A New Relaxation Framework for Quadratic Assignment Problems based on Matrix Splitting

Jiming Peng * Hans Mittelmann† Xiaoxue Li‡

January 13, 2010

Abstract

Quadratic assignment problems (QAPs) are known to be among the hardest discrete optimization problems. Recent study shows that even obtaining a strong lower bound for QAPs is a computational challenge. In this paper, we first discuss how to construct new simple convex relaxations of QAPs based on various matrix splitting schemes. Then we introduce the so-called symmetric mappings that can be used to derive strong cuts for the proposed relaxation model. We show that the bounds based on the new models are comparable to some strong bounds in the literature. Promising experimental results based on the new relaxations will be reported.

Key words. Quadratic Assignment Problem (QAP), Semidefinite Programming (SDP), Matrix Splitting, Relaxation, Lower Bound.

*Corresponding author. Department of Industrial and Enterprise System Engineering , University of Illinois at Urbana-Champaign. Urbana, IL, 61801.

†School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287-1804, USA. Email:mittelmann@math.asu.edu.

‡Department of Industrial and Enterprise System Engineering , University of Illinois at Urbana-Champaign. Urbana, IL, 61801.

1 Introduction

The standard quadratic assignment problem takes the following form

$$\min_{X \in \Pi} \mathrm{Tr}\big(\mathrm{AXBX}^{\mathrm{T}}\big) \tag{1}$$

where $A, B \in \Re^{n \times n}$, and Π is the set of permutation matrices. This problem was first introduced by Koopmans and Beckmann [19] for facility location. The model covers many scenarios arising from various applications such as in chip design [15], image processing [29], and communications [5, 26]. For more applications of QAPs, we refer to the survey papers [7, 21] where many interesting QAPs from numerous fields are listed. Due to its broad range of applications, the study on QAPs has caught a great amount of attention of many experts from different fields [21]. It is known that QAPs are among the hardest discrete optimization problems. For example, a QAP with $n = 30$ is typically recognized as a great computational challenge [4].

A popular technique for finding the exact solution of the QAPs is branch and bound (B&B) method. Crucial in a typical B&B approach is how a strong bound can be computed at a relatively cheap cost. Various relaxations and bounds for QAPs have been proposed in the literature. Roughly speaking, these bounds can be categorized into two groups. The first group includes several bounds that are not very strong but can be computed efficiently such as the well-known Gilmore-Lawler bound (GLB)[11], the bound based on projection [13] (denoted by PB) and the bound based on convex quadratic programming (denoted by QPB) [3]. The second group contains strong bounds that require expensive computation such as the bounds derived from lifted integer linear programming [1, 14] and bounds based on SDP relaxation [31, 27].

In this paper, we are particularly interested in bounds based on SDP relaxations. Note that if both A and B are symmetric, by using the *Kronecker-product*, we have

$$\mathrm{Tr}\big(\mathrm{AXBX}^{\mathrm{T}}\big) = \mathrm{x}^{\mathrm{T}}(\mathrm{B} \otimes \mathrm{A})\mathrm{x} = \mathrm{Tr}\big((\mathrm{B} \otimes \mathrm{A})\mathrm{xx}^{\mathrm{T}}\big), \quad \mathrm{x} = \mathrm{vec}(\mathrm{X})$$

where vec(X) is obtained from X by stacking its columns into a vector of order n^2. Many existing SDP relaxations of QAPs are derived by relaxing the rank-1 matrix xx^T to be positive semidefinite with additional constraints on the matrix elements. As pointed out in [21, 27], the SDP bounds are tighter compared with bounds based on other relaxations, but usually much more expensive to compute due to the large number $O(n^4)$ of variables and constraints in the standard SDP relaxation induced by the vectorization of the permutation matrix X. To release such a computational challenge, Burer et al [6] adopted an augmented Lagrangian method to solve the relaxed SDP. In [27], Rendl and Sotirov applied the bundle method to solve the SDP relaxation of QAPs. In [18], Klerk and Sotirov exploited the symmetry in the underlying problem to compute the lower bound derived from the standard SDP relaxation. These works improved the computational process of estimating the lower bound of QAPs based on the underlying models, but they did not reduce the complexity of the relaxation model itself or had to refer to other expensive procedures to reduce the complexity of the relaxation model. The huge number (n^4) of variables and constraints in these expensive relaxations makes it very hard to solve these expensive relaxations for a medium size QAP with the current computational facility.

Recently Ding and Wolkowicz [8] introduced a new SDP relaxation of QAP based on matrix lifting, which is based on the following fact

$$\begin{pmatrix} I \\ X \\ XB \end{pmatrix} (I\ X^T\ BX^T) \succeq 0.$$

Since X is an assignment matrix, we can derive the following SDP relaxation

$$\begin{array}{ll} \min & \mathrm{Tr(AY)} \\ & \begin{pmatrix} I & X^T & Z^T \\ X & I & Y \\ Z & Y & W \end{pmatrix} \succeq 0, \quad Z = XB; \\ & \mathrm{diag}\,(Y) = X \mathrm{diag}\,(B) \quad Ye = XBe, \\ & \mathrm{diag}\,(W) = X \mathrm{diag}\,(B^2), \quad We = XB^2 e, \\ & Xe = X^T e = e, \quad X \geq 0. \end{array} \tag{2}$$

In [8], additional cuts based on the singular value decomposition of the matrix A are added to strengthen the relaxation. The model in [8] has only n^2 constraints and thus can be solved by using open source SDP solvers for QAPs of size $n \leq 30$, though it still remains a computational challenge for $n \geq 30$. In our recent work [24], we proposed a new SDP relaxation for special classes of QAPs associated with the Hamming distance matrix of a hypercube or the Manhattan distance matrix of rectangular grids. It has been observed that the lower bound based on the new relaxation can be computed efficiently and is rather strong compared with other existing bounds in the literature for the underlying QAPs.

The model in [24] is based on the following observation: the eigenvalues of the matrix XBX^T are independent of the permutation matrix X. Moreover, based on the special structure of the matrices in the underlying QAPs, we can decompose the matrix XBX^T into two parts, i.e., $XBX^T = \alpha E - Y$ where α is a parameter defined by B, $E = ee^T$ where $e \in \Re^n$ is a vector whose elements have value 1, and Y is a positive semidefinite matrix. By exploring the intrinsic relations among Y, E and B, we have derived a simple and concise relaxation for the underlying QAPs that can be solved efficiently by open-source SDP solvers and the obtained lower bound has been observed to be very strong.

One limitation of the approach in [24] is that it depends heavily on the structure of the data matrix B and thus can be applied only to special classes of QAPs. The main purpose of the present paper is to introduce a new framework of relaxing generic QAPs to convex optimization problems. The models are based on matrix splitting, which can be viewed as a generalization of the decomposition technique introduced in [24]. Secondly, we also introduce a new class of mappings, the so-called symmetric mappings, which can be used to construct strong cuts for the relaxation models proposed in this paper.

The paper is organized as follows. In Section 2, we first introduce generic SDP relaxation models of QAPs based on matrix splitting and then we present several matrix splitting schemes resulting in numerous SDP relaxations of QAPs. In Section 3, we discuss how to enhance the SDP relaxation by using strong cuts derived by applying the so-called symmetric mapping to the underlying QAP. A procedure to generate new convex relaxations of QAPs is introduced in Section 4 and numerical experiments based on the new models will be reported in Section 5. We conclude our paper with some remarks in Section 6.

A few sentences about the notation. Throughout this paper, we use upper case letters to denote matrices and lower case letters for vectors. For a given symmetric matrix B, $\operatorname{diag}(B)$ denotes the vector consisting of the diagonal elements of B. For a vector d, $\operatorname{diag}(d)$ denotes the diagonal matrix whose (i,i)-element is d_i. For a given matrix B, $|B|$ denotes the matrix whose elements take the absolute value of the corresponding element of the matrix B at the same position, i.e.,

$$|B| = [|b|_{ij}], \quad |b|_{ij} = |b_{ij}|, \quad i,j = 1, \cdots, n. \tag{3}$$

B_{off} denotes the matrix consisting of all the off-diagonal elements of B, i.e., $B_{off} = B - \operatorname{diag}(b_{11}, b_{22}, \cdots, b_{nn})$. $\max(B)$ (or $\min(B)$) denotes the vector whose i-th component is the maximal element (or minimal element) in the i-th row (denoted by $B_{i,:}$) of B. $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$ denote the largest eigenvalue and the smallest eigenvalue, respectively.

2 New SDP Relaxations for QAPs based on Matrix Splitting

In this section, we first describe how to derive SDP relaxations of QAPs based on various matrix splitting schemes. The section consists of three parts. In the first subsection, we introduce two generic frameworks for deriving SDP relaxations of QAPs based on various matrix splitting schemes. In the second part we present two new positive semidefinite (PSD) splitting schemes. In the last subsection, we introduce the so-called non-PSD splitting scheme, and discuss how to derive SDP relaxations of QAPs based on non-PSD splitting scheme.

2.1 Two generic SDP relaxation models for QAPs based on matrix splitting

As pointed out in the introduction, various matrix splitting techniques have been widely used to solve linear systems of equations and other classes of problems. Several splitting schemes such as Jacobi, Gauss-Seidel, and successive overrelaxation (SOR) methods have been well studied [12]. However, these schemes can not be directly used to derive SDP relaxations of QAPs.

To construct new SDP relaxations of QAPs, let us recall the fact that for any permutation matrix X and any matrix B, the matrix XBX^T has the same set of eigenvalues as that of B. Moreover, if B is positive semidefinite, so is XBX^T. Let us denote $Y = XBX^T$. Then we can impose the constraint $Y \succeq 0$ if and only if $B \succeq 0$. This gives a straightforward way to relax the original QAP as an SDP problem. To deal with the case when B might not be positive semidefinite, we first introduce the following definition.

Definition 2.1. *The matrix pair* (B^+, B^-) *is called a positive semidefinite splitting of the matrix* B *if it satisfies the following relation*

$$B = B^+ - B^-, \quad B^+, B^- \succeq 0. \tag{4}$$

The above definition indicates that there exist many PSD splitting schemes. To see this, let us assume that (B^+, B^-) is a PSD splitting of B. Then for any positive semidefinite matrix M, the matrix pair $(B^+ + M, B^- + M)$ is also a PSD splitting of B. From Definition 2.1 one can conclude that for a given matrix B and a permutation matrix X, the matrix pair (B^+, B^-) is a PSD splitting of B if and only if the matrix pair $(Y^+ = XB^+X^T, Y^- = XB^-X^T)$ is a PSD splitting of the matrix XBX^T.

If a PSD splitting of the matrix B is available, then we can derive the following SDP relaxation

$$\begin{array}{ll} \min & \mathrm{Tr}\big(\mathrm{A}(\mathrm{Y}^+ - \mathrm{Y}^-)\big) \\ s.t. & \text{constraints on } Y^+, Y^- \text{ and } X. \end{array} \tag{5}$$

An alternative to the scheme (4) is to choose a splitting that satisfies the following conditions

$$B = B^+ - B^-, \quad B^- \succeq 0; \qquad X^T B^+ X = B^+, \forall X \in \Pi. \tag{6}$$

Suppose that the matrix B can be split into a matrix pair (B^+, B^-) that satisfies relation (6), then we obtain the following SDP relaxation

$$\begin{array}{ll} \min & \mathrm{Tr}\big(\mathrm{A}(\mathrm{B}^+ - \mathrm{Y}^-)\big) \\ s.t. & \text{constraints on } Y^- \text{ and } X. \end{array} \tag{7}$$

In what follows we present several PSD matrix splitting schemes that can be used in the above two models. We leave the discussion on how to specify the constraints in the above models to later Sections.

2.2 Two new PSD matrix splitting schemes

In this subsection we introduce two new matrix splitting schemes that can help us to construct SDP relaxations of QAPs. We start with a very specific splitting scheme that has been used in our early work [24]. For a given matrix B, let us consider the following splitting

$$B = \alpha E - B^-, \quad B^- \succeq 0, \alpha \geq 0. \tag{8}$$

Such a splitting scheme is particularly attractive, because for any permutation matrix X, we have

$$\mathrm{Tr}\big(\mathrm{AXBX^T}\big) = \mathrm{Tr}\big(\mathrm{AX}(\alpha\mathrm{E} - \mathrm{B^-})\mathrm{X^T}\big) = \alpha\mathrm{Tr}\big(\mathrm{e^T Ae}\big) - \mathrm{Tr}\big(\mathrm{AXB^- X^T}\big).$$

In such a case, it suffices to focus only on the matrix B^- to obtain some simple yet strong relaxations of the underlying QAP. We also point out that in [24] we proved that when the matrix B in the underlying QAPs is the Hamming distance matrix of a hypercube or the Manhattan distance matrix of rectangular grids, then it can be split in the way above with a carefully chosen parameter α defined by the associated hypercube or the rectangular grids.

It will be interesting to investigate under what conditions, the matrix B can be split as in (8). For this purpose, we propose to solve the following auxiliary SDP

$$\begin{array}{ll} \min & t \\ s.t. & tE - B \succeq 0, t \geq 0. \end{array} \tag{9}$$

If the above SDP is feasible, then we can use $\alpha = t^*$, the optimal value of the above optimization problem, to split the matrix B into two parts. In the next subsections we shall discuss how to cope with the scenario when the problem (9) is infeasible.

2.2.1 Matrix splitting based on singular value decomposition

As mentioned earlier, if the problem (9) is infeasible, then we can not use the splitting like $B = \alpha E - B^-$ with $\alpha \geq 0$. In this subsection, we discuss another special PSD splitting of B based on the singular value decomposition of B. Let Q be an orthogonal matrix whose columns are the eigenvectors of the matrix B associated with the eigenvalues $\{\lambda_1, \cdots, \lambda_n\}$, i.e., $B = \sum_{i=1}^n \lambda_i q_i q_i^T$ where q_i is the i-th column of Q. Let us denote

$$B^+ = \sum_{i:\lambda_i \geq 0} \lambda_i q_i q_i^T, \quad B^- = -\sum_{i:\lambda_i < 0} \lambda_i q_i q_i^T.$$

We then have

$$B = B^+ - B^-, \quad \mathrm{Tr}\big(\mathrm{B^+ B^-}\big) = 0, \quad \mathrm{B^+, B^-} \succeq 0. \tag{10}$$

Similarly we call (B^+, B^-) the orthogonal PSD splitting of B. From (10) one can easily see that for any permutation matrix X, the matrix pair (XB^+X^T, XB^-X^T) is an orthogonal PSD splitting of the matrix XBX^T.

We remark that in general an orthogonal PSD splitting of form (8) may not exist. However, as observed in [24], when B is the Hamming distance matrix of a hypercube in a certain space, then these two splitting schemes are identical.

Let (B^+, B^-) be an orthogonal PSD splitting of B and let us define

$$Y^+ = XB^+X^T, \quad Y^- = XB^-X^T.$$

It follows from relation (10) that

$$\mathrm{Tr}\big(\mathrm{Y}^{+}\mathrm{Y}^{-}\big) = 0, \quad \mathrm{Y}^{+} \succeq 0, \mathrm{Y}^{-} \succeq 0.$$

From the above relation we obtain

$$\|Y_{i,:}^{+} + Y_{i,:}^{-}\| = \|Y_{i,:}\|, \quad i = 1, \cdots, n. \tag{11}$$

where $\|\cdot\|$ is the 2-norm of a vector. The above relation can be used to construct valid constraints that can strengthen the lower bound.

2.2.2 Matrix splitting based on Laplace operator

As pointed out earlier, there are various matrix splitting schemes to represent a given matrix B as the difference of two positive semidefinite matrices, i.e., $B = B^{+} - B^{-}$. In this section we focus on a special case where B^{+} is a diagonal matrix. For this, we need some basic concepts that are widely used in graph modelling and theory. Suppose B is the adjacency matrix of some completely connected graph (G). The so-called Laplace matrix of the graph G is defined by

$$L(G) = D - B, \quad D = diag(d_1, \cdots, d_n), \quad d_i = \sum_{j=1}^{n} b_{ij}, \quad i = 1, \cdots, n. \tag{12}$$

It is easy to see that $L(G)$ is positive semidefinite, and e is an eigenvector corresponding to its smallest eigenvalue 0. A well-known result in graph theory by Fiedler [9, 10] states that the graph G is connected if and only if the second smallest eigenvalue (denoted by $\alpha(G)$) of $L(G)$ is positive. Note that we still have $L(G) \succeq 0$ and $L(G)e = 0$ when B is a distance matrix of some graph. In particular, it is straightforward to prove the following result.

Theorem 2.2. *Suppose that B is a distance matrix of a graph G and $\alpha(G) > 0$ is the second smallest eigenvalue of the matrix $L(G)$. For any $X \in \Pi$, we have $XL(G)X^T - \alpha(G)(I - \frac{1}{n}E) \succeq 0$.*

Note that the above theorem can be extended to the scenario that B is not associated with a particular graph. To see this, recall that the matrix $L(G)$ is positive semidefinite whenever the matrix B has only nonnegative elements. We next argue that without loss of generality, we can always assume that B does not contain negative elements. In case B has negative elements, we can consider a new data matrix $\bar{B} = B + \alpha E$ for sufficiently large constant α such that all the elements of the new matrix $\bar{B}$ are nonnegative. Due to the choice of $\bar{B}$, it is easy to verify that the solution sets of problem (1) with two matrices B and $\bar{B}$ are identical. Similar arguments can also be applied to matrix A. Therefore, in the remaining part of this subsection, we assume that both matrices in problem (1) have only nonnegative elements.

Now let us define

$$B^{+} = D, B^{-} = L(G), \quad Y^{+} = XB^{+}X^{T}, Y^{-} = XB^{-}X^{T},$$

where D and $L(G)$ are defined by (12). It follows from Theorem 2.2 that

$$Y^{-} - \alpha(G)(I - \frac{1}{n}E) \succeq 0. \qquad (13)$$

The above constraint can be added to model (5) to improve its bound.

We remark that the matrix splitting scheme based on the Laplace operator can also be used to enhance some ILP reformulations of QAPs. For example, when B is the adjacency matrix of a hypercube in a finite space, in [23] we had derived a new ILP reformulation of the original QAP based on the graphical features of the associated hypercube. Since the hypercube is completely connected, we can add a simple SDP constraint to the existing ILP model. Similar ideas can be applied to all QAPs where the matrix B is the adjacency matrix of a connected graph.

2.3 Non-positive semi-definite (Non-PSD) matrix splitting

In the previous subsections, we have discussed how to represent a matrix B as the difference of two positive semidefinite matrices. A particular case of interest is the splitting $B = \alpha E - B^{-}$ where α is a carefully chosen parameter depending on B, and B^{-} is positive semi-definite. Unfortunately, such a highly desirable splitting is impossible for many QAPs unless the associate matrix B enjoys certain properties such as the Hamming distance matrix or the Manhattan distance matrix [24]. In this subsection, we introduce a similar matrix splitting scheme $B = \alpha(E - I) - B^{-}$, where α is a parameter to be determined and B^{-} is positive semidefinite. Note that such a desirable splitting scheme might not exist for generic QAPs. In this subsection we concern mainly about QAPs that satisfy the following condition

Assumption 2.3. *At least one matrix (A or B) in the underlying QAP has zeros on its diagonal.*

It should be pointed out that the above assumption is quite reasonable and most QAP instances from the QAP library indeed satisfy such a condition. For convenience of discussion, in the remaining part of this section, we assume the matrix A has zeros on its diagonal.

Under Assumption 2.3, we can easily show that for any diagonal matrix D, one has

$$\mathrm{Tr}\big(\mathrm{AXBX^T}\big) = \mathrm{Tr}\big(\mathrm{AX(B-D)X^T}\big), \quad \forall \mathrm{X} \in \Pi.$$

Based on the above relation, we propose to split the matrix B into the following form

$$B - D = \alpha(E - I) - B^{-}, \quad B^{-} = D + \alpha(E - I) - B \succeq 0, \qquad (14)$$

where D is a diagonal matrix to be found and α is a parameter. A critical issue here is how to choose appropriate α and D so that the resulting relaxation can provide a strong bound. Note that there are two issues to be addressed in such a splitting scheme. First, since the matrix $E - I$ is invariant under any permutation, a large value of the parameter α is desirable. On the other hand, the diagonal matrix D should have elements as small as possible, since otherwise, the relaxation might become too loose. Taking the above two issues into consideration, we propose to solve the following optimization problem

$$\min_{d} \quad -\alpha + \sum_{i=1}^{n} d_i \tag{15}$$

$$s.t. \quad D + \alpha(E - I) - B \succeq 0, \quad D = \mathrm{diag}\,(d). \tag{16}$$

It should be pointed out that for simplicity of the model, we can also impose the constraint that $D = \beta I$ for some parameter $\beta > 0$.

3 Symmetric Mapping and Valid Constraints of QAPs

In this section we discuss how to find strong cuts that can strengthen the relaxation model (5), or in other words, to construct tighter valid constraints on the elements of $Y = XBX^T$. For this, we first recall the so-called symmetric functions [22]:

Definition 3.1. *A function $f(v) : \Re^n \to \Re$ is said to be symmetric if for any permutation matrix $X \in \Pi$, the relation $f(v) = f(Xv)$ holds.*

There are many symmetric functions. Two simple symmetric functions are the minimum and maximum function defined by

$$\min(v) = \min_{i \in \{1,\cdots,n\}} v_i, \quad \max(v) = \max_{i \in \{1,\cdots,n\}} v_i.$$

Moreover, it is easy to see that for any univariate function $h(\cdot)$, the additive function $H(v) = \sum_{i=1}^n h(v_i)$ is also symmetric. For example, for $v \in \Re^n$, its L_p norm

$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p\right)^{\frac{1}{p}}$$

is also symmetric. For a given matrix M and a function $f : \Re^n \to \Re$, we define the following mapping

$$\mathcal{F}(M) = \begin{pmatrix} f(M_{1,:}) \\ f(M_{2,:}) \\ \vdots \\ f(M_{n,:}) \end{pmatrix} \tag{17}$$

where $M_{i,:}$ denotes the i-th row of M. Similarly we call the mapping $\mathcal{F}(M)$ symmetric if $f(\cdot)$ is symmetric. Now we are ready to state the first main result in this section.

Theorem 3.2. *Suppose that the mapping $\mathcal{F}$ defined by (17) be symmetric. Then for every $X \in \Pi$ and $M \in \Re^{n\times n}$, we have*

$$\mathcal{F}(XMX^T) = X\mathcal{F}(M). \tag{18}$$

Proof. Since $X \in \Pi$, the i-th row $Y_{i,:}$ of the matrix $Y = XMX^T$ has the same set of elements as the i-th row of the matrix XM. Due to the symmetry of f, we have

$$f(Y_{i,:}) = f((XM)_{i,:}),$$

which further yields the conclusion in the theorem. □

Theorem 3.2 allows us to add tight constraints to the relaxation model (5). To see this, recall the fact that $Y = XBX^T$. Therefore, for any mapping $\mathcal{F}$ defined by (17) with a symmetric function f, we have

$$\mathcal{F}(Y) = X\mathcal{F}(B).$$

If we further choose f to be a convex function, e.g., $f(v) = \|v\|_2^2$, then the above relation can be relaxed to

$$f(Y_{i,:}) \leq (X\mathcal{F}(B))_i, \quad \forall i = 1, \cdots, n.$$

For concave symmetric f, the relation (18) can be relaxed to

$$f(Y_{i,:}) \geq (X\mathcal{F}(B))_i, \quad \forall i = 1, \cdots, n.$$

It should be mentioned that in our early work [24], we have used three symmetric functions $\min(v), \max(v)$ and $f(v) = \sum_{i=1}^n v_i$ to construct valid constraints.

We next discuss how to add additional cuts based on the L_p norm for some $p > 1$. For simplicity of discussion, let us assume that B is a matrix with some negative elements. Let us recall the notation $|B|$ and let $Z = X\,|B|\,X^T$. We then have

$$-z_{ij} \leq y_{ij} \leq z_{ij}, \qquad \forall i, j = 1, \cdots, n; \tag{19}$$

$$\operatorname{diag}(Z) = X\operatorname{diag}(|B|), \qquad \mathcal{L}_p(Z) = X\mathcal{L}_p(|B|). \tag{20}$$

On the other hand, one can easily verify the following relation:

$$(X\min(|B_{off}|))_i \leq z_{ij} \leq (X\max(|B|))_i, \quad \forall i \neq j = 1, \cdots, n.$$

Moreover, the matrix $Z-Y$ has the same spectrum as that of the matrix $|B|-B$. Let $\lambda_{\max}(|B|-B)$ and $\lambda_{\min}(|B|-B)$ be the maximal and minimal eigenvalues of $|B|$, respectively. Then we have

$$\lambda_{\max}(|B|-B)I - Z + Y \succeq 0, \quad Z - Y - \lambda_{\min}(|B|-B)I \succeq 0. \tag{21}$$

4 A Procedure for Constructing SDP Relaxations of QAPs

In this section, we describe a procedure for constructing the convex relaxation of QAPs, which combine all the ideas discussed in Sections 2 and 3 to derive the SDP relaxation of QAPs. We remind the readers that B_{off} denotes the matrix consisting of all the off-diagonal elements of B.

Constructing Convex Relaxations of QAP

- **S.1** Choose a splitting scheme to separate a selected matrix, i.e., $B = B^+ - B^-$ so that B^- is positive semidefinite and B^+ is either invariant under permutation or positive semidefinite;
- **S.2** Set $Y^+ = XB^+X^T, Z^+ = X|B^+_{off}|X^T, Y^- = X^TB^-X^T$ and $Z^- = X|B^-_{off}|X^T$. Add the SDP constraints for matrices Y^+, Y^-, Z^+ and Z^- as described in the previous sections;
- **S.3** Select the symmetric mappings $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_k$;
- **S.4** Relax the constraints $\mathcal{F}_i(Y^+) = X\mathcal{F}_i(B^+), \mathcal{F}_i(Z^+) = X\mathcal{F}_i(|B^+|)$ for all the selected symmetric mappings, and relax similar constraints for Y^- and Z^- too.

It should be pointed out that in the first step S.1 of the procedure, we might need to solve different problems depending on the selected splitting schemes. For example, if we choose the splitting scheme based on SVD, then we need to perform the SVD for the selected matrix. If the PSD splitting based on Laplace operator is chosen, then we need to find the second smallest eigenvalue of the corresponding Laplace matrix. If we choose the non-PSD splitting described, we then need to solve problem (15).

We next give two examples of the SDP relaxation model generated by the above procedure. Let us choose the splitting scheme defined by (14), which requires to solve problem (15) to find the parameter α and the diagonal matrix D. In total four mappings based on symmetric functions including the minimum, the maximum, the L_1 norm and the square of the L_2 norm are used to construct the relaxation model. For notational convenience, we use $\mathcal{L}_2$ to denote the mapping defined by (17) with the L_2 norm. The resulting SDP relaxation of the original QAP can be written as the following:

$$
\begin{aligned}
\min \quad & \alpha e^T Ae - \mathrm{Tr}\big(\mathrm{AY}^-\big) && (22)\\
s.t. \quad & Y^- - XB^-X^T \succeq 0; && (23)\\
& \lambda_{\max}(\big|B^-\big| - B)I - Z^- + Y^- \succeq 0,\ Z^- - Y^- - \lambda_{\min}(\big|B^-\big| - B)I \succeq 0; && (24)\\
& \mathrm{diag}\,(Y^-) = X\mathrm{diag}\,(B^-), \quad Y^-e = XB^-e; && (25)\\
& \mathrm{diag}\,(Z^-) = 0, \quad Z^-e = X|B^-_{off}|e; && (26)\\
& (X\min(B^-_{off}))_i \le y^-_{i,j} \le (X\max(B^-_{off}))_i, \qquad \forall i \ne j; && (27)\\
& -z^-_{ij} \le y^-_{ij} \le z^-_{ij}, \qquad \forall i \ne j; && (28)\\
& (X\min(|B^-_{off}|))_i \le z^-_{ij} \le (X\max(|B^-_{off}|))_i, \quad \forall i \ne j; && (29)\\
& \mathcal{L}_2(Z^-) \le X\mathcal{L}_2(|B^-_{off}|); && (30)\\
& X \ge 0, \quad Xe = X^Te = e. && (31)
\end{aligned}
$$

Note that the SDP constraint (23) in the above can also be simplified to

$$Y^- \succeq 0. \tag{32}$$

Correspondingly, we call the resulting SDP the simplified model in this work. On the other hand, we can also apply some PSD splitting scheme to the matrix Z^- to get tighter SDP constraints on Z^- than the relation (24). Adding these extra SDP constraints can improve the bound to some extent. However, since there are many different options here and it is hard to find the best choice among all these options, we thus use only the above-described basic model.

We next present the SDP relaxation model derived from the so-called orthogonal PSD matrix splitting based on the SVD of the associated matrices. Let (B^+, B^-) be an orthogonal PSD splitting of B and $Y^+ = XB^+X^T, Y^- = XB^-X^T$. For notational convenience, we also define $\tilde{B} = B^+ + B^-$ and $\tilde{Y} = Y^+ + Y^-$. Again, we use the following symmetric mappings $\max, \min, \mathcal{L}_1$ and $\mathcal{L}_2$ defined by (17). The resulting SDP model can be described as follows:

$$
\begin{aligned}
\min \quad & \mathrm{Tr}\big(\mathrm{A}(\mathrm{Y}^+ - \mathrm{Y}^-)\big) && (33)\\
s.t. \quad & Y = Y^+ - Y^-, \quad \tilde{Y} = Y^+ + Y^-; && (34)\\
& Y^+ - XB^+X^T \succeq 0, \quad Y^- - XB^-X^T \succeq 0; && (35)\\
& \mathrm{diag}\,(Y^+) = X\mathrm{diag}\,(B^+), \quad Y^+e = XB^+e; && (36)\\
& \mathrm{diag}\,(Y^-) = X\mathrm{diag}\,(B^-), \quad Y^-e = XB^-e; && (37)\\
& (X\min(B^+))_i \le y^+_{i,j} \le (X\max(B^+))_i, \qquad \forall i \ne j; && (38)\\
& (X\min(B^-))_i \le y^-_{i,j} \le (X\max(B^-))_i, \qquad \forall i \ne j; && (39)\\
& (X\min(B))_i \le y_{ij} \le (X\max(B))_i, \quad \forall i \ne j; && (40)\\
& (X\min(\bar{B}))_i \le \tilde{y}_{ij} \le (X\max(\bar{B}))_i, \quad \forall i \ne j; && (41)\\
& \mathcal{L}_2(Y^+) \le X\mathcal{L}_2(B^+), \quad \mathcal{L}_2(Y^-) \le X\mathcal{L}_2(B^-); && (42)\\
& \mathcal{L}_2(Y) \le X\mathcal{L}_2(B), \quad \mathcal{L}_2(\tilde{Y}) \le X\mathcal{L}_2(\tilde{B}); && (43)\\
& X \ge 0, \quad Xe = X^Te = e. && (44)
\end{aligned}
$$

It is possible to add more linear constraints similar to what we adopted in model (22)-(31) based on non-PSD splitting. However, for simplicity of the model, we only present the above sample model.

We note that problem (2) is precisely the so-called MSDR1 model in [8]. With a close look, one can easily verify that we have imposed tighter constraints on the matrix argument Y than in the MSDR1 model. We therefore have the following result.

Theorem 4.1. *Let μ^*_{SDP} be the lower bound derived by solving the relaxation model (22)-(31) or (33)-(44), and μ^*_{MSDR1} the bound provided by solving problem (2) (the MSDR1 model in [8]). Then we have*

$$\mu^*_{SDP} \geq \mu^*_{MSDR1}.$$

It should be pointed out that in [8], the authors had compared the bounds based on model (2) and its enhanced variants against some bounds constructed from the eigenvalues of the associated matrices A and B [13, 28], and concluded that the bounds based on (2) and its variants are stronger than the eigenvalue based bounds in [28]. Consequently, the bound provided by either model (22)-(31) or model (33)-(44) is also stronger than the bound in [28].

5 Numerical Experiments

In this section, we report some numerical results based on our new relaxation models. We examine the performance of two models: the full model (35)-(44) based on the orthogonal PSD splitting (or SVD of the associated matrix) (denoted by F-SVD) and its simplified version (denoted by S-SVD) where we simply require that $Y^+, Y^- \succeq 0$. In all cases, we swap the positions of A and B in model (5) and both bounds from the corresponding two relaxations are reported as well. Since in our early work [24] we had presented bounds for all QAP problems from QAPLIB and other references that have a Hamming distance matrix, in this section we will only report on results for other QAPLIB problems. We compare our bounds with other three bounds: GLB, PB and QPB, provided by the three cheap relaxation models respectively. We also compare our bounds with three strong bounds in the literature including the MSDR3 in [8], the LP and SDP relaxations (denoted by N, N_+ respectively) based on the lifting and projection technique [6]. In particular, we would like to point out that the SDP bound N_+ has been recognized as one of the strongest bounds for QAPs in the literature [6]. We also mention that when applied to QAPs associated with Manhattan and Hamming distance matrices, the resulting SDP relaxations based on the orthogonal PSD splitting and non-PSD splitting in this paper are stronger than the relaxation models introduced in [24] where we have compared our relaxation models with other relaxation models in the literature.

In our experiments, all the problems whose size are below 70 were solved on a 2.67GHz Intel Core 2 Quad with 8 GB memory and the larger ones on a

2.2GHz AMD Opteron dual Quad with 64 GB. We used Matlab R2009a, CVX 1.2 and the solver SDPT3 within CVX to solve all the problems.

In Table 1, we list the comparing results of the relaxation models introduced in this paper with the other three strong bounds for some QAP instances whose sizes are between 30 and 50. In all the tables in this section, we use the fat font to highlight the strongest bound among all the bounds in comparison. In the columns of F-SVD, we first list the relative gap computed as follows

$$R_{gap} = 1 - \frac{\text{Bound}}{\text{Optimal or best known feasible objective value}},$$

where the bound is provided by model (33)-(44) and the ordering of the matrices A and B is the same as given in the QAPLIB file [1]. The relative gap based on the reversed ordering of the matrices A and B are listed in the following line. The results for the columns of S-SVD and MSDR3 are listed in the same manner. In the column $N//N_+$, since the resulting bounds are independent of the ordering of the two associated matrices, we first list the relative gap provided by model N [6], followed by the results for model N_+. The CPU time to compute these bounds are also listed in seconds as well. It is worthwhile mentioning that among all the tested problems, only the instances kra30a, kra30b,ste30a, ste36c, lipa80a and lipa90b have been solved to optimality as reported in the QAP library [7].

As one can see from Table 1, in many cases, both bounds based on the F-SVD and S-SVD models are stronger than the bounds provided by MSDR3, while the CPU time to solve the F-SVD and S-SVD models are much less than those for MSDR3. The bounds provided by model N_+ is the strongest in many cases. However, for some QAP instances such as Tai30b and Tai40b, if we select the better bound based on the different ordering of the matrix pair (A, B), then the selected bound based on the F-SVD model is better than the bound from the N_+ model. This observation further implies that it is difficult (or almost impossible) to theoretically compare the bounds based on the relaxation models presented in this work and other strong relaxation models in the literature. It should be pointed out that when $n = 50$, it is extremely expensive to compute the bounds based on the N and N_+ models. Thus we did not report these two bounds for Tai50b. We also mention that when we try to solve the MSDR3 model for instance tai30b, SeDuMi terminates with the message "no sensible solution/direction found". Therefore, we mark it as a failed case in the table. Finally, we point out that the CPU time used by the simplified model to obtain a lower bound is less than half of the CPU time used by the full model, while the gap between these two bounds is marginal. This suggests for large scale QAP instances, we could use the simplified model to get a good bound.

In Table 2, we compare our bounds with three cheap bounds (GLB, PB and QPB) for QAP instances whose size is above 70. Due to memory restriction, we use only the simple model based on orthogonal PSD splitting. For sizes above

[1] We observed in our experiments that sometimes the SDP solver in CVX stops because of slow progress or a large gap. In such a case, we use the procedure described in [17] (coded by Kim-chuan Toh[30]) to find a rigorous lower bound for our relaxation model which further yields a lower bound for the underlying QAP.

	F-SVD		S-SVD		MSDR3		$N//N_+$	
Prob.	R_{gap}	CPU	R_{gap}	CPU	R_{gap}	CPU	R_{gap}	CPU
kra30a	23.81%	121	23.97%	45	24.86%	1234	14.53%	10552
	16.86%	141	17.07%	67	18.51%	1503	**2.5%**	29470
kra30b	25.43%	128	25.52%	53	26.4%	1201	16.12%	10396
	17.8%	145	18.25%	66	20%	1504	**4.07%**	28593
ste36a	35.12%	261	35.41%	122	44.89%	2129	16.84%	32006
	19.58%	256	19.96%	115	45.72%	5222	**5.32%**	164205
ste36c	29.31%	265	29.38%	126	36.93%	1994	14.75%	33790
	18.24%	344	18.35%	193	64.72%	5393	**3.92%**	228351
tai30b	85.32%	220	86.06%	66	Fail	Fail	77.9%	15022
	14.75%	183	15.74%	67	86.69%	1223	18.05%	55151
tai35b	70.36%	348	71.75%	116	82.44%	4577	65.16%	33940
	21.68%	326	22.95%	137	89.94%	3852	**15%**	208550
tai40b	69.25%	623	71.1%	239	75.14%	11141	74.05%	39356
	14.57%	576	15.45%	193	67.75%	10631	15.18%	375776
tai50b	73.74%	1719	74.17%	643	64.31%	54838	too	to
	16.86%	1485	17.58%	787	61.78%	53753	expensive	compute

Table 1: Selected Bounds for QAPs of size between 30 and 50.

150 we further remove the constraints on the matrix $\tilde{Y} = Y^+ + Y^-$ in the model to conserve memory. We also removed all the constraints derived from various valid cuts except the simple constraints on the lower and upper bounds of the elements of Y^+ and Y^-. For problem Tai256c, we keep only the lower and upper bound of the elements of Y. In Table 2, we use the asterisks to indicate these instances for which the further simplified relaxation model has been used to compute the lower bound. Since the CPU times for the GLB and PB bounds are very small, and the QPB bounds were computed by N. Brixius on a different platform, we do not list the CPU time in Table 2. However, we would like to point out that in our experiments, all the bounds in the table were obtained within twenty hours.

As one can see from Table 2 except for the two Lipa# instances where the GLB bound is the strongest, the better bound based on S-SVD selected from the two ordering of the associated matrices is the strongest. In particular, for several large scale QAP instances (Tai#b), the bounds obtained in this paper have been listed in the QAP library as the tightest lower bounds known so far. For purpose of reference, we list all these bounds in Table 3 and compare them with the best know feasible solutions for the corresponding QAP instances that have not been confirmed to be solved to optimality in the literature.

We also note that there are a few failed cases in Table 2 such as the Tai# instances. With a close look, we can find that for these instances, the two matrices A and B have completely different structure. For example, for the matrix B in the Tai# instances have many zero elements, and a few very large

	S-SVD	Other Relaxations		
Prob.	R_{gap}	GLB	PB	QPB
lipa80a	0.2% 2.48%	**0.12%**	2.44%	2.49%
lipa90b	0.3% 0.14%	**0**	0.34%	0.27%
sko81	18.8% **6.18%**	33.75%	9.58%	8.45%
sko90	18.92% **5.87%**	34.62%	8.27%	7.35%
sko100a	19.24% **5.57%**	34.9%	8.31%	7.37%
sko100d	19.71% **5.85%**	35.87%	8.42%	7.32%
tai80b	*Fail* **17.35%**	89.1%	40.25%	25.02%
tai100b	*Fail* **20.36%**	85.27%	44.05%	76.13%
tai150b*	89.83% **14.35%**	87.37%	22.32%	43.73%
tai256c*	*Fail* **2.03%**	73.22%	8.06%	8.06%
tho150*	53.51% **7.71%**	49.3%	9.62%	8.28%
wil100	10.85% **3.29%**	22.74%	4.47%	3.89%

Table 2: Selected bounds for QAPs of size above 70.

nonzero elements while all the off-diagonal elements of the matrix A are nonzero. In such a case, it is reasonable to expect that the relaxation model based on the splitting of the matrix A would give a better bound, as confirmed in our experiments. On the other hand, since we use the standard Matlab routine to perform the SVD and split the matrix, possibly the ill-conditioning in the associated matrices caused some instability for the relaxation model.

On the other hand, we note that for a given relaxation model of QAPs, the growing behavior of the lower bounds along the branching tree also plays an important role in the B&B approach. In what follows, we examine the growing behavior of the lower bounds based on the full non-PSD splitting model (22) for the problem Nug12 as in [8][2]. Table 4 shows in position (i, j) the bound that results from setting $X_{i,j} = 1$ for the test problem NUG12 whose optimal

[2]It should be mentioned that the distance matrix B in Nug# instances have a nice symmetry that can be explored to help in the branching process. In [24] we have discussed how to use such a symmetry to simplify the relaxation model and speed-up the solving process. However, for simple illustration, we do not explore such a symmetry in this paper.

Problem	L-bound	Best known	Problem	L-bound	Best known
sko81	85371	90998	sko90	108752	115534
sko100a	143532	152002	sko100d	140821	149576
tai40b	544404660	637250948	tai50b	381474035	458821517
tai80b	676424154	818415043	tai100b	944476208	1185996137
tai150b	427329160	498896643	tai256c	43849646	44759294
tho150	7506110	8133398	wil100	264059	273038

Table 3: Selected lower bounds for unsolved QAPs.

526	525	525	526	529	538	538	529	526	525	525	526
527	533	533	527	537	547	547	537	527	533	533	527
524	527	527	524	528	538	538	528	524	527	527	524
533	523	523	533	527	535	535	527	533	523	523	533
524	537	537	524	534	555	555	534	524	537	537	524
531	528	528	531	529	543	543	529	531	528	528	531
537	524	524	537	529	525	525	529	537	524	524	537
540	525	525	540	532	527	527	532	540	525	525	540
537	525	525	537	528	533	533	528	537	525	525	537
532	534	534	532	532	547	547	532	532	534	534	532
548	529	529	548	537	522	522	537	548	529	529	548
523	526	526	523	528	536	536	528	523	526	526	523

Table 4: Bounds for Nug12 when $x_{i,j} = 1$

solution has the value 578. It is worthwhile mentioning that the bound at the root of the tree is 520.

Since the max bound is at position $(5, 6)$. Following a similar idea as in [3], we set $X_{5,6} = 1$ and all the other elements in the 5-th row and 6-th column are zeros. Table 5 lists all the bounds resulting from such a setting. As one can see from Table 5, the bounds grow very fast and many of the bounds are already larger or close to the optimum and thus at the next level all these nodes might be fathomed.

For example, based on Table 5, if one knows in advance that 578 is an upper bound, then any node that already has value 578 or greater would be fathomed. One can further branch on a node at position $(2, 5)$ with an objective value 577. In such a case, we have observed in our experiments that all the resulting bounds are greater than or equal 578, which indicates all the nodes can be fathomed. Last we point out that we also tested the growth behavior of the simplified relaxation model and observed that the bounds of the simplified relaxation model along the branching tree does not grow as fast as that of the full relaxation model.

564	569	573	566	577	0	592	578	564	569	573	566
563	574	578	573	577	0	597	584	563	574	578	573
570	571	570	562	582	0	586	573	570	571	570	562
576	564	569	572	576	0	574	570	576	564	569	572
0	0	0	0	0	0	0	0	0	0	0	0
572	564	583	597	566	0	581	588	572	564	583	597
572	569	562	570	581	0	569	574	572	569	562	570
578	574	565	572	588	0	579	577	578	574	565	572
584	576	564	568	588	0	581	569	584	576	564	568
563	571	573	583	569	0	588	582	563	571	573	583
584	567	564	581	582	0	562	575	584	567	564	581
572	568	565	557	583	0	573	565	572	568	565	557

Table 5: Bounds for Nug12 when $X_{5,6} = 1, x_{i,j} = 1, \forall i \neq 5, j \neq 6$

6 Conclusions

In this paper, we have proposed a new mechanism to derive convex programming relaxations for generic QAPs based on various matrix splitting schemes. A new class of mappings, the so-called symmetric mappings, is introduced to construct valid cuts for the resulting programming relaxation. The new convex programming relaxation works in the space $\Re^{n\times n}$ which is a substantial improvement over the standard SDP relaxation of SDP that works in $\Re^{n^2\times n^2}$. Promising experiments illustrate that rather strong bounds for large scale QAPs can be obtained. The test on Nug12 shows that the bounds of our relaxation model grows very fast along the branching tree and thus it is very promising to develop a B&B approach based on the new relaxation models.

There are several different ways to extend our results. First of all, there exist various matrix splitting schemes. In our experiments, we have observed that various splitting schemes lead to different bounds. The choice of the splitting scheme might also play an important role in the solving process of the relaxed problem. For example, when applied to special class of QAPs associated with certain graphs, model (22)-(31) (denoted by non-PSD) can provide stronger bounds within less CPU time. It is of interest to investigate how to find a splitting scheme that can provide the strongest lower bound among all the possible PSD splitting schemes. It is also interesting to see whether one can combine various splitting schemes to improve the lower bound or speed up the solving process. Secondly, given the efficiency of the model, it is possible to develop an effective branch-bound type method for solving the underlying QAPs. More study is needed to address these issues.

Acknowledgement: The authors would like to thank the anonymous referees and editors for their useful suggestions that lead to substantial improvements in this revision. They are grateful to many colleagues who have generously supported this work. In particular, Nathan Brixius provided the QPB bounds of some test problems in this paper, Samuel Burer allowed us to use his code to compute the N and N_+ models. Kim-chuan Toh shared his code to compute a rigorous bound for SDP and Yichuan Ding and Henry Wolkowicz provided the code for MSDR3. The research of this work is supported by AFOSR grant FA9550-09-1-0098 and NSF grant DMS 09-15240 ARRA.

References

[1] W.P. Adams and T.A. Johnson. Improved Linear Programming-Based Lower Bounds for the Quadratic Assignment Problem. In P.M. Pardalos and H. Wolkowicz, editors, Quadratic Assignment and Related Problems, DIMACS 25 Series in Discrete Mathematics and Theoretical Computer Science, pages 43–75, 1994.

[2] W.P. Adams and H.D. Sherali. A tight linearization and an algorithm for zero-one quadrtic programming problems. *Management Science*, 32, 1274–1290, 1986.

[3] K.M. Anstreicher and N.W. Brixius, A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming*, 80, 341–357, 2001.

[4] K.M. Anstreicher, N.W. Brixius, J.-P. Goux, and J.Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, 91, 563–588, 2002.

[5] G. Ben-David and D. Malah. Bounds on the performance of vector-quantizers under channel errors. *IEEE Transactions on Information Theory*, 51(6), 2227–2235, 2005.

[6] S. Burer and D. Vandenbussche. Solving lift-and-project relaxations of binary integer programs. *SIAM Journal on Optimization*, 16, 726–750, 2006.

[7] R.E. Burkard, E. ela, S.E. Karisch and F. Rendl. QAPLIB-A quadratic assignment problem library. *J. Global Optimization*, 10, 391–403, 1997.

[8] Y. Ding and H. Wolkowicz. A matrix-lifting semidefinite relaxation for the quadratic assignment problem. Technical report CORR 06-22, University of Waterloo, Ontario, Canada, October, 2006.

[9] M. Fiedler. Algebraic connectivity of Graphs. *Czechoslovak Mathematical Journal*, 23(98), 1973.

[10] M. Fiedler. Laplacian of graphs and algebraic connectivity. *Combinatorics and Graph Theory*, 25, 57–70, 1989.

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

[12] Golub, G. and Loan, C. V. (1996) *Matix Computation.* John Hopkins University Press.

[13] S.W. HADLEY, F. RENDL and H. WOLKOWICZ. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research*, 17, 727–739, 1992.

[14] Hahn, P. and Grant, T. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research*, 46, 912–922, 1998.

[15] M. Hannan and J.M. Kurtzberg. Placement techniques. In *Design Automation of Digital Systems: Theory and Techniques*, M.A. Breuer Ed., vol(1), Prentice-hall: Englewood Cliffs, 213–282, 1972.

[16] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press,1991.

[17] C. Jansson and C. Keil, Rigorous error bounds for the optimal value in semidefinite programming. *SIAM J. Numerical Analysis*, 46, 188–200, 2007.

[18] E. de Klerk and R. Sotirov. Exploring group summetry in semidefinite programming relaxations of the quadratic assignment problem. *Mathematical Programming*, 122(2), 225–246, 2010.

[19] T. Koopmans and M. Beckmann. Assignment problems and the location of economic activities, *Econometrica*, 25, 53–76, 1957.

[20] E. Lawler. The quadratic assignment problem. *Management Science*, 9, 586–599, 1963.

[21] E.M. Loiola, N.M. Maia de Abreu, P.O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European J. Oper. Res.*, 176(2), 657–690, 2007.

[22] I.G. Macdonald. *Symmetric functions and Hall polynomials.* The Clarendon Press, Oxford University Press, New York, 1995.

[23] H. Mittelmann, J. Peng and X. Wu. Graph Modeling for Quadratic Assignment Problem Associated with the Hypercube. Technical Report. University of Illinois at Urbana-Champaign, 2007. *http://www.optimization-online.org/DB_HTML/2007/06/1674.html.*

[24] H. Mittelmann and J. Peng. Estimating Bounds for Quadratic Assignment Problems Associated with Hamming and Manhattan Distance Matrices based on Semidefinite Programming. Technical Report. University of Illinois at Urbana-Champaign, 2008. *http://www.optimization-online.org/DB_HTML/2008/05/1980.html.*

[25] 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.

[26] L.C. Potter and D.M. Chiang, Minimax nonredundant channel coding. *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, 804–811, February/March/April 1995.

[27] F. Rendl and R. Sotirov. Bounds for the quadrtic assignment problem using bundle method. *Mathematical Programming, Series B*, 109(2-3), 505–524, 2007.

[28] F. Rendl and H. Wolkowicz. Applications of pasrametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming, Series A*, 53(1), 63–78,1992.

[29] E.D. Taillard. Comparison of iterative searches for the quadratic assingnment problem. *Location Science*, 3, 87–105, 1995.

[30] K.C. Toh. Private communication. 2009.

[31] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Combinatorial Optimization*, 2, 71–109, 1998.