# A Low Dimensional Semidefinite Relaxation for the Quadratic Assignment Problem *

Yichuan Ding[†] Henry Wolkowicz[‡]

April 19, 2008

University of Waterloo
Department of Combinatorics & Optimization
Waterloo, Ontario N2L 3G1, Canada
Research Report CORR 2006-22

**Abstract**

The quadratic assignment problem (***QAP***) is arguably one of the hardest of the NP-hard discrete optimization problems. Problems of dimension greater than 25 are still considered to be large scale. Current successful solution techniques use branch and bound methods, which rely on obtaining *strong and inexpensive* bounds.

In this paper we introduce a new semidefinite programming (***SDP***) relaxation for generating bounds for the ***QAP*** in the trace formulation $\min_{X\in\Pi} \operatorname{trace} AXBX^T + CX^T$. We apply majorization to obtain a relaxation of the orthogonal similarity set of the matrix $B$. This exploits the matrix structure of ***QAP*** and results in a relaxation with $O(n^2)$ variables, a much smaller dimension than other current ***SDP*** relaxations. We compare the resulting bounds with several other computationally inexpensive bounds, such as the convex quadratic programming relaxation (***QPB***). We find that our method provides stronger bounds on average and is adaptable for branch and bound methods.

**Keywords:** Quadratic assignment problem, semidefinite programming relaxations, interior-point methods, large scale problems.

---

*This report is available with URL: http://orion.math.uwaterloo.ca/~hwolkowi/reports/ABSTRACTS.html

[†]Research supported by The Natural Sciences and Engineering Research Council of Canada. Email y7ding@stanford.edu. (This author is currently at Stanford University, Department of Management Science and Engineering.)

[‡]Research supported by The Natural Sciences and Engineering Research Council of Canada. Email hwolkowicz@uwaterloo.ca

# Contents



# List of Tables



# 1 Introduction

In this paper we introduce a new efficient bound for the Quadratic Assignment Problem (***QAP***). We use the *Koopmans-Beckmann* trace formulation

$$(\boldsymbol{QAP}) \quad v_{\mathbf{QAP}} := \min_{X \in \Pi} \ \operatorname{trace} AXBX^T + CX^T,$$

where $A$, $B$ and $C$ are $n$ by $n$ real matrices, and $\Pi$ denotes the set of $n$ by $n$ permutation matrices. Throughout this paper we assume the symmetric case, i.e., that both $A$ and $B$ are symmetric matrices. The ***QAP*** is considered to be one of the hardest NP-hard problems to solve in practice. However, many important combinatorial optimization problems can be formulated as a ***QAP***. Examples include: the traveling salesman problem, VLSI design, keyboard design, and the graph partitioning problem. The ***QAP*** is well described by the problem of allocating a set of $n$ facilities to a set of $n$ locations while minimizing the quadratic objective arising from the distance between the locations in combination with the flow between the facilities. Recent surveys include [28, 32, 34, 27, 19, 12, 13, 11, 31].

Solving ***QAP*** to optimality usually requires a branch and bound (***B&B***) method. Essential for these methods are strong, inexpensive bounds at each node of the branching tree. In this paper, we study a new bound obtained from a semidefinite programming (***SDP***) relaxation. This relaxation uses only $O(n^2)$ variables and $O(n^2)$ constraints. But, it yields a bound provably better than the so-called *projected eigenvalue bound* (***PB***) [12], and it is competitive with the recently introduced *quadratic programming bound* (***QPB***), [1].

## 1.1 Outline

In Section 1.2 we continue with preliminary results and notation. In Section 1.3 we review some of the known bounds in the literature. Our main results appear in Section 2. Here we compare relaxations that use a *vector lifting* of the matrix $X$ into the space of $n^2 \times n^2$ matrices with a *matrix lifting* that remains in $\mathcal{S}^n$, the space of $n \times n$ symmetric matrices. We then parametrize and characterize the *orthogonal similarity set of* $B$, $\mathcal{O}(B)$, using majorization results on the eigenvalues of $B$, see Theorem 2.1. This results in three ***SDP*** relaxations, $\mathbf{MSDR}_1$ to $\mathbf{MSDR}_3$. We conclude with numerical tests in Section 3.

## 1.2 Notation and Preliminaries

For two real $m \times n$ matrices $A, B \in \mathcal{M}^{mn}$, $\langle A, B\rangle = \operatorname{trace} A^T B$ is the trace inner product; $\mathcal{M}^{nn} = \mathcal{M}^n$, denotes the set of $n$ by $n$ square real matrices; $\mathcal{S}^n$ denotes the space of $n \times n$ symmetric matrices, while $\mathcal{S}^n_+$ denotes the cone of positive semidefinite matrices in $\mathcal{S}^n$. We let $A \succeq B$ denote the Löwner partial order, $A - B \in \mathcal{S}^n_+$.

The linear transformation $\operatorname{diag} M$ denotes the vector formed from the diagonal of the matrix $M$; the adjoint linear transformation is $\operatorname{diag}^* v = \operatorname{Diag} v$, i.e., the diagonal matrix formed from the vector $v$. We use $A \otimes B$ to denote the Kronecker product of $A$ and $B$, and use $x = \operatorname{vec}(X)$ to denote the vector in $\mathbb{R}^{n^2}$ obtained from the columns of $X$. Then, see e.g., [17],

$$\operatorname{trace} AXBX^T = \langle AXB, X\rangle = \langle \operatorname{vec}(AXB), x\rangle = x^T(B \otimes A)x. \tag{1.1}$$

We let $\mathcal{N}$ denote the cone of nonnegative (elementwise) matrices, $\mathcal{N} := \{X \in \mathcal{M}^n : X \geq 0\}$; $\mathcal{E}$ denotes the set of matrices with row and column sums 1, $\mathcal{E} := \{X \in \mathcal{M}^n : Xe = X^T e = e\}$, where $e$ is the vector of ones; $\mathcal{D}$ denotes the set of doubly stochastic matrices, $\mathcal{D} = \mathcal{E} \cap \mathcal{N}$. The *minimal product* of two vectors is

$$\langle x, y\rangle_- := \min_{\sigma,\pi} \sum_{i=1}^n x_{\sigma(i)} y_{\pi(i)},$$

where the minimum is over all permutations, $\sigma, \pi$, of the indices $\{1, 2, \ldots, n\}$. Similarly, we define the maximal product of $x, y$, $\langle x, y\rangle_+ := \max_{\sigma,\pi} \sum_{i=1}^n x_{\sigma(i)} y_{\pi(i)}$. We denote the vector of eigenvalues of a matrix $A$ by $\lambda(A)$.

**Definition 1.1** *Let* $x, y \in \mathbb{R}^n$. *By abuse of notation, we denote* $x$ majorizes $y$ *or* $y$ is majorized by $x$ *with* $x \succeq y$ *or* $y \preceq x$. *Let the components of both vectors be sorted in* *nonincreasing order, i.e.,* $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \ldots \geq x_{\sigma(n)}$*,* $y_{\pi(1)} \geq y_{\pi(2)} \geq \ldots \geq y_{\pi(n)}$*. Following e.g., [21],* $x \succeq y$ *if and only if*

$$\begin{array}{rcll} \sum_{i=1}^p x_{\sigma(i)} & \geq & \sum_{i=1}^p y_{\pi(i)}, & p = 1, 2, \ldots, n-1, \\ \sum_{i=1}^n x_{\sigma(i)} & = & \sum_{i=1}^n y_{\pi(i)}. & \end{array}$$

In [21], it is shown that $x \succeq y$ if and only if there exists $S \in \mathcal{D}$ with $Sx = y$. Note that for fixed $y$, the constraints $x \succeq y$ is not a convex constraint; but $x \preceq y$ is a convex constraint and it has an equivalent LP formulation, e.g., [16].

## 1.3 Known Relaxations for QAP

One of the earliest and least expensive relaxations for ***QAP*** is based on a Linear Programming (***LP***) formulation, e.g., Gilmore-Lawler(GLB) [10, 7]; related dual-based ***LP*** bounds are ***KCCEB*** in e.g., [18, 29, 7, 15]. These inexpensive formulations are able to handle problems with $n$ with $n$ approximately 20 [10, 20]. However, the bounds are usually too weak to handle larger problems. Strengthened bounds are based on: eigenvalue and parametric eigenvalue bounds (***EB***) [9, 31]; projected eigenvalue bounds ***PB*** [12, 8]; convex quadratic programming bounds ***QPB*** [1]; and ***SDP*** bounds [30, 34]. For recent numerical results that use these bounds, see e.g., [1, 30].

Note that $\Pi = \mathcal{O} \cap \mathcal{E} \cap \mathcal{N}$, i.e. the addition of the orthogonal constraints changes the doubly stochastic matrices to permutation matrices. This illustrates the power of nonlinear quadratic constraints for ***QAP***. Using the quadratic constraints, we can see that ***SDP*** arises naturally from Lagrangian relaxation, see e.g., [25]. Alternatively, one can *lift* the problem using the positive semidefinite matrix $\begin{pmatrix} 1 \\ \mathrm{vec}\,(X) \end{pmatrix} \begin{pmatrix} 1 \\ \mathrm{vec}\,(X) \end{pmatrix}^T$ into the symmetric matrix space $\mathcal{S}^{n^2+1}$. One then obtains deep cuts for the convex hull of the lifted permutation matrices. However, this vector-lifting ***SDP*** relaxation requires $O(n^4)$ variables and hence is expensive to use. Problems with $n > 25$ become impractical for branch and bound methods.

Recently, Anstreicher and Wolkowicz [3] proved that strong (Lagrangian) duality holds for the following quadratic program with orthogonal constraints, i.e.,

$$v_{\mathbf{EB}} = \min_{XX^T = X^TX = I} \mathrm{trace}\,(AXBX^T),$$

whose optimal value gives the eigenvalue bound ***EB***. The Lagrangian dual is

$$v_{\mathbf{EB}} = \max_{S,T \in \mathcal{S}^n} \min_{x \in \mathbb{R}^{n^2}} \{\mathrm{trace}\,(S) + \mathrm{trace}\,(T) + x^T(B \otimes A - I \otimes S - T \otimes I)x\}. \tag{1.2}$$

The inner minimization problem results in the *hidden semidefinite constraint*

$$B \otimes A - I \otimes S - T \otimes I \succeq 0.$$

Under this constraint, the inner minimization program is attained at $x = 0$. As a result of strong duality, the equivalent dual program

$$v_{\mathbf{EB}} = \max_{S,T \in \mathcal{S}^n} \{\text{trace}\,(S) + \text{trace}\,(T) : B \otimes A - I \otimes S - T \otimes I \succeq 0\} \qquad (1.3)$$

has the same value as the primal program, i.e. both yield $\boldsymbol{EB}$. In [1], the authors use the optimal solutions from (1.3), add linear constraints $Xe = X^T e = e$ and $X \geq 0$, and set $S = S^*, T = T^*$ in the inner minimization problem in (1.2). This results in a (parametric) convex quadratic programming bound ($\boldsymbol{QPB}$), i.e.,

$$(\boldsymbol{QPB}) \qquad \begin{array}{rl} v_{\mathbf{QPB}}(S^*, T^*) := \min_X & \text{trace}\, S^* + \text{trace}\, T^* \\ & +\text{vec}\,(X)^T (B \otimes A - I \otimes S^* - T^* \otimes I)\text{vec}\,(X) \\ \text{s.t.} & X \geq 0, Xe = X^T e = e. \end{array}$$

$\boldsymbol{QPB}$ is inexpensive to compute and, under some mild assumptions, strictly stronger than $\boldsymbol{PB}$. Currently, $\boldsymbol{QPB}$ is arguably the most competitive bound, if we take into account the trade-off between the quality of the bound and the expense in the computation. The use of $\boldsymbol{QPB}$, along with the Condor high-throughput computing system, has resulted in the solution for the first time of several large $\boldsymbol{QAP}$ problems from the $\boldsymbol{QAP}$LIB library, [6], [1], [2].

In this paper, we propose a new relaxation for $\boldsymbol{QAP}$, which has comparable complexity to $\boldsymbol{QPB}$. Moreover, our numerical tests show that this new bound obtains better bounds than $\boldsymbol{QPB}$ when applied to problem instances from the $\boldsymbol{QAP}$LIB library.

# 2 SDP Relaxation and Quadratic Matrix Programming

## 2.1 Vector Lifting SDP Relaxations, VSDR

Consider the following quadratic constrained quadratic program

$$(\boldsymbol{QCQP}) \qquad \begin{array}{rll} v_{\mathbf{QCQP}} := & \min & (x^T Q_0 x + c_0^T x) + \beta_0 \\ & \text{s.t.} & (x^T Q_j x + c_j^T x) + \beta_j \leq 0, \quad j = 1, \ldots, m \\ & & x \in \mathbb{R}^n, \end{array}$$

where for all $j$, we have $Q_j \in \mathcal{S}^n, c_j \in \mathbb{R}^n, \beta_j \in \mathbb{R}$. To find approximate solutions to $\boldsymbol{QCQP}$, one can homogenize the quadratic functions to get the equivalent quadratic forms $q_j(x, x_0) = x^T Q_j x + c_j^T x x_0 + \beta_j x_0^2$, along with the additional constraint $x_0^2 = 1$. The homogenized forms can be linearized using the vector $\begin{pmatrix} x_0 \\ x \end{pmatrix} \in \mathbb{R}^{n+1}$, i.e.,

$$\begin{array}{rcl} q_j(x,x_0) & = & \begin{pmatrix} x_0 \\ x \end{pmatrix}^T \begin{pmatrix} \beta_j & \frac{1}{2}c_j^T \\ \frac{1}{2}c_j & Q_j \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix} \\ & = & \text{trace} \begin{pmatrix} \beta_j & \frac{1}{2}c_j^T \\ \frac{1}{2}c_j & Q_j \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix}^T \\ & = & \text{trace} \begin{pmatrix} \beta_j & \frac{1}{2}c_j^T \\ \frac{1}{2}c_j & Q_j \end{pmatrix} \begin{pmatrix} x_0^2 & x_0x^T \\ x_0x & xx^T \end{pmatrix} \\ & = & \text{trace} \begin{pmatrix} \beta_j & \frac{1}{2}c_j^T \\ \frac{1}{2}c_j & Q_j \end{pmatrix} \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \end{array} \tag{2.4}$$

where $\begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} = \begin{pmatrix} x_0 \\ x \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix}^T \succeq 0$. Here $Y$ represents $xx^T$ and the constraint $Y = xx^T$ is relaxed to $xx^T \preceq Y$. Equivalently, we can use the Schur complement and get the lifted linear constraint

$$Z = \begin{pmatrix} 1 & x^T \\ x & Y \end{pmatrix} \succeq 0, \tag{2.5}$$

i.e. we can identify $y = x$. The objective function is now linear, trace $\begin{pmatrix} \beta_0 & \frac{1}{2}c_0^T \\ \frac{1}{2}c_0 & Q_0 \end{pmatrix} Z \leq 0$; and the constraints in $\boldsymbol{QCQP}$ are relaxed to linear inequality constraints, trace $\begin{pmatrix} \beta_j & \frac{1}{2}c_j^T \\ \frac{1}{2}c_j & Q_j \end{pmatrix} Z \leq 0, j = 1, \ldots, m$. In this paper, we call this a *vector-lifting semidefinite relaxation, (*$\boldsymbol{VSDR}$*)*, and we note that the unknown variable $Z \in \mathcal{S}^{n+1}$.

## 2.2 Matrix Lifting SDP Relaxation, MSDR

Consider $\boldsymbol{QCQP}$ with matrix variables

$$(\boldsymbol{MQCQP}) \qquad \begin{array}{rll} v_{\mathbf{MQCQP}} := & \min & \text{trace}\,(X^TQ_0X + C_0X^T) + \beta_0 \\ & \text{s.t.} & \text{trace}\,(X^TQ_jX + C_jX^T) + \beta_j \leq 0, \quad j = 1, \ldots, m \\ & & X \in \mathcal{M}^{mn}. \end{array}$$

Let: $x := \text{vec}\,(X)$, $c := \text{vec}\,(C)$, $\delta_{ij}$ denote the Kronecker delta, and $E_{ij} = e_ie_j^T \in \mathcal{M}^n$ be the zero matrix except with 1 at the $(i,j)$ position. Note that the orthogonality constraint $XX^T = I$ is equivalent to $x^T(I \otimes E_{ij})x = \delta_{ij}, \forall i,j$; and $X^TX = I$ is equivalent to $x^T(E_{ij} \otimes I)x = \delta_{ij}, \forall i,j$. Using *both* of the redundant constraints $XX^T = I$ and $X^TX = I$ strengthens the $\boldsymbol{SDP}$ relaxation, see [3]. We can now rewrite $\boldsymbol{QAP}$ using the Kronecker product and see that it is a special case of $\boldsymbol{MQCQP}$ with linear and quadratic equality constraints, and with nonnegativity constraints, i.e.,

$$\begin{array}{rlll} v_{\mathbf{QAP}} = & \min & x^T(B \otimes A)x + c^Tx & \\ & \text{s.t.} & x^T(I \otimes E_{ij})x = \delta_{ij}, & \forall i,j \\ & & x^T(E_{ij} \otimes I)x = \delta_{ij}, & \forall i,j \\ & & Xe = X^Te = e & \\ & & x \geq 0. & \end{array} \tag{2.6}$$

Note that $x = \mathrm{vec}\,(X)$ from (2.6) is in $\mathbb{R}^{n^2}$. Relaxing the quadratic objective function and/or the quadratic orthogonality constraints results in a linearized/lifted constraint (2.5). This uses the relaxation $Y \succeq xx^T$, and we end up with $Z = \begin{pmatrix} 1 & x^T \\ x & Y \end{pmatrix} \in \mathcal{S}^{n^2+1}$, a prohibitively *large* matrix. However, we can use a different approach and exploit the structure of the problem. We can replace the constraint $y = xx^T$ with the constraint $Y = XX^T$ and then relax it to $Y \succeq XX^T$. This is equivalent to the linear semidefinite constraint $\begin{pmatrix} I & X^T \\ X & Y \end{pmatrix} \succeq 0$. The size of this constraint is significantly smaller. We call this a *matrix-lifting semidefinite relaxation* and denote it $\boldsymbol{MSDR}$. The relaxation for $\boldsymbol{MQCQP}$ with $X \in \mathcal{M}^{mn}$ is

$$(\boldsymbol{MSDR}) \qquad \begin{array}{rll} v_{\mathbf{MSDR}} := & \min & \mathrm{trace}\,(Q_0Y + C_0X^T) + \beta_0 \\ & \text{s.t.} & \mathrm{trace}\,(Q_jY + C_jX^T) + \beta_j \leq 0, \quad j = 1, \ldots, m \\ & & \begin{pmatrix} I & X^T \\ X & Y \end{pmatrix} \succeq 0 \\ & & X \in \mathcal{M}^{mn}, Y \in \mathcal{S}^m. \end{array}$$

If $m \leq n$ and the Slater constraint qualification holds, then $\boldsymbol{MSDR}$ solves $\boldsymbol{MQCQP}$, $v_{\mathbf{MQCQP}} = v_{\mathbf{MSDR}}$, see [4, 5]. Otherwise, the bound from $\boldsymbol{MSDR}$ is not tight in general.

To apply this to $\boldsymbol{QAP}$, we first reformulate it as a $\boldsymbol{MQCQP}$ by moving $B$ from the objective using the constraint $R = XB$. Recall that $\Pi = \mathcal{O} \cap \mathcal{E} \cap \mathcal{N}$.

$$\begin{array}{rll} v_{\mathbf{QAP}} = & \min & \mathrm{trace} \begin{pmatrix} X \\ R \end{pmatrix}^T \begin{pmatrix} 0 & \frac{1}{2}A \\ \frac{1}{2}A & 0 \end{pmatrix} \begin{pmatrix} X \\ R \end{pmatrix} + \mathrm{trace}\, CX^T \\ & \text{s.t.} & R = XB \\ & & XX^T - I = X^TX - I = 0 \\ & & Xe = X^Te = e \\ & & X \geq 0, \quad X \in \mathcal{M}^n. \end{array} \tag{2.7}$$

To linearize the objective function we use

$$\mathrm{trace} \begin{pmatrix} X \\ R \end{pmatrix}^T \begin{pmatrix} 0 & \frac{1}{2}A \\ \frac{1}{2}A & 0 \end{pmatrix} \begin{pmatrix} X \\ R \end{pmatrix} = \mathrm{trace} \begin{pmatrix} 0 & \frac{1}{2}A \\ \frac{1}{2}A & 0 \end{pmatrix} \begin{pmatrix} X \\ R \end{pmatrix} \begin{pmatrix} X \\ R \end{pmatrix}^T,$$

and the lifting

$$\begin{pmatrix} X \\ R \end{pmatrix} \begin{pmatrix} X \\ R \end{pmatrix}^T = \begin{pmatrix} XX^T & XR^T \\ RX^T & RR^T \end{pmatrix} = \begin{pmatrix} I & Y \\ Y & Z \end{pmatrix}. \tag{2.8}$$

This defines the symmetric matrices $Y, Z \in \mathcal{S}^n$, where we see $Y = RX^T = X(X^TR)X^T = XBX^T \in \mathcal{S}^n$. We can then relax this to get the convex quadratic constraint

$$G(X, R, Y, Z) := \begin{pmatrix} XX^T & XR^T \\ RX^T & RR^T \end{pmatrix} - \begin{pmatrix} I & Y \\ Y & Z \end{pmatrix} \preceq 0. \tag{2.9}$$

A Schur complement argument shows that the convex quadratic constraint (2.9) is equivalent to the linear conic constraint

$$\begin{pmatrix} I & X^T & R^T \\ X & I & Y \\ R & Y & Z \end{pmatrix} \succeq 0. \tag{2.10}$$

The above discussion yields the ***MSDR*** relaxation for ***QAP***

$$(\boldsymbol{MSDR_0}) \qquad \begin{array}{lll} v_{\mathbf{QAP}} \geq & \min & \operatorname{trace} AY + \operatorname{trace} CX^T \\ & \text{s.t.} & R = XB \\ & & Xe = X^T e = e \\ & & X \geq 0 \\ & & \begin{pmatrix} I & X^T & R^T \\ X & I & Y \\ R & Y & Z \end{pmatrix} \succeq 0 \\ & & X, R \in \mathcal{M}^n, Y, Z \in \mathcal{S}^n, \end{array} \tag{2.11}$$

where $Y$ represents/approximates $RX^T = XBX^T$ and $Z$ represents/approximates $RR^T = XB^2X^T$. Since $X$ is a *permutation matrix*, we conclude that the diagonal of $Y$ is the $X$ permutation of the diagonal of $B$ (and similarly for the diagonals of $Z$ and $B^2$)

$$\operatorname{diag}(Y) = X\operatorname{diag}(B), \quad \operatorname{diag}(Z) = X\operatorname{diag}(B^2). \tag{2.12}$$

Also, given that $Xe = X^Te = e$ and $Y = XBX^T, Z = XB^2X^T$ for all $X, Y, Z$ feasible for the original ***QAP*** , we conclude that

$$Ye = XBe, Ze = XB^2e.$$

We may add these additional *redundant constraints* to the above ***MSDR*** . These constraints essentially replace the orthogonality constraints. We get the first version of our ***SDP*** relaxation:

$$(\mathbf{MSDR}_1) \qquad \begin{array}{lll} \mu^*_{\mathbf{MSDR1}} := & \min & \operatorname{trace} AY + \operatorname{trace} CX^T \\ & \text{s.t.} & R = XB \\ & & Xe = X^T e = e \\ & & \left\{ \begin{array}{l} \operatorname{diag}(Y) = X\operatorname{diag}(B) \\ \operatorname{diag}(Z) = X\operatorname{diag}(B^2) \\ Ye = XBe \\ Ze = XB^2e \end{array} \right\} \\ & & X \geq 0 \\ & & \begin{pmatrix} I & X^T & R^T \\ X & I & Y \\ R & Y & Z \end{pmatrix} \succeq 0 \\ & & X, R \in \mathcal{M}^n, Y, Z \in \mathcal{S}^n. \end{array}$$

**Proposition 2.1** *Let $B$ be nonsingular. In addition, suppose that $(X, Y, Z)$ solves* $\mathbf{MSDR}_1$ *and satisfies $Z = XB^2(X)^T$. Then $X$ is optimal for* $\boldsymbol{QAP}$ *.*

*Proof:* Via the Schur Complement, we know that the semidefinite constraint in $\mathbf{MSDR}_1$ is equivalent to

$$\begin{pmatrix} I - XX^T & Y - XBX^T \\ Y - XBX^T & Z - XB^2X^T \end{pmatrix} \succeq 0. \tag{2.13}$$

Therefore, $XX^T \preceq I$, $X^TX \preceq I$. Moreover, $X$ satisfies $Xe = X^Te = e, X \geq 0$. Now, multiplying both sides of $\operatorname{diag}(Z) = X\operatorname{diag}(B^2)$ from the left by $e^T$ yields $\operatorname{trace} Z = \operatorname{trace} B^2$. Since $Z = XB^2X^T$, we conclude that $\operatorname{trace} Z = \operatorname{trace} XB^2X^T = \operatorname{trace} B^2$, i.e., $\operatorname{trace} B^2(I - X^TX) = 0$. Since $B$ is nonsingular, we conclude that $B^2 \succ 0$. Therefore, $I - X^TX \succeq 0$ implies that $I = X^TX$. Thus the optimizer $X$ is orthogonal and doubly stochastic ($X \in \mathcal{E} \cap \mathcal{N}$). Hence $X$ is a permutation matrix.

Moreover, (2.13) and $Z - XB^2X^T = 0$ implies the off-diagonal block $Y - XBX^T = 0$. Thus, we conclude that the bound $\mu^*_{MSDR}$ from ($\mathbf{MSDR}_1$ ) is tight. ■

**Remark 2.1** *The assumption that $B$ is nonsingular is made without loss of generality, since we could shift $B$ by a* small *positive multiple of the identity matrix, say $\epsilon I$, while simultaneously subtracting $\epsilon(\operatorname{trace} A)$. i.e.,*

$$\begin{aligned} \operatorname{trace}(AXBX^T + CX^T) &= \operatorname{trace}(AX(B + \epsilon I)X^T - \epsilon AXX^T + CX^T) \\ &= \operatorname{trace}(AX(B + \epsilon I)X^T + CX^T) - \epsilon \operatorname{trace} A \end{aligned}$$

### 2.2.1 The Orthogonal Similarity Set of $B$

Our numerical tests in Section 3 below show that the bound from $\mathbf{MSDR}_1$ is often weaker than that from $\boldsymbol{QPB}$ . In this section we include additional redundant constraints in order to strengthen $\mathbf{MSDR}_1$ . Using majorization given in Definition 1.1, the now characterize the *convex hull of the orthogonal similarity set of* $B$, denoted $\operatorname{conv}\mathcal{O}(B)$.

**Theorem 2.1** *Let*

$$\begin{aligned} S_1 &:= \operatorname{conv}\mathcal{O}(B) = \operatorname{conv}\{Y \in \mathcal{S}^n : Y = XBX^T, X \in \mathcal{O}\}, \\ S_2 &:= \{Y \in \mathcal{S}^n : \operatorname{trace}\bar{A}Y \geq \langle \lambda(\bar{A}), \lambda(B)\rangle_-, \forall \bar{A} \in \mathcal{S}^n\}, \\ S_3 &:= \{Y \in \mathcal{S}^n : \operatorname{diag}(X^TYX) \preceq \lambda(B), \forall X \in \mathcal{O}\}, \\ S_4 &:= \{Y \in \mathcal{S}^n : \lambda(Y) \preceq \lambda(B)\}. \end{aligned} \tag{2.14}$$

*Then $S_1$ is the convex hull of the orthogonal similarity set of $B$, and $S_1 = S_2 = S_3 = S_4$.*

**Proof.**

1. $\underline{S_1 \subseteq S_2}$**:** Let $Y \in S_1, \bar{A} \in \mathcal{S}^n$. Then

$$\operatorname{trace}\bar{A}Y \geq \min_{Y \in \operatorname{conv}\mathcal{O}(B)} \operatorname{trace}\bar{A}Y = \min_{X \in \mathcal{O}} \operatorname{trace}\bar{A}XBX^T = \langle \lambda(\bar{A}), \lambda(B)\rangle_-,$$

by the well-known minimal inner-product result, e.g., [31][9].

2. $\underline{S_2 \subseteq S_3}$: Let $U \in \mathcal{O}$, $p \in \{1, 2, \ldots, n-1\}$, and let $\Gamma_p$ denote the index set corresponding to the $p$ smallest entries of $\operatorname{diag}(U^T Y U)$. Define the support vector $\delta^p \in \mathbb{R}^n$ of $\Gamma_P$ by

$$(\delta^p)_i = \begin{cases} 1 & \text{if } i \in \Gamma_p \\ 0 & \text{otherwise.} \end{cases}$$

Then, for $A_p := U \operatorname{Diag}(\delta^p) U^T$, we get

$$\begin{array}{rcl} \langle \delta^p, \operatorname{diag}(U^T Y U) \rangle & = & \langle \operatorname{Diag}(\delta^p), U^T Y U \rangle \\ & = & \langle U \operatorname{Diag}(\delta^p) U^T, Y \rangle \\ & = & \langle A_p, Y \rangle \\ & \geq & \langle \delta^p, \lambda(B) \rangle_-, \end{array}$$

by definition of $S_2$. Since choosing $\bar{A} = \pm I$ implies $\operatorname{trace} Y = \operatorname{trace} B$, the inclusion follows.

3. $\underline{S_3 \subseteq S_4}$: Let $Y \in S_3$, and $Y = V \operatorname{Diag}(\lambda(Y)) V^T$, $V \in \mathcal{O}$, be its spectral decomposition. Since $U \in \mathcal{O}$ implies that $\operatorname{diag}(U^T Y U) \preceq \lambda(B)$, we may take $U = V$ and deduce

$$\lambda(Y) = \operatorname{diag}(V^T Y V) \preceq \lambda(B).$$

4. $\underline{S_4 \subseteq S_1}$: To obtain a contradiction, suppose $\lambda(\hat{Y}) \preceq \lambda(B)$, but $\hat{Y} \notin \operatorname{conv} \mathcal{O}(B)$. Since $\mathcal{O}$ is a compact set, we conclude that the continuous image $\mathcal{O}(B) = \{Y : Y = XBX^T, X \in \mathcal{O}\}$ is compact. Hence, its convex hull $\operatorname{conv} \mathcal{O}(B)$ is compact as well. Therefore, a standard hyperplane separation argument implies that there exists $\bar{A} \in \mathcal{S}^n$, such that

$$\langle \bar{A}, \hat{Y} \rangle < \min_{Y \in \operatorname{conv}(\mathcal{O}(B))} \langle \bar{A}, Y \rangle = \min_{Y \in \mathcal{O}(B)} \langle \bar{A}, Y \rangle = \langle \lambda(\bar{A}), \lambda(B) \rangle_-.$$

As a result,

$$\langle \lambda(\bar{A}), \lambda(\hat{Y}) \rangle_- \leq \langle \bar{A}, \hat{Y} \rangle < \langle \lambda(\bar{A}), \lambda(B) \rangle_-.$$

Without loss of generality, suppose that the eigenvalues $\lambda(\bullet)$ are in nondecreasing order. Then the above minimum product inequality could be written as

$$\sum_{i=1}^{n} \lambda_i(\bar{A}) \lambda_{n-i+1}(\hat{Y}) < \sum_{i=1}^{n} \lambda_i(\bar{A}) \lambda_{n-i+1}(B),$$

which implies

$$0 > \sum_{i=1}^{n} \lambda_i(\bar{A}) (\lambda_{n-i+1}(\hat{Y}) - \lambda_{n-i+1}(B)).$$

Since $\lambda_i(\bar{A}) = \sum_{j=1}^{i-1} (\lambda_{j+1}(\bar{A}) - \lambda_j(\bar{A})) + \lambda_i(\bar{A})$, we can rewrite the above inequality as

$$\begin{array}{rcl} 0 & > & \sum_{i=1}^{n} (\sum_{j=1}^{i-1} (\lambda_{j+1}(\bar{A}) - \lambda_j(\bar{A})) + \lambda_i(\bar{A}))(\lambda_{n-i+1}(\hat{Y}) - \lambda_{n-i+1}(B)) \\ & = & \sum_{j=1}^{n-1} (\lambda_{j+1}(\bar{A}) - \lambda_j(\bar{A})) \sum_{i=j+1}^{n} (\lambda_{n-i+1}(\hat{Y}) - \lambda_{n-i+1}(B)) \\ & & \quad + \lambda_1(\bar{A}) \sum_{i=1}^{n} (\lambda_i(\hat{Y}) - \lambda_i(B)). \end{array}$$

Notice $\lambda(\hat{Y}) \preceq \lambda(B)$ implies $e^T\lambda(\hat{Y}) = e^T\lambda(B)$, so $\lambda_1(\bar{A})\sum_{i=1}^n(\lambda_i(\hat{Y}) - \lambda_i(B)) = 0$. Thus, we have the following inequality:

$$0 > \sum_{j=1}^{n-1}(\lambda_{j+1}(\bar{A}) - \lambda_j(\bar{A}))\sum_{i=j+1}^{n}(\lambda_{n-i+1}(\hat{Y}) - \lambda_{n-i+1}(B)). \tag{2.14}$$

However, by assumption $\lambda_{j+1}(\bar{A}) \geq \lambda_j(\bar{A})$, and by the definition of $\lambda(\hat{Y})$ majorized by $\lambda(B)$,

$$\sum_{i=j+1}^{n}\lambda_{n-i+1}(\hat{Y}) = \sum_{t=1}^{n-j}\lambda_t(\hat{Y}) \geq \sum_{t=1}^{n-j}\lambda_t(B) = \sum_{i=j+1}^{n}\lambda_{n-i+1}(B)$$

which contradicts (2.14).

■

**Remark 2.2** *Based on our Theorem 2.1* [1]*, Xia [33] recognized that the sets S1-S4 in* (2.14) *admit a semidefinite formulation, i.e.,*

$$S_1 = S_5 := \left\{Y \in S^n : Y = \sum_{i=1}^n \lambda_i(B)Y_i, \sum_{i=1}^n Y_i = I_n, \operatorname{trace} Y_i = 1, Y_i \succeq 0, i = 1,\ldots,n\right\}.$$

*He then proposed an orthogonal bound, denoted (**OB2** ), from the optimal value of the **SDP***

$$v_{\mathbf{OB2}} := \min_{X\geq 0, Xe=X^Te=e, Y\in S_5} \operatorname{trace}(AY + CX^T).$$

*This new bound is provably stronger than the current convex quadratic programming bound (**QPB** ).*

*We failed to recognize this point in our initial work. Instead, motivated by Theorem 2.1, we now propose an inexpensive bound that is stronger than **QPB** for most of the problem instances we tested.*

### 2.2.2 Strengthened MSDR Bound

Suppose that $A = U_A \operatorname{Diag}(\lambda(A))U_A^T$ denotes the orthogonal diagonalization of $A$ with the vector of eigenvalues $\lambda(A)$ in nonincreasing order; we assume that the vector of eigenvalues $\lambda(B)$ is in nondecreasing order. Let

$$\delta^p := \{\overbrace{1,1,\ldots,1}^{p}, 0, 0, \ldots, 0\}, p = 1, 2, \ldots n-1.$$

We add the following cuts to $\mathbf{MSDR}_1$ ,

$$\langle \delta^p, \operatorname{diag}(U_A^T Y U_A)\rangle \geq \langle \delta^p, \lambda(B)\rangle, \quad p = 1, 2, \ldots, n-1, \tag{2.15}$$

[1] Xia [33] references our Theorem 2.1 from an earlier version of our paper.

and get a new relaxation

$$
(\boldsymbol{MSDR_2}) \qquad
\begin{array}{rll}
\mu^*_{MSDR2} := & \min & \langle A, Y\rangle + \langle C, X\rangle \\
& \text{s.t.} & Xe = X^T e = e \\
& & \operatorname{diag}(Y) = X \operatorname{diag}(B) \\
& & \operatorname{diag}(Z) = X \operatorname{diag}(B^2) \\
& & Ye = XBe \\
& & Ze = XB^2 e \\
& & \langle \delta^p, \operatorname{diag}(U_A^T Y U_A)\rangle \geq \langle \delta^p, \lambda(B)\rangle, p = 1, 2, \ldots, n-1 \\
& & X \geq 0 \\
& & \begin{pmatrix} I & X^T & B^T X^T \\ X & I & Y \\ XB & Y & Z \end{pmatrix} \succeq 0 \\
& & X \in \mathcal{M}^n, Y, Z \in \mathcal{S}^n
\end{array}
$$

The cuts (2.15) approximate the majorization constraint

$$
\operatorname{diag}(U_A^T Y U_A) \preceq \lambda(B). \tag{2.16}
$$

From (2.16), we have

$$
\langle A, Y\rangle = \operatorname{trace}(\operatorname{Diag}(\lambda(A)) U_A^T Y U_A) = \langle \lambda(A), \operatorname{diag}(U_A^T Y U_A)\rangle \geq \langle \lambda(A), \lambda(B)\rangle_-.
$$

This inequality yields a comparison between the bounds $\boldsymbol{MSDR_2}$ and $\boldsymbol{EB}$.

**Lemma 2.1** *The bound from* $\boldsymbol{MSDR_2}$

$$
u^*_{MSDR2} \geq \langle \lambda(A), \lambda(B)\rangle_- + \min_{Xe = X^T e = e, X \geq 0} \langle C, X\rangle,
$$

*the eigenvalue bound,* $\boldsymbol{EB}$.

**Proof.** It is enough to show that the first terms on both sides of the inequality satisfy

$$
\langle A, Y\rangle \;\geq\; \langle \lambda(A), \lambda(B)\rangle_-
$$

for any $Y$ feasible in $\boldsymbol{MSDR_2}$. Note that

$$
\langle A, Y\rangle = \langle U_A \operatorname{Diag}(\lambda(A)) U_A^T, Y\rangle = \langle \lambda(A), \operatorname{diag}(U_A^T Y U_A)\rangle.
$$

Since $\lambda(A)$ is a nonincreasing vector, and $\lambda(B)$ is nondecreasing, we have $\langle \lambda(B), \lambda(A)\rangle = \langle \lambda(B), \lambda(A)\rangle_-$. Also,

$$
\lambda(A) = \sum_{p=1}^{n-1} (\lambda_p(A) - \lambda_{p+1}(A))\delta^p + \lambda_n(A) e.
$$

Therefore, since $\mathrm{diag}\,(Y) = X\mathrm{diag}\,(B)$ and $e^TX = e^T$, we have

$$\langle A, Y\rangle = \sum_{p=1}^{n-1} (\lambda_p(A) - \lambda_{p+1}(A))\langle \delta^p, \mathrm{diag}\,(U_A^T Y U_A)\rangle + \lambda_n(A)\langle e, \lambda(B)\rangle.$$

Since $\langle \delta^p, \mathrm{diag}\,(U_A^T Y U_A)\rangle \geq \langle \delta^p, \lambda(B)\rangle$ holds for any feasible $Y$, we have

$$\begin{array}{rcl} \langle A, Y\rangle & \geq & \sum_{p=1}^{n-1} (\lambda_p(A) - \lambda_{p+1}(A))\langle \delta^p, \lambda(B)\rangle + \lambda_n(A)\langle e, \lambda(B)\rangle \\ & = & \sum_{p=1}^{n-1} ((\lambda_p(A) - \lambda_{p+1}(A)) \sum_{i=1}^{p} \lambda_i(B)) + \lambda_n(A) \sum_{i=1}^{n} \lambda_i(B) \\ & = & \sum_{i=1}^{n} \lambda_i(B)(\sum_{p=i}^{n-1} \lambda_p(A) - \lambda_{p+1}(A)) + \lambda_n(A)) \\ & = & \sum_{i=1}^{n} \lambda_i(B)\lambda_i(A) \\ & = & \langle \lambda(B), \lambda(A)\rangle_-. \end{array}$$

■

## 2.3 Projected Bound OB2

The row and column sum equality constraints of ***QAP*** , $\mathcal{E} = \{X \in \mathcal{M}^n : Xe = X^Te = e\}$, can be eliminated using a nullspace method. (In the following proposition, $\mathcal{O}$ refers to the orthogonal matrices of appropriate dimension.)

**Proposition 2.2 ([12])** *Let $V \in \mathcal{M}^{n,n-1}$ be full column rank and satisfy $V^Te = 0$. Then $X \in \mathcal{E} \cap \mathcal{O}$ if and only if*

$$X = \frac{1}{n}E + V\hat{X}V^T, \text{ for some } \hat{X} \in \mathcal{O}.$$

■

After substituting for $X$, and using $\hat{A} = V^TAV, \hat{B} = V^TBV$, the ***QAP*** can now be reformulated as the projected version (***PQAP*** )

$$(PQAB) \qquad \begin{array}{ll} \min & \mathrm{trace}\left(\hat{A}\hat{X}\hat{B}\hat{X}^T + \frac{1}{n}\hat{A}\hat{X}\hat{B}E + \frac{1}{n}\hat{A}E\hat{B}\hat{X}^T + \frac{1}{n^2}\hat{A}E\hat{B}E\right) \\ \text{s.t.} & \hat{X}\hat{X}^T = \hat{X}^T\hat{X} = I \\ & X(\hat{X}) = \frac{1}{n}E + V\hat{X}V^T \geq 0. \end{array}$$

We now define $\hat{Y} = \hat{X}\hat{B}\hat{X}^T$ and $\hat{Z} = \hat{Y}\hat{Y} = \hat{X}\hat{B}\hat{B}\hat{X}^T$; and we replace $X$ with $\frac{1}{n}E + V\hat{X}V^T$. Then the two terms $XBX$ and $XBVV^TBX^T$ admit the linear representations

$$XBX^T = V\hat{X}\hat{B}\hat{X}^TV^T + \frac{1}{n}EBV\hat{X}^TV^T + \frac{1}{n}V\hat{X}V^TBE + \frac{1}{n^2}E\hat{B}E$$

and

$$XBVV^TBX^T = V\hat{Z}V^T + \frac{1}{n}EBVV^TBVX^TV^T + \frac{1}{n}VXV^TBVV^TBE + \frac{1}{n^2}EBVV^TBE,$$

respectively. In $\boldsymbol{MSDR_2}$ , we use $Y$ to represent/approximate $XBX^T$, and use $Z$ to represent/approximate $XBBX^T$. However, $XBBX^T$ cannot be linearly represented with $\hat{X}$ and $\hat{Y}$. Therefore, in the projected version, we have to let $Z$ represent $XBVV^TBX^T$ instead of $XBBX^T$, and we replace the corresponding diagonal constraint with $\operatorname{diag}(Z) = X\operatorname{diag}(BVV^TB)$.

Based on these definitions, $\boldsymbol{PQAP}$ has the following quadratic matrix programming formulation:

$$\begin{array}{ll}
\min & \operatorname{trace}(AY + CX^T) \\
\text{s.t.} & \operatorname{diag} Y = X\operatorname{diag}(B) \\
& \operatorname{diag} Z = X\operatorname{diag}(BVV^TB) \\
& X(\hat{X}) = V\hat{X}V^T + \frac{1}{n}E \\
& Y(\hat{X},\hat{Y}) = V\hat{Y}V^T + \frac{1}{n}EBV\hat{X}^TV^T + \frac{1}{n}V\hat{X}V^TBE + \frac{1}{n^2}E\hat{B}E \\
& Z(\hat{X},\hat{Z}) = V\hat{Z}V^T + \frac{1}{n}EBVV^TBVX^TV^T + \frac{1}{n}VXV^TBVV^TBE + \frac{1}{n^2}EBVV^TBE \\
& \hat{R} = \hat{X}\hat{B} \\
& \begin{pmatrix} I & \hat{Y} \\ \hat{Y} & \hat{Z} \end{pmatrix} = \begin{pmatrix} \hat{X}\hat{X}^T & \hat{X}\hat{R}^T \\ \hat{R}\hat{X}^T & \hat{R}\hat{R}^T \end{pmatrix} \\
& X(\hat{X}) \geq 0 \\
& \hat{X}, \hat{R} \in \mathcal{M}^{n-1}, \hat{Y}, \hat{Z} \in S^{n-1}.
\end{array} \tag{2.17}$$

We can now relax the quadratic constraint

$$\begin{pmatrix} I & \hat{Y} \\ \hat{Y} & \hat{Z} \end{pmatrix} = \begin{pmatrix} \hat{X}\hat{X}^T & \hat{X}\hat{R}^T \\ \hat{R}\hat{X}^T & \hat{R}\hat{R}^T \end{pmatrix}$$

with the convex constraint

$$\begin{pmatrix} I & \hat{X}^T & \hat{R}^T \\ \hat{X} & I & \hat{Y} \\ \hat{R} & \hat{Y} & \hat{Z} \end{pmatrix} \succeq 0.$$

As in $\boldsymbol{MSDR_2}$ , we now add the following cuts for $\hat{Y} \in \operatorname{conv}\mathcal{O}(\hat{X})$

$$\langle \delta^p, \operatorname{diag}(U_{\hat{A}}^T\hat{Y}U_{\hat{A}}\rangle \geq \langle \delta^p, \lambda(\hat{B})\rangle, p = 1, 2, \ldots, n-2,$$

where $\hat{A} = U_{\hat{A}}\operatorname{Diag}(\lambda(\hat{A}))U_{\hat{A}}^T$ is the spectral decomposition of $\hat{A}$, and $\lambda_1(\hat{A}) \leq \lambda_2(\hat{A}) \leq \ldots \leq \lambda_n(\hat{A})$. $\delta^p$ follows the definition in Section 2.2.1, i.e., $\delta^p \in R^{n-1}, \delta^p = \{0, 0, \ldots, 0, 1, \ldots, 1\}$.

Our final projected relaxation $\mathbf{MSDR_3}$ is

$$
(\mathbf{MSDR_3}) \qquad
\begin{array}{rcll}
\mu^*_{\mathbf{MSDR_3}} & := & \min & \langle A, Y(\hat X,\hat Y)\rangle + \langle C, X(\hat X)\rangle \\
 & & \text{s.t.} & \operatorname{diag}(Y(\hat X,\hat Y)) = X(\hat X)\operatorname{diag}(B) \\
 & & & \operatorname{diag}(Z(\hat X,\hat Z)) = X(\hat X)\operatorname{diag}(BVV^TB) \\
 & & & \langle \delta^p, \operatorname{diag}(U_{\hat A}^T\hat Y U_{\hat A}\rangle \ge \langle \delta^p, \lambda(\hat B)\rangle, p = 1,2,\ldots,n-2 \\
 & & & X(\hat X) \ge 0 \\
 & & & \begin{pmatrix} I & \hat X^T & \hat B^T\hat X^T \\ \hat X & I & \hat Y \\ \hat X\hat B & \hat Y & \hat Z \end{pmatrix} \succeq 0 \\
 & & & \hat X \in \mathcal{M}^{n-1}, \hat Y, \hat Z \in S^{n-1},
\end{array}
$$

where:

$$
\begin{array}{rcl}
X(\hat X) & = & \frac{1}{n}E + V\hat X V^T; \\
Y(\hat X,\hat Y) & = & V\hat Y V^T + \frac{1}{n}EBV\hat X^T V^T + \frac{1}{n}V\hat X V^T BE + \frac{1}{n^2}E\hat B E; \\
Z(\hat X,\hat Z) & = & V\hat Z V^T + \frac{1}{n}EBVV^TBVX^TV^T + \frac{1}{n}VXV^TBVV^TBE + \frac{1}{n^2}EBVV^TBE.
\end{array}
$$

Note that the constraints $Ye = XBe$, $Ze = XB^2e$ are no longer needed in $\mathbf{MSDR_3}$.

In $\mathbf{MSDR_3}$, all the constraints act on the lower dimensional space. Such strategy, i.e., *first projected back, then cut*, has also been used in projected eigenvalue bound(***PB***) and quadratic programming bound(***QPB***). It is numerically superior than directly adding cuts for high-dimensional image space of the projection, e.g., projected eigenvalue bound(***PB***) is much stronger than the eigenvalue bound(***EB***). For this reason, we propose $\mathbf{MSDR_3}$ instead of $\boldsymbol{MSDR_2}$.

**Lemma 2.2** *Let $\mu^*_{PB}$ denote the projected eigenvalue bound. Then*

$$\mu^*_{\mathbf{MSDR_3}} \ge \mu^*_{PB}.$$

**Proof.** Since $\mathbf{MSDR_3}$ has constraints

$$\langle \delta^p, \operatorname{diag}(U_{\hat A}^T\hat Y U_{\hat A}\rangle \ge \langle \delta^p, \lambda(\hat B)\rangle, p = 1,2,\ldots,n-2,$$

we need only prove that $\operatorname{trace} \hat A\hat Y \ge \langle \lambda(\hat A), \lambda(\hat B)\rangle_-$. This proof is the same as the proof for $\operatorname{trace} AY \ge \langle \lambda(A), \lambda(B)\rangle_-$ in Lemma 2.1. ■

**Remark 2.3** *Every feasible solution to the original **QAP** satisfies $Y = XBX^T, X \in \Pi$. This implies that $Y$ could be obtained from a permutation of the entries of $B$. Moreover, the diagonal entries of $B$ remain on the diagonal after a permutation. Denote the off-diagonal entries of $B$ by $\operatorname{0ffDiag}(B)$. We see that, for each $i, j = 1,2,\ldots,n, i \ne j$, the following cuts are valid for any feasible $Y$:*

$$\min\left[\operatorname{0ffDiag}(B)\right) \le Y_{ij} \le \max(\operatorname{0ffDiag}(B)\right]. \tag{2.18}$$

*It is easy to verify that if the elements of* $\operatorname{OffDiag}(B)$ *are all equal, then* $\boldsymbol{QAP}$ *can be solved by* $\mathbf{MSDR}_1$ *,* $\boldsymbol{MSDR_2}$ *or* $\mathbf{MSDR}_3$ *, using the constraints in* (2.18).

*If* $B$ *is diagonally dominant, than for any permutation* $X$*, we have* $Y = XBX^T$ *is diagonal dominant. This property generates another series of cuts. These results could be to used to add cuts for* $Z = XB^2X^T$ *as well.*

# 3 Numerical Results

## 3.1 QAPLIB Problems

In Table 1 we present a comparison of $\mathbf{MSDR}_3$ with several other bounds applied to instances from $\boldsymbol{QAP}$LIB , [6]. The first column OPT denotes the exact optimal value. The following columns contain the: $\boldsymbol{GLB}$ , Gilmore-Lawler bound [10]; $\boldsymbol{KCCEB}$ , dual linear programming bound [18, 15, 14]; $\boldsymbol{PB}$ , projected eigenvalue bound [12]; $\boldsymbol{QPB}$ , convex quadratic programming bound [1]; and $\boldsymbol{SDR1}$ , $\boldsymbol{SDR2}$ , $\boldsymbol{SDR3}$ , the vector-lifting semidefinite relaxation bounds [34] computed by the bundle method [30]. The last column is our $\mathbf{MSDR}_3$ bound. All output values are rounded up to the nearest integer.

To solve $\boldsymbol{QAP}$ , the minimization of $\operatorname{trace} AXBX^T$ and $\operatorname{trace} BXAX^T$ are equivalent. But for the relaxation $\mathbf{MSDR}_3$ , exchanging the roles of $A$ and $B$ results in two different formulations and bounds. In our tests we use the maximum of the two formulations for $\mathbf{MSDR}_3$ . We then stay with the better formulation throughout the branch and bound process, so that we do not double the computational work.

From Table 1, we see that the relative performance of the $\boldsymbol{LP}$ -based bounds $\boldsymbol{GLB}$ and $\boldsymbol{KCCEB}$ is unpredictable. For some instances both are weaker than the least expensive $\boldsymbol{PB}$ bound; while for other instances the $\boldsymbol{KCCEB}$ bound is better than the more expensive $\boldsymbol{SDR2}$ . The *average* performance of the bounds can be ranked as follows:

$$\boldsymbol{PB} < \boldsymbol{QPB} < \mathrm{MSDR}_3 \approx \boldsymbol{SDR1} < \boldsymbol{SDR2} < \boldsymbol{SDR3} .$$

In Table 2 we present the number of variables and constraints used in each of the relaxations. Our bound $\mathbf{MSDR}_3$ uses only $O(n^2)$ variables and only $O(n^2)$ constraints. If we solve $\mathbf{MSDR}_3$ with an interior point method, the complexity of computing the Newton direction in each iteration is $O(n^6)$. And, the number of iterations of an interior point method is bounded by $O(n \ln \frac{1}{\epsilon})$ [24]. Therefore, the complexity of computing $\mathbf{MSDR}_3$ with an interior point methods is $O(n^7 \ln \frac{1}{\epsilon})$. Note that the computational complexity for the most expensive $\boldsymbol{SDP}$ formulation, $\boldsymbol{SDR3}$ , is $O(n^{14} \ln \frac{1}{\epsilon})$, where $\epsilon$ is the desired accuracy. Thus $\mathbf{MSDR}_3$ is significantly less expensive than $\boldsymbol{SDR3}$ . Though $\boldsymbol{QPB}$ is less expensive than $\mathbf{MSDR}_3$ in practice, the complexity as a function of $n$ is the same.

Table 3 lists the CPU time (in seconds) for $\mathbf{MSDR}_3$ for several of the *Nugent instances* [26]. (We used a SUN SPARC 10 and the SeDuMi[2] $\boldsymbol{SDP}$ package.)

[2]sedumi.mcmaster.ca

| Problem | OPT | GLB | KCCEB | PB | QPB | SDR1 | SDR2 | SDR3 | **MSDR$_3$** |
|---|---|---|---|---|---|---|---|---|---|
| esc16a | 68 | 38 | 41 | 47 | 55 | 47 | 49 | 59 | 50 |
| esc16b | 292 | 220 | 274 | 250 | 250 | 250 | 275 | 288 | 276 |
| esc16c | 160 | 83 | 91 | 95 | 95 | 95 | 111 | 142 | 123 |
| esc16d | 16 | 3 | 4 | -19 | -19 | -19 | -13 | 8 | 1 |
| esc16e | 28 | 12 | 12 | 6 | 6 | 6 | 11 | 23 | 14 |
| esc16g | 26 | 12 | 12 | 9 | 9 | 9 | 10 | 20 | 13 |
| esc16h | 996 | 625 | 704 | 708 | 708 | 708 | 905 | 970 | 906 |
| esc16i | 14 | 0 | 0 | -25 | -25 | -25 | -22 | 9 | 0 |
| esc16j | 8 | 1 | 2 | -6 | -6 | -6 | -5 | 7 | 0 |
| had12 | 1652 | 1536 | 1619 | 1573 | 1592 | 1604 | 1639 | 1643 | 1595 |
| had14 | 2724 | 2492 | 2661 | 2609 | 2630 | 2651 | 2707 | 2715 | 2634 |
| had16 | 3720 | 3358 | 3553 | 3560 | 3594 | 3612 | 3675 | 3699 | 3587 |
| had18 | 5358 | 4776 | 5078 | 5104 | 5141 | 5174 | 5282 | 5317 | 5153 |
| had20 | 692 | 6166 | 6567 | 6625 | 6674 | 6713 | 6843 | 6885 | 6681 |
| kra30a | 88900 | 68360 | 75566 | 63717 | 68257 | 69736 | 68526 | 77647 | 72480 |
| kra30b | 91420 | 69065 | 76235 | 63818 | 68400 | 70324 | 71429 | 81156 | 73155 |
| Nug12 | 578 | 493 | 521 | 472 | 482 | 486 | 528 | 557 | 502 |
| Nug14 | 1014 | 852 | n.a. | 871 | 891 | 903 | 958 | 992 | 918 |
| Nug15 | 1150 | 963 | 1033 | 973 | 994 | 1009 | 1069 | 1122 | 1016 |
| Nug16a | 1610 | 1314 | 1419 | 1403 | 1441 | 1461 | 1526 | 1570 | 1460 |
| Nug16b | 1240 | 1022 | 1082 | 1046 | 1070 | 1082 | 1136 | 1188 | 1082 |
| Nug17 | 1732 | 1388 | 1498 | 1487 | 1523 | 1548 | 1619 | 1669 | 1549 |
| Nug18 | 1930 | 1554 | 1656 | 1663 | 1700 | 1723 | 1798 | 1852 | 1726 |
| Nug20 | 2570 | 2057 | 2173 | 2196 | 2252 | 2281 | 2380 | 2451 | 2291 |
| Nug21 | 2438 | 1833 | 2008 | 1979 | 2046 | 2090 | 2244 | 2323 | 2099 |
| Nug22 | 3596 | 2483 | 2834 | 2966 | 3049 | 3140 | 3372 | 3440 | 3137 |
| Nug24 | 3488 | 2676 | 2857 | 2960 | 3025 | 3068 | 3217 | 3310 | 3061 |
| Nug25 | 3744 | 2869 | 3064 | 3190 | 3268 | 3305 | 3438 | 3535 | 3300 |
| Nug27 | 5234 | 3701 | n.a. | 4493 | n.a. | n.a. | 4887 | 4965 | 4621 |
| Nug30 | 6124 | 4539 | 4785 | 5266 | 5362 | 5413 | 5651 | 5803 | 5446 |
| rou12 | 235528 | 202272 | 223543 | 200024 | 205461 | 208685 | 219018 | 223680 | 207445 |
| rou15 | 354210 | 298548 | 323589 | 296705 | 303487 | 306833 | 320567 | 333287 | 303456 |
| rou20 | 725522 | 599948 | 641425 | 597045 | 607362 | 615549 | 641577 | 663833 | 609102 |
| scr12 | 31410 | 27858 | 29538 | 4727 | 8223 | 11117 | 23844 | 29321 | 18803 |
| scr15 | 51140 | 44737 | 48547 | 10355 | 12401 | 17046 | 41881 | 48836 | 39399 |
| scr20 | 110030 | 86766 | 94489 | 16113 | 23480 | 28535 | 82106 | 94998 | 50548 |
| tai12a | 224416 | 195918 | 220804 | 193124 | 199378 | 203595 | 215241 | 222784 | 202134 |
| tai15a | 388214 | 327501 | 351938 | 325019 | 330205 | 333437 | 349179 | 364761 | 331956 |
| tai17a | 491812 | 412722 | 441501 | 408910 | 415576 | 419619 | 440333 | 451317 | 418356 |
| tai20a | 703482 | 580674 | 616644 | 575831 | 584938 | 591994 | 617630 | 637300 | 587266 |
| tai25a | 1167256 | 962417 | 1005978 | 956657 | 981870 | 974004 | 908248 | 1041337 | 970788 |
| tai30a | 1818146 | 1504688 | 1565313 | 1500407 | 1517829 | 1529135 | 1573580 | 1652186 | 1521368 |
| tho30 | 149936 | 90578 | 99855 | 119254 | 124286 | 125972 | 134368 | 136059 | 122778 |

Table 1: Comparison of bounds for ***QAP***LIB instances

| 7 Methods | GLB | KCCE | PB | QPB | SDR1 | SDR2 | SDR3 | **MSDR**$_3$ |
|---|---|---|---|---|---|---|---|---|
| Variables | $O(n^4)$ | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^4)$ | $O(n^4)$ | $O(n^4)$ | $O(n^2)$ |
| Constraints | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^3)$ | $O(n^4)$ | $O(n^2)$ |

Table 2: Complexity of Relaxations

| Instances | Nug12 | Nug15 | Nug18 | Nug20 | Nug25 | Nug27 | Nug30 |
|---|---|---|---|---|---|---|---|
| CPU time(s) | 15.1 | 57.6 | 203.9 | 534.9 | 3236.4 | 5211.3 | 12206.0 |
| Number of iterations | 18 | 19 | 22 | 26 | 27 | 25 | 29 |

Table 3: CPU time and iterations for computing **MSDR**$_3$ on the Nugent problems

### 3.2 MSDR$_3$ in a Branch and Bound Framework

When solving general discrete optimization problems using ***B&B*** methods, one rarely has advance knowledge that helps in branching decisions. But, we now see that **MSDR**$_3$ helps in choosing a row and/or column for branching in our ***B&B*** approach for solving ***QAP*** .

If $X$ is a permutation matrix, then the diagonal entries $\operatorname{diag}(Z) = X\operatorname{diag}(BVV^TB)$ are a permutation of the diagonal entries of $BVV^TB$. In fact, the converse is true under a mild assumption.

**Proposition 3.1** *Assume the $n$ entries of* $\operatorname{diag}(BVV^TB)$ *are all distinct. If* $(X^*, Y^*, Z^*)$ *is an optimal solution to* **MSDR**$_3$ *that satisfies* $\operatorname{diag}(Z^*) = P\operatorname{diag}(BVV^TB)$, *for some* $P \in \Pi$, *then* $(X^*, Y^*, Z^*)$ *solves* ***QAP*** *exactly.*

**Proof.** Without loss of generality, assume the entries of $b := \operatorname{diag}(BVV^TB)$ are strictly increasing, i.e., $b_1 < b_2 < \ldots < b_n$. By the feasibility of $X^*, Z^*$, we have $\operatorname{diag}(Z^*) = X^*b$. Also, we know $\operatorname{diag}(Z^*) = Pb$, for some $P \in \Pi$. Therefore, $X^*b = Pb$ holds as well. Now assume $P_{i1} = 1$. Then $\sum_{j=1}^n X^*_{ij}b_j = b_1$. Since $\sum_{j=1}^n X^*_{ij} = 1$ and $X^*_{ij} \geq 0, j = 1, 2, \ldots, n$, we conclude that $b_1$ is a convex combination of $b_1, b_2, \ldots, b_n$. However, $b_1$ is the strict minimum in $b_1, b_2, \ldots, b_n$. This implies that $X^*_{i1} = 1$. The conclusion follows for $P = X^*$ by finite induction, after we delete column 1 and row $i$ of $X$. ■

As a consequence of Proposition 3.1, we may consider the original ***QAP*** problem in order to determine an optimal assignment of entries of $\operatorname{diag}(BVV^TB)$ to $\operatorname{diag}(Z)$, where each entry of $\operatorname{diag}(BVV^TB)$ requires a branch and bound process to determine its assigned position. For entries with large difference from the mean of $\operatorname{diag}(BVV^TB)$, the assignments are particularly important, because a change of their assigned positions usually leads to significant differences in the corresponding objective value. Therefore, in order to fathom more nodes early, our ***B&B*** strategy first processes those entries with large differences from the mean of $\operatorname{diag}(BVV^TB)$.

**Branch and Bound Strategy 3.1** *Let* $b := \operatorname{diag}(BVV^TB)$. *Branch on the $i$-th column of $X$ where $i$ corresponds to the element $b_i$ that has the largest deviation from the mean of the* *elements of b. (If this strategy results in several elements close in value, then we randomly pick one of them.)*

For example, Nug12 yields

$$\operatorname{diag}(BVV^TB)^T = (\,23 \quad 14 \quad 14 \quad 23 \quad 17.67 \quad 8.67 \quad 8.67 \quad 17.67 \quad 23 \quad 14 \quad 14 \quad 23\,).$$

Therefore, the 6 (or 7)-th entry has value 8.67; this has the largest difference from the mean value 16.72. Table 4 presents the $\mathbf{MSDR}_3$ bounds in the first level of the branching tree for Nug12. The first and second column presents the results for branching on elements from the 6-th column of $X$ first. The other columns provide a comparison with branching from other columns first. On average, branching with the 6-th column of $X$ first generates tighter bounds, thus allowing for more nodes to be fathomed.

| nodes | bounds | nodes | bounds | nodes | bounds |
|---|---|---|---|---|---|
| $X_{1,6}=1$ | 523 | $X_{1,1}=1$ | 508 | $X_{1,2}=1$ | 512 |
| $X_{2,6}=1$ | 528 | $X_{2,1}=1$ | 509 | $X_{2,2}=1$ | 513 |
| $X_{3,6}=1$ | 520 | $X_{3,1}=1$ | 507 | $X_{3,2}=1$ | 508 |
| $X_{4,6}=1$ | 517 | $X_{4,1}=1$ | 515 | $X_{4,2}=1$ | 510 |
| $X_{5,6}=1$ | 537 | $X_{5,1}=1$ | 512 | $X_{5,2}=1$ | 519 |
| $X_{6,6}=1$ | 529 | $X_{6,1}=1$ | 517 | $X_{6,2}=1$ | 513 |
| $X_{7,6}=1$ | 507 | $X_{7,1}=1$ | 516 | $X_{7,2}=1$ | 507 |
| $X_{8,6}=1$ | 519 | $X_{8,1}=1$ | 524 | $X_{8,2}=1$ | 513 |
| $X_{9,6}=1$ | 522 | $X_{9,1}=1$ | 524 | $X_{9,2}=1$ | 514 |
| $X_{10,6}=1$ | 527 | $X_{10,1}=1$ | 514 | $X_{10,2}=1$ | 513 |
| $X_{11,6}=1$ | 506 | $X_{11,1}=1$ | 527 | $X_{11,2}=1$ | 510 |
| $X_{12,6}=1$ | 504 | $X_{12,1}=1$ | 510 | $X_{12,2}=1$ | 516 |
| mean | 519.9 | mean | 515.3 | mean | 512.3 |

Table 4: Results for the first level branching for Nug12

# 4 Conclusion

We have presented new bounds for $\boldsymbol{QAP}$ that are based on a matrix-lifting (rather than a vector-lifting) semidefinite relaxation. By exploiting the special doubly stochastic and orthogonality structure of the constraints, we obtained a series of cuts to further strengthen the relaxation. The resulting relaxation $\mathbf{MSDR}_3$ is provably stronger than the projected eigenvalue bound $\boldsymbol{PB}$ , and is comparable with the $\boldsymbol{SDR}\mathbf{1}$ bound and the quadratic programming bound $\boldsymbol{QPB}$ in our empirical tests. Moreover, due to the matrix-lifting property of the bound, it only use $O(n^2)$ variables and $O(n^2)$ constraints. Hence the complexity is comparable with e.g. relaxations based on $\boldsymbol{LP}$ .

Our $\mathbf{MSDR}_3$ relaxation and bound is in particularly efficient for matrices with special structure, e.g., if $B$ is a Hamming distance matrix of a hypercube or a Manhattan distance matrix from rectangular grids, e.g., [22].

Based on our work, additional new relaxations have been proposed which have strong theoretical properties, e.g., the bound ***OB2*** in [33]. Another recent application is decoding in multiple antenna system, see [23].

# References


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

[2] K.M. ANSTREICHER, N.W. BRIXIUS, J.-P. GOUX, and J. LINDEROTH. Solving large quadratic assignment problems on computational grids. *Math. Program.*, 91(3, Ser. A):563–588, 2002.

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

[4] A. BECK. Quadratic matrix programming. *SIAM J. Optim.*, to appear:– (electronic), 2006.

[5] A. BECK and M. TEBOULLE. Global optimality conditions for quadratic optimization problems with binary constraints. *SIAM J. Optim.*, 11(1):179–188 (electronic), 2000.

[6] R.E. BURKARD, S. KARISCH, and F. RENDL. QAPLIB – a quadratic assignment problem library. *European J. Oper. Res.*, 55:115–119, 1991. www.opt.math.tu-graz.ac.at/qaplib/.

[7] Z. DREZNER. Lower bounds based on linear programming for the quadratic assignment problem. *Comput. Optim. Appl.*, 4(2):159–165, 1995.

[8] J. FALKNER, F. RENDL, and H. WOLKOWICZ. A computational study of graph partitioning. *Math. Programming*, 66(2, Ser. A):211–239, 1994.

[9] G. FINKE, R.E. BURKARD, and F. RENDL. Quadratic assignment problems. *Ann. Discrete Math.*, 31:61–82, 1987.

[10] P.C. GILMORE. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal on Applied Mathematics*, 10:305–313, 1962.

[11] S.W. HADLEY, F. RENDL, and H. WOLKOWICZ. Bounds for the quadratic assignment problems using continuous optimization. In *Integer Programming and Combinatorial Optimization*, pages 237–248, Waterloo, Ontario, Canada, 1990. University of Waterloo Press.

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

[13] S.W. HADLEY, F. RENDL, and H. WOLKOWICZ. Symmetrization of nonsymmetric quadratic assignment problems and the Hoffman-Wielandt inequality. *Linear Algebra Appl.*, 167:53–64, 1992. Sixth Haifa Conference on Matrix Theory (Haifa, 1990).

[14] P. HAHN and T. GRANT. A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method. Paper, Sci-Tech Services, 1416 Park Rd, Elverson, PA 19520, 1998.

[15] P. HAHN and T. GRANT. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Oper. Res.*, 46(6):912–922, 1998.

[16] G.H. HARDY, J.E. LITTLEWOOD, and G. POLYA. *Inequalities*. Cambridge Univesity Press, London and New York, 1934. 2nd edition 1952.

[17] R.A. HORN and C.R. JOHNSON. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[18] S.E. KARISCH, E. ÇELA, J. CLAUSEN, and T. ESPERSEN. A dual framework for lower bounds of the quadratic assignment problem based on linearization. *Computing*, 63(4):351–403, 1999.

[19] S.E. KARISCH, F. RENDL, and H. WOLKOWICZ. Trust regions and relaxations for the quadratic assignment problem. In *Quadratic assignment and related problems (New Brunswick, NJ, 1993)*, pages 199–219. Amer. Math. Soc., Providence, RI, 1994.

[20] E. LAWLER. The quadratic assignment problem. *Management Sci.*, 9:586–599, 1963.

[21] A.W. MARSHALL and I. OLKIN. *Inequalities: Theory of Majorization and its Applications*. Academic Press, New York, NY, 1979.

[22] H. MITTELMANN and J. PENG. Estimating bounds for quadratic assignment problems associated with hamming and manhattan distance matrices based on semidefinite programming. *Private Communications*, 2007.

[23] A. MOBASHER and A.K. KHANDANI. Matrix-lifting semi-definite programming for decoding in multuple antenna systems. *The 10th Canadian Workshop on Information Theory (CWIT'07), Edmonton, Alberta, Canada*, 2007.

[24] R.D.C. MONTEIRO and M.J. TODD. Path-following methods. In *Handbook of Semidefinite Programming*, pages 267–306. Kluwer Acad. Publ., Boston, MA, 2000.

[25] Y.E. NESTEROV, H. WOLKOWICZ, and Y. YE. Semidefinite programming relaxations of nonconvex quadratic optimization. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 361–419. Kluwer Acad. Publ., Boston, MA, 2000.

[26] C.E. NUGENT, T.E. VOLLMAN, and J. RUML. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[27] P. PARDALOS, F. RENDL, and H. WOLKOWICZ. The quadratic assignment problem: a survey and recent developments. In P. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems (New Brunswick, NJ, 1993)*, pages 1–42. Amer. Math. Soc., Providence, RI, 1994.

[28] P. PARDALOS and H. WOLKOWICZ, editors. *Quadratic assignment and related problems*. American Mathematical Society, Providence, RI, 1994. Papers from the workshop held at Rutgers University, New Brunswick, New Jersey, May 20–21, 1993.

[29] P.M. PARDALOS, K.G. RAMAKRISHNAN, M.G.C. RESENDE, and Y. LI. Implementation of a variance reduction-based lower bound in a branch-and-bound algorithm for the quadratic assignment problem. *SIAM J. Optim.*, 7(1):280–294, 1997.

[30] F. RENDL and R. SOTIROV. Bounds for the quadratic assignment problem using the bundle method. *Math. Programming*, to appear.

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

[32] H. WOLKOWICZ. Semidefinite programming approaches to the quadratic assignment problem. In *Nonlinear assignment problems*, volume 7 of *Comb. Optim.*, pages 143–174. Kluwer Acad. Publ., Dordrecht, 2000.

[33] Y. XIA. New semidefinite relaxations for the quadratic assignment problem. *Private Communications*, 2007.

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