y_{j,k}$ and $y_{k,k}$. We also use $Y_{12} = y_1 e^{\mathrm{T}} - Y_{11}$ to rewrite $y_{i,k}$ and $y_{i,j}$, which leads to the inequalities

$$y_{i,j} + y_{k,j} \leq y_{j,j} + y_{i,k}, \quad 1 \leq i, j, k \leq n. \tag{11}$$

Similarly, to reformulate the inequalities

$$y_{i,i} + y_{j,j} + y_{k,k} \leq y_{i,j} + y_{i,k} + y_{j,k} + 1, \quad n+1 \leq i, j, k \leq 2n,\ j \neq k \neq i,$$

we exploit $Y_{22} = J + Y_{11} - y_1 e^{\mathrm{T}} - e y_1^{\mathrm{T}}$ to rewrite all y-variables. This results with the constraints:

$$y_{i,i} + y_{j,j} + y_{k,k} \leq y_{i,j} + y_{i,k} + y_{j,k} + 1, \quad 1 \leq i, j, k \leq n,\ j \neq k \neq i. \tag{12}$$

Continuing in a similar way, we get that the constraints (6)–(7) can be reduced to the constraints (11)–(12), i.e., to the same inequalities for the smaller index set. We summarize the previous results in the following theorem.

Theorem 6. *The SDP relaxation* (9) *with additional constraints* (10) *is equivalent to the SDP relaxation* (3) *with additional BQP constraints* (4)*–*(7)*.*

Thus, this paper presents reformulated and simplified the strongest SDP relaxation for the the bisection problem, and suggest its strengthening. In the following section we test our simplified and strengthened SDP relaxation on several graphs from the literature.

5 Numerical results

In this section we present numerical results that verify the quality of the SDP relaxation (9), as well as the relaxation obtained after adding the BQP constraints (10) to (9). All relaxations were solved with Mosek [19] using the Yalmip interface [18] on an Intel Xeon, E5-1620, 3.70 GHz with 32 GB memory.

The instances we use belong to the various classes of graphs from the literature. In particular, in Table 1 and Table 2 we consider the following graphs.

- `compiler design instances` were introduced in [14]. We denote them by `cd.xx.yy`.

- `kkt instances` originate from nested dissection approaches for solving sparse symmetric linear systems, see [12]. We denote them by `kkt_name`.
- `mesh instances` come from an application of the finite element methods, see [28]. We denote them with the initials `mesh.xx.yy`.
- `VLSI design instances` are derived from data in the layout of electronic circuits. For details see [7]. We denote them with the initials `vlsi.xx.yy`.

In the above instances `xx` denotes the number of vertices, and `yy` the number of edges in the graph. Table 1 reads as follows. In the first three columns, we list the graphs, number of vertices in the graph, and corresponding m, respectively. In the fourth to six column we present the SDP bounds (2), (9), and the SDP bound (9) with additional BQP constraints (10), respectively. Bounds in the column six are obtained by adding the most violated inequalities of type (10) to the SDP relaxation (9). The cutting plane scheme adds at most $2n$ violated valid constraints in each iteration and performs at most 20 iterations. In the last column of Table 1 we list upper bounds obtained by a tabu search heuristics, see also [23].

All lower bounds in Table 1 are rounded up to the closest integer. Note that for only three out of twenty-one instances we can not prove optimality.

Table 1: Computational results for the bisection problem.

instance	$\|V\|$	m^{T}	(2)	(9)	(9)+(10)	u.b.
cd.30.47	30	$(20, 10)$	110	114	114	114
cd.30.56	30	$(20, 10)$	156	169	169	169
cd.45.98	45	$(25, 20)$	576	631	631	631
cd.47.99	47	$(25, 22)$	471	514	537	537
cd.47.101	47	$(25, 22)$	326	361	382	382
cd.61.187	61	$(40, 21)$	774	798	798	798
kkt_lowt01	82	$(42, 40)$	5	5	13	13
kkt_putt01	115	$(59, 56)$	20	22	28	29
mesh.35.54	35	$(22, 13)$	2	3	4	4
mesh.69.212	69	$(40, 29)$	2	2	4	4
mesh.70.120	70	$(50, 20)$	2	4	6	6
mesh.74.129	74	$(70, 4)$	1	4	4	4
mesh.137.231	137	$(100, 37)$	1	3	6	6
mesh.148.265	148	$(120, 28)$	1	5	6	6
vlsi.34.71	34	$(22, 12)$	4	6	6	6
vlsi.37.92	37	$(30, 7)$	3	6	6	6
vlsi.38.105	38	$(20, 18)$	84	86	110	110
vlsi.42.132	42	$(20, 22)$	97	99	120	120
vlsi.48.81	48	$(40, 8)$	4	12	12	18
vlsi.166.504	166	$(100, 66)$	12	23	24	24
vlsi.170.424	170	$(100, 70)$	35	37	37	48

In Table 2 we list the computational times required for solving the SDP relaxations and instances from Table 1. In the same table we include the computational times for solving the Slater feasible version of the SDP relaxation (3), see [29]. We do not compute the SDP bound (9)+(10), if (9) provides an optimal solution. In such cases, we write 'n.a.'. Since the computational times for computing each of the lower bounds for cd.30.47, cd.30.56, cd.45.98 are below one second, we omit these results from the table.

Table 2 shows that there is only a marginal time difference for solving (2), (9), and (3) for graphs up to 61 vertices. The results show that for graphs with more than 61 vertices, the SDP relaxation (2) requires noticeable less computational effort than the other relaxations. Computational times in Table 2 verify that there is an advantage in solving the relaxation (9) for larger graphs ($n > 115$) than solving (3).

Table 2 indicates that sometimes the running time for solving (9)+(10) is a few times longer than the running time for solving (9). For example, we compute the bound (9) for vlsi.166.504 ($n = 166$) in 892 s, and (9)+(10) in 2079 s. See also results for cd.47.101, mesh.148.265. However, the difference between the two running times can be significant. In particular, we compute the SDP bound (9) for kkt_putt01 ($n = 115$) in 106 seconds, and (9)+(10) in 21 minutes. (To approximately solve the SDP relaxation (3) + (4)–(7) using the cutting plane schema for kkt_putt01 it takes about 47 minutes). In general, the computational time for solving (9)+(10) depends on the number of the violated BQP constraints (10) after each iteration of the cutting plane scheme.

Table 2: Computational times in seconds for the bisection problem.

instance	$\|V\|$	m^{T}	(2)	(9)	(3)	(9)+(10)
cd.47.99	47	$(25, 22)$	1	2	2	7
cd.47.101	47	$(25, 22)$	1	2	3	6
cd.61.187	61	$(40, 21)$	3	4	4	n.a.
kkt_lowt01	82	$(42, 40)$	10	21	21	442
kkt_putt01	115	$(59, 56)$	66	106	114	1289
mesh.35.54	35	$(22, 13)$	1	1	1	3
mesh.69.212	69	$(40, 29)$	6	9	11	83
mesh.70.120	70	$(50, 20)$	4	12	9	55
mesh.74.129	74	$(70, 4)$	7	12	14	n.a.
mesh.137.231	137	$(100, 37)$	167	296	345	6574
mesh.148.265	148	$(120, 28)$	241	550	566	990
vlsi.38.105	38	$(20, 18)$	1	1	1	15
vlsi.42.132	42	$(20, 22)$	1	1	1	20
vlsi.48.81	48	$(40, 8)$	1	2	2	11
vlsi.166.504	166	$(100, 66)$	371	892	1081	2079
vlsi.170.424	170	$(100, 70)$	486	943	1252	7789

In [3], the authors strengthened the SDP relaxations (2) and (3) by adding two constraints that correspond to assigning two vertices of the graph to different parts of the partition. In particular, they show that such strengthening performs well on highly symmetric graphs when other relaxations provide weak or trivial bounds. In [3], it was also shown how to aggregate the triangle and independent set constraints for highly symmetric graphs in order to add them to the SDP relaxation (2). Our numerical results show that the SDP relaxation (9) with additional inequalities (10) provides bounds that are competitive to those from [3].

In particular, in Table 3 we list bounds for highly symmetric graphs considered in [3]. `Pappus`, `Desargues`, and `Biggs-Smith` graphs are distance-regular graphs, $J(7,2)$ is the Johnson graph. The first three columns in Table 3 read similar as the first three columns in Table 1. In the fourth (resp. sixth) column we list values of the SDP bound (9) (resp. bound (9) with additional inequalities (10)) for different graphs. The fifth column of Table 3 lists the best obtained bounds from [3]; that is for the `Pappus` graph the relaxation (2) with all triangle inequalities, for `Desargues` the relaxation (3) with constraints that fix two vertices of the graph, for $J(7,2)$ the relaxation (2) with independent set inequalities, and for `Biggs-Smith` the relaxation (2) with all triangle inequalities.

Table 3 shows that our new relaxation (9)+(10) provides lower bounds that are competitive with other bounding approaches known for highly symmetric graphs. Although most of the bounds presented in [3] can be solved within a few seconds, it is not clear a priori which of the bounding approaches should be implemented for a given graph. However, the SDP relaxation (9)+(10) provides lower bounds that equal the best bounds among all approaches studied in [3]. To compute the SDP bound (9) for `Biggs-Smith` it takes 22 s, and to compute the bound (9)+(10) for $J(7,2)$ (resp. `Biggs-Smith`) it takes 4 s (resp. 773 s). All other bounds in Table 3 are obtained within a second. To compute the SDP bound (9) for highly symmetric graphs we didn't exploit symmetry reduction as described in [3] although this can be done in a similar way. By doing as described in [3], one can compute SDP bounds (9) from Table 3 very fast. The interested reader is invited to verify this.

Table 3: Bounds for the bisection on highly symmetric graphs.

G	$\lvert V\rvert$	m^{T}	(9)	b.b. [3]	(9)+(10)	u.b.
`Pappus`	18	$(10,8)$	6	7	7	8
`Desargues`	20	$(15,5)$	5	6	6	7
$J(7,2)$	21	$(11,10)$	37	40	40	40
`Biggs-Smith`	102	$(70,32)$	10	15	15	18

6 Conclusion

In this paper we present an SDP relaxation for the graph bisection problem with a matrix variable of order n, where n is the order of the graph. To derive our relaxation we exploit the fact that variables corresponding to one set in the bisection uniquely determine variables of the other set. We prove that our relaxation is equivalent to the strongest known SDP relaxation for general graphs that is obtained by using vector lifting. This result is in the line of the similar results for some other optimization problems. Namely, for the graph equipartition problem there exists an SDP relaxation with a matrix variable of order equal to the order of the graph, which is equivalent to the strongest vector lifting-based SDP relaxation, see [26].

To strengthen our SDP relaxation we add facet defining inequalities of the boolean quadric polytope, which enables us to compute strongest SDP bounds for the GBP and for graphs with $n \leq 200$ vertices in reasonable time.

Since our relaxation has strictly feasible solutions it can be directly solved as it is, which makes it attractive for a branch and bound framework. However, this will be part of our future research.

Acknowledgements: The author would like to thank two anonymous referees for suggestions that led to an improvement of this paper.

References

[1] Biswas, R., Hendrickson, B., & Karypis, G.: Graph partitioning and parallel computing. *Parallel Comput.*, 26(12), 1515–1517 (2000)

[2] Dai, W., & Kuh, E.: Simultaneous floor planning and global routing for hierarchical building-block layout. IEEE Trans. Comput.-Aided Des. *Integrated Circuits* & *Syst.*, CAD-6, 5, 828–837 (1987)

[3] van Dam, E.R., & Sotirov, R.: Semidefinite programming and eigenvalue bounds for the graph partition problem, *Math. Program., Ser B*, 151(2), 379–404 (2015)

[4] De Klerk, E., Pasechnik, D.V., Sotirov, R., & Dobre, C.: On semidefinite programming relaxations of maximum k-section. *Math. Program. Ser. B*, 136(2), 253–278 (2012)

[5] De Klerk, E., de Oliveira Filho, F.M., & Pasechnik, D.V.: Relaxations of combinatorial problems via association schemes, in *Handbook of Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*, Miguel Anjos and Jean Lasserre (eds.), pp. 171–200, Springer (2012)

[6] Feige, U., & Langberg, M.: Approximation algorithms for maximization problems arising in graph partitioning. *J. Algorithm*, 41, 174–211 (2001)

[7] Ferreira, C.E., Martin, A., de Souza, C.C., Weismantel, R., & Wolsey, L.A.: The node capacitated graph partitioning problem: a computational study. *Math. Program.*, 81, 229–256 (1998)

[8] Fiduccia, C.M., & Mattheyses, R.M.: A linear-time heuristic for improving network partitions. Proceedings of the 19th Design Automation Conference, 175–181 (1982)

[9] Garey, M.R., Johnson, D.S., & Stockmeyer, L.: Some simplified NP-complete graph problems. *Theoret. Comput. Sci.* 1(3), 237-267 (1976)

[10] Gijswijt, D.: Matrix algebras and semidefinite programming techniques for codes. PhD thesis, University of Amsterdam, The Netherlands (2005)

[11] Han, Q., Ye, Y., & Zhang, J.: An improved rounding method and semidefinite relaxation for graph partitioning. *Math. Program.*, 92, 509–535 (2002)

[12] Helmberg, C.: A cutting plane algorithm for large scale semidefinite relaxations. In: *Padberg Festschrift The Sharpest Cut*, M. Grötschel (Ed.). MPS-SIAM 233–256 (2004)

[13] Hendrickson, B., & Kolda, T.G.: Partitioning rectangular and structurally nonsymmetric sparse matrices for parallel processing. *SIAM J. Sci. Comput.*, 21(6), 2048–2072 (2000)

[14] Johnson, E., Mehrotra, A., & Nemhauser, G.: Min-cut clustering. *Math. Program.*, 62, 133–152 (1993)

[15] Karisch, S.E., Rendl, F., & Clausen, J.: Solving graph bisection problems with semidefinite programming. *INFORMS J. Comput.*, 12, 177–191 (2000)

[16] Lengauer, T.: *Combinatorial algorithms for integrated circuit layout.* Wiley, Chicester (1990)

[17] Li, M., Andersen, D.G., & Smola, A.J.: Graph partitioning via parallel submodular approximation to accelerate distributed machine learning, `arXiv:1505.04636v1` (2015)

[18] Löfberg, J.: YALMIP: A toolbox for modeling and optimization in MATLAB. *Proc. of the CACSD Conference, Taipei, Taiwan*, 284–289 (2004) `http://users.isy.liu.se/johanl/yalmip/`

[19] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28) (2015) `http://docs.mosek.com/7.1/toolbox/index.html`

[20] Padberg, M.W.: The boolean quadric polytope: Some characteristics, facets and relatives. *Math. Program.*, 45, 139–172 (1989)

[21] Pirim, H., Ekçioğlu, B., Perkins, A., & Yüceer, Ç.: Clustering of high throughput gene expression data. *Comput Oper Res.*, 39(12), 3046-3061 (2012)

[22] Rendl, F., & Sotirov, R.: The min-cut and vertex separator problem, Preprint (2016)

[23] Rolland, E., Pirkul, H., & Glover, F.: Tabu search for graph partitioning. *Annals of Operations Research*, 63(2), 209-232 (1996)

[24] Sanchis, L.: Multiple-way network partitioning. *IEEE Trans. Comput.*, 38, 62–81, (1989)

[25] Simon, H.D.: Partitioning of unstructured problems for parallel processing. *Comput. Syst. Eng.*, 2, 35–148 (1991)

[26] Sotirov, R.: SDP relaxations for some combinatorial optimization problems. In M.F. Anjos and J.B. Lasserre (Eds.), *Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications*, 795–820 (2012)

[27] Sotirov, R.: An efficient semidefinite programming relaxation for the graph partition problem, *INFORMS J. Comput.*, 26(1), 16–30, (2014)

[28] de Souza, C.C., Keunings, R., Wolsey, L.A., & Zone, O.: A new approach to minimizing the frontwidth in finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 111, 323-334 (1994)

[29] Wolkowicz, H., & Zhao, Q.: Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97, 461-479 (1999)