Estimating Bounds for Quadratic Assignment Problems Associated with Hamming and Manhattan Distance Matrices based on Semidefinite Programming

Hans Mittelmann*   Jiming Peng [†]

September 17, 2010

Abstract

Quadratic assignment problems (QAPs) with a Hamming distance matrix for a hypercube or a Manhattan distance matrix for a rectangular grid arise frequently from communications and facility locations and are known to be among the hardest discrete optimization problems. In this paper we consider the issue of how to obtain lower bounds for those two classes of QAPs based on semidefinite programming (SDP). By exploiting the data structure of the distance matrix B, we first show that for any permutation matrix X, the matrix $Y = \alpha E - XBX^T$ is positive semi-definite, where α is a properly chosen parameter depending only on the associated graph in the underlying QAP and $E = ee^T$ is a rank one matrix whose elements have value 1. This results in a natural way to approximate the original QAPs via SDP relaxation based on the matrix splitting technique. Our new SDP relaxations have a smaller size compared with other SDP relaxations in the literature and can be solved efficiently by most open source SDP solvers. Experimental results show that for the underlying QAPs of size up to n=200, strong bounds can be obtained effectively.

Key words. Quadratic Assignment Problem (QAP), Semidefinite Programming (SDP), Singular Value Decomposition (SVD), Relaxation, Lower Bound.

1   Introduction

The standard quadratic assignment problem takes the following form

$$\min_{X \in \Pi} \operatorname{Tr}\left(AXBX^T\right) \qquad (1)$$

---

*Department of Mathematics and Statistics, Arizona State University, Tempe, AZ 85287-1804, USA. Email:mittelmann@math.asu.edu.

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

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 [20] for facility location. The model covers many scenarios arising from various applications such as in chip design [15], image processing [30], and communications [6, 25]. For more applications of QAPs, we refer to the survey paper [8] where many interesting QAPs from numerous fields are listed. Due to its broad range of applications, the study of QAPs has attracted a great deal of attention from many experts in different fields [22]. 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 [5].

In this work we focus on two special classes of QAPs where the matrix B is the Hamming distance matrix for a hypercube in space $\Re^d$ for a positive integer d, or the Manhattan distance matrix for a rectangular grid. The first class of problems arises frequently from channel coding in communications where the purpose is to minimize the total channel distortion caused by the channel noise [6, 9, 25, 33]. For ease of reference, we first give a brief introduction to this class of problems. Vector Quantization (VQ) is a widely used technique in low bit rate coding of speech and image signals. An important step in a VQ-based communication system is index assignment where the purpose is to minimize the distortion caused by channel noise. Mathematically, the index assignment problem in a VQ-based communication system can be defined as follows[6, 25]

$$\min_{X \in \Pi} \mathrm{Tr}\big(AX(HP + P^T H)X^T\big), \tag{2}$$

where A is the channel probability matrix, P is a diagonal matrix based on the resource statistics, and H is the Hamming distance matrix of a binary codebook $\mathcal{C} = \{c_1, \cdots, c_n\}$ of length d ($n = 2^d$). The above problem has caught the attention of many researchers in the communication community and many results have been reported [25, 33]. For example, it was shown in [25] that the problem is NP-hard in general. However, under the conditions that the channel is Binary Symmetric (BSC) and the quantizer is a uniform scalar with a uniform source, i.e.,

$$A = [a_{ij}], a_{ij} = t^{h_{ij}}(1 - t)^{d - h_{ij}}, \quad P = \frac{1}{n}I,$$

then the problem can be solved in polynomial time [33]. In [6], Ben-David and Malah used the projection method [13] to estimate lower and upper bounds for the above problem. The problem (2) is usually of large size. Taking for example a communication system with an 8-bit quantizer, the resulting QAP has a size of $n = 512$. This poses a great challenge in solving problem (2). As we will see in our later discussion, even obtaining a strong lower bound for problem (2) is nontrivial.

The second class of problems that we shall discuss in the present work arises usually from facility location applications [8, 20, 22, 24], and is well-known to be one of the hardest QAPs in the literature [22]. For more background on this type of problem, we refer to [22, 24].

2

A popular technique for finding the exact solution of a QAP is the 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) [12, 21], the bound based on projection [13] and the bound based on convex quadratic programming [4]. The second group contains strong bounds that require expensive computation such as the bounds derived from lifted integer linear programming [1, 2, 14] and bounds based on SDP relaxation [27, 36].

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

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

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. For convenience, we call such a relaxation the classical SDP relaxation of QAPs. As pointed out in [22, 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 classical SDP relaxation. In [19], de Klerk and Sotirov exploited the symmetry in the underlying problem to compute the bounds based on the standard SDP relaxation. In [7], Burer et al. proposed an augmented Lagrangian method to more effectively solve the relaxed SDP. Various relations between several known SDP bounds for QAP are discussed in [26]. In [10], Ding and Wolkowicz proposed a new SDP relaxation for QAPs with $O(n^2)$ variables and constraints to further reduce the computational cost.

Note that since the two classes of QAPs considered in this paper are associated with matrices of specific structure, it is highly desirable to use the specific data structure in the problem to design an algorithm to solve the problem or estimate lower bounds. The purpose of this work is to utilize the algebraic properties of the distance matrices to derive new SDP relaxations for the underlying QAPs. A distinction between our SDP relaxations and the existing SDP relaxations of QAPs lies in the lifting process. While most existing SDP relaxations of QAPs cast the underlying problem as a generic quadratic optimization model with discrete constraints to derive the SDP relaxation, we try to characterize the matrix XBX^T for any permutation matrix X by using the properties of the matrix B.

Our approach is inspired by the following observation: the eigenvalues of the matrix XBX^T are independent of the permutation matrix X. Moreover, as we shall see later, if B has some special structure such as the Hamming distance matrix associated with a hypercube, then for any permutation matrix X, the

matrix $XBX^T - \alpha E$ is negative semi-definite for a properly chosen parameter α, where E is a matrix whose elements have value 1. This provides a new way to construct SDP relaxations for the underlying QAPs. Compared with the existing SDP relaxations for QAPs in the literature, our new model is more concise and yet still able to provide a strong bound for the underlying QAP.

Secondly, we consider the issue of how to further improve the efficiency of the SDP relaxation model without much sacrifice in the quality of the lower bound. For this purpose, we remove some constraints to derive a simplified variant of the new SDP model. The simplified SDP model can be solved efficiently by most open source SDP solvers. Experiments on large scale QAPs show that the bound from the new simplified SDP relaxation is comparable to the bound provided by the original SDP model.

The paper is organized as follows. In Section 2, we explore the properties of the distance matrix associated with either Hamming distances on a hypercube in a finite dimensional space or Manhattan distances on a rectangular grid. In Section 3, we discuss how to use the theoretical properties of the matrices to construct new SDP relaxations for the underlying QAPs. We also discuss some simplified and enhanced variants of the SDP models. Numerical experiments are reported in Section 4, and finally we conclude our paper with some remarks in Section 5.

2   Properties of the matrices associated with Hamming and Manhattan distances

We start with a simple description of the geometric structure of a hypercube in space $\Re^d$. First we note that corresponding to every vertex v of the hypercube is a binary code c_v of length d, i.e., $c_v \in \{0,1\}^d$. The Hamming distance between two binary codewords corresponding to two vertices of the hypercube is defined by

$$\delta(v_i, v_j) = \sum_{k=1}^{d} |c_{v_i}^k - c_{v_j}^k|. \tag{3}$$

Moreover, for every vertex v of the hypercube, let us define

$$S(v, l) = \{v' \in V : \delta(v, v') = l\}, \tag{4}$$

and let $|S(.,.)|$ denote the cardinality of the set. Then we have

$$|S(v, l)| = C(d, l) = \frac{d!}{l!(d-l)!}, \forall\, l = 1, \cdots, d, \tag{5}$$

where $C(d, l)$ is a binomial coefficient and $d!$ is the factorial of d.

Definition 2.1. *The matrix $H^d \in \Re^{n \times n}$ with $n = 2^d$ is the Hamming distance matrix of a hypercube in $\Re^d$ if*

$$H^d = [h_{ij}^d], \quad h_{ij}^d = \delta(v_i, v_j),$$

where $V = \{v_i : i = 1, \cdots, n\}$ is the vertex set of the hypercube in $\Re^d$.

Theorem 2.2. [1] *Let $H^d \in \Re^{n \times n}$ $(n = 2^d)$ be the Hamming distance matrix defined by Definition 2.1 whose eigenvalues $\lambda_1, \cdots, \lambda_n$ are listed in decreasing order. Then we have*

$$\lambda_1 = \tfrac{dn}{2}, \quad \lambda_2 = \cdots = \lambda_{n-d} = 0, \qquad \lambda_{n-d+1} = \cdots = \lambda_n = -\frac{n}{2} \qquad (6)$$

$$H^d e = \lambda_1 e, \qquad\qquad e = (1, \cdots, 1)^T \in \Re^n. \qquad (7)$$

Proof. For any index $i \in \{1, \cdots, n\}$, it is easy to see that

$$\sum_{j=1}^{n} h_{ij}^d = \sum_{l=1}^{d} l C(d, l) = d 2^{d-1},$$

which further implies the second statement of the theorem. Since the matrix H^d is nonnegative and $I + H^d > 0$, it follows from Lemma 8.4.1 in [17]that H^d is irreducible. Using the Perron-Frobenius theorem (see Theorem 8.4.4 in [17]), we can conclude

$$\lambda_2 < \lambda_1. \qquad (8)$$

Next we progress to prove the first statement of the theorem by mathematical induction. When $d = 1$, it is easy to see that for any permutation matrix $X \in \Re^{2 \times 2}$, we have

$$X H^1 X^T = H^1.$$

In such a case, the theorem holds trivially. Now let us assume the conclusions of the theorem are true for $d = k$ $(n/2 = 2^k)$. We consider the case where $d = k + 1$ and $n = 2^{k+1}$. Note that every matrix defined in Definition 2.1. can be represented as $X H^d X^T$ for some specific matrix H^d and a permutation matrix X. Thus it suffices to consider the eigenvalues of any specific matrix H^d constructed from a hypercube. Now let us consider the matrix $H^k \in \Re^{\frac{n}{2} \times \frac{n}{2}}$ defined by Definition 2.1 associated with a hypercube in $\Re^k$ and a set of binary codes of length k $\{c^1, \cdots, c^{n/2}\}$. We can construct a new set of n binary codes of length $k + 1$ as follows

$$c_v^i = 0 \oplus c^i, \quad c_v^{i+n/2} = 1 \oplus c^i, \quad i = 1, \cdots, n/2.$$

Here the symbol $\oplus$ denotes the operation of directly adding the two binary strings to form a new string. Based on such a construction and Definition 2.1, we have

$$H^d = H^{k+1} = \begin{pmatrix} H^k & E + H^k \\ E + H^k & H^k \end{pmatrix}. \qquad (9)$$

---

[1]We thank one anonymous reviewer of an earlier version of this paper who pointed out that our result can also be proved by using combinatorial algebra and Krawtchouk polynomials. When we revised our paper, we also found an interesting technical note [35] where the author used Kronecker products to provide an alternative proof of the theorem. For self-completeness, we still include a direct proof here.

Here $E \in \Re^{2^k \times 2^k}$ is the all-1 matrix. Similarly, we use $e \in \Re^{2^k}$ to denote the all-1 vector in the remaining part of the proof. Next, consider an eigenvector $(u_1^T, u_2^T)^T \in \Re^{2^{k+1}}$ of the matrix H^{k+1} corresponding to its eigenvalue λ. It follows immediately that

$$H^k u_1 + H^k u_2 + e^T u_2 e = \lambda u_1; \qquad (10)$$
$$H^k u_1 + H^k u_2 + e^T u_1 e = \lambda u_2. \qquad (11)$$

The above relations yield

$$(\lambda I - 2H^k)(u_1 + u_2) = e^T(u_1 + u_2)e.$$

Because e is also an eigenvector of the matrix $\lambda I - H^k$, we have

$$e^T(\lambda I - 2H^k)(u_1 + u_2) = (\lambda - k2^k)e^T(u_1 + u_2) = 2^{d-1}e^T(u_1 + u_2).$$

Therefore, we must have either $\lambda = (k+1)2^k = d2^{d-1}$ or $e^T(u_1 + u_2) = 0$. In the first case, we know from the second conclusion of the theorem that λ is the largest eigenvalue of the matrix H^{k+1} corresponding to its all-1 eigenvector in $\Re^{k+1}$. In the second case, it must hold

$$(\lambda I - 2H^k)(u_1 + u_2) = 0,$$

which further implies that either $\lambda/2$ is an eigenvalue of H^k with an eigenvector $u_1 + u_2$ or $u_1 + u_2 = 0$. If $u_1 + u_2 = 0$, from relations (10)-(11) we can conclude that u_1 is a multiple of e, and $(e^T, -e^T)^T$ is an eigenvector of H^{k+1} with eigenvalue $\lambda = -2^k$. It remains to consider the case when $\lambda/2$ is an eigenvalue of H^k with an eigenvector $u_1 + u_2$. Therefore, we have either $\lambda = 0$ or $\lambda = -2 \times 2^{k-1} = -2^k = -2^{d-1}$. This shows that all the negative eigenvalues of H^d equal to -2^{d-1} when $d = k + 1$. The multiplicities of the eigenvalues follow from the fact that the Hamming matrix has zero diagonal and thus the summation of all its eigenvalues equals zero. This proves the first statement of the theorem. $\qquad\square$

A direct consequence of Theorem 2.2 is:

Corollary 2.3. *Suppose that H is the Hamming distance matrix of the hypercube in $\Re^d$ defined in Definition 2.1. Then the projection matrix of H onto the null space of e^T is negative semidefinite. Moreover, matrix $H_1 = \frac{d}{n}E - \frac{2}{n}H$ is the projection matrix of rank d.*

We next give a technical result that will be used in our later analysis, which is a refinement of Theorem 2.5 in [23].

Lemma 2.4. *Let H^d be the Hamming distance matrix defined by Definition 2.1. Then there exist permutation matrices $X_1, \cdots, X_n$ satisfying*

$$\sum_{i=1}^{n} X_i = E, \quad X_i H^d X_i^T = H^d, \forall i = 1, \cdots, n. \qquad (12)$$

Proof. Without loss of generality, we can assume that H^d is the adjacency matrix of the hypercube corresponding to the natural labeling, i.e., v_i corresponds to the binary coding of the number $i - 1$. Let c_v denote the binary codeword of a vertex v. Let $\mathcal{I}$ be an index set $\mathcal{I} \subseteq \{1, 2, \cdots, d\}$ and its complement $\bar{\mathcal{I}}$ defined by $\bar{\mathcal{I}} = \{1, 2, \cdots, d\} \setminus \mathcal{I}$. We can separate the codeword corresponding to every vertex into two parts according to the index sets $\mathcal{I}$ and $\bar{\mathcal{I}}$, i.e.,

$$c_v = c_v^{\mathcal{I}} \oplus c_v^{\bar{\mathcal{I}}}.$$

We then perform the following transformation

$$c_v = c_v^{\mathcal{I}} \oplus c_v^{\bar{\mathcal{I}}} \mapsto c_{v'}^{\mathcal{I}} \oplus c_v^{\bar{\mathcal{I}}}, \tag{13}$$

where $c_{v'}^{\mathcal{I}}$ is the complement of $c_v^{\mathcal{I}}$ derived by switching the component in $c_v^{\mathcal{I}}$ from 0 to 1 and vice versa. From (3), one can easily verify that for any vertex pair (v_i, v_j), the following relation holds

$$\delta(v_i, v_j) = \delta(c_{v_i}^{\mathcal{I}}, c_{v_j}^{\mathcal{I}}) + \delta(c_{v_i}^{\bar{\mathcal{I}}}, c_{v_j}^{\bar{\mathcal{I}}}) = \delta(c_{v_i'}^{\mathcal{I}}, c_{v_j'}^{\mathcal{I}}) + \delta(c_{v_i}^{\bar{\mathcal{I}}}, c_{v_j}^{\bar{\mathcal{I}}}).$$

Let $X_{\mathcal{I}}$ be the corresponding permutation matrix induced by the transformation (13). Our above discussion implies that

$$X_{\mathcal{I}} H^d X_{\mathcal{I}}^T = H^d.$$

Now applying the transformation (13) to all the subsets in the index set $\{1, \cdots, d\}$ [2], we obtain n permutation matrices as stated in the lemma. $\square$

Next we explore the properties of a Manhattan distance matrix. For any positive integer k, let us consider the following matrix

$$M_k = [m_{ij}] \in \Re^{k \times k}, m_{ij} = |i - j|. \tag{14}$$

Such a matrix can be viewed as a submatrix of a Hamming distance matrix corresponding to a binary code book of length k [3]. It can also be viewed as the Manhattan distance matrix of a grid on a straight line. From Theorem 2.2 we obtain the following Corollary.

Corollary 2.5. *For a given matrix M_k defined by (14), the matrix $\bar{M}_k = \frac{k-1}{2} E_k - M_k$ is positive semidefinite.*

Now let us consider the Manhattan distance matrix for a two-dimensional rectangular grid. Without loss of generality, we can assume the rectangle has k rows and l columns with a set $\mathcal{V} = \{v_1, \cdots, v_n\}$ of $n = k \times l$ nodes. Note that the Manhattan distance can be represented as the sum of two distances: the row and column distances. Because the set of eigenvalues of the matrix

---

[2]Here we can cast the empty set $\emptyset$ as a subset of indices corresponding to the identity matrix I.

[3]For example, consider a code book $\mathcal{C} = \{c_i : i = 1, \cdots, k\}$, where c_i is a binary string of length k such that its first i-digits are 1s, and the remaining digits are zeros. One can easily show that $\delta(c_i, c_j) = m_{ij}$.

XBX^T is invariant for any permutation matrix X, by performing permutations if necessary, we can assume the subset of nodes $\mathcal{V}_i^r = \{v_{(i-1)l+j} : j = 1, \cdots, l\}$ are from the i-th row of the rectangular grids, and the subset of nodes $\mathcal{V}_j^c = \{v_{(i-1)l+j} : i = 1, \cdots, k\}$ are from the j-th column of the rectangular grids. In such a case, the row distance matrix B_r can be written as the Kronecker product of two matrices

$$B_r = M_k \otimes E_l,$$

where M_k is a matrix defined by (14). Now recalling Corollary 2.5, we know that the matrix $\frac{k-1}{2}E_k - M_k$ is positive semidefinite. Moreover, it is straightforward to verify the following relation

$$\frac{k-1}{2}E_n - B_r = \left(\frac{k-1}{2}E_k - M_k\right) \otimes E_l.$$

Since the Kronecker product of two positive semidefinite matrices is also positive semidefinite (Corollary 4.2.13, pp. 245-246, [17]), it follows immediately that the matrix $\frac{k-1}{2}E_n - B_r$ is positive semidefinite. Similarly we can also show that $\frac{k-1}{2}E_n - B_c$ is positive semidefinite where B_c is the column distance matrix defined by

$$B_c = E_k \otimes M_l.$$

We thus have the following result.

Theorem 2.6. *Suppose M is a Manhattan distance matrix for a rectangular grid with k rows and l columns. Then the matrix $\bar{M} = \frac{k+l-2}{2}E_n - M$ is positive semidefinite.*

3 New SDP Relaxations for QAPs

In this section, we describe our new SDP relaxations for QAPs associated with a Hamming or Manhattan distance matrix. The section consists of three parts. In the first subsection we introduce the new SDP relaxation for QAPs with a Hamming distance matrix and compare it with other SDP relaxations for the underlying problems in the literature. In the second subsection, we discuss the case for QAPs with a Manhattan distance matrix. In the last subsection, we discuss how to further improve and simplify the relaxations in the first two subsections based on computational considerations. The bounds from the simplified models will be analyzed as well.

3.1 New SDP relaxations for QAPs with a Hamming distance matrix

We first discuss the case when B is the Hamming distance matrix. Let $Y = \frac{d}{n}E_n - \frac{2}{n}XBX^T$. From Corollary 2.3, we have $Y^2 = Y$, which can be further

relaxed to $I \succeq Y \succeq 0$. On the other hand, since the matrix $\frac{d}{n}E_n - \frac{2}{n}B$ is also a projection matrix and X is a permutation matrix, we have

$$Y = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right)X^T = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right)^2 X^T.$$

Defining $Z = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right)$, we have

$$\left(\begin{array}{c} X \\ Z \end{array}\right)\left(\begin{array}{c} X \\ Z \end{array}\right)^T = \left(\begin{array}{cc} XX^T & XZ^T \\ ZX^T & ZZ^T \end{array}\right) = \left(\begin{array}{cc} I & Y \\ Y & Y \end{array}\right). \tag{15}$$

Following the matrix-lifting procedure in [10], we relax such a relation to

$$\left(\begin{array}{ccc} I & X^T & Z^T \\ X & I & Y \\ Z & Y & Y \end{array}\right) \succeq 0. \tag{16}$$

Because $XZ^T = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right)X^T$, the above constraint can be further reduced to

$$\left(\begin{array}{cc} I & Z^T \\ Z & Y \end{array}\right) \succeq 0, \quad I - Y \succeq 0.$$

We next impose constraints on the elements of the matrix Y. Because all the elements on the diagonal of B are zeros, so are the elements on the diagonal of the matrix XBX^T. Moreover, since all the off-diagonal elements of B are positive integers, we thus have $y_{ij} \leq \frac{d-2}{n}$. Therefore, we can relax the QAP with a Hamming distance matrix to the following SDP:

$$\min \quad \frac{d}{2}\text{Tr}(AE) - \frac{n}{2}\text{Tr}(AY) \tag{17}$$

$$s.t. \quad \text{diag}\,(Y) = \frac{d}{n}e, \qquad Ye = 0; \tag{18}$$

$$y_{ij} \leq \frac{d-2}{n}, \qquad \forall j \neq i \in \{1, \cdots, n\}; \tag{19}$$

$$\left(\begin{array}{cc} I & Z^T \\ Z & Y \end{array}\right) \succeq 0, \quad I - Y \succeq 0; \tag{20}$$

$$Z = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right), \quad X \geq 0,\ X^T e = Xe = e. \tag{21}$$

Because we employ the same matrix-lifting technique to derive the SDP relaxation for the underlying QAPs, it is interesting to compare our model with the strongest SDP relaxation model (MSDR3) in [10]. It is easy to see that from any feasible solution of our SDP model, we can construct a feasible solution to the MSDR3 model in [10]. We thus have the following result.

Theorem 3.1. *Let μ^*_{SDP} be the lower bound derived by solving (17)-(21) with a Hamming distance matrix, and μ^*_{MSDR3} the bound derived from the MSDR3 model in [10]. Then we have*

$$\mu^*_{SDP} \geq \mu^*_{MSDR3}.$$

Proof. To prove the theorem, we first observe that there are three major differences between model (17)-(21) and MSDR3: The first one is that the constraints (18)-(20) are replaced by the following constraints

$$\begin{pmatrix} I & X^T & Z^T \\ X & I & Y \\ Z & Y & W \end{pmatrix} \succeq 0, \quad \text{diag}\,(Y) = X\text{diag}\,(B) \qquad (22)$$

$$Ye = XBe, \quad \text{diag}\,(W) = X\text{diag}\,(B^2), We = XB^2e. \qquad (23)$$

Secondly, MSDR3 decomposes the assignment matrix X into two parts $X = E/n + VQV^T$ where $V \in \Re^{n \times (n-1)}$ is a matrix of full column rank satisfying $V^T e = 0$, and Q is an orthogonal matrix. Constraints based on such a decomposition are then added to the matrix Q to ensure $X \geq 0$. Thirdly, additional cuts based on the eigenvalues of the matrices A and B are added to strengthen the relaxation.

One can easily verify that the constraints in model (17) are tighter than the constraints (22)-(23) with respect to matrix Y. Secondly, since we have already projected the matrix B onto the null space of e, the further decomposition of the permutation matrix X won't change our model and thus its bound can not be improved by the use of such a decomposition.

To describe the cuts in MSDR3, we use the eigenvalue decomposition of A defined by $A = U\Gamma U^T$, where $\Gamma = \text{diag}\,(\lambda_1(A), \cdots, \lambda_n(A))$ is a diagonal matrix and $\lambda_i(A), i = 1, \cdots, n$ are the eigenvalues of A sorted in non-increasing order, and U is an orthogonal matrix whose i-th column is the eigenvector of A corresponding to $\lambda_i(A)$. For convenience, we also assume that $\lambda_i(B), i = 1, \cdots, n$ are the eigenvalues B listed in non-increasing order. Then the cuts in MSDR3 are defined as follows

$$\text{Tr}\left(\Gamma_k[U(\frac{dE - nY}{2})U]_k\right) \geq \sum_{i=1}^{k} \lambda_i(A)\lambda_{n-i+1}(B) \quad k = 1, \cdots, n-2, \qquad (24)$$

where Γ_k and $[U(\frac{dE-nY}{2})U]_k$ are the principal submatrix consisting of the first k rows and columns of Γ and $U(\frac{dE-nY}{2})U^T$ respectively. Using Theorem 2.2, we can rewrite the relation (24) as

$$\text{Tr}\left(\Gamma_k[U(\frac{dE - nY}{2})U]_k\right) \geq -\frac{n}{2}\sum_{i=1}^{\min(k,d)} \lambda_i(A) \quad k = 1, \cdots, n-2. \qquad (25)$$

The above relation holds trivially when $I \succeq Y \succeq 0$ as required by (20). $\qquad \square$

3.2 New SDP relaxations for QAPs with a Manhattan distance matrix

In this subsection we describe our new SDP relaxations for QAPs with the Manhattan distance matrix for a rectangular grid with k rows and l columns.

10

As in Section 2, we first decompose the matrix B into two matrices, i.e., $B = B_r + B_c$, where B_r and B_c are the distance matrices based on the row and column positions of the nodes in the rectangular grid, respectively. Without loss of generality, we can further assume that

$$\bar{B}_r = \frac{k-1}{2}E_n - B_r = \left(\frac{k-1}{2}E_k - M_k\right) \otimes E_l,$$

$$\bar{B}_c = \frac{l-1}{2}E_n - B_c = E_k \otimes \left(\frac{l-1}{2}E_l - M_l\right).$$

It follows from Corollary 2.5 that both $\bar{B}_r$ and $\bar{B}_c$ are positive semi-definite.

Let $\tilde{B}_r$ and $\tilde{B}_c$ denote the symmetric square root of the matrix $\bar{B}_r$ and $\bar{B}_c$, respectively [4]. We have

$$\begin{aligned}
\text{Tr}(AXBX^T) \quad &= \frac{k+l-2}{2}\text{Tr}(AE_n) - \text{Tr}(AX(\bar{B}_r + \bar{B}_c)X^T) \\
&= \frac{k+l-2}{2}\text{Tr}(AE_n) - \text{Tr}\left(AX(\tilde{B}_r^2 + \tilde{B}_c^2)X^T\right).
\end{aligned}$$

Let us define $Y^r = X\bar{B}_rX^T, Y^c = X\bar{B}_cX^T$. It follows immediately that

$$Y^r = X\tilde{B}_r^2X^T, Y^c = X\tilde{B}_c^2X^T. \tag{26}$$

The above relations can be relaxed to

$$Y^r - X\tilde{B}_r^2X^T \succeq 0, \quad Y^c - X\tilde{B}_c^2X^T \succeq 0.$$

We next discuss how to efficiently compute the square roots $\tilde{B}_r$ and $\tilde{B}_c$. For this we first cite a classic result on Kronecker product (Lemma 4.2.10, [17]).

Lemma 3.2. *Let $A_1, A_2 \in \Re^{m \times m}, B_1, B_2 \in \Re^{n \times n}$. Then we have*

$$(A_1 \otimes B_1)(A_2 \otimes B_2) = (A_1A_2) \otimes (B_1B_2).$$

Based on the above lemma, we have

$$\tilde{B}_r = \tilde{M}_k \otimes \left(\frac{1}{\sqrt{l}}E_l\right), \tilde{B}_c = \left(\frac{1}{\sqrt{k}}E_k\right) \otimes \tilde{M}_l,$$

where $\tilde{M}_k, \tilde{M}_l$ are the square roots of the matrices $\frac{k-1}{2}E_k - M_k$ and $\frac{l-1}{2}E_l - M_l$, respectively.

We next discuss how to impose some constraints on the elements of Y^r and Y^c. It is easy to see that all the elements on the diagonal of Y^r have the same values as those of $\bar{B}_r$. We next consider the sum of all the elements in any row of Y^r. Since $Y^r = X\bar{B}_rX^T$, we have

$$Y^r e = X\left(\frac{k-1}{2}E_n - B_r\right)X^T e = \frac{(k-1)n}{2}e - XB_re.$$

---

[4] We mention that it is also possible to use the Cholesky decomposition of $\bar{B}_r$ and $\bar{B}_c$.

Similar relations for Y^c can also be obtained. Because

$$(B_r + B_c)_{ij} \geq 1, \forall i \neq j \in \{1, \cdots, n\},$$

it follows

$$y_{ij}^r + y_{ij}^c \leq \frac{k + l - 4}{2}, \forall j \neq i \in \{1, \cdots, n\}.$$

Based on the above discussion, we can relax a QAP with a Manhattan distance matrix for a $k \times l$ rectangular grid to the following SDP:

$$\min \quad \frac{k + l - 2}{2} \mathrm{Tr}(AE) - \mathrm{Tr}(A(Y^r + Y^c)) \tag{27}$$

$$s.t. \quad Y^r e = \frac{(k-1)n}{2} e - X B_r e, \quad \mathrm{diag}\,(Y^r) = \frac{k-1}{2} e;$$

$$Y^c e = \frac{(l-1)n}{2} e - X B_c e, \quad \mathrm{diag}\,(Y^c) = \frac{l-1}{2} e;$$

$$y_{ij}^r + y_{ij}^c \leq \frac{k + l - 4}{2}, \qquad \forall j \neq i \in \{1, \cdots, n\};$$

$$\begin{pmatrix} I & Z_r^T \\ Z_r & Y^r \end{pmatrix} \succeq 0, \qquad \begin{pmatrix} I & Z_c^T \\ Z_c & Y^c \end{pmatrix} \succeq 0;$$

$$Z_r = X \tilde{B}_r, Z_c = X \tilde{B}_c, \qquad X \geq 0, X^T e = X e = e.$$

Let (Y^r, Y^c) be a feasible solution to model (27) and let us define $Y^* = \frac{k+l-2}{2} E - Y^r - Y^c$. One can easily see that the matrix Y^* also satisfies constraints (22)-(23) with respect to matrix argument Y. Therefore, model (27) can also be viewed as a simplified and improved version of the MSDR1 model in [10].

It should be pointed out that some QAPs (like nug16a, nug17, nug18) from the QAP library [8] were constructed out of the larger instances by deleting several nodes from some rows or columns. In order to apply model (27), we need to first compute the row (or column) distance matrix of the original rectangular grid. Then we remove the rows and columns corresponding to the construction of the underlying QAP instance. In such a case, we can not represent the row (or column) distance matrix B_r (or B_c) as a Kronecker product. Nevertheless, we can still show that the matrices $\bar{B}_r = \frac{k-1}{2} E_n - B_r, \bar{B}_c = \frac{l-1}{2} E_n - B_c$ remain positive semi-definite.

3.3 Further enhancements and simplifications

In this subsection, we first discuss how to further enhance the SDP relaxation models proposed earlier in this section by incorporating simple linear constraints from the GLB model into our model. To see this, let us recall the GLB model, which can be defined as the following linear assignment problem

$$\min \quad \mathrm{Tr}(WX) \tag{28}$$

$$X \geq 0, Xe = X^T e = e. \tag{29}$$

Here W is a matrix constructed from the matrices A and B by the following rule [12]. Let $a_{i,:}$ and $b_{:,i}$ be the i-th row and i-th column of matrices A and B, respectively. We have

$$w_{ij} = a_{ii}b_{jj} + \sum_{k=1}^{n-1} a_{ik}^* b_{kj}^*,$$

where $a_{ik}^*, k = 1, \cdots, n-1$ are the sorted elements of $a_{i,:}$ excluding a_{ii} in a decreasing order, and $b_{kj}^*, k = 1, \cdots, n-1$ are the sorted elements of $b_{:,j}$ excluding b_{jj} in an increasing order. Let $Y = XBX^T$ where B is the original matrix in (1), then we have

$$a_{i,:}y_{:,i} \geq w_{i,:}x_{:,i}, \quad \forall i = 1, \cdots, n.$$

We can add the above constraints to the existing SDP relaxation model to improve its lower bound.

On the other hand, we also note that although the SDP relaxation models proposed earlier in this section are much more concise and simpler than other SDP relaxations for QAPs in the literature, they might still involve relatively intensive computation for large scale QAP instances with $n \geq 150$. We next discuss how to simplify the relaxation models in Sections 3.1 and 3.2. To start, let us first examine the SDP model (17)-(21) where we impose the following constraints on the relation between the matrix Y and the assignment matrix X

$$\begin{pmatrix} I & Z^T \\ Z & Y \end{pmatrix} \succeq 0, \quad Z = X\left(\frac{d}{n}E_n - \frac{2}{n}B\right).$$

If we replace the above constraint by $Y \succeq 0$, then we can simplify model (17)-(21) to the following

$$\min \quad \frac{d}{2}\text{Tr}(AE) - \frac{n}{2}\text{Tr}(AY) \tag{30}$$

$$\begin{aligned} s.t. \quad & \text{diag}(Y) = \frac{d}{n}e, \quad && Ye = 0, \\ & y_{ij} \leq y_{ii} - \frac{2}{n}, \quad && \forall j \neq i \in \{1, \cdots, n\}; \\ & Y \succeq 0, \quad && I - Y \succeq 0. \end{aligned}$$

Using the special structure of the Hamming distance matrix as explored in Lemma 2.4, we can prove the following interesting result.

Theorem 3.3. *Let μ_{SDP}^2 be the lower bound derived by solving (30) for a QAP with a Hamming distance matrix and μ_{SDP}^1 be the bound derived from model (17). Then we have*

$$\mu_{SDP}^1 = \mu_{SDP}^2.$$

Proof. From Lemma 2.4 we can conclude that there exist n optimal permutation matrices satisfying the relation (12). This implies that if we specify the SDP constraints in model (17) to

$$\begin{pmatrix} I & Z^T \\ Z & Y \end{pmatrix} \succeq 0, \quad I - Y \succeq 0, \quad Z = \frac{E}{n}\left(\frac{d}{n}E_n - \frac{2}{n}B\right);$$

13

then the bound computed from the resulting SDP relaxation is tighter than the bound computed from model (17). However, from Theorem 2.2, we have $Z = \frac{E}{n}\left(\frac{d}{n}E_n - \frac{2}{n}B\right) = 0$. Therefore, we can further reduce the above SDP constraint to

$$I \succeq Y \succeq 0,$$

which are precisely the SDP constraints in the simplified model (30). $\qquad\square$

We can similarly relax the SDP constraint in (27) to derive the following simpler SDP:

$$\min \quad \frac{k+l-2}{2}\text{Tr}(AE) - \text{Tr}(A(Y^r + Y^c)) \tag{31}$$

$$s.t. \quad Y^r e = \frac{(k-1)n}{2}e - XB_r e, \qquad \text{diag}\,(Y^r) = \frac{k-1}{2}e;$$

$$Y^c e = \frac{(l-1)n}{2}e - XB_c e, \qquad \text{diag}\,(Y^c) = \frac{l-1}{2}e;$$

$$y^r_{ij} + y^c_{ij} \leq \frac{k+l-4}{2}, \qquad \forall j \neq i \in \{1, \cdots, n\};$$

$$Y^r \succeq 0, Y^c \succeq 0, \qquad X \geq 0, X^T e = Xe = e.$$

Suppose that (Y^r, Y^c) is a feasible solution to model (31) and let us define $Y^* = \frac{k+l-2}{2}E - Y^r - Y^c$. One can easily see that the matrix Y^* also satisfies constraints (22)-(23) with respect to matrix argument Y. Therefore, we can also cast model (31) as a simplified/improved version of the MSDR1 model in [10].

We mention that in our recent paper [23], we have shown (see Theorem 2.6 of [23]) that for QAPs associated with the hypercube (or Hamming distance matrices), the optimal solution can be attained at a permutation matrix X^* with $x^*_{11} = 1$. Therefore, we can further add such a constraint to strengthen the bound.

4 Numerical Experiments

In this section, we report some numerical results based on our models. Our experiments consist of two parts. In the first part, we report the bounds for QAPs with the Hamming distance matrix of a hypercube. In the second part, we present our numerical experiments for QAPs with the Manhattan distance matrix for a rectangular grid. As pointed out in the introduction, most expensive relaxations of QAPs can only be applied to small scale instances. Because of this observation, in this section we only compare our bounds with the bounds that can be computed effectively such as the GLB bound [12, 21], the bound based on projection (denoted by PB) [13] and the bound based on convex quadratic programming (denoted by QPB) [4]. We note that overall the CPU times to compute the other three bounds are shorter than that for our model. Since our emphasis is on obtaining improved bounds at reasonable speed and we are using available software and a customized program for the other bounds, we list only the CPU time for our model.

For QAPs with a Hamming distance matrix, we test our model on four different choices of the matrix A. The first one is the matrix used in [32] defined by

$$a_{ij} = \frac{\Delta^3 \sqrt{1-\rho^2}}{(2\pi\sigma^6)^{\frac{3}{2}}} \sum_{l=1}^{n} \exp^{-\frac{\delta^2}{2\sigma^2}((1-\rho^2)(i-n_1)^2 + (j-n_1-\rho*(i-n_1))^2 + (l-n_1-\rho*(i-n_1))^2)},$$

where $n_1 = \frac{n+1}{2}$, and Δ is the step size to quantize the source, σ is the variance of the Gaussian Markov source with zero mean and ρ its correlation coefficient. In our experiment, we set the step size $\Delta = 0.4, \sigma = 1$ and $\rho = 0.1, 0.9$ respectively. These two different choices of ρ represent the scenarios of the source with dense correlation and non-dense correlation. The second test problem is the so-called Harper code [16] with

$$a_{ij} = |i - j|.$$

We also tested our model on a random matrix A and the so-called vector quantization problem (denoted by VQ) [34] provided by our colleague Dr. Xiaolin Wu from McMaster University. Our experiments are done on an AMD Opteron with 2.4GHz CPU and 12 GB memory. We use the latest version of CVX [11] and SDPT3 [31] under Matlab R2008b to solve our problem. In Table 1, we list the lower bound (L-bound in the table) computed from our model, the CPU time in seconds to obtain the bound, and the upper bound (U-bound in the table) obtained by using the Tabu search described as in [30]. We use such a heuristics to find an upper bound because, except for the case $n = 16$, no global solutions to the underlying problems have been reported in the literature. For comparison purpose, we also include three inexpensive bounds: the GLB, the PB and the QPB bounds in the table. For each instance, the strongest bound is indicated in boldface. We also mention that in Table 1, the problem eng1 (eng9) refers to the engineering problem with $\rho = 0.1(0.9)$ respectively. The experimental results with the hypercube in $\Re^d$ with $d = 4, 5, 6, 7$ are summarized in the table. It should be pointed out that since the computational cost for solving the model (17) is more expensive than that of the simplified model (30), while the lower bounds computed from these two models are the same as shown in Theorem 3.3, we use only the latter model to compute the lower bound.

From Table 1, we can see that except for some instances of the problems eng1 and eng9, the bounds provided by our model for other test problems are the strongest among the four inexpensive bounds. One can also find that our bound is very close to the global optimal solution of the underlying problem for most test instances. For the test problems eng1 and eng9, we observed that the relative gap between the lower and upper bounds increases as n increases. One possible explanation for this is that for large n, most elements of the matrix A have usually very small values. On the other hand, from its definition we know that the elements of Y will also have small absolute values. These two points might lead to some numerical instability when we solve the SDP relaxation.

We next report our experiments for the ESC problems from the QAP library [8][5]. Note that as announced in the QAP library [8], there are several

---

[5]We thank Kurt Anstreicher for reminding us that our model can be applied to the ESC

| | | Model (30) | | Other Bounds | | | |
Problem	size	L-Bounds	CPU	GLB	PB	QPB	U-Bound
Harper	16	**2741.7**	1	2252	2737.3	2737.3	2752
	32	**27328**	3	21530	27321.1	27321.1	27360
	64	**262160**	56	.926658	262177	262177	262260
	128	**2446944**	1491	1910244	2446750	2446750	2479944
eng1	16	**1.5452**	1	1.23545	1.54439	1.54439	1.58049
	32	**1.24196**	4	1.11861	1.10647	1.19687	1.58528
	64	.926658	56	**1.04878**	-.287054	-.106264	1.58297
	128	.881738	1688	**.99196**	-3.57411	-3.21253	1.56962
eng9	16	**.930857**	1	.82764	.883709	.883825	1.02017
	32	1.03724	3	**1.06144**	.682844	.682844	1.40941
	64	.887776	68	**1.00596**	-1.19555	.0523582	1.43201
	128	.846574	2084	**.95404**	-5.58127	-2.57681	1.43198
rand	16	**733.5**	1	658.7777	724.284	724.284	745.2
	32	**6372**	3	5623.157	6244.76	6305.78	6486
	64	**60965**	54	54324.09	60654.9	60655	62035
	128	**1163220**	2937	521664.6	564947	580956	1177020
VQ	16	**244.11**	1	201.0883	243.487	243.487	245.27
	32	**294.49**	3	226.1443	293.633	293.633	297.29
	64	**352.4**	45	262.2678	351.955	351.955	353.5
	128	**393.29**	2719	286.1720	393.084	393.084	399.09

Table 1: Inexpensive bounds for QAPs associated with hypercubes

ESC instances whose optimal solutions have not been confirmed. In such a case, we use the symbol '*' to denote the best feasible solution found so far in the column of optimum values. From Table 2 we can see that our lower bounds can be computed very efficiently. We also mention that in all the tables, whenever all the elements of the associated matrices A and B have only integer values, we list the value obtained by rounding up the lower bound (obtained from the relaxation model) to the smallest integer above it. From Table 2, one can verify that our bounds are usually stronger than the other three bounds. In particular, for several instances such as ESC16a-c,ESC16h, ESC32b-d, ESC32h, our bound is very strong and comparable to the strongest bound reported in the literature [8]. On the other hand, we also would like to point out that in several cases, our bound is not that strong. With a closer look at the data, we noted that for all the cases where our bound is not strong, the coefficient matrix A is very sparse and has only a few nonzero elements. Since we did not exploit the structure of the matrix A in our model, it is not surprising that we could not obtain strong bounds for these cases. It is worthwhile mentioning that we can also compute the bounds based on the enhanced model introduced in Section 3.3 by incorporating the linear constraints in the GLB model [12, 21] into the SDP model proposed in this work with a little extra effort. Based on our experiments, such an enhanced model is able to provide better bounds when A is very sparse at the cost of slightly more CPU time.

We now present our experiments on several large scale QAPs with a Manhattan distance matrix of rectangular grids. For these problems, the best known bounds were reported in [18], and we use the best known feasible solution from the QAP library as the upper bound. We list the best known lower and upper bounds in the last column of the table. Table 3 lists the results of both models (27) and (31) for QAPs of size from 42 to 100 [6].

As we can see from Table 3, the bound provided by the simple model (31) is quite strong and comparable with the bound by model (27). In a few cases (indicated by the boldface), the bound obtained from the simple model (31) even improves over the best known bound for the underlying QAP in the literature [8]. From Table 3 one can see that our bounds are tighter than the other three inexpensive bounds for all instances.

It should be pointed out that for QAP instances whose size is above 100, we could not apply the model (27) due to memory restriction. In Table 4 we list only the lower bound computed by solving the simple model (31). The first test problem is from [8] while the rest are from [29]. All the tested problems from [29] have a size of $n = 200$. As one can see from Table 4, the new bound is the tightest among all the listed inexpensive bounds. It is also worthwhile mentioning that to the best of our knowledge, for the large scale instances in Table 4, no lower bounds based on expensive relaxation models have been reported in the literature.

We also tested our model on QAPs with a Manhattan distance matrix of

---

instances as pointed out in [3].

[6] Due to space consideration, we list only the CPU time for the simplified model (31) in the table.

	Model (30)		Other Bounds			
Prob.	L-Bound	CPU	GLB	QB	QPB	Optimum
esc16a	**57**	1	38	47	55	68
esc16b	**284**	1	220	250	250	292
esc16c	**135**	1	83	95	95	160
esc16d	**4**	4	3	-19	-19	16
esc16e	**18**	1	12	6	6	28
esc16g	**19**	1	12	9	9	26
esc16h	**927**	1	625	708	708	996
esc16i	**0**	1	**0**	-25	-25	14
esc16j	**1**	1	**1**	-6	-6	8
esc32a	19	7	**35**	-150	-150	130
esc32b	**103**	7	96	64	64	168
esc32c	**578**	6	350	464	464	642
esc32d	**152**	7	106	97	97	200
esc32e	**0**	5	**0**	-165	-165	2
esc32g	**0**	5	**0**	-76	-76	6
esc32h	**381**	7	257	245	245	438
esc64a	0	63	**47**	-243	-243	116*
esc128	0	2140	**2**	-999	-999	64*

Table 2: Inexpensive bounds for the ESC instances

	Our models		Other Bounds			
Prob.	Model (27)	Model (31)/CPU	GLB	PB	QPB	Best known bounds
wil50	46526	46467/22	38069	45730	46129	[47098,48816]
sko42	14393	14377/12	11311	13829	13992	[14934,15812]
sko49	21449	21427/21	16161	20714	20999	[22004,23386]
sko56	31711	31641/34	23321	30700	31087	[32610,34458]
sko64	45013	44946/65	32522	43889	44434	[45736,48498]
sko72	61854	61764/135	44280	60401	60971	[62691,66256]
sko81	85194	85056/262	60283	82276	83304	[86072,90998]
sko90	108496	108312/489	75531	105982	107036	[108493,115534]
sko100a	**143068**	**142843**/760	98953	139364	140793	[142668,152002]
sko100b	**144826**	**144571**/754	99028	141250	142992	[143872,153890]
sko100c	139059	138870/939	95979	135010	136591	[139402,147862]
sko100d	**140577**	**140385**/865	95921	136978	138630	[139898,149576]
sko100e	140053	139823/884	95551	136995	138576	[140105,149150]
sko100f	**140288**	**140077**/921	96016	136859	138500	[139452,149036]
wil100	263710	263406/814	210949	260826	262428	[263909,273038]

Table 3: Inexpensive bounds for large scale QAPs: Part I

18

Prob.	CPU	Bound	GLB	PB	QPB	Upper bound
tho150	6801	**7537980**	4123652	7350920	7459810	8133398
stu1	34214	**22425300**	18283015	22153511	22258800	22642312
stu2	37795	**14551200**	9917321	14287321	14402300	14882423
stu3	37769	**6807470**	3389030	6580161	6675240	7095880
stu4	38657	**1526710**	497207	1393886	1448860	1694716
stu5	43215	**1828430**	580610	1585348	1646680	2000485
stu6	36910	**9300150**	5501973	9023365	9121280	9575346
stu7	35019	**4289370**	1865596	4085400	4162170	4488544
stu8	38659	**952830**	290013	837993	875209	1089094
stu9	35078	**12002100**	7430481	11706223	11824600	12347759
stu10	39656	**7534160**	3829222	7219091	7323370	7865751
stu11	39537	**3447140**	1311009	3210504	3290470	3685347
stu12	37766	**719307**	207686	562363	602774	845597
stu13	38707	**6404970**	3275523	6168578	6254000	6660709
stu14	36881	**3990840**	1661830	3696276	3771760	4251108
stu15	43202	**1828430**	580610	1585348	1646680	2000485
stu16	44020	**354328**	98908	187026	217990	437802

Table 4: Bounds for large scale QAPs: Part II

rectangular grids whose size is below 40. These QAP instances were solved on a 2.67GHz Intel Core 2 computer with 4GB memory. We compare our result with the method in [10] as well as other three inexpensive models.

As we can see from Table 5 [7], except for a few cases [8], the bounds provided by models (27) and (31) are tighter than the bound provided by MSDR3 and other inexpensive relaxations, and the bounds from models (27) and (31) are very close while the simple model (31) is more efficient. We also observed in our experiments that the CPU time for MSDR3 grows very fast as the size of the instance increases. For example, it took more than hours to compute the lower bound for the test problems whose size are 30 or above.

5   Conclusions

In this paper, we have proposed new SDP relaxations for QAP having the Hamming distance matrix for a hypercube in $\Re^n$ or the Manhattan distance matrix for a rectangular grid. By exploiting the special structure of the underlying

---

[7] We mentioned that the bounds we report for MSDR3 are slightly different from those given in [10]. This is because in [10], the authors applied their model to two scenarios by swapping A and B, and select the best bound from these two scenarios. However, in our paper, since we elaborate more on the structure of B, we could not exploit such a possibility. Consequently, for purpose of consistence, we only list the bound obtained in our experiment.

[8] An anonymous reviewer pointed out that the bound from model (29) is worse than the MSDR3 on Nug16a, Nug17 and Nug18 where the distance matrices do not correspond to complete rectangular grids.

19

	Model (27)		Model (31)		Other bounds				
Prob.	CPU	bound	Bound	CPU	MSDR3	GLB	PB	QPB	Optimum
Nug12	2	**509**	509	1	501	493	472	482	578
Nug14	3	**930**	927	1	917	852	871	891	1014
Nug15	4	**1044**	1041	1	1016	963	973	996	1150
Nug16a	5	1439	1433	1	**1460**	1314	1403	1448	1610
Nug16b	4	**1102**	1102	1	1085	1022	1045	1071	1240
Nug17	5	1527	1520	1	**1548**	1388	1486	1529	1732
Nug18	7	1719	1715	1	**1725**	1554	1663	1705	1930
Nug20	9	**2299**	2298	1	2287	2057	2196	2254	2570
Nug21	12	**2163**	2152	1	2095	1833	1979	2055	2438
Nug22	15	**3226**	3185	1	3118	2483	2966	3080	3596
Nug24	19	**3114**	3106	1	3055	2676	2959	3028	3488
Nug25	21	**3331**	3326	1	3290	2869	3190	3272	3744
Nug27	38	**4727**	4690	1	4602	3701	4493	4588	5234
Nug28	40	**4656**	4639	1	4553	3786	4433	4519	5166
Nug30	59	**5490**	5478	2	5418	4539	5265	5365	6124
scr12	2	**28110**	**28110**	1	18766	27858	4727	8585	31410
scr15	5	**43823**	43583	1	23735	44737	10354	12479	51140
scr20	11	**86181**	85964	1	42114	86766	16113	23960	110030
tho30	55	**128815**	127943	2	118010	90578	119255	124684	149936
Ste36a	129	7108	7010	4	4423	**7124**	-11771	-10155	9526
tho40	212	**206622**	205571	5	183150	143804	191042	197131	240516

Table 5: **Bounds for small scale QAPs**

distance matrix (denoted by B), we show that for a properly chosen parameter α depending on the size of the hypercube or the rectangular grids, the matrix $\alpha E - B$ is positive semidefinite. This leads to a new way to relax the underlying QAP to an SDP. For large scale QAP instances, the proposed relaxation models can be solved effectively by most open source SDP solvers. Experimental results illustrate that the bound provided by our new models is tighter than the bounds based on other inexpensive relaxations for most test problems.

There are several different ways to extend our results. First of all, it is of interest to investigate whether the proposed approach in the present work can be adapted to derive new SDP relaxations for general QAPs. Secondly, given the efficiency of the model, it may be possible to develop an effective branch-and-bound type method for solving the underlying QAPs. Thirdly, we note that at the optimal solution of the original QAP, the matrix XBX^T has a low rank (rank d for the case of Hamming distance, and $k + l - 2$ for the Manhattan distance case). Therefore, it may be interesting to investigate how to find a low rank solution to the relaxed SDP. Finally, like in [3], we also observed in our experiments that when the matrix A is very sparse, then the bound from our model might not be very strong. It will be interesting to explore how we can incorporate the sparse structure of the underlying matrix into our SDP relaxation model. Further study is needed to address these issues.

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. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming (Series B)*, 97: 27-42, 2003.

[4] 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.

[5] K.M. Anstreicher, N.W. Brixius, J.-P. Goux, and J.Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming* 91(3):563-588, 2002.

[6] 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.

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

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

[9] N.T. Cheng and N.K. Kingsburg. Robust zero-redundancy vector quantization for noisy channels. In *Proceedings of the International Conference on Communications*, Boston, 1338–1342, 1989.

[10] 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.

[11] M. Grant, S. Boyd and Y. Ye. CVX Users' Guide, 2007. *http://www.stanford.edu/~boyd/cvx/cvx_usrguide.pdf*

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

[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] L.H. Harper. Optimal assignments of numbers to vertices. *J. of the Society for Industrial and Applied Mathematics*, 12, 131–135, 1964.

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

[18] S.E. Karisch and F. Rendl. Lower bounds for the quadratic assignment problem via triangle decompositions. *Mathematical Programming*, 71(2):137-152, 1995.

[19] 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.

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

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

[22] 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.

[23] H. Mittelmann, J. Peng and X. Wu. Graph Modeling for Quadratic Assignment Problem Associated with the Hypercube. *AIP Conference Proceedings* 1146, 65-79, 2009.

[24] 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.

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

[26] J. Povh and F. Rendl. Copositive and Semidefinite Relaxations of the Quadratic Assignment Problem. *Discrete Optimization*, Vol. 6, 231-24, 2009.

[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] M. Scriabin and R.C. Vergin. Comparison of computer algorithms and visual based methods for plant layout. *Management Science*, 22, 172–187, 1975.

[29] T. Stützle and S. Fernandes. New Benchmark Instances for the QAP and the Experimental Analysis of Algorithms. In *J. Gottlieb und G. Raidl, editor, Evolutionary Computation in Combinatorial Optimization: 4th European Conference*, EvoCOP 2004, volume 3004 in *Lecture Notes in Computer Science*, pages 199-209, Springer Verlag, Berlin, Germany, 2004.

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

[31] K.C. Toh, R.H. Tütüncü and M. Todd. On the implementation and usage of SDPT3. 2006. *http://www.math.nus.edu.sg/˜mattohkc/guide4-0-draft.pdf*

[32] X. Wang, X. Wu and S. Dumitrescu. On optimal index assignment for MAP decoding of Markov sequences, Technical report, Department of Electrical and Computer Engineering, McMaster University, Ontario, Canada, 2006.

[33] K. Zeger and A. Gersho. Pseudo-Gray Code. *IEEE Trans. on Communications*, 38(12), 2147–2158, 1990.

[34] Xiaolin Wu. Private communication. Department of Electrical and Computer Engineering, McMaster University, Ontario, Canada, 2007.

[35] Xiaoyan Zeng. Result for Hamming Distance Matrix. Technical note. Department of Applied Mathematics, Illinois Institute of Technology. See *http://www.iit.edu/z̃engxia/paper/kronecker.pdf*.

[36] 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.