Annals of Operations Research 50(1994)387-410

Lower bounds for the quadratic assignment problem

Y. Li

Computer Science Department, The Pennsylvania State University, University Park, PA 16802, USA

P.M. Pardalos

Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA

K.G. Ramakrishnan and M.G.C. Resende

Mathematical Sciences Research Center, AT&T Bell Laboratories, Murray Hill, NJ 07974, USA

We investigate the classical Gilmore–Lawler lower bound for the quadratic assignment problem. We provide evidence of the difficulty of improving the Gilmore–Lawler bound and develop new bounds by means of optimal reduction schemes. Computational results are reported indicating that the new lower bounds have advantages over previous bounds and can be used in a branch-and-bound type algorithm for the quadratic assignment problem.

Keywords: Quadratic assignment problem, branch-and-bound, lower bound, combinatorial optimization.

AMS(MOS) subject classifications: AMS(MOS) 68Q25, 90B80, 90C27.

1. Introduction

We consider the quadratic assignment problem (QAP) in the Koopmans and Beckmann form [18]. Given a positive integer n and two $n \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, the problem is to find a permutation p of the set $\{1,2,\ldots,n\}$ that minimizes

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}.$$

The QAP belongs to a class of combinatorial optimization problems with many practical applications. However, only small instances ($n \le 15$) of QAP have been solved to optimality in practice. The QAP, of which the traveling salesman problem is a special case, is NP-complete [35]. Furthermore, unless $P = NP$, there is no polynomial algorithm guaranteed to find an ε-approximate solution [13]. In this paper, we denote the QAP associated with matrices A and B by QAP(A,B).

In the framework of the facility location problem, there are n facilities (locations), the matrix $A = (a_{ij})$ corresponds to the flow matrix, where a_{ij} represents the flow of material from facility i to facility j for $i, j = 1, \ldots, n$, the matrix $B = (b_{ij})$ is the distance matrix, with b_{ij} representing the distance from location i to location j for $i, j = 1, \ldots, n$. The objective function, to be minimized, is the cost associated with the assignment of the n facilities to the n locations (Koopmans and Beckmann [18]). In addition to its application in the facility location problem, the QAP has been found useful in scheduling, backboard wiring in electronics, and other applications. Applications of the QAP can be found in [3,7,18,19].

The QAP may be formulated in many equivalent forms. On can formulate it as a quadratic 0–1 programming problem, a global concave minimization problem [29,30], or an integer program.

Extensive research has been done on the quadratic assignment problem, focusing on both heuristic solutions (see [4–6,24,37,38]) and exact solutions (see [2,23,28]). Recently, there has been work on generating test problems with known optimal solutions for the quadratic assignment problem (see [22,27]).

In this paper, we are concerned with lower bounds for the QAP. In section 2, we review existing lower bounds for the QAP. In section 3, we discuss the difficulty in improving the classical Gilmore–Lawler bound. In section 4, we present new lower bounds for the QAP. In section 5, we report computational results.

2. Previous lower bounds for the QAP

Lower bounds are key to the success of a branch-and-bound algorithm in combinatorial optimization. The ideal lower bounds should be sharp and should be fast to compute.

For the QAP, there are three categories of lower bounds. The first category includes the classical Gilmore–Lawler bound (GLB) [14,20] and related bounds. The second category includes the eigenvalue-based bounds [11,16,17,31]. The rest of the bounds are mostly based on reformulations of the QAP and generally involve solving a number of linear assignment problems (e.g. [1,8–10,12]). In the following, we briefly discuss the three categories of lower bounds. The new lower bounds we propose belong to the first category.

2.1. GILMORE–LAWLER BOUND

We define the following minimal and maximal vector products

$$\langle x, y \rangle_- = \min_{P \in \Pi} \langle x, Py \rangle, \quad \langle x, y \rangle_+ = \max_{P \in \Pi} \langle x, Py \rangle,$$

where the set Π denotes the set of all $n \times n$ permutation matrices and $x, y \in \mathbb{R}^n$. Since there is a one-to-one correspondence between the set of permutations of $\{1, \ldots, n\}$ and the set of $n \times n$ permutation matrices, we denote both sets by Π.

For a given QAP(A, B), we denote the Gilmore–Lawler bound by GLB(A, B). The bound is based on the minimal scalar products defined as follows:

Let $a_i, b_i, i = 1, \ldots, n$ represent the row vectors of matrices A, B, respectively. Let $\hat{a}_i$ be the vector consisting of the $(n-1)$ components of a_i, not including a_{ii}. Let $\hat{b}_i$ be the vector consisting of the $(n-1)$ components of b_i not including b_{ii} Define a matrix $L = (l_{ij})$ as follows

$$l_{ij} = a_{ii}b_{jj} + \langle \hat{a}_i, \hat{b}_j \rangle_-, \quad i, j = 1, \ldots, n.$$

Then GLB(A,B) is defined to be the solution to the linear assignment problem (LAP) with cost matrix L, i.e.

$$\mathrm{GLB}(A, B) = \min_{p \in \Pi} \sum_{i=1}^{n} l_{ip(i)}.$$

The GLB is the most widely used lower bound for the QAP (see [28,23]). However, the GLB deteriorates fast as the size of the QAP increases. For example, for the Nugent test problem set [25], the GLB for the problem of size 6 is about 5 percent below the optimal and the GLB for the problem of size 30 is about 25 percent below the best known value.

Due to this reason, there have been efforts in improving these lower bounds by means of reduction (see [2,11,33,34]). The rationale is to shift out as much information as possible from the quadratic term. In particular, one lower bound studied in Finke et al. [11] is obtained by subtracting from each column the minimum entry (we denote it by MCCR). However, the reduction techniques used in the literature have not consistently outperformed GLB, as indicated by the results in [11].

2.2. EIGENVALUE BOUNDS (EVB)

Bounds based on eigenvalues of the flow and distance matrices A and B have been proposed in a series of papers by Finke et al. [11], Hadley et al. [16,17], and Rendl and Wolkowicz [31]. Those bounds are based on the following theorem (see [11]).

THEOREM 2.1

Let A and B be symmetric matrices, $\lambda_1 \le \lambda_2 \ldots \le \lambda_n$ be the eigenvalues of A, and $\mu_1 \le \mu_2 \ldots \le \mu_n$ be those of B. For any $p \in \Pi$, we have

$$\sum_{i=1}^{n} \lambda_i \mu_{n-i+1} \le \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)} \le \sum_{i=1}^{n} \lambda_i \mu_i.$$

Theorem 2.1 gives us a lower bound based on eigenvalues of A and B,

$$EVB(A,B) = \sum_{i=1}^{n} \lambda_i \mu_{n-i+1}.$$

Certain reductions of the original matrices have to be performed before using the eigenvalues to obtain lower bounds for the QAP. Two such bounds, EVB1 and EVB2, were developed in [11]. Two other bounds, EVB3 and IVB, were developed in [16]. Rendl and Wolkowicz [31] recently proposed the MEVB lower bound, based on eigenvalue decomposition in conjunction with a steepest ascent algorithm. The bound is obtained iteratively, each iteration taking $O(n^3)$ time. Note that the above eigenvalue based bounds are applicable only to symmetric quadratic assignment problems for which the flow and distance matrices are symmetric. An extension to the asymmetric QAP was done in Hadley et al. [16]. However, no computational results were reported.

Although this class of bounds can be computed in time comparable to that of the GLB computations, they are sharper than GLB only in a few limited cases. For example, for the Nugent test problem, the GLB is better for $n = 6, 8, 12, 15$. In some cases, the eigenvalue bounds can even be negative.

2.3. OTHER LOWER BOUNDS

Assad and Xu [1] proposed the AX bound for a class of quadratic 0-1 programs, including the QAP. The bound is obtained iteratively, where $n^2 + 1$ assignment of size n are solved in each iteration. Hence, the running time to compute is $O(kn^5)$ where k is the number of iterations.

Christofides and Gerrard [9] proposed the XG lower bound by solving $O(n^4)$ linear assignment problems, corresponding to pairs of assignments, resulting in a $O(n^7)$ procedure.

Frieze and Yadegar [12] obtained two lower bounds by solving the Lagrangian relaxation of a related linear integer formulation of QAP. The bounds are denoted by FY1 and FY2.

Finally, Carraresi and Malucelli [8] proposed a new lower bound (CM) for the QAP through an iterative process. In each iteration, at most $O(n^2)$ linear assignment problems related to an equivalent reformulation of the QAP are solved. Hence, the procedure has a time complexity of $O(kn^5)$ where k is the number of iterations used. One disadvantage with this category of lower bounds is that, they are not computed efficiently, and thus are not effective for branch-and-bound type algorithms.

3. Improving GLB is nontrivial

In this section, we present some evidence of the difficulty of improving GLB. We begin by stating the following theorem.

THEOREM 3.1

Let $f^*_{A,B}$ be the unknown optimal objective function value for QAP(A,B). It is NP-complete to check if GLB(A,B) = $f^*_{A,B}$.

Proof

This problem is in NP, since a linear assignment problem must be solved to compute GLB and since QAP is in NP. Now we transform the Hamiltonian Circuit problem (HC) to this problem. Given an instance of HC, an undirected, connected graph $G = (V,E)$, with n vertices $V = \{1, \ldots, n\}$, we construct the following QAP, where the flow and distance matrices A and B are of size $n \times n$. A is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{for } i = 1, \ldots, n-1, \quad j = i+1, \\ 1 & \text{for } i = n, \quad j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

A is the adjacency matrix of a cycle of length n. The distance matrix B is defined to be

$$b_{ij} = \begin{cases} 0 & \text{for } j = i, \quad i \in V, \\ 1 & \text{for } (i,j) \in E, \\ 2 & \text{otherwise.} \end{cases}$$

First, GLB(A,B) = n due to the fact that all ordered products are equal to 1. Second, we show that there is a Hamiltonian Circuit in the graph G if and only if $f^*_{A,B} = n$.

If there is a Hamiltonian Circuit in G, then the circuit can be regarded as a permutation $p = (i_1, i_2, \ldots, i_n)$, where i_k, $k = 1, \ldots, n$, are the vertices of the tour in the order specified by p. Let $f_{A,B}(p)$ be the objective function of QAP(A,B) for permutation p. Then

$$f_{A,B}(p) = n.$$

Since GLB(A,B) = n, we have $f_{A,B}(q) \geq n$ for any permutation q. Hence

$$f^*_{A,B} = \text{GLB}(A,B).$$

Conversely, if $f^*_{A,B}$ = GLB(A,B), let p be the optimal permutation. Then the tour defined by the permutation p is a Hamiltonian Circuit. □

From the above theorem, we have the following corollary, that answers positively the question raised in Li and Pardalos [22] about the complexity of the class of quadratic assignment problems whose GLBs are equal to their optimal objective function values.

COROLLARY 3.1

If $P \neq NP$, then there is no polynomial algorithm to find an optimal permutation for a QAP whose GLB is equal to the optimal objective function value.

We next show that by splitting the flow and distance matrices, it is impossible to improve the GLB of QAP. The following lemmas are useful in proving the theorem.

LEMMA 3.1

Given the two vectors x_1, x_2 such that $x = x_1 + x_2$, then

$$\langle x, y\rangle_- \geq \langle x_1, y\rangle_- + \langle x_2, y\rangle_-,$$

for all vectors y.

Proof

Let P be the permutation such that $\langle x, y\rangle_- = \langle x, Py\rangle$. Since

$$\langle x, Py\rangle = \langle x_1, Py\rangle + \langle x_2, Py\rangle \geq \langle x_1, y\rangle_- + \langle x_2, y\rangle_-,$$

we have that

$$\langle x, y\rangle_- \geq \langle x_1, y\rangle_- + \langle x_2, y\rangle_-.$$

□

COROLLARY 3.2

Given four vectors x_1, x_2, y_1, y_2 such that $x = x_1 + x_2$ and $y = y_1 + y_2$, then

$$\langle x, y\rangle_- \geq \langle x_1, y_1\rangle_- + \langle x_2, y_1\rangle_- + \langle x_1, y_2\rangle_- + \langle x_2, y_2\rangle_-.$$

THEOREM 3.2

Given a QAP(A, B) and two matrices A_1 and A_2 *such that* $A = A_1 + A_2$. Then

$$\mathrm{GLB}(A, B) \geq \mathrm{GLB}(A_1, B) + \mathrm{GLB}(A_2, B)$$

for any distance matrix B.

Proof

This can be regarded as a generalization of lemma 3.1. Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be the rows of A and B, respectively. Furthermore, let $a_1^{(1)}, \ldots, a_n^{(1)}$ and $a_1^{(2)}, \ldots, a_n^{(2)}$ be the rows of A_1 and A_2, respectively. Let p be the optimal permutation of the linear assignment problem associated with Gilmore–Lawler bound. Let q_i, $i = 1, \ldots, n$, be a permutation such that

$$\langle a_i, Q_i\, b_{p(i)} \rangle = \langle a_i, Q_i\, b_{p(i)} \rangle_- ,$$

where Q_i is the permutation matrix for q_i. Then, following lemma 3.1, we have

$$\begin{aligned} \mathrm{GLB}(A,B) &= \sum_{i=1}^{n} \langle a_i, Q_i\, b_{p(i)} \rangle \\ &= \sum_{i=1}^{n} \langle a_i^{(1)}, Q_i\, b_{p(i)} \rangle_- + \sum_{i=1}^{n} \langle a_i^{(2)}, Q_i\, b_{p(i)} \rangle_- \\ &\geq \mathrm{GLB}(A_1, B) + \mathrm{GLB}(A_2, B). \end{aligned}$$

□

COROLLARY 3.3

Given a QAP(A,B) and matrices A_1, A_2, B_1, B_2 such that $A = A_1 + A_2$, and $B = B_1 + B_2$, then

$$\mathrm{GLB}(A,B) \geq \mathrm{GLB}(A_1,B_1) + \mathrm{GLB}(A_2,B_2) + \mathrm{GLB}(A_1,B_2) + \mathrm{GLB}(A_2,B_1).$$

This corollary shows that one cannot improve GLB by splitting the flow or distance matrices. The complexity result on the GLB implies that given a QAP(A,B) and its corresponding GLB(A,B), it is NP-complete to decide if the optimal solution is strictly greater than the lower bound. Therefore, it appears to be nontrivial to improve the GLB. These two results, together with the efficient computation of GLB, help to explain why GLB remains the most widely used lower bound for a branch-and-bound type algorithm for the QAP.

4. The new lower bounds

In this section, we propose new lower bounds based on optimal reduction schemes for the QAP. The schemes we propose can be regarded as extensions of the reduction techniques in the literature [2,11,33,34].

For convenience of discussion, let us clarify some notation to be used. For a given QAP(A,B), consider a partition of A into two matrices $A_1 = (a_{ij}^{(1)})$ and $A_2 = (a_{ij}^{(2)})$ such that $A = A_1 + A_2$ and partition of B into two matrices $B_1 = (b_{ij}^{(1)})$ and $B_2 = (b_{ij}^{(2)})$ such that $B = B_1 + B_2$. For each pair $(i,j) = 1, \ldots, n$, consider the following minimization problem

$$\min \sum_{k=1}^{n} a_{ik}^{(1)} b_{jp(k)}^{(1)} + \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j} + \sum_{k=1}^{n} a_{ki} b_{p(k)j}^{(2)} - \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j}^{(2)}, \tag{4.1}$$

where $p \in \Pi$ and $p(i) = j$. Define an $n \times n$ matrix $L = (l_{ij})$ where l_{ij} is the optimal objective function value of (4.1). Now, we can define a new lower bound based on the following theorem.

THEOREM 4.1

Let the matrix L be defined as above. Then the solution of the linear assignment problem with cost matrix L is a lower bound for the corresponding QAP.

Proof

Let $p \in \Pi$. Then

$$\begin{aligned}
f_{A,B}(p) &= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_{ij}^{(1)} + a_{ij}^{(2)}\right)\left(b_{p(i)p(j)}^{(1)} + b_{p(i)p(j)}^{(2)}\right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(1)} b_{p(i)p(j)}^{(1)} + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(1)} \\
&\quad + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(1)} b_{p(i)p(j)}^{(2)} + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(2)} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(1)} b_{p(i)p(j)}^{(1)} + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(1)} + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(2)} \right) \\
&\quad + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(1)} b_{p(i)p(j)}^{(2)} + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(2)} \right) - \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(2)} b_{p(i)p(j)}^{(2)} \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} a_{ij}^{(1)} b_{p(i)p(j)}^{(1)} + \sum_{j=1}^{n} a_{ji}^{(2)} b_{p(j)p(i)} \right. \\
&\quad \left. + \sum_{j=1}^{n} a_{ji} b_{p(j)p(i)}^{(2)} - \sum_{j=1}^{n} a_{ji}^{(2)} b_{p(j)p(i)}^{(2)} \right) \\
&\geq \sum_{i=1}^{n} l_{ip(i)}.
\end{aligned}$$

Hence the result holds. □

It is the primary objective of this paper to study the class of lower bounds derived from the above theorem. The classical Gilmore–Lawler bound is a special case in which both matrices A and B are not partitioned. Different ways of partitioning the matrices A and B (we also refer to this as *reduction*) yield different lower bounds. The common reduction techniques used in the literature choose A_2 and B_2 with constant column sums (which we call constant columns). We refer to such techniques as constant column reductions. There are two important questions here:

(A) How should the matrices be partitioned such that the resulting lower bound is maximized.

(B) How can one solve (4.1) efficiently.

In the remainder of this section, we try to answer these two questions and introduce the new lower bounds.

4.1. OPTIMAL REDUCTION OF FLOW AND DISTANCE MATRICES

In the literature, the column reduction techniques are restricted to the case in which the partition of matrices A and B is done in such a way that A_2 and B_2 have constant columns. Here, we consider the general case in which the matrices A_2 and B_2 may not have constant columns. Problem **A** is an unconstrained nonconvex optimization problem, and the optimal solutions are intractable. We derive a partitioning scheme that results in an approximate solution of **A**.

Let $M = (m_{ij})$ be a matrix in $\mathbb{R}^{m \times n}$. We treat a row vector m_i, $1 \le i \le m$, of M as a $1 \times n$ matrix and a column vector m_j^{T}, $1 \le j \le n$ as a $n \times 1$ matrix. For convenience of discussion, we use the following notations of average $\gamma(M)$, variance $V(M)$, and total variance $T(M, \lambda)$ for M:

$$\gamma(M) = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij},$$

$$V(M) = \sum_{i=1}^{m} \sum_{j=1}^{n} (\gamma(M) - m_{ij})^2,$$

$$T(M, \lambda) = \lambda \sum_{i=1}^{m} V(m_i) + (1-\lambda) V(M), \quad \text{for} \quad 0 \le \lambda \le 1.$$

We were motivated by the following observation: for QAP(A,B), the smaller the variances of A and B are, the tighter the GLB is. Furthermore, if the row variances of A and B are zero, then GLB is equal to the optimal objective function value. In addition to the above observations, from the definition of l_{ij}, we also desire small variances in the rows and columns of the partitional matrices. Hence, we consider the following problems:

(C1) Find a matrix Δ_A and partition A as A_1 and A_2, where $A_1 = A + \Delta_A$ and $A_2 = -\Delta_A$, such that the variances of A_1 and A_2, the sum of variances of the rows of A_1, and the sum of variances of the rows of A_2 are minimized.

(**C2**) Find a matrix Δ_B and partition B as B_1 and B_2, where $B_1 = B + \Delta_B$ and $B_2 = -\Delta_B$, such that the variances of B_1 and B_2, the sum of variances of the rows of B_1, and the sum of variances of the rows of B_2 are minimized.

Problems **C1** and **C2** share the same solution procedure and we consider solving **C1** in the remaining part of this section. We can formulate problem **C1** as the following minimization problem in terms of the total variances of matrices $A + \Delta_A$ and $-\Delta_{A^T}$ and the parameter θ, $0 \le \theta \le 1$:

$$\begin{aligned} \min \quad & \theta T(A + \Delta_A, \lambda) + (1-\theta) T(-\Delta_{A^T}) \\ \text{where} \quad & \Delta_A \in \mathbb{R}^{n \times n}. \end{aligned} \tag{4.2}$$

The above problem involves minimizing certain weighted sum of variances of n vectors and the variance of an $n \times n$ matrix. To solve **C1**, let us consider the problem of finding the partition $D = (D + \Delta) - \Delta$ of a $m \times n$ matrix $D = (d_{ij})$ such that the variance of $D + \Delta$ is minimized. Let $\Delta = (\delta_{ij})$.

In fact, the variance of the matrix $D + \Delta$ is

$$V(D+\Delta) = \sum_{i=1}^{m} \sum_{j=1}^{n} (\gamma(D+\Delta) - (d_{ij} + \delta_{ij}))^2. \tag{4.3}$$

To minimize $V(D + \Delta)$, we need to find a Δ satisfying the following constraints:

$$\frac{\partial V(D+\Delta)}{\partial \delta_{ij}} = 0, \quad i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{4.4}$$

Although there are $m \times n$ equations, the following analysis reveals that we can have closed form solution of the system of equation:

$$\sum_{k=1}^{m} \sum_{l=1}^{n} \alpha_{ij,kl}^{D} \delta_{kl} = \beta_{ij}^{D}, \quad i = 1, \ldots, m \;\; j = 1, \ldots, n, \tag{4.5}$$

where $\alpha_{ij,kl}^{D}, \beta_{ij}^{D}$, $i,k = 1, \ldots, m$, $j,l = 1, \ldots, n$, are computed according to

$$\alpha_{ij,kl}^{D} = \begin{cases} (mn-1)/mn & \text{if } (k,l) = (i,j), \\ -1/mn, & \text{if } (k,l) \ne (i,j), \end{cases} \quad \beta_{ij}^{D} = \gamma(D) - d_{ij}.$$

LEMMA 4.1

Systems (4.4) and (4.5) are equivalent and the solution is

$$\delta_{ij} = d_{mn} - d_{ij} + \delta_{mn}, \quad i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{4.6}$$

Proof

Systems (4.4) is equivalent to

$$\frac{1}{2}\frac{\partial V(D+\Delta)}{\partial \delta_{ij}}=0, \quad i=1,\dots,m, \ j=1,\dots,n.$$

Note that

$$\gamma(D+\Delta)=\frac{1}{mn}\sum_{k=1}^{m}\sum_{l=1}^{n}(d_{kl}+\delta_{kl}) \text{ and } \frac{1}{mn}\sum_{k=1}^{n}\sum_{l=1}^{n}(\gamma(D)-d_{kl})=0.$$

Denote the term $\gamma(D+\Delta)-(d_{ij}+\delta_{ij})$ by T_{ij}, then

$$\frac{1}{2}\frac{\partial V(D+\Delta)}{\partial \delta_{ij}}=\left(\frac{1}{mn}-1\right)T_{ij}+\frac{1}{mn}\sum_{(k,l)\neq(i,j)}T_{kl}, \quad i=1,\dots,m, \ j=1,\dots,n.$$

Hence, the system of linear equations is equivalent to

$$\sum_{k=1}^{m}\sum_{l=1}^{n}\alpha^{D}_{ij,kl}\,\delta_{kl}=\beta^{D}_{ij}, \quad i=1,\dots,m \ \ j=1,\dots,n, \tag{4.7}$$

where

$$\begin{aligned}
\alpha^{D}_{ij,ij} &= \left(\frac{1}{mn}-1\right)\left(\frac{1}{mn}-1\right)+\frac{1}{mn}\sum_{(r,s)\neq(i,j)}\frac{1}{mn}\\
&= \left(\frac{1}{mn}-1\right)^{2}+\frac{mn-1}{(mn)^{2}}=\frac{mn-1}{mn},\\
\alpha^{D}_{ij,kl} &= \left(\frac{1}{mn}-1\right)\frac{1}{mn}+\frac{1}{mn}\sum_{(r,s)\neq(i,j)\,\text{or}\,(k,l)}\frac{1}{mn}+\frac{1}{mn}\left(\frac{1}{mn}-1\right)\\
&= -\frac{1}{mn}, \quad \text{if } (k,l)\neq(i,j),\\
\beta^{D}_{ij} &= -\left(\frac{1}{mn}-1\right)(\gamma(D)-d_{ij})-\frac{1}{mn}\sum_{(k,l)\neq(i,j)}(\gamma(D)-d_{kl})\\
&= -\frac{1}{mn}\sum_{k=1}^{n}\sum_{l=1}^{n}(\gamma(D)-d_{kl})+(\gamma(D)-d_{ij})=\gamma(D)-d_{ij}.
\end{aligned}$$

By direct computation, one can verify that the solution satisfies the system of linear equations and yields a minimum of 0 for the related minimization problem. □

Note that δ_{mn} in (4.6) can take any value. We set $\delta_{mn}=0$ in the computational experiments.

With the above results, we can minimize the total variance $T(A+\Delta,\lambda)$. To do that we solve

$$\begin{aligned} \min \quad & \lambda \sum_{i=1}^{n} V(a_i + \delta_i) + (1-\lambda) V(A+\Delta), \\ \text{where} \quad & \Delta \in \mathbb{R}^{n \times n}. \end{aligned} \tag{4.8}$$

Following the discussion on minimizing the variance, we solve the following system of linear equations in order to solve the problem (4.8),

$$\lambda \frac{\partial V(a_i+\delta_i)}{\partial \delta_{ij}} + (1-\lambda) \frac{\partial V(A+\Delta)}{\partial \delta_{ij}} = 0, \quad i,j = 1, \ldots, n, \tag{4.9}$$

or equivalently,

$$\lambda \sum_{l=1}^{n} \alpha_{ij,il}^{a_i} + (1-\lambda) \sum_{k=1}^{n} \sum_{l=1}^{n} \alpha_{ij,kl}^{A} \delta_{kl} = \lambda \beta_{ij}^{a_i} + (1-\lambda) \beta_{ij}^{A}, \tag{4.10}$$

where $\alpha_{ij,il}^{a_i}, \beta_{ij}^{a_i}, \alpha_{ij,kl}^{A}, \beta_{ij}^{A}, i,j,k,l = 1, \ldots, n$, are computed as follows:

$$\alpha_{ij,il}^{a_i} = \begin{cases} (n-1)/n, & \text{if } l = j, \\ -1/n, & \text{if } l \neq j, \end{cases} \quad \beta_{ij}^{a_i} = \gamma(a_i) - a_{ij}; \tag{4.11}$$

$$\alpha_{ij,il}^{A} = \begin{cases} (n^2-1)/n^2, & \text{if } (k,l) = (i,j), \\ -1/n^2, & \text{if } (k,l) \neq (i,j), \end{cases} \quad \beta_{ij}^{A} = \gamma(A) - a_{ij}. \tag{4.12}$$

THEOREM 4.2

Systems (4.9) and (4.10) are equivalent. Furthermore, the solution is

$$\delta_{ij} = a_{nn} - a_{ij} + \delta_{nn}, \quad i,j = 1, \ldots, n. \tag{4.13}$$

Proof

The equivalence of the systems follows from lemma (4.1). One can substitute the solution in the system (4.10) to verify correctness. □

Note that the above is independent of the value of λ. When we consider the constant column reduction technique, we solve the above system of linear equations with additional constraints imposing that the columns in matrix Δ are constant. The resulting Δ gives us an optimal reduction of the original matrix. The new system of equations can be written as follows:

$$\sum_{i=1}^{n} \left(\lambda \sum_{l=1}^{n} \alpha_{ij,il}^{a_i} + (1-\lambda) \sum_{k=1}^{n} \sum_{l=1}^{n} \alpha_{ij,kl}^{A} \delta_{kl} \right) = \sum_{i=1}^{n} \left(\lambda \beta_{ij}^{a_i} + (1-\lambda) \beta_{ij}^{A} \right) \quad j = 1, \ldots, n. \tag{4.14}$$

THEOREM 4.3

The solution of (4.14) is

$$\delta_{ij} = \gamma\left(a_n^{\mathrm{T}}\right) - \gamma\left(a_j^{\mathrm{T}}\right) + \delta_{nn}, \quad i, j = 1, \ldots, n, \tag{4.15}$$

where a_i^{T}, $i = 1, \ldots, n$, are the columns of matrix A.

Proof

Let $x_j = \delta_{ij}$, $i, j = 1, \ldots, n$. We can rewrite the system (4.14) as follows:

$$\sum_{l=1}^{n} h_{lj} x_l = e_j, \quad j = 1, \ldots, n,$$

where the coefficients h_{lj} and constants e_j can be derived from (4.14). In fact, using the formulae in (4.11) and (4.12), we have

$$h_{lj} = \lambda \sum_{i=1}^{n} \alpha_{ij,il}^{a_i} + (1-\lambda) \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_{ij,kl}^{A}, \quad l, j = 1, \ldots, n.$$

If $l = \mathrm{j}$, then

$$h_{lj} = \lambda \sum_{i=1}^{n} \frac{n-1}{n} - (1-\lambda) \sum_{i=1}^{n} \sum_{k=1, k \neq i}^{n} \frac{1}{n^2} + (1-\lambda) \sum_{i=1}^{n} \sum_{k=i}^{n} \frac{n^2-1}{n^2}$$

$$= \lambda(n-1) + (1-\lambda)(n-1) = n-1.$$

If $l \neq j$, then

$$h_{lj} = \lambda \sum_{i=1}^{n} \frac{-1}{n} + (1-\lambda) \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{-1}{n^2} = -1.$$

Finally, for $j = 1, \ldots, n$, we have

$$e_j = \sum_{i=1}^{n} \left(\lambda \beta_{ij}^{a_i} + (1-\lambda) \beta_{ij}^{A}\right)$$

$$= \lambda \sum_{i=1}^{n} (\gamma(a_i) - a_{ij}) + (1-\lambda) \sum_{i=1}^{n} (\lambda(A) - a_{ij})$$

$$= n\gamma(A) - n\gamma(a_j^{\mathrm{T}}).$$

Similar to the case of lemma 4.1, one can derive the desired solution. □

After the above discussion, we can consider the solution of (4.2). Unfortunately, we do not have a closed form solution for (4.2). Observe that if the addition term of $T(-\Delta^T, \lambda)$ is replaced by $T(-\Delta, \lambda)$, the resulting solution is exactly the scaling of the solutions in (4.13) and (4.15). Hence, we provide the following approximate reduction schemes **R1** and **R2**.

(**R1**) if Δ is not restricted, then let

$$\delta_{ij} = \theta(a_{nn} - a_{ij}) + \delta_{nn}, \quad i, j = 1, \ldots, n.$$

(**R2**) if Δ is restricted to have constant columns, let

$$\delta_{ij} = \theta(\gamma(a_n^T) - \gamma(a_j^T)) + \delta_{nn}, \quad i, j = 1, \ldots, n.$$

4.2. COMPUTING MATRIX L

Now, let us consider problem **B**, where one must compute l_{ij} $i, j = 1, \ldots, n$. If we use constant column reduction, matrices A_2 and B_2 have constant columns and the problem is easy to solve. One can compute l_{ij} as follows:

$$l_{ij} = \langle \hat{a}_i^{(1)}, \hat{b}_j^{(1)} \rangle_- + a_{1i}^{(2)} \sum_{k=1, k\neq j}^{n} b_{kj} + b_{1j}^{(2)} \sum_{k=1, k\neq i}^{n} a_{ki} - (n-1) a_{1i}^{(2)} b_{1j}^{(2)} + a_{ii} b_{jj}, \quad i, j = 1, \ldots, n.$$

For the general case, A_2 and B_2 may not have constant columns. Before we proceed to the details of computing l_{ij}, we need some notation.

Given two sets of vectors $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ in $\mathbb{R}^m$. Let $x_i = (x_1^{(i)}, \ldots, x_m^{(i)})$ and $y_i = (y_1^{(i)}, \ldots, y_m^{(i)})$, $i = 1, \ldots, n$. Let us first consider the following minimization of multidimensional vector product (MVP) problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^{n} \langle x_k, y_{p(k)} \rangle \\ \text{where} \quad & p \in \Pi, x_k \in \mathbb{R}^m, \; y_k \in \mathbb{R}^m, \quad k = 1, \ldots, n. \end{aligned} \tag{4.16}$$

Then, computing l_{ij} is essentially a special case of MVP with $m = 4$ and

$$x_k = \left(a_{ik}^{(1)}, a_{ki}^{(2)}, a_{ki}, -a_{ki}^{(2)}\right), \quad k = 1, \ldots, n, \; k \neq i,$$
$$y_k = \left(b_{jk}^{(1)}, b_{kj}, b_{kj}^{(2)}, -b_{kj}^{(2)}\right), \quad k = 1, \ldots, n, \; k \neq j.$$

Hence, l_{ij} can be computed by the following methods:

(**M1**) Compute l_{ij} by solving a linear assignment problem with cost matrix H where $h_{ij} = \langle x_i, y_j \rangle$, $i, j = 1, \ldots, n$.

(**M2**) Compute l_{ij} as the sum of four independent minimal vector products.

Method **M1** is time consuming since one needs to solve n^2 linear assignment problems of size n each in order to compute L, resulting in an $O(n^5)$ procedure. Method **M2** is fast but the solution quality may not be as good as that of method **M1**. In our computational tests, l_{ij} is computed based on method **M2** when we do not apply constant column reduction. However, one can do a better job when using method **M2**. A closer look at the problem (4.1) reveals that the problem can be reduced to a 3-dimensional MVP with

$$x_k = \left(a_{ik}^{(1)}, a_{ki}^{(2)}, a_{ki}^{(1)}\right), \quad k = 1, \ldots, n,$$
$$y_k = \left(b_{jk}^{(1)}, b_{kj}, b_{kj}^{(2)}\right), \quad k = 1, \ldots, n.$$

Furthermore, we can give an alternative way of computing lower bounds to the solution of (4.1) quickly. Consider the 2-dimensional MVP:

$$\begin{aligned} \min \quad & \sum_{k=1}^{n} \langle x_k, y_{p(k)} \rangle \\ \text{where} \quad & p \in \Pi, x_k \in \mathbb{R}^2, y_k \in \mathbb{R}^2, \quad k = 1, \ldots, n. \end{aligned} \tag{4.17}$$

Let

$$t_1 = \left(x_1^{(1)}, \ldots, x_1^{(n)}\right), \quad t_2 = \left(x_2^{(1)}, \ldots, x_2^{(n)}\right),$$
$$u_1 = \left(y_1^{(1)}, \ldots, y_1^{(n)}\right), \quad u_2 = \left(y_2^{(1)}, \ldots, y_2^{(n)}\right).$$

Define the following four vectors

$$v_1 = t_1 + t_2, \quad v_2 = t_1 - t_2,$$
$$w_1 = u_1 + u_2, \quad w_2 = u_1 - u_2.$$

We have the following lemma.

LEMMA 4.2

Given two sets of vectors $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ in $\mathbb{R}^2$, define v_1, v_2, w_1, w_2 as above. Let p be a permutation, then

$$\sum_{k=1}^{n} \langle x_k, y_{p(k)} \rangle \geq \frac{1}{2}(\langle v_1, w_1 \rangle_- + \langle v_2, w_2 \rangle_-).$$

Proof

Let p be any permutation and P be the corresponding permutation matrix, then

$$\begin{aligned}\langle v_1, w_1\rangle_- + \langle v_2, w_2\rangle_- &\le \langle v_1, Pw_1\rangle + \langle v_2, Pw_2\rangle \\ &= \langle t_1 + t_2, P(u_1 + u_2)\rangle + \langle t_1 + t_2, P(u_1 - u_2)\rangle \\ &= 2(\langle t_1, Pu_1\rangle + \langle t_2, Pu_2\rangle) = 2\sum_{i=1}^{n} \langle x_k, y_{p(k)}\rangle. \qquad \square\end{aligned}$$

Consequently, we can use vectors v_1, v_2, w_1, w_2 to provide a lower bound for the 2-dimensional MVP. Similarly, for the 3-dimensional MVP:

$$\begin{aligned}&\min \quad \sum_{k=1}^{n} \langle x_k, y_{p(k)}\rangle \\ &\text{where} \quad p \in \Pi, x_k \in \mathbb{R}^3,\ y_k \in \mathbb{R}^3,\quad k = 1, \dots, n.\end{aligned} \tag{4.18}$$

Let

$$t_1 = \left(x_1^{(1)}, \dots, x_1^{(n)}\right),\quad t_2 = \left(x_2^{(1)}, \dots, x_2^{(n)}\right),\quad t_3 = \left(x_3^{(1)}, \dots, x_3^{(n)}\right),$$

$$u_1 = \left(y_1^{(1)}, \dots, y_1^{(n)}\right),\quad u_2 = \left(y_2^{(1)}, \dots, y_2^{(n)}\right),\quad u_3 = \left(y_3^{(1)}, \dots, y_3^{(n)}\right).$$

Define the following six vectors:

$$\begin{aligned}v_1 &= t_1 + t_2, & v_2 &= t_1 + t_3, & v_3 &= t_2 + t_3,\\ w_1 &= u_1 + u_2 - u_3, & w_2 &= u_1 - u_2 + u_3, & w_3 &= -u_1 + u_2 + u_3.\end{aligned}$$

We can state the following lemma.

LEMMA 4.3

For two given sets of vectors $x_1, \dots, x_n$ and $y_1, \dots, y_n$ in $\mathbb{R}^3$, define $v_1, v_2, v_3, w_1, w_2, w_3$ as above. Let p be a permutation, then

$$\sum_{k=1}^{n} \langle x_k, y_{p(k)}\rangle \ge \frac{1}{2}(\langle v_1, w_1\rangle_- + \langle v_2, w_2\rangle_- + \langle v_3, w_3\rangle_-).$$

Proof

Let p be any permutation and P be the corresponding permutation matrix, then

$$\begin{aligned}&\langle v_1, w_1\rangle_- + \langle v_2, w_2\rangle_- + \langle v_3, w_3\rangle_- \\ &\quad\le \langle v_1, Pw_1\rangle + \langle v_2, Pw_2\rangle + \langle v_3, Pw_3\rangle \\ &\quad= \langle t_1 + t_2, P(u_1 + u_2 - u_3)\rangle + \langle t_1 + t_3, P(u_1 - u_2 + u_3)\rangle\end{aligned}$$

$$+\langle t_2 + t_3, P(-u_1 + u_2 + u_3)\rangle$$
$$= 2(\langle t_1, Pu_1\rangle + \langle t_2, Pu_2\rangle + \langle t_3, Pu_3\rangle) = 2\sum_{i=1}^{n} \langle x_k, y_{p(k)}\rangle. \qquad \square$$

The above lemma provides an alternative lower estimate for the solution of (4.1). This lower bound may be better than directly computing the minimal vector products in some cases. Hence, one can compute both and take the larger bound.

4.3. TWO NEW LOWER BOUNDS

One new lower bound that we propose in this paper is to use the reduction scheme **R1**. We denote this lower bound by LB1(θ). The other new lower bound that we propose is to use the reduction scheme **R2**. This lower bound is denoted LB2(θ). Both new lower bounds are dependent on the parameter θ in (4.2). Note that LB1(0.0) = GLB(A,B) and LB1(1.0) = GLB(A^T,B^T).

For LB1(θ), we found in our computational experiments that $\theta = 0.5$ is a good choice. For LB2(θ), we used $\theta = 1.0$. The latter was expected since the column variance of the matrix Δ is already zero when computing LB2(θ).

The new lower bounds can be computed quite efficiently. Computing the matrix Δ to partition matrices A and B takes only $O(n^2)$ time. By presorting the rows of the flow and distance matrices A and B, one can compute $l_{ij}, i, j = 1, \ldots, n$, in $O(n^3)$ running time using method **M2**. Hence, the total running time is $O(n^3)$, which is the same as that for computing GLB. Furthermore, the constant factor is small.

5. Computational results

We report the computational results on the new lower bounds, LB1(θ) and LB2(θ), and compare the new bounds with existing lower bounds. As we stated in the previous section, we choose $\theta = 0.5$ for LB1(θ) and $\theta = 1.0$ for LB2(θ). In the tables for reporting computational results, we simply use LB1 and LB2 to denote LB1(0.5) and LB2(1.0). The new lower bounding procedure was implemented in FORTRAN and the computational experiments were concluded on a Sun Sparcstation II. One of the eigenvalue bounds (EVB1) was also implemented for the purpose of comparison. In addition to the existing lower bounds discussed earlier and the new lower bounds, we also implemented the trivial lower bounds obtained by fixing k facilities, for $k = 1,2,3$. The corresponding lower bounds are denoted by TB(k), $k = 1,2,3$.

The test problems used include the following classes of problems

I. Nugent test problems [25]: 6 problems of sizes 6, 8, 12, 15, 20, 30. The problems are symmetric.

II. Random symmetric problems with uniform distribution.

III. Random symmetric problems with hyperexponential distribution.

IV. Random asymmetric problems with uniform distribution.

V. Random asymmetric problems with hyperexponential distribution.

The uniform random number generator was taken from [26]. The hyperexponential random number generator made use of the uniform random number generator described in [36] and used exponential distributions in parallel [15]. The computational results are summarized in tables 1 to 3.

Table 1 contains the performance of the existing bounds and the new lower bounds on the Nugent test problems. Since the number of bounds is large, the table is split into two parts. The last line of the table contains the computational complexity of the lower bounds. Note, that the procedure to compute EVB3 is iterative. According to Hadley et al. [16], EVB3 took 20 to 30 iterations before outperforming IVB.

The bounds in part 1 of table 1 are considered inexpensive since the computational complexities of those bounds are the same as that for GLB. For those bounds, there is no clear trend as to which one is the best for the Nugent test problems. However, we note that LB1 and LB2 (as well as MCCR) outperforms GLB for $n = 30$.

We consider the bounds in part 2 of table 1 as expensive since the complexity of the bounding procedure is at least $O(n^4)$. Note, here we listed MEVB in part 2 because the maximum number of iterations used in [31] was large ($k = 200$). We observe that for the bounds in part 2 of table 1, TB(3) and TB(2) are clearly the best. Hence, we are more interested in comparing the bounds in part 1 of table 1.

In the next two tables, we focus on the comparison of the inexpensive lower bounds. In particular, we compare GLB, MCCR, and EVB1 with the new bounds LB1 and LB2. Table 2 shows the results on 16 problems in classes II, III, IV, and V. Four problems are taken from each class, one for each size of 5, 10, 20, 40. Note that EVB1 is valid only for symmetric QAPs.

Table 1

Part 1. Comparison of bounds on Nugent test problems.

n	BKV	GLB	MCCR	EVB1	EVB2	EVB3	IVB	LB1	LB2
6	86	82	82	70	73	70	69	82	82
8	214	186	186	160	164	174	167	186	186
12	578	493	493	446	448	495	472	493	493
15	1150	963	963	927	934	989	973	963	963
20	2570	2057	2057	2075	2085	2229	2196	2057	2057
30	6124	4539	4558	4982	5005	5349	5265	4558	4558
		$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^3)$

Table 1

Part 2. Comparison of bounds on Nugent test problems.

n	MEVB	CM(k)	AX(k)	CG	FY1	FY2	TB(1)	TB(2)	TB(3)
6	70	82(3)	82(1)	82	86	82	82	82	86
8	174	188(6)	188(3)	190	194	187	186	190	195
12	495	496(7)	495(3)	500	–	–	494	498	503
15	989	972(6)	972(6)	–	–	–	964	976	984
20	2229	2067(8)	2071(4)	–	–	–	2062	2092	2120
30	5349	–	–	–	–	–	4554	4599	4671
	$O(n^3)$	$O(kn^5)$	$O(kn^5)$	$O(n^7)$	–	–	$O(n^4)$	$O(n^5)$	$O(n^6)$

The results in table 2 indicate that LB2 is the overall best bound. It is better than EVB1 except for $n = 10$ in class II. It is better than GLB except for $n = 10$ in class II and $n = 5$ in class V. It is better than LB1 except for $n = 10$ in class II and $n = 40$ in class IV. It is always better than MCCR. LB1 is the second best bound.

In table 3, we provide the comparison of GLB, MCCR, EVB1, and the new lower bounds over 40 randomly generated problems in each of the classes II, III, IV, and V. For each class, 10 problems were generated for each size of 5, 10, 20, and 40. The bounds were reported relative to the GLB. To reveal the performance of the lower bounds in details, we have three parts for the table.

Table 2

Comparison of bounds on random test problems.

Class	n	GLB	MCCR	EVB1	LB1	LB2
II	5	424	373	426	424	436
	10	1260	1212	1251	1260	1245
	20	25636	25135	25429	25636	25708
	40	402455	399161	400409	402455	402620
III	5	326	322	322	326	330
	10	1310	1200	1287	1310	1310
	20	25829	25450	25611	25829	25845
	40	399562	397041	397542	399562	399699
IV	5	184	202	–	212	220
	10	1452	1423	–	1470	1478
	20	24585	23945	–	24678	24723
	40	409879	405602	–	411376	410798
V	5	219	203	–	208	216
	10	1508	1427	–	1491	1513
	20	28381	28085	–	28382	28464
	40	428244	424572	–	429283	429750

Table 3

Part 1. Minimum ratio over GLB.

Class	n	GLB	MCCR	EVB1	LB1	LB2
II	5	1.000	0.880	0.939	1.000	0.997
	10	1.000	0.924	0.959	1.000	0.988
	20	1.000	0.969	0.989	1.000	1.000
	40	1.000	0.992	0.995	1.000	1.000
III	5	1.000	0.857	0.887	1.000	0.948
	10	1.000	0.906	0.970	1.000	0.995
	20	1.000	0.970	0.989	1.000	0.997
	40	1.000	0.989	0.995	1.000	1.000
IV	5	1.000	0.927	–	0.981	1.000
	10	1.000	0.915	–	0.994	1.006
	20	1.000	0.972	–	1.000	1.003
	40	1.000	0.990	–	1.001	1.002
V	5	1.000	0.887	–	0.936	0.961
	10	1.000	0.888	–	0.997	1.001
	20	1.000	0.974	–	1.001	1.006
	40	1.000	0.990	–	1.002	1.002

Table 3

Part 2. Average ratio over GLB.

Class	n	GLB	MCCR	EVB1	LB1	LB2
II	5	1.000	0.982	1.010	1.000	1.039
	10	1.000	0.943	0.981	1.000	1.000
	20	1.000	0.983	0.992	1.000	1.002
	40	1.000	0.995	0.995	1.000	1.001
III	5	1.000	0.954	0.969	1.000	1.005
	10	1.000	0.948	0.987	1.000	1.007
	20	1.000	0.981	0.991	1.000	1.001
	40	1.000	0.994	0.996	1.000	1.001
IV	5	1.000	1.015	–	1.044	1.066
	10	1.000	0.961	–	1.008	1.018
	20	1.000	0.982	–	1.005	1.008
	40	1.000	0.992	–	1.003	1.004
V	5	1.000	0.996	–	0.997	1.034
	10	1.000	0.963	–	1.014	1.025
	20	1.000	0.983	–	1.004	1.008
	40	1.000	0.993	–	1.002	1.003

Table 3

Part 3. Maximum ratio over GLB.

Class	n	GLB	MCCR	EVB1	LB1	LB2
II	5	1.000	1.100	1.088	1.000	1.090
	10	1.000	0.962	0.998	1.000	1.023
	20	1.000	0.996	0.995	1.000	1.005
	40	1.000	0.997	0.997	1.000	1.002
III	5	1.000	1.037	1.045	1.000	1.057
	10	1.000	1.001	1.003	1.000	1.024
	20	1.000	0.991	0.993	1.000	1.005
	40	1.000	0.996	0.997	1.000	1.001
IV	5	1.000	1.116	–	1.152	1.196
	10	1.000	1.018	–	1.047	1.062
	20	1.000	0.992	–	1.009	1.013
	40	1.000	0.997	–	1.004	1.004
V	5	1.000	1.058	–	1.078	1.116
	10	1.000	1.022	–	1.050	1.061
	20	1.000	0.994	–	1.008	1.012
	40	1.000	0.996	–	1.003	1.005

In part 1 of table 3, the entry for a lower bound for problems of a fixed size contains the minimum of the ratio of this bound over the GLB for the 10 problems. In part 2 of table 3, the entry contains the average of the ratios for the 10 problems. In part 3 of table 3, the entry contains the maximum of the ratios for the 10 problems.

Part 1 of table 3 indicates that LB2 is again the best bound among all the bounds in comparison. In particular, LB2 is better than MCCR and EVB in all cases. LB2 is better than GLB for the asymmetric problems (classes IV and V, however, it is worse than GLB for the symmetric problems (classes II and III). LB1 and GLB are tied on the second best. However, part 1 of table 3 is not very conclusive since only the worst case among 10 problems of a fixed size in a class is considered. Now, let us look at the other parts of table 3.

In parts 2–3 of table 3, the results clearly indicated that LB2 and LB1 are the best bounds in all cases. In comparison with EVB1, LB2 is in general about 0.5–1 percent above EVB1 on the average. In some cases, LB2 is better than GLB for more than 2 percent. For all classes and sizes, LB2 is slightly better than GLB on the average. With respect to MCCR, one can see that it is worse than GLB on the average and worse than LB1 and LB2. EVB1 also failed in this competition with GLB in the average sense.

6. Concluding remarks

The computational results in the previous section indicate that the new lower bounds are better than GLB, EVB1, MCCR on the random problems, and are competitive on the Nugent test problems. Since the new bounds are very fast to compute, they should be incorporated in a branch-and-bound type algorithm for the QAP.

It is interesting to observe that the new bounds are consistently better than all other bounds tested for the random problems with hyperexponential distribution (problem classes III and V). Hyperexponential distributions are characterized by a coefficient of variation greater than 1. Thus, the variance in the flow and distance matrices are high for problems in classes III and V, where all random problem instances tested had coefficients of variation in the range 1 to 5. Since the basic tenet embodied in our bounds is based on variance reduction, the results indicate that our bounds should perform better than the others tested if the coefficient of variation is higher, say, in the 100's. We report on results for instances with high variances in [21], where we present a branch and bound implementation using these new bounds.

Some questions still remain. The reduction schemes proposed are more effective when the variances of the flow and distance matrices are large. When the variances of the matrices are small, the proposed lower bounds degenerate to GLB. Hence, one question is how to design reduction schemes which can be effective when the variances of matrices are small. Another question is, for QAPs with certain structures, how to provide better reduction schemes than those proposed in this paper. For example, the proposed bounds tie with GLB on the Nugent test problems of sizes $n \leq 20$. We know that the distance matrices for the Nugent test problems are derived from grid graphs. There may exist better reduction schemes than those proposed here.

References

[1] A. Assad and W. Xu, On lower bounds for a class of quadratic {0,1} programs, Oper. Res. Lett. 4(1985)175–180.

[2] R. Burkard, Die Störungsmethode zur Lösung quadratischer Zuordnungsprobleme, Oper. Res. Verfahren 16(1973)84–108.

[3] R. Burkard, Locations with spatial interactions: the quadratic assignment problem, in: *Discrete Location Theory*, ed. P.B. Mirchandani and R.L. Francis (Wiley, Berlin, 1990) chapter 9.

[4] R. Burkard and T. Bonniger, A heuristic for quadratic Boolean programs with applications to quadratic assignment problems, Euro. J. Oper. Res. 13(1983)374–386.

[5] R. Burkard and F. Rendl, A thermodynamically motivated simulation procedure for combinatorial optimization problems, Euro. J. Oper. Res. 17(1984)169–174.

[6] R. Burkard and K. Stratmann, Numerical investigations on quadratic assignment problems, Naval Res. Log. Quarterly 25(1978)129–148.

[7] R. Burkard and J. Offerman, Entwurf von Schreibmachinentastaturen mittels quadratischer Zuordnungsprobleme, Z. Oper. Res. 21(1977)B121–B132.

[8] P. Carraresi and F. Malucelli, A new lower bound for the quadratic assignment problem. Oper. Res. 40(1992)S22–S27.
[9] N. Christofides and M. Gerrard, A graph theoretic analysis of bounds for the quadratic assignment problem, in: *Studies on Graphs and Discrete Programming,* ed. P. Hansen (North-Holland, 1981) pp. 61–68.
[10] C. Edwards, A branch and bound algorithm for the Koopmans–Beckman quadratic assignment problem, Math. Progr. Study 13(1980)35–52.
[11] G. Finke, R. Burkard and F. Rendl, Quadratic assignment problems, Ann. Discr. Math. 31(1987)61–82.
[12] A. Frieze and J. Yadegar, On the quadratic assignment problem, Discr. Appl. Math, 5(1983)89–98.
[13] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman, New York, 1979).
[14] P. Gilmore, Optimal and suboptimal algorithms for the quadratic assignment problem J. SIAM 10(1962)305–313.
[15] G. Gordon, *System Simulation* (Prentice–Hall, Englewood Cliffs, NJ, 1969).
[16] S. Hadley, F. Rendl and H. Wolkowicz, Bounds for the quadratic assignment problem using continuous optimization tehcniques, in: *Integer Programming and Combinatorial Optimization* (University of Waterloo Press, 1990) pp. 237–248.
[17] S. Hadley, F. Rendl and H. Wolkowicz, A new lower bound via projection for the quadratic assignment problem, Math. Oper. Res. 17(1992)727–739.
[18] T. Koopmans and M. Beckmann, Assigned problems and the location of economic activities, Econometrica 25(1957)53–76.
[19] J. Krarup and P. Pruzan, Computer-aided layout design, Math. Progr. Study 9(1978)75–94.
[20] E. Lawler, The quadratic assignment problem, Manag. Sci. 9(1963)586–599.
[21] Y. Li, P. Pardalos, K. Ramakrishnan and M. Resende, A branch-and-bound algorithm for the quadratic assignment problem, Technical Report, AT&T Bell Laboratories, Murray Hill, NJ 07974-2070 (December 1992).
[22] Y. Li and P.M. Pardalos, Generating quadratic assignment test problems with known optimal permutations, Comp. Optim. Appl. (1992), to appear.
[23] Y. Li and P.M. Pardalos, Parallel algorithms for the quadratic assignment problem, in: *Recent Advances in Optimization and Parallel Computing* (Elsevier, Amsterdam, 1922) pp. 177–189.
[24] K. Murthy, P. Pardalos and Y. Li, A local search algorithm for the quadratic assignment problem, Informatica 3(1992).
[25] C. Nugent, T. Vollmann and J. Ruml, An experimental comparison of techniques for the assignment of facilities to locations, J. Oper. Res. 16(1969)150–173.
[26] L. Nyhoff and S. Leestma, *Fortran 77 for Engineers and Scientist,* 2nd ed. (Macmillan, New York, 1988).
[27] G. Palubetskis, Generation of quadratic assignment test problems with known optimal solutions, Zh. Vychisl. Mat. Fiz. 28(1988)1740–1743 (in Russian).
[28] P. Pardalos and J. Crouse, A parallel algorithm for the quadratic assignment problem, in: *Proc. Supercomputing 1989 Conf.* (ACM Press, 1989) pp. 351–360.
[29] P. Pardalos and G. Rodgers, Parallel branch and bound algorithms for unconstrained quadratic 0–1 programming, in: *Impact of Recent Advances on Operations Research* (North-Holland, 1989) pp. 131–143.
[30] P. Pardalos and J. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science, No. 268 (Springer, 1987).
[31] F. Rendl and H. Wolkowicz, Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem, Math. Progr. 53(1992)63–78.
[32] C. Roucairol, Affection quadratique, Ph.D. Thesis, Université Pierre et Marie Curie, Paris VI (1976).

[33] C. Roucairol, A reduction method for quadratic assignment problems, Oper. Res. Verfahren 32(1979)183–187.
[34] C. Roucairol, Un nouvel algorithme pour le problème d'affectation quadratique, RAIRO 13(1979)275–301.
[35] S. Sahni and T. Gonzalez, P-complete approximation problems, J. ACM 23(1976)555–565.
[36] L. Schrage, A more portable Fortran random number generator, ACM Trans. Math. Software 5(1979)132–138.
[37] J. Skorin-Kapov, Tabu search applied to the quadratic assignment problem, ORSA J. Comp. 2(1990)33–45.
[38] E. Taillard, Robust tabu search for the quadratic assignment problem, Parallel Comp. 17(1991)443–455.