DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 00, 0000

[1]

The Quadratic Assignment Problem: A Survey and Recent Developments

PANOS M. PARDALOS, FRANZ RENDL, AND HENRY WOLKOWICZ

ABSTRACT. Quadratic Assignment Problems model many applications in diverse areas such as operations research, parallel and distributed computing, and combinatorial data analysis. In this paper we survey some of the most important techniques, applications, and methods regarding the quadratic assignment problem. We focus our attention on recent developments.

1. Introduction

Given a set $\mathcal{N} = \{1, 2, \ldots, n\}$ and $n \times n$ matrices $F = (f_{ij})$ and $D = (d_{kl})$, the quadratic assignment problem (QAP) can be stated as follows:

$$\min_{p \in \Pi_{\mathcal{N}}} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{p(i)p(j)} + \sum_{i=1}^{n} c_{ip(i)},$$

where $\Pi_{\mathcal{N}}$ is the set of all permutations of $\mathcal{N}$. One of the major applications of the QAP is in location theory where the matrix $F = (f_{ij})$ is the flow matrix, i.e. f_{ij} is the flow of materials from facility i to facility j, and $D = (d_{kl})$ is the distance matrix, i.e. d_{kl} represents the distance from location k to location l [**62, 67, 137**]. The cost of simultaneously assigning facility i to location k and facility j to location l is $f_{ij}d_{kl}$. The objective is to find an assignment of all facilities to all locations (i.e. a permutation $p \in \Pi_{\mathcal{N}}$), such that the total cost of the assignment is minimized. Throughout this paper we often refer to the QAP in the context of this location problem.

In addition to its application in facility location problems, the QAP has been found useful in such applications as scheduling [**88**], the backboard wiring problem in electronics [**240**], parallel and distributed computing [**24**], and statistical data analysis [**118**]. Other applications may be found in [**77, 138, 159**].

The term "quadratic" comes from the reformulation of the problem as an optimization problem with a quadratic objective function. There is a one-to-one correspondence between $\Pi_{\mathcal{N}}$ and the set of $n \times n$ permutation matrices

[1]This paper and a separate bibliography file (bib file) is available by anonymous ftp at orion.uwaterloo.ca in the directory pub/henry/qap.

1991 *Mathematics Subject Classification.* Primary 90B80, 90C20, 90C35, 90C27; Secondary 65H20, 65K05.

Key words and phrases. Combinatorial optimization, quadratic assignment problem, graph partitioning, survey, exact algorithms, heuristics, algorithms, test problems, bibliography.

$X = (x_{ij})_{n \times n}$. The entries of each such matrix must satisfy:

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1, \dots, n, \tag{1.1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = 1, \dots, n, \tag{1.2}$$

$$x_{ij} \in \{0, 1\}, \ \ i = 1, \dots, n, \ j = 1, \dots, n, \tag{1.3}$$

$$x_{ij} = \begin{cases} 1 & \text{if facility } i \text{ is assigned to location } j \\ 0 & \text{otherwise.} \end{cases}$$

With the above constraints on x, we have the following equivalent formulation for the quadratic assignment problem, working on the space of permutation matrices,

$$\min \ \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{i,j=1}^{n} c_{ij} x_{ij}. \tag{1.4}$$

The paper is organized as follows. In Section 2 we present the mathematical tools and techniques that have proven to be useful for QAP. This includes various formulations of the problem and representations of the feasible set. The various representations of the feasible set and objective function lead directly to tractable relaxations. We include optimality conditions and representations of derivatives for QAP and its relaxations. In Section 3, we present several applications of QAP, both theoretical and practical. We also include generalizations. The computational complexity is described in Section 4. A survey of current numerical methods is presented in Section 5. Test problem generation with known optimal permutation is discussed in Section 6. Concluding remarks are made in Section 7.

2. Mathematics of QAP

In this section we outline the mathematical tools and techniques that are useful and interesting for QAP.

2.1. Formulations. Several formulations have been used in the literature to study the QAP. We outline several of these formulations now. (Please see [**100**] for more details and more formulations.)

2.1.1. *Koopmans-Beckmann.* The QAP was introduced in 1957 by Koopmans and Beckmann [**137**] using the formulation presented above in equations (1.1) to (1.4). This model was formulated to study the assignment of a set of economic activities to a set of locations.

2.1.2. *Trace.* For simplicity, we mainly use the *trace formulation* in this paper

$$\mathbf{Q}AP \qquad \min_{X\in\Pi} \ f(X) = trace\,(AXB + C)X^t,$$

where $.^t$ denotes transpose, Π is the set of permutation matrices, and $trace$ stands for the sum of the diagonal elements. We assume A and B to be real symmetric $n \times n$ matrices and $C \in \Re^{n\times n}$. This formulation was introduced in [**65, 66**]. It allows for easy manipulation and relaxation of the data and truly illustrates the n dimensional nature of the problem.

Several elementary properties of the trace can be exploited. For example: the trace of a symmetric matrix is the sum of the eigenvalues; and the trace of a product satisfies $trace\,MN = trace\,NM = trace\,N^tM^t$. Moreover, the trace provides a valid *inner product* on the space of real (or complex) $m \times n$ matrices

$$\langle M, N\rangle = trace\,M^*N,$$

where $\cdot^*$ stands for complex conjugate.

2.1.3. *Kronecker Product.* The trace formulation of the objective function is a compact form of representing the quadratic form with the matrix X as the variable. The Hessian of this quadratic form is the *tensor product* or *Kronecker product*

$$A \otimes B = (a_{ij}b_{kl}) = (a_{ij}B)\,,$$

i.e. the matrix formed from all possible products of elements from A and B. We can use the notation that $vec\,(X) \in \Re^{n^2}$ denotes the vector formed from the columns of the matrix X. Then the objective function is equivalent to

$$f(X) = \mathrm{vec}\,(X)^t(A \otimes B)\mathrm{vec}\,(X) + \mathrm{vec}\,(C)^t\mathrm{vec}\,(X).$$

Derivatives and algebraic manipulations can all be done via the Kronecker product. For example, the eigenvalues of the Kronecker product are the n^2 eigenvalues formed from all possible products of the eigenvalues of A and B. However, using the Kronecker product hides the structure of the problem and increases the complexity in that we do not take advantage of the hidden fact that we are really working on an n dimensional problem. For that reason, the Kronecker product is rarely used and we do not study it further. (See [**96**] for details on manipulations and calculus involving Kronecker products.)

2.2. The Feasible Set and Perturbations. The feasible set for QAP consists of all the possible assignments of n objects to n locations. The extreme points, or vertices, of the *bipartite perfect matching polytope* (as defined by the nonnegative vectors $x = (x_{ij}) \in \Re^{n^2}$ satisfying the constraints (1.1) and (1.2)) are the incidence vectors of all possible assignments. Alternatively, in the trace formulation, the feasible set consists of all the permutation matrices. It is well known that the permutation matrices satisfy

$$\begin{aligned} \Pi &= \mathcal{O} \cap \mathcal{E} \cap \mathcal{N}, \\ &= \mathcal{S} \cap \mathcal{E} \cap \mathcal{N}, \end{aligned} \tag{2.1}$$

where: $\mathcal{O} = \{X : X^tX = I\}$ is the set of *orthogonal* matrices; $\mathcal{S} = \{X : trace\, X^tX = n\}$; while $\mathcal{E} = \{X : Xu = X^tu = u\}$ is the set of all matrices having *row and column sums equal to one*; and $\mathcal{N} = \{X : X \geq 0\}$ is the set of *nonnegative* matrices. We let $\mathcal{D} = \mathcal{E} \cap \mathcal{N}$ denote the set of *doubly stochastic* matrices. A well known theorem of Birkoff [**21**] states that the convex hull of the permutation matrices is the set of doubly stochastic matrices, $conv\, \Pi = \mathcal{D}$. Thus the set of doubly stochastic matrices corresponds to the bipartite perfect matching polytope. (For more details on the above, see e.g. [**28**].)

The properties of the feasible set allow for perturbations of the objective function without changing the optimal solution of the original problem. These perturbations do change relaxations for the problem and so are important in improving bounding techniques. Two standard perturbations, constant row and column perturbations and diagonal perturbations, are known to have this property. Specifically, suppose that $e, f, r, s \in \Re^n$ and define

$$A(e,r) = A + eu^t + ue^t + \mathrm{d}iag\,(r),$$

$$B(f,s) = B + fu^t + uf^t + \mathrm{d}iag\,(s),$$

$$\begin{aligned} C(e,f,r,s) \;\; = \;\; & C + 2Auf^t + 2eu^tB - 2nef^t - 2\textstyle\sum_k e_k uf^t \\ & + \mathrm{d}iag\,(A)s^t + r\ \mathrm{d}iag\,(B)^t - 2es^t - 2rf^t - rs^t, \end{aligned}$$

where $\mathrm{d}iag$ changes a vector to a diagonal matrix and, conversely, it changes a matrix to a vector formed from the diagonal elements. Then

$$trace\ (AXB + C)X^t = trace\,(A(e,r)XB(f,s) + C(e,f,r,s))X^t,\ \forall X \in \Pi.$$

(Note that symmetry is preserved by these transformations.)

If we keep the constraint $X \in \mathcal{E}$, i.e. X has row and column sums equal to 1, then the constant row and column perturbations are redundant and can be ignored, i.e. only the diagonal perturbations need be used. (See e.g. [**102**] for details. The question of which perturbations are needed, under which relaxations, is discussed in [**198**].)

2.3. Relaxations. Since QAP is an NP-hard problem, the equivalent expression of the constraints in (2.1) are very useful. We immediately get representations for relaxations. These relaxations remove the combinatorial nature of the problem and allow for solutions using continuous optimization techniques. The strategy is to relax the objective function and/or the constraints, in order to get a tractable problem. These problems do not provide useful approximations in general. But, we can then find the best relaxation of a family of these problems over the above mentioned perturbations. (More details are provided when we discuss lower bounds in Section 5.2.)

2.3.1. *Linearization.* If the quadratic term of QAP vanishes, then we have an ordinary linear assignment problem which can be solved very efficiently. In general, the QAP can be relaxed to a (0,1)-linear integer program. This can be done by introducing new binary variables $y_{ijkl} = x_{ij}x_{kl}$. These new variables replace the occurrence of quadratic terms in the objective function. New constraints are added to ensure consistency with the original problem. Typically, the convex hull of the constraint set is used in order to obtain an ordinary linear programming problem. See e.g. [**143, 155, 154, 132, 8, 18, 55, 34, 81, 2, 100**].

2.3.2. *Quadratic Programming.* As stated above in Section 2.2, perturbations of the objective function can be done without changing the optimum of QAP. But these perturbations can be very useful in forming relaxations. Since diagonal perturbations are allowed, we can perturb the diagonal elements of A and B in order to make the Hessian of the objective function convex. We can then take the convex hull of the feasible set, i.e. we relax the feasible set to the doubly stochastic matrices $\mathcal{D}$. This results in a standard quadratic programming problem that can be solved by well known methods. In fact, we do not need to make the Hessian positive semidefinite on all matrices X, but rather only on the span of the feasible set, i.e. on the span of the doubly stochastic matrices, or equivalently, on the span of matrices with row and column sums equal to 1.

However, this relaxation does not fully exploit the structure of the problem, since it treats the objective function as a quadratic form over $\Re^{n^2}$.

2.3.3. *Trust Region Subproblems.* If we use the second representation in (2.1) and relax the constraint set to $X \in \mathcal{S} \cap \mathcal{E}$, then we do not have to worry about convexity of the objective function, i.e. we obtain a tractable problem called a trust region subproblem. (In [**241**] it is shown that these problems are really implicit convex problems, since their dual problems are concave maximization problems.) However, these problems still do not exploit the structure of QAP.

2.3.4. *Parametrization of Permutation and Orthogonal Groups.* The vector e of ones is both a right and left eigenvector corresponding to an eigenvalue of 1, for every permutation matrix. This fact can be exploited to project the feasible set of QAP onto the span of the doubly stochastic matrices while not losing the special trace structure of the objective function.

Let the $n \times (n-1)$−matrix V be such that

$$V^t e = 0; \; V^t V = I_{n-1}.$$

The columns of V therefore constitute an orthonormal basis of $\{e\}^\perp$. Further, let

$$v := \frac{e}{||e||}; \; P = [v \,\vdots\, V] \in \mathcal{O}.$$

Thus $Q := VV^t = I - vv^t$ describes the orthogonal projection on $\{e\}^\perp$. The parametrization of the permutation matrices follows. (See [**102**].)

PROPOSITION 2.1. *Let X be $n \times n$ and Y be $(n-1) \times (n-1)$. Suppose that X and Y satisfy*

$$X = P \begin{bmatrix} 1 & 0 \\ 0 & Y \end{bmatrix} P^t. \tag{2.2}$$

Then

$$X \in \mathcal{E},$$

$$X \in \mathcal{N} \iff VYV^t \geq -vv^t,$$

$$X \in \mathcal{O}_n \iff Y \in \mathcal{O}_{n-1}.$$

Conversely, if $X \in \mathcal{E}$, then there is a Y such that (2.2) holds.

Relaxing the constraints to $X \in \mathcal{O}$ or to $X \in \mathcal{O} \cap \mathcal{E}$ removes the combinatorial nature of the problem. The resulting problem can be split into an eigenvalue problem for the quadratic part of the objective function and a standard linear programming problem for the linear part. (This is discussed in detail in the section on bounds, Section (5.2.2).)

In addition, one can parametrize the orthogonal constraint $X \in \mathcal{O}$ by noting that it is equivalent to the matrix exponential $X = \exp^S$ for some skew symmetric matrix S, see e.g. [**172, 60**]. This results in an unconstrained problem over the space of skew symmetric matrices.

2.3.5. *Semidefinite Programming.* The orthogonal constraint can be relaxed to $XX^t \preceq I$, i.e. XX^t is negative semidefinite. This relaxation is discussed in detail in an accompanying paper in these proceedings [**129**].

2.4. Derivatives and Optimality Conditions. QAP is an NP-hard problem. It is therefore not surprising that verifying optimality is also an NP-hard problem. In fact, even checking local optimality is a hard problem. (See Section 4.2.) However there are tractable optimality conditions for the relaxations.

2.4.1. *Differentials.* We first present the derivatives of the functions involved in QAP. Let

$$k(X) = XBX^t, \; g(X) = X^tX - I, \; f(X) = trace\, AXBX^t.$$

Then the corresponding differentials in the (matrix) direction h are

$$\begin{aligned} dk(X;h) &= XBh^t + hBX^t; \\ dg(X;h) &= Xh^t + h^tX; \\ df(X;h) &= trace\, A(dk(X;h)) = trace\, A(XBh^t + hBX^t). \end{aligned}$$

2.4.2. *Optimality Conditions.* First, consider the relaxation of QAP when $C = 0$ and the constraint set consists of the orthogonal matrices. From the above notation this corresponds to

$$\min f(X) \text{ subject to } g(X) = 0.$$

The Lagrangian for this problem is

$$f(X) + trace\, Sg(X),$$

where the Lagrange multiplier S is a symmetric matrix. We can differentiate the Lagrangian using the above differentials. If we set the derivative to 0, we get the condition that $AXB + XS = 0$ or $X^tAXB = -S$. We conclude from $S = S^t$, that X^tAX and B commute and so are mutually diagonalizable. This yields the minimum scalar product of the eigenvalues used in the bounds in Theorem 5.1. An improved bound can be obtained by projecting the feasible set onto the span of the doubly stochastic matrices. This uses a parametrization of the permutation matrices, see 2.3.4. (More details can be found in [**210, 102, 129**].)

2.4.3. *Global Optimality.* If we relax QAP to a quadratic (convex) program, i.e. we relax the constraint set to the set of doubly stochastic matrices $\mathcal{D}$ while perturbing the objective function to make it convex on the span of $\mathcal{D}$, then the Karush-Kuhn-Tucker optimality conditions characterize optimality. This is well known for convex programming problems where a constraint qualification holds. The primal relaxed problem is equivalent to the min-max of the Lagrangian

$$(2.3) \quad \min_{X \in \mathcal{N}} \max_{\lambda_1, \lambda_2 \in \Re^n} trace\ \left[(AXB + C)X^t\right] + \lambda_1^t(Xe - e) + \lambda_2^t(X^te - e),$$

while the dual is the max-min problem

$$(2.4) \quad \max_{\lambda_1, \lambda_2 \in \Re^n} \min_{X \in \mathcal{N}} trace\ \left[(AXB + C)X^t\right] + \lambda_1^t(Xe - e) + \lambda_2^t(X^te - e).$$

The above relaxation provides lower bounds for QAP. Thus, for each perturbation defined in Section 2.2, with the above convexity assumption on the span of $\mathcal{D}$, the dual problem provides lower bounds for QAP because we can effectively characterize global optimality for it. This is no longer true if we do not make the convexity assumptions on the objective function.

Statements about global optimality for nonconvex problems, such as QAP itself, are much harder to make. A characterization for general problems can be found in [**114**].

2.4.4. *Local Optimality.* For the general quadratic programming relaxation, even in the nonconvex case, i.e. for a general objective function constrained to D, we still get necessary and sufficient local optimality conditions. This is due to the quadratic nature of the problem, i.e. the second order optimality conditions are necessary and sufficient. However, this is not the case for problems with nonnegativity constraints. It has been shown in [**191**] that the problem of checking local optimality (and the problem of checking if a local minimum is strict) in quadratic programming with linear constraints is NP-hard. Pardalos and Vavasis [**192**] have also shown that quadratic programming with one negative eigenvalue (all others zero) remains an NP-hard problem.

2.5. Nonsymmetric QAP. Applications for QAP usually involve symmetric matrices A, B. If one of A or B is symmetric, then we can still get an equivalent symmetric QAP by symmetrizing the other, e.g. replace B by $(B+B^t)/2$. If both A and B are not symmetric, then we can still symmetrize the quadratic form by using the Kronecker product, but we then lose the trace structure of the problem.

However, even if both A and B are not symmetric, we can still obtain meaningful bounds by moving into the space of Hermitian matrices. See [**103**].

3. Applications, Generalizations and special cases

Applications for QAP are many and varied. Several are mentioned in the introduction above. We also point out that [**33**] summarizes recently published applications of quadratic assignment problem.

We will describe now first some generalizations of the quadratic assignment problem, and then discuss some interesting special cases.

3.1. The 3-index Assignment Problem. The three-index (or 3-dimensional) assignment problem of order n can be stated as a (0,1)-programming problem of the following form:

$$(3.1)\qquad \begin{array}{ll} \min & \sum\{c_{ijk}x_{ijk} : i \in I, j \in J, k \in K\}, \\ \text{s.}t. & \sum\{x_{ijk} : j \in J, k \in K\} = 1, \ \ \forall i \in I, \\ & \sum\{x_{ijk} : i \in I, k \in K\} = 1, \ \ \forall j \in J, \\ & \sum\{x_{ijk} : i \in I, j \in J\} = 1, \ \ \forall k \in K, \\ & \quad x_{ijk} \in \{0,1\}, \quad \forall i, j, k, \end{array}$$

where I, J and K are disjoint index sets with $|I| = |J| = |K| = n$. Note that the number of variables of the three-dimensional assignment problem of order n is n^3.

From the complexity point of view, it has been shown that the three-dimensional assignment problem is NP-hard [**131**]. Most ot the proposed algorithms for this problem are implicit enumeration methods. Some of the proposed algorithms include those of Vlach [**247**], Pierskalla [**196, 197**] and Leue [**145**]; a primal-dual algorithm described by Hansen and Kaufman [**106**]; a branch and bound algorithm using a Lagrangian dual and subgradient optimization implemented by Fröhlich [**83**] and discussed by Burkard and Fröhlich [**38**]. Also see Burkard and Rudolf [**42**]. More recently, Balas and Saltzman [**11**] developed a branch and bound algorithm that also uses facet-defining inequalities in a Lagrangian fashion with subgradient optimization.

Let A be the coefficient matrix of the constraint set of (3.1). Then $R = I \cup J \cup K$ is the row index set of A. Let S be the column index set of A. Let G_A be the intersection graph of A, i.e., the graph that has a vertex for every column of A and an edge for every pair of non-orthogonal columns. Let

$$P = \{x \in R^{n^3} : Ax = e, x \geq 0\},$$

where $e = (1, ..., 1)^t \in R^{3n}$. Then

$$P_I = conv\,\{x \in \{0,1\}^{n^3} : x \in P\}$$

is the three-index assignment polytope of order n.

Balas and Saltzman [**10**] started to study the facial structure of P_I. They gave an $O(n^4)$ procedure to detect whether there is a clique facet of P_I, violated by a given noninteger point x. In [1], Balas and Qi gave an $O(n^3)$ procedure to do this. Since the number of variables of (3.1) is n^3, an $O(n^3)$ separation algorithm for a facet class of P_I is linear-time and its complexity is best possible. Balas and Qi [**9**], Gwan and Qi [**99**] also gave linear-time separation algorithms for other two facet classes of P_I, identified in [**10**]. More recent results can be found in [**202**]].

Other papers on the three-index assignment problem include [**43, 68, 80, 81**] and [**219**].

3.2. The Quadratic Semiassignment Problem. The quadratic semiassignment problem (QSA) unifies some interesting combinatorial optimization problems. The general problem has the form:

$$\begin{array}{lll} & \min & \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ik}x_{jk} \\ (3.2) & \text{s.}t. & \sum_{k=1}^{m} x_{jk} = 1, \;\; j = 1,\ldots,n \\ & & x_{jk} \in \{0,1\}, \quad \forall j,k, \end{array}$$

Some special cases of this problem include the clustering problem, the equipartition problem, and the m-coloring problem on graphs [**234**], [**235**].

Given n objects and an $n \times n$ dissimilarity matrix $C = (c_{ij})$, the "clustering problem" is to find a partition of the objects into m classes (clusters) which minimizes the sum of the dissimilarities between objects belonging to the same class.

The "equipartition problem" is the following: Given n objects with weights w_i, $i = 1, \ldots, n$, find a partition of the objects into m classes so as to minimize the variance of the class weights. This problem can be formulated as a quadratic semiassignment problem with $c_{ij} = w_i w_j$ for all i and j.

The "m-coloring problem" is also a special case of the (QSA) problem. Given a graph $G(V, A)$, the graph admits a coloration of its vertices with m colors (adjacent vertices have different colors) if and only if the following problem

$$
\begin{array}{lll}
 & \min & \sum_{k=1}^{m} \sum_{(i,j)\in A} x_{ik} x_{jk} \\
(3.3) & & \sum_{k=1}^{m} x_{jk} = 1, \;\; j = 1, \ldots, n \\
 & \text{s.}t. & x_{jk} \in \{0,1\}, \quad \forall j,k,
\end{array}
$$

has optimal function value zero.

3.3. The Biquadratic Assignment Problem. Recently Burkard, Cela and Klinz [**35**] (see the paper in this volume) introduced a fourth order generalization of the quadratic assignment problem. Let $A = (a_{i,j,k,l})$ and $B = (b_{m,p,s,t})$ be two arrays of n^4 elements. Then the *biquadratic assignment problem* asks to minimize

$$\sum_{i,j,k,l,m,p,s,t} a_{i,j,k,l} b_{m,p,s,t} x_{im} x_{jp} x_{ks} x_{lt}$$

over all permutation matrices X. This problem arises in the field of VLSI synthesis. In [**35**] various formulations of this problem are described. Also, lower bounds and some methods to construct instances with known optimal solution are presented. There are still many open problems related to this generalized model of the quadratic assignment problem. In particular it would be interesting to explore eigenvalue related techniques to this problem.

After having described some relatives of the quadratic assignment problem, which typically are at least as hard or harder to solve we will now focus on specially structured quadratic assignment problems which lead to simplifications of the problem.

3.4. Special Cases. The quadratic assignment problem can be formulated very naturally in a graph theoretical context. This formulation was investigated first by Christofides and Gerrard [**54**], and later by Bokhari [**23**] and Rendl [**208**]. We review this formulation and present several applications to other optimization problems on graphs that can be derived from this formulation.

Let $G = (V, E)$ and $G' = (V', E')$ be graphs. G is isomorphic to G' ($G \approx G'$ for short), if there exists an adjacency preserving bijection $p : V \mapsto V'$, i.e.

$$ij \in E \iff p(i)p(j) \in E'.$$

We denote by $\Pi(G, G')$ the set of adjacency preserving mappings between G and G'. Furthermore we denote by $M(G, G')$ the set of all subgraphs H of G', which are isomorphic to G,i.e.

$$M(G, G') = \{H : H \text{ subgraph of } G', H \approx G\}.$$

The graph theoretic formulation of quadratic assignment problem was proposed by Christofides and Gerrards as follows.

Let G and G' be graphs with edge weights $a : E \mapsto \Re,\; b : E' \mapsto \Re$.

$$\min_{H \in M(G,G')} \min_{\pi \in \Pi(G,H)} \sum_{ij \in E} a_{ij} b_{\pi(i)\pi(j)}.$$

(To avoid trivialities one has to assume $|V| \leq |V'|$.) Note that if both G and G' are isomorphic to the complete graph K_n, then $M(G, G') = \{G'\}$ and $\Pi(G, G')$ contains all permutations of n elements, leading to the standard quadratic assignment problem.

The graph theoretic formulation provides many possibilities to generate special cases of the standard quadratic assignment problem. The complexity issues related to these special cases will be addressed in subsection 4. Here we will relate other difficult optimization problems on graphs to the quadratic assignment problem.

It is well known that the traveling salesman problem and the matching problem can be formulated as a special quadratic assignment problem, see [**75**]. Perhaps less known are the connections of the quadratic assignment problem to the *bandwidth problem in graphs.* To see this connection we first introduce the bandwidth problem (we refer to [**201**] for a survey on the topic):

Let G be an undirected (and unweighted) graph on n nodes. A permutation π of n elements is called a *labeling* of the nodes of G. The bandwidth of a labeling π is defined as

$$\max_{ij \in E} |\pi(i) - \pi(j)|.$$

The bandwidth σ of G is the minimum of this number over all labelings. In terms of matrices, the bandwidth problem asks for a simultaneous permutation of the rows and columns of the adjacency matrix of G such that all nonzero entries are as close as possible to the main diagonal.

Suppose the bandwidth of a given graph G is at most k. Let us denote by $P_{n,k}$ the graph on n vertices with edges ij whenever $|i-j| \leq k$. Then clearly G must be isomorphic to some subgraph of $P_{n,k}$. Conversely, if the bandwidth of G is larger than k, then there cannot exist a subgraph of $P_{n,k}$ which is isomorphic to G. If we denote the adjacency matrix of G by A and the adjacency matrix of $P_{n,k}$ by B, then we conclude:

The bandwidth of G is at most k if and only if $\max_{\pi \in \Pi} \sum_{ij} a_{ij} b_{\pi(i)\pi(j)} = 2|E|$. Therefore if some upper bound on this quadratic assignment problem has a value less than $2|E|$ for a fixed value of k, one immediately concludes that the bandwidth is larger than k. This idea was used in [**110**] to derive lower bounds on the bandwidth of graphs. In particular the following simple lower bound on the bandwidth is proved.

$$\sigma(G) \geq n\lambda_2(L)/\lambda_n(L) \; - \; 1$$

Here L denotes the Laplacian matrix of the graph, which is related to A by $L = Diag(Ae) - A$. (Recall that Ae is the vector of row sums of A.)

A similar approach works also for the problem of bounding the *1-sum of a graph*, which is the same as the bandwidth except that one minimizes

$$\sum_{ij \in E} |\pi(i) - \pi(j)|.$$

over all permutations π. Finally the problem of determining whether a *vertex separator* of some given size k exists in a graph can be modeled as a quadratic assignment problem. We refer to further details in [**110**].

In [**166, 209**] it is pointed out that the general *graph partition problem* can be modeled as a quadratic assignment problem. (See Section 4.3 for details.) It turns out however, that exploiting the special structure of the partition problem leads to more powerful results than treating this problem as a quadratic assignment problem.

4. Complexity Issues and Asymptotic Behavior

From the computational point of view the QAP is one of the most difficult problems to solve. In this section several aspects regarding the complexity of the QAP are discussed. Although computational complexity characterizes worst case instances, it also plays an important role in developing new algorithms for solving combinatorial optimization problems, analyzing their intrinsic difficulty, and revealing surprising connections among problems and their solutions.

4.1. Computational Complexity. In 1976, Sahni and Gonzalez showed that the QAP is NP-complete, which implies that finding a polynomial-time algorithm to solve it is unlikely [**221**]. In addition, they have also shown that QAP belongs even to the hardest core of this complexity class, in the sense that the problem of finding an ϵ-approximate solution of QAP remains NP-complete.

Many well known NP-Complete problems, such as the traveling salesman problem (TSP), the graph partitioning problem (GP), the maximum clique problem (MCP), can be easily formulated as special cases of the QAP:

- The traveling salesman problem (TSP): The distance matrix corresponds to the distance matrix of the TSP, the flow matrix corresponds to the adjacency matrix of a cycle of length n.
- The graph partitioning problem (GP): The distance matrix corresponds to the adjacency matrix of the GP, the flow matrix corresponds to the adjacency matrix of two disjoint complete graphs of size $n/2$ (assuming n is even).
- The maximum clique problem (MCP): To identify the existence of a clique of size k, one constructs a distance matrix corresponding to the adjacency matrix of the graph for the MCP, a flow matrix corresponds to the adjacency matrix of a clique of size k. The maximum clique can be found by solving a set of n QAPs, one for each $k, 1 \leq k \leq n$.

There are a few polynomial-time solvable special cases for the QAP. Christofides and Gerrard [**53**] investigated the conditions under which the QAP can be solved in polynomial time. They show that if both matrices A and B are weighted adjacency matrices of a tree, the problem can be solved in a dynamic programming fashion, in polynomial time. But if only 1 of the 2 matrices is a weighted adjacency matrix of a tree, the problem remains to be NP-complete since the TSP can be reduced to such a form.

Moreover, the following case is solved in $O(n \log n)$ running time by Adolphson and Hu in 1973 [**53**]: consider the case in which one matrix represents the weighted adjacency matrix of a tree, while the other one represents the distance matrix of a grid graph $G = (V, E)$, where the distances between nodes i and j is defined as follows

$$b_{ij} = \begin{cases} 1, \text{ if } (i,j) \in E, \\ \text{length of the short path between } i \text{ and } j, \text{ if } (i,j) \notin E. \end{cases}$$

Other polynomial-time solvable cases include the case in which one of the matrix is the weighted adjacency matrix of a double star (see Christofides and Gerrard [**53**]). When both distance and flow matrices are weighted adjacency matrices of series-parallel graphs containing no bipartite graph $K_{2,2}$, then again the corresponding QAP is solved in polynomial time [**208**].

4.2. Complexity of Local Search. Next we show that finding a locally optimal permutation is a difficult problem from the complexity point of view.

A local search algorithm starts with an initial feasible solution and successively moves to neighboring solutions until no further improvement is possible. To characterize the complexity of solving combinatorial optimization problems such as the QAP with local search algorithms, a Polynomial-time Local Search (PLS) class has been defined [**126**] that captures the structure of NP problems at the level of their feasible solutions and neighborhoods. Similar to NP-completeness, the concept of PLS-completeness has been defined to capture the class of the hardest problems in PLS. For certain NP-complete problems, the corresponding PLS problems have already been shown to be PLS-complete [**126, 223**]. In regard to the complexity of local search, see also [**181**] and [**191**].

4.2.1. *A K-L Type Local Search Algorithm.* Next we describe a new local search algorithm for the QAP and establish the connection between the new algorithm and the Kernighan-Lin heuristic algorithm for the (GP).

The local search algorithm for the QAP starts with a random permutation as a current permutation. For a current permutation p_0, a sequence of permutations, $p_1, \ldots, p_l$, is constructed in a greedy sense. Each of the permutations in the sequence is obtained from the previous one by swapping (interchanging) two assignments and has cost lower than the current permutation. A local search is performed in the sequence, replacing the current permutation by the permutation with the lowest cost in the sequence (the algorithm stops if the sequence is empty for the current permutation). In the description of the local search algorithm below, instead of using the cost $C(p_k)$ of a permutation p_k in the sequence of permutations corresponding to a current permutation p_0, we use the cumulative gain $G(k)$ of the permutation p_k, where $G(k) = C(p_0) - C(p_k)$. Hence, the larger the cumulative gain of a permutation, the lower the cost is.

Algorithm 1: A Local Search Algorithm for the QAP
Input: n, $n \times n$ matrices F, D, and a permutation p of size n.
Output: A local optimal permutation p for the QAP.

(i) Set $p_0 = p$ and calculate its cost $C(p_0)$. Set $i = 0, g_i = 0$, and $G(i) = 0$, where g_i and $G(i)$ are the step gain and the cumulative gain, respectively.
(ii) $i = 1$. Initially, select a pair of facilities such that, by exchanging their locations, a positive step gain is obtained, i.e., $g_1 = C(p_0) - C(p_1) > 0$. If no such pair exists then go to 7, otherwise set $G(1) = g_1$.
(iii) $i = i+1$. For each pair of facilities not already selected, evaluate the step gain by exchanging their locations. Then, select the pair with maximum gain $g_i = C(p_{i-1}) - C(p_i)$. If all facilities have been selected then set $i = i - 1$ and go to 5.
(iv) Compute the cumulative gain, $G(i) = \sum_{k=1}^{k=i} g_k$. If $G(i) > 0$; then go to 3.
(v) Select k, such that $G(k)$ is maximum for $0 \le k \le i$.
(vi) If $k > 0$ then set $p_0 = p_k$ and go to 2.
(vii) We have reached a local optimum for the QAP. Set $p = p_0$ and output p and $C(p)$.

Now let us review the (KL) heuristic algorithm for the GP for an undirected graph $G(V, E)$ (assuming $|V| = 2n$) with edge weights $w(e)$, $e \in E$. For convenience, a partition of the set V always means a partition into two sets (A, B) with $|A| = |B| = |V|/2$ in the rest of the paper. Then, the problem GP is to find a partition (A, B) of the set V with the minimum cost $C(A, B)$, which is defined to be the sum of the weights of all edges between A and B. As one of the most successful heuristic algorithms for the GP, the Kernighan-Lin heuristic starts with a random partition of the set V. A sequence of partitions, $(A_1, B_1), \ldots, (A_l, B_l)$, is constructed for a current partition (A_0, B_0) in a greedy sense. Each partition $(A_k, B_k), 1 \le k \le l$, in the sequence is obtained from the previous one (A_{k-1}, B_{k-1}) by swapping one vertex in A_{k-1} with one vertex in B_{k-1} and has cost lower than the current partition. A local search is performed in the set of partitions of this sequence, replacing the current partition by the partition with the lowest cost in the sequence (the algorithm stops if the sequence is empty for the current partition). Similar to the description of Algorithm 1, we use the cumulative gain $G(k)$ for a partition p_k.

Algorithm 2: Kernighan-Lin heuristic for the GP
Input: $n, G = (V, E)$ with $|V| = 2n$, $W = (w_{ij})$, and a partition (A, B) of V.
Output: A locally optimal partition (A, B) of V

(i) Set $A_0 = A$ and $B_0 = B$, obtain its cost $C(A_0, B_0)$. Set $i = 0, g_i = 0$, and $G(i) = 0$, where g_i and $G(i)$ are step gain and cumulative gain, respectively.
(ii) i = 1. Initially, select a pair of vertices $a_1 \in A_0$ and $b_1 \in B_0$ such that, by swapping them, the resulting partition (A_1, B_1) produces a positive step gain g_1, i.e., $g_1 = C(A_0, B_0) - C(A_1, B_1) > 0$. If such a pair does not exist then go to 7, otherwise set $G(1) = g_1$.
(iii) $i = i+1$. Among the vertices not selected so far, choose a pair $a_i \in A_{i-1}$ and $b_i \in B_{i-1}$ and swap them to obtain A_i and B_i with maximum step gain $g_i = C(A_{i-1}, B_{i-1}) - C(A_i, B_i)$. If all the vertices have been selected then set $i = i - 1$ and go to 5.
(iv) Compute the cumulative gain $G(i) = \sum_{k=1}^{k=i} g_k$. If $G(i) > 0$; then go to 3.
(v) Choose k, such that $G(k)$ is maximum for $0 \leq k \leq i$.
(vi) If $k > 0$ then set $A_0 = A_k$ and $B_0 = B_k$ and go to 2.
(vii) We have reached a local optimum for the GP; set $A = A_0$ and $B = B_0$. Output A, B and $C(A, B)$.

Comparing the above algorithm with the local search algorithm for the QAP, one can easily see the similarity between them. Instead of working with partitions in the GP, we work with permutations in the QAP. The reduction from the GP to the QAP in the next section reveals why the adaptation of (KL) algorithm to the QAP can be effective. Furthermore, extensive computational results in section 3 indicate that the proposed local search algorithm (Algorithm 1) performs very well.

4.3. PLS-completeness and the QAP. For many combinatorial optimization problems, local search gives rise to some of the most successful heuristics. A classical example in this regard is the Linear Programming Problem for which the Simplex method can be viewed as a local search algorithm, in which a local search step is to go from the current basis to an adjacent basis which differs from the current one by one column vector. Based on the pivoting rule, worst-case examples can be constructed that force the Simplex method to take exponential time. Whether there can be a pivoting rule under which the Simplex method takes only polynomial time is a major open question.

In order to characterize the complexity of such local search algorithms, a new complexity class, the Polynomial-time Local Search class, was introduced and studied in [**126**]. A problem P is in PLS if, for each instance $x \in I$ (the set of all instances), we have a set of feasible solutions $F(x)$ such that it is easy to decide whether $s \in F(x)$ for any solution s. Then, given $x \in I$, we can produce a feasible solution $s \in F(x)$ in polynomial time. Next, given $x \in I$ and $s \in F(x)$ we can compute the cost $C(s,x)$ of s in polynomial time. In addition, every solution $s \in F(x)$ has a set of neighboring solutions $N(s,x)$. Finally, given $x \in I$ and $s \in F(x)$, we can test in polynomial time whether s is locally optimal, and if not, produce a solution belonging to $N(s,x)$ with a better cost value (A solution s is *locally optimal* if it does not have a strictly better neighbor).

More formally, a local search problem P in PLS is defined as follows: Given an input x, find a locally optimal solution $s \in F(x)$. For the problem P, the following three polynomial time algorithms should also exist.

(i) Algorithm A, on input $x \in I$, computes an initial feasible solution $s_0 \in F(x)$.
(ii) Algorithm B, on input $x \in I$ and $s \in F(x)$ computes $C(s,x)$.
(iii) Algorithm C, on input $x \in I$ and $s \in F(x)$, either determines that s is locally optimal or finds a better solution in $N(s,x)$.

A problem $P \in$ PLS is PLS-reducible to another problem $Q \in$ PLS, if there are polynomial time computable functions f and g, such that f maps an instance x of P to an instance $f(x)$ of Q and for any locally optimal solution s for $f(x)$, $g(s,x)$ produces a locally optimal solution for x. A problem P in PLS is PLS-complete, if every other problem in PLS is PLS-reducible to P.

An example of a PLS-complete problem is the GP with the (KL) neighborhood structure defined in Algorithm 2 [**126**]. The (KL) neighborhood of a partition for the GP can be defined as follows. A swap of a partition (A,B) is a partition $(A^{'},B^{'})$, where A and $A^{'}$ have a symmetric difference of 2, i.e., $(A^{'},B^{'})$ is obtained from (A,B) by swapping one element of A with one element of B. $(A^{'},B^{'})$ is a greedy swap if $C(A,B) - C(A^{'},B^{'})$ is maximized over all swaps of (A,B). If in fact $(A^{'},B^{'})$ is the lexicographically smallest over all greedy swaps, we say that $(A^{'},B^{'})$ is the lexicographic greed swap of (A,B). Let (A_i,B_i) be a sequence of partitions, each of which is a swap of the one preceding it, starting from (A_0,B_0). We call it monotonic, if the differences of $A_i - A_0$ and $B_i - B_0$ are monotonically increasing (that is, no vertex is switched back to its original set (A_0,B_0)). Finally, we say that a partition $(A^{'},B^{'})$ is a neighbor of (A,B) if it occurs in the unique maximal monotonic sequence of lexicographically greedy swaps starting with (A,B). Note that such a sequence will consist of $|V|/2+1$ partitions, with the last one equal to (B,A). Thus, each partition has $|V|/2$ neighbors. The algorithm performs local search over this neighborhood structure, replacing the current partition by the partition with the lowest cost in the neighborhood.

In the remaining part of this section, we show that the QAP with the neighborhood structure defined in Algorithm 1 is PLS-complete by reduction from the GP with the (KL) neighborhood structure. First, we show that the local search problem for the QAP is in PLS. Since the set of feasible solutions of the QAP is the set of permutations, an initial feasible solution can be produced in linear time. Computing the cost of a permutation for the QAP can be done in polynomial time. The neighborhood structure defined for the QAP in Algorithm 1 is quite similar to the (KL) neighborhood structure for the GP. For a given permutation for the QAP, there are $\lfloor n/2 \rfloor$ neighbors. We can determine in polynomial time if the permutation is locally optimal, and if not, produce a better permutation among the $\lfloor n/2 \rfloor$ neighboring permutations. Hence, with this neighborhood structure, finding a local optimum for the QAP is in PLS.

To prove PLS-completeness, we show that the GP is PLS-reducible to the QAP. Given an instance of the GP of size $2n$, we can create an instance of the QAP with the same size in polynomial time. Furthermore, for each local optimal permutation of the QAP, there is a natural local optimal partition for the corresponding GP. More specifically, suppose for the GP, the graph $G = (V, E)$ has edge weights $w(e)$ and vertex set V with $|V| = 2n$. We construct, in polynomial time, an instance of the QAP with $2n \times 2n$ matrices $F = (f_{ij})$ and $D = (d_{kl})$ defined below:

$$f_{ij} = w(i,j) \text{ if } \ (i,j) \in E; \text{ otherwise } \ f_{ij} = 0,$$

$$d_{kl} = 0 \text{ if } k,l \in \ A \text{ or } k,l \in \ B; \text{ otherwise } d_{lk} = 1,$$

$$\text{where } A = \{1, 2, \ldots, n\}, B = \{n+1, n+2, \ldots, 2n\}.$$

This reduction defines a one-one correspondence between a permutation p_k of the QAP with a partition (A_k, B_k) of the vertex set V of the corresponding GP. The set of facilities allocated to locations 1 to n in p_k constitutes the set A_k. The set of facilities allocated to locations $n+1$ to $2n$ in p_k constitutes the set B_k. The cost of p_k for the QAP is exactly twice the cost of the partition (A_k, B_k) for the GP. Let the partition corresponding to a permutation p_0 be (A_0, B_0), then a permutation p_k is a neighboring permutation of p_0 if, and only if, (A_k, B_k) is a neighboring partitions of (A_0, B_0). Hence, for any local optimal permutation of the QAP, the corresponding partition is a local optimal partition for the GP and can be recovered in polynomial time. By definition, the local search problem for the QAP with the neighborhood structure defined in Algorithm 1 is PLS-complete.

We should also mention that, at present, there are no known local criteria in deciding how good a local optimal solution is, in relation to the global optimum. From the complexity point of view, it can be shown that, if there exists a polynomial time algorithm for checking whether a given permutation is globally optimal, then P = NP [**177**].

4.4. Asymptotic Behavior. A nice feature of the QAP is that the relative difference between the worst and optimal solutions becomes arbitrarily small with a probability tending to 1 as the problem size tends to infinity. Burkard and Finke discovered this behavior for the QAP in the plane, i.e., the distance matrix B corresponds to Euclidean distances in the plane. Later they showed that this behavior holds also for the QAP in general in 1985. The result can be stated in the following theorem.

THEOREM 4.1. *For $i, j, k, l \in \{1, \cdots n\}$, let c_{ijkl} be identically distributed random variables in $[0,1]$ with expected value E and variance $\sigma^2 > 0$. For every fixed permutation $p \in \Pi$, let $c_{ip(i)jp(j)}$ be independently distributed. For given $\epsilon > 0$ and $0 < \epsilon_0 \leq \sigma^2$ and $0 < (E+\epsilon_0)/(E-\epsilon_0) \leq 1+\epsilon$,*

$$P(\frac{F_C^+}{F_C^-} < 1+\epsilon) \geq 1 - 2n!e^{-\lambda_0 n^2},$$

where $\lambda_0 = 2((\epsilon_0\sigma)/(\epsilon_0 + 2\sigma^2))^2, \lim_{n\to\infty} n!e^{-\lambda_0 n^2} = 0$, and F_C^+ and F_C^- are the maximum and minimum objective function values for the QAP with cost matrix C.

Several other researchers, including Frenk, van Houweninge, and Rinnooy Kan [**78**], and Rhee [**211, 212**] improved the order of convergence and showed that the convergence holds almost everywhere.

5. Methods of Solution

5.1. Exact Algorithms. In this section, we describe different methods used to find an optimal solution of QAP. The methods include dynamic programming, cutting plane, and branch-and-bound techniques.

Among these methods, branch-and-bound is the most successful one, on which this section is focused the most. Currently, problems of size greater than 15 are generally difficult to solve. This is due to the inherent difficulty of the QAP, characterized by the lack of sharp lower bounding techniques for moderate and large size problems. For this reason, a separate section is devoted to lower bounding techniques for the QAP.

Cutting plane methods for the QAP were introduced by Bazaraa and Sherali [**19**]. Although the computational experience was not satisfactory, such methods can be used to find good suboptimal solutions, see e.g., Burkard and Bönniger [**34**]. In Bazaraa and Sherali [**20**], cutting plane procedures were investigated for solving the concave quadratic minimization formulation of the QAP. Several heuristics derived from the cutting plane procedures produce good quality solutions in early stage of the search procedure.

Christofides and Benavent [**52**] used a special dynamic programming approach for the special case of the QAP in which the flow matrix is the weighted adjacency matrix of a tree. Problems of sizes up to 30 were solved.

Branch-and-bound is a general technique for solving combinatorial optimization problems. To solve the QAP with branch-and-bound, currently there are 3 types of algorithms: single assignment algorithms, pair assignment algorithms, and the relative positioning algorithm. They all start with the empty permutation as the initial partial permutation; during the execution of the algorithms, the partial permutation is extended to a full permutation. Single assignment algorithms date back to Gilmore [**89**] and were extended to the general QAP by Lawler [**143**]. In this approach, as described in the work by Gilmore and Lawler, a facility is assigned to a location at each node of the branch-and-bound search tree. Some of the earliest branch and bound algorithms for solving QAPs are described in [**36**], [**66**], [**180**] and [**218**].

Pair assignment algorithms were developed by Gavett and Plyter [**86**], Land [**141**], and Nugent et al. [**171**], etc. At each node of the branch-and-bound search tree, a fixed pair of facilities is allocated to a pair of locations. The last algorithm, the relative positioning algorithm, was developed by Mirchandani and Obata [**160**]. In their approach, the levels of the branch-and-bound search tree do not correspond to the assignments of facilities to locations. The partial permutations at each level are determined in terms of distances between facilities, i.e., their relative positions.

Numerical experiences indicate that among the 3 types of branch-and-bound algorithms the single assignment algorithms are the best. The pair assignment algorithms were shown to be not computationally efficient. The authors of the relative positioning algorithm claimed favorable behavior of the algorithm for problems with sparse matrices.

5.2. Lower Bounds. Lower bounds are keys to the success of a branch-and-bound type algorithm in combinatorial optimization. The ideal lower bounds should be sharp and should be fast to compute. For the QAP, there are roughly 3 categories of lower bounds. The first category includes the classical Gilmore-Lawler bound (GLB) and related bounds [**89, 143**]. The second category includes the eigenvalue based bounds [**74, 102, 101, 210, 103**]. The rest of the bounds are mostly based on reformulations of the QAP and generally involves solving a number of linear assignment problems [**5, 47, 54, 82**]. It is generally acknowledged that the eigenvalue based bounds are the best but also the most expensive to compute. In the following, a brief discussion of the 3 categories of lower bounds is given.

5.2.1. *Gilmore-Lawler Bound (GLB) and Related Bounds.* The GLB is computed by using the *minimal vector product* and the *maximal vector product*, denoted $\langle x, y\rangle_-$ and $\langle x, y\rangle_+$, defined below

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

where the set Π denotes the set of all permutations of N and $x, y \in R^n$. In fact, $\langle x, y\rangle_-$ can be computed as the inner product of x^+ and y^-, where x^+ is obtained by ordering the components of x ascendingly and y^- is obtained by ordering the components of y descendingly. $\langle x, y\rangle_+$ can be computed similarly.

Let $a_i, b_i,\ i = 1, ..., n$, 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

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

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

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

5.2.2. *Eigenvalue Based Bounds.* 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. [**74**], Hadley et al. [**102, 101**], and Rendl and Wolkowicz [**210**]. These bounds, denoted by EVB, are based on the trace formulation of the QAP, see (2.1.2).

A lower bound for the quadratic part of QAP, based on eigenvalues of A and B, is given by relaxing the constraint set of permutation matrices to $\mathcal{O}$. This results in the following theorem (see [**70, 74**]).

THEOREM 5.1. *Let A and B be symmetric matrices, and $\lambda_1 \le \lambda_2 ... \le \lambda_n$ be the eigenvalues of A, and $\mu_1 \le \mu_2 ... \le \mu_n$ be the eigenvalues of B. For any $p \in \Pi$,*

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

The linear part of QAP is bounded exactly by solving a *linear sum assignment problem*, denoted LSAP(C). We get the following bound for QAP

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

Certain reductions of the original matrices have to be performed before using the eigenvalues to obtain lower bounds for the QAP. These involve diagonal, or constant row and column, perturbations of the matrices. Several such lower bounds, EVB1, EVB2, EVB3, and IVB, were developed in [**74, 101**].

The strengthened relaxation of the constraint set of permutation matrices to $\mathcal{O} \cap \mathcal{E}$ was done in [**101**]. This relaxation proved to be particularly efficient and made the constant row and column reductions redundant. Rendl and Wolkowicz [**210**] recently proposed a new lower bound (MEVB) based on eigenvalue decomposition in conjunction with a steepest ascent algorithm. The bound is obtained iteratively, with each iteration taking $O(n^3)$ running time. This latter technique attempted to combine the bounds for the linear and quadratic parts.

(A further attempt to avoid taking the sum of two minima obtained by treating the quadratic and linear parts separately is given in the paper by Karisch, Rendl, and Wolkowicz in these proceedings.)

5.2.3. *Reformulation Based Bounds.* There are also other types of lower bounds for the QAP. Assad and Xu [**5**] proposed a bound (AX) for a class of quadratic 0-1 programs, including the QAP. The bound is obtained iteratively. In each iteration, $n^2 + 1$ assignment problems of size n are solved. Hence the running time to compute the bound is $O(kn^5)$ where k is the number of iteration. Christofides and Gerrard [**54**] proposed a lower bound (CG) by solving $O(n^4)$ linear assignment problems corresponding to pairs of assignments, resulting in a $O(n^7)$ procedure. Frieze and Yadegar [**82**] obtained 2 lower bounds by solving the Lagrangian relaxation of a related linear integer formulation of the QAP. The bounds are denoted by FY1 and FY2 respectively. Finally, Carraresi and Malucelli [**47**] 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 with 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.

5.2.4. *A New Class of Lower Bounds.* A class of lower bounds based on optimal reduction schemes for the QAP was proposed in Li, Pardalos, Ramakrishnan and Resende [**149**]. 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 a 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), $i,j = 1,...,n$, consider the following minimization problem

$$(5.1) \quad \min \quad \sum_{k=1}^{n} a_{ik}^{(1)} b_{jp(k)}^{(1)} + \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j} + \sum_{k=1}^{n} a_{ki} b_{p(k)j}^{(2)} - \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j}^{(2)}$$
$$\text{where} \quad p \in \Pi \text{ and } p(i) = j.$$

Define a $n \times n$ matrix $L = (l_{ij})$ where l_{ij} is the optimal objective function value of (5.1). The following theorem defines a new lower bound [**149**, Theorem 4.1].

THEOREM 5.2. *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.*

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.

Let $M = (m_{ij})$ be a matrix in $R^{n \times n}$. We treat a row vector m_i, $1 \leq i \leq n$, of M as a $1 \times n$ matrix and a column vector m_j^t, $1 \leq j \leq n$ as a $n \times 1$ matrix.

For convenience of discussion, we use the following notation for average $\gamma(M)$, variance $V(M)$, and total variance $T(M,\lambda)$ of M:

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

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

Note that the statistical total variance used here is the convex combination of two variances, (a) average row variance, (b) variance of the entire matrix, i.e.

$$\begin{aligned} \overline{T} &= \frac{\lambda}{n}\sum_{i=1}^{n} V(m_i) + (1-\lambda)V(M) \\ &= \frac{\lambda}{n}\sum_{i=1}^{n}(\gamma(m_i) - m_{ij})^2 + \frac{\lambda}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma(M) - m_{ij})^2 \\ &= \frac{1}{n^2}T(M,\lambda). \end{aligned}$$

In our reduction scheme, we considered the partition $A = A_1 + A_2$, where $A_1 = A + \Delta$ and $A_2 = -\Delta$, 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. This minimization problem has been formulated as follows:

$$\begin{array}{ll} \min & \theta T(A+\Delta,\lambda) + (1-\theta)T(-\Delta^t,\lambda) \\ \text{such that} & \Delta \in R^{n\times n} \end{array}$$

where $0 \le \theta \le 1$.

Motivated by the observation that for QAP(A,B) the smaller the variances of A and B are the tighter the GLB is, the following reduction schemes were proposed in Li, Pardalos, Ramakrishnan and Resende [**149**]:

$\mathcal{R}$-1) $a_{ij}^{(1)} = a_{ij} - \theta(a_{nn} - a_{ij})$ and $a_{ij}^{(2)} = \theta(a_{nn} - a_{ij}), i,j = 1,...,n.$

$\mathcal{R}$-2) $a_{ij}^{(1)} = a_{ij} - \theta(\gamma(a_n^t) - \gamma(a_j^t))$ and $a_{ij}^{(2)} = \theta(\gamma(a_n^t) - \gamma(a_j^t)),\ i,j = 1,...,n.$

One new lower bound proposed in [**149**] is to use the reduction scheme $\mathcal{R}$-1. We denote this lower bound by LB1(θ). The other new lower bound that we propose is to use the reduction scheme $\mathcal{R}$-2. This lower bound is denoted LB2(θ). Both new lower bounds are dependent on the parameter θ. 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 $l_{ij}, i,j =$

$1, ..., n$, in $O(n^3)$ [**149**]. Hence the total running time is $O(n^3)$, which is the same as that for computing GLB. Furthermore, the constant factor is small.

Recently, Jansen [**122**] has derived an analytical closed form solution to (5.2–5.2). That solution is given by

$$\begin{aligned} \delta_{ij} &= \theta\lambda\frac{1-\theta}{1-\theta\lambda}\gamma(a_i) + \frac{\theta(1-\lambda)+\theta\lambda^2(1-\theta)-\theta^2\lambda^2(1-\theta)}{(1-\theta\lambda)(1-\lambda+\theta\lambda)}\gamma(A) \\ &\quad -\frac{\lambda\theta(1-\theta)}{1-\lambda+\theta\lambda}\gamma(a_j^t) - \theta a_{ij}. \end{aligned}$$

Note that, as $\theta \to 1$, this partitioning scheme approaches the constant column reduction partitioning. Experimentally, we have observed that the column reduction partitioning scheme is more effective and is easier to implement. For an efficient implementation of the lower bound and computational results of a branch and bound algorithm that uses the new bound see [**148**].

5.3. Suboptimal Algorithms. As exact algorithms can only solve small size instances of the QAP, and finding an ϵ-approximate solutions of the QAP remains NP-complete, heuristics with good performance in both solution time and solution quality are highly desirable. Research in this direction abounds in literature. Basically, there are 5 types of heuristics including construction methods which start from an empty permutation p and expand p to a suboptimal permutation according to certain criteria, limited enumeration methods which perform partial enumeration with the expectation that good solutions are generally found in early stages of enumeration, improvement methods which start from a permutation p and try to get an improved permutation by using some techniques, simulation approaches and genetic algorithms which belong to stochastic search techniques.

5.3.1. *Construction Methods.* Construction methods, as the name suggests, construct a suboptimal permutation step by step. The basic idea was first introduced by Gilmore [**89**]. The general scheme for a construction method is as follows. It starts with a partial permutation p which is empty. Then p is expanded by repetitively selecting a pair of assignment (i, j) such that $i \notin M$ and $j \notin p(M)$ according to certain heuristic, where M is the index set containing the indices of p for which the corresponding assignments are done and $p(M)$ is the set $\{p(i) \mid i \in M\}$. This process is repeated until p becomes a complete permutation.

One of the oldest heuristics used is the CRAFT (Computerized Relative Allocation of Facilities Technique) [**4, 231, 30**]. This is a well-known heuristic for designing the layout of facilities that has been in use for over 25 years. Given a set of departments, locations, a matrix of flows between departments, and a matrix of costs to transport one item between two departments a unit distance, CRAFT iteratively improves an initial, user-supplied, layout by a series of department exchanges. At each step CRAFT considers either all possible 2-

way, 3-way, or both 2-way and 3-way exchanges. It chooses the exchange that provides the most improvement in minimizing total cost, and then repeats the process until no improving exchange can be found.

5.3.2. *Limited Enumeration Methods.* The argument for such type of methods is that a good suboptimal solution may be produced early in an enumerative search procedure, while finding the optimal solution takes much longer time. For example, it was observed in an experiment that one of the Nugent test problems, of size n equal to 15, the optimal solution was found after 23.48 seconds on a CDC Cyber 76, yet it took more than an hour to prove the optimality of the solution.

There are several ways to limit enumeration of the set of all possible permutations. One simple way is to put a time limit on the search procedure. Enumeration stops when a predetermined time limit is reached or there is no improvement within a certain time limit. Another way is to decrease the requirements for optimality. For example, whenever an improvement is not obtained after a given time period, the upper bound is decreased by a certain specified percentage, resulting in deeper cuts in the enumeration tree. Although it is possible that the optimal solution may be cut off, the enumeration process is speeded up. Furthermore, one can estimate that the optimal value differs from the suboptimal one by not more than some specified percentage.

5.3.3. *Improvement Methods.* The majority of the heuristic solution methods for the QAP falls into the category of improvement methods. An improvement method starts with a solution and tries to move to a better solution while improving the solution quality. *Local search algorithms* and *tabu search* belong to such type of methods. There is rich literature on local search algorithms for suboptimal solutions for the QAP. Local search algorithms are iterative algorithms. At each iteration, a local search algorithm generates the next solution by finding a better solution in the neighborhood of the current solution. The algorithm terminates when there is no better solution in the neighborhood of the current solution.

Tabu search was introduced by Glover [**90, 91**] as a technique to overcome local optimality in combinatorial search. The underlying idea is to limit the search directions for each search step to obtain good quality solutions in an effective way. This approach has been applied successfully to a number of combinatorial optimization problems including the TSP. Adaptations of tabu search to the QAP have been studied by Skorin-Kapov [**238**] and Taillard [**242**]. The basic idea is as follows.

To improve a given initial permutation tabu search seeks, among the set of permutations obtained by a pair exchange of assignments, a permutation with the best heuristic evaluation. In the simplest case, such an evaluation dictates the choice of a permutation which give the best objective function value. Every choice of a neighboring permutation represents an exchange of a pair of facilities.

A *tabu list* with a specified size is maintained to forbid certain choices of neighboring permutations, i.e., to forbid certain exchanges between facilities. The tabu list is essentially a list of pair exchanges occurring in the search process. As certain valuable exchanges of facilities may be prohibited by the tabu list, an *aspiration function* is introduced to allow exchanges prohibited by the tabu list to be chosen if they are judged to be valuable.

5.3.4. *Simulation Approaches.* An interesting analogy between problems in combinatorial optimization and statistical mechanics has been developed and has proven useful in solving some traditional optimization problems, such as computer design, partitioning, component placement, wiring, and the traveling salesman problem. The analogy has resulted in a methodology, termed *simulated annealing*, which is used to overcome local optimality (see Kirkpatrick, Gelatti, and Vecchi [**135**]).

The term "annealing" refers to the process of a thermal system by first melting at high temperatures and then lowering the temperatures slowly based on an annealing schedule. The process is continued until the vicinity of the solidification temperature is reached, where the system is allowed to reach the "ground state" (the lowest energy state of the system). Simulated annealing is a Monte Carlo approach to simulate the behavior of this system to achieve thermal equilibrium at a given temperature in a given annealing schedule. This analogy has been applied in solving combinatorial optimization problems. According to the above authors:

Iterative improvement, commonly applied to such problems, is much like the microscopic rearrangement process modeled by statistical mechanics, with the cost function playing the role of energy. However, accepting only rearrangements that lower the cost function of the system is like extremely rapid quenching high temperatures to $T = 0$. So, it should not be surprising that resulting solutions are usually metastable. The Metropolis procedure from statistical mechanics provides a generalization of iterative improvement in which controlled uphill steps can also be incorporated in the search for a better solution.

Simulated Annealing is applied to the QAP [**253**] as follows:

- Given any feasible solution (a permutation of locations in relation to the facilities), randomly select two facilities, make a pair exchange and evaluate the consequent change (δf) in the total cost (f)
- Repeat the above step as long as $\delta f < 0$. Otherwise, select a random variable x from a uniform distribution $U(0,1)$. If $x < P(\delta f) = EXP(-\delta f/t_i)$ (where P represents the probability obtained from the exponential distribution (EXP)), then accept the pair exchange and repeat the process. Here t_i represents the annealing schedule temperature at stage i where $t_1 > t_2 > \ldots > t_r$ represents the annealing schedule. For example $t_i = 10 \times (0.9)^{(i-1)}$.
- The system remains at stage i until a predetermined number of pair

exchanges have been considered before going to the next stage.
- If all the temperatures in the annealing schedule have been used, i.e. if $i > r$, then stop.

In the search process for an optimal permutation for the QAP, the simulated annealing searches through the 2-change neighborhood of a present permutation. It uses Monte Carlo sampling to occasionally accept solutions which increase rather than decrease the objective function value. Such a choice of neighboring permutation is counter to the normal *steepest descent* strategy. However, it is argued in the analogy that by taking such controlled ascent steps, the optimization algorithm needs not get stuck on poor solutions.

Burkard and Rendl [**41**] applied simulated annealing to the QAP and reported favorable computational results. Wilhelm and Ward [**253**] further investigated the procedure. Essential to the success of the adaptation of simulated annealing to the QAP is the annealing schedule as discussed in their work [**41, 253**]. In the paper [**183**], computational results with four heuristics, the CRAFT, simulating annealing, tabu search, and a local search based on graph partitioning are reported.

5.3.5. *Genetic Algorithms.* Genetic algorithms, like simulated annealing, form another type of stochastic search technique. While simulated annealing is based on thermodynamic process, genetic algorithms are based on the mechanics of natural selection and natural adaptation. A genetic algorithm maintains a population consisting of a subset of individuals (solutions). Through means of biased selection and genetic operations, the algorithm replaces a population with a new population of individuals with better fitness values on the average.

Genetic algorithms have been developed by John Holland et al. at the University of Michigan in 1975. However, genetic algorithms did not have a major influence at that time. With the advent of parallel computers, there has been increasing interest in genetic algorithms since they are inherently parallel. A number of researchers have tried to apply genetic algorithms to solve combinatorial optimization problems, such as the the graph partitioning problem and the traveling salesman problem [**161**].

5.4. Greedy Randomized Adaptive Search Procedures (GRASP). GRASP is an iterative randomized sampling technique in which each iteration provides an approximate solution to the problem at hand. The incumbent solution over all GRASP iterations is kept as the final result. There are two phases within each GRASP iteration: the first constructs an initial solution via an adaptive randomized greedy function; the second applies a local search technique to the constructed solution in hope of finding an improvement. A comprehensive survey of GRASP can be found in [**73**].

In [**71**] GRASP has been applied to a quadratic assignment problem that models the positioning of intermodal highway trailers on railcars. The GRASP is incorporated within a branch and bound algorithm to compute optimal solutions. In Li, Pardalos et al (see the paper in this volume), the GRASP has been applied to solve the general QAP. The GRASP was tested on 88 instances of QAPs (most of which are from QAPLIB) and found the best known solution of almost all the instances, and improved on the best known solution in a few cases.

6. Test Problem Generation

In this section we consider the problem of generating QAPs with a known optimal permutation. The problems will be of the form: Given a set $N = \{1, 2, \ldots, n\}$ and two $(n \times n)$ matrices $F = (f_{ij})$ and $D = (d_{kl})$, find a permutation p of the set N that minimizes:

$$Q_{FD}(p) = \sum_i \sum_j f_{ij} d_{p(i)p(j)}.$$

Test problem generators are very important in computational optimization. The efficiency of an algorithm for a given problem is determined by several criteria including the accuracy of the solution, the speed of the algorithm, and the effectiveness of the algorithm with respect to different problem classes. However, for many difficult problems, existing theory cannot itself provide measurement for these criteria. Hence, empirical computational experimentation is necessary. Evaluation and test of an algorithm can be done by using test problems with a known optimal solution. Test problems also provide a standard platform on which different algorithms for the same problem can be compared. For general references on this subject, see e.g., Pardalos [**178, 179**], Pardalos and Rosen [**190**], and Floudas and Pardalos [**76**].

6.1. Palubetskis' Generator for QAPs with a Known Solution. Next, we discuss the generation of test problems for the quadratic assignment problem. One of the first methods for constructing test problems with a known optimal permutation was proposed by Palubetskis [**175**]. Assume that the distance matrix is taken from a grid graph.
Input: w, a value to initialize the F matrix, and $z < w$, to obtain random values between $[0, z]$.
Output: Matrices F and D and an optimal permutation p^*.

(i) Construct the matrix $D = (d_{ij})$, of which the elements are the distances between the knots of the two dimensional grid $r \times s$, where $rs = n$, using rectilinear distances. If (i, j) are neighboring knots, then $d_{ij} = 1$.

(ii) Set $F = (f_{ij})$ where $f_{ij} = w$ (an input parameter to the algorithm). Compute $g_{ij} = 2 - d_{ij}$.
(iii) While for any $i, j = 1, \ldots, n$ such that $g_{ij} \leq 0$.
(iv) Choose the pair l, m, such that $d_{lm} = \max\{d_{ij}\}$ where the max is taken over every (i, j) for which $g_{ij} \leq 0$. If no such pair exists, then go to 8.

(v) Randomly select a grid point k on one of the shortest ways from l to m, such that, $\mid d_{lk} - d_{mk} \mid \leq 1$. Then, choose randomly, $\Delta \in [0, z < w]$ where w and z are the input parameters to the algorithm.
(vi) Set $f_{lm} := \Delta, f_{lk} := f_{lk} + (w - \Delta), f_{mk} := f_{mk} + (w - \Delta)$ and $g_{lm} := g_{lk} := g_{mk} := 1$.
(vii) Endwhile.
(viii) Finally, generate the random permutation $p^* = p^*(i), i = 1, 2, \ldots, n$, where P^* will be the optimal permutation. Form the matrix $F = (c_{ij})$ in which $f_{ij} = f_{uv}$ where $i = p^*(u)$ and $j = p^*(v)$.
(ix) Output F, D and p^* and optimal cost $w(\sum\sum d_{ij})$.

Next, we provide the proof of correctness of the above algorithm [**165**]. Before applying Step #8, the identity permutation is an optimal permutation for the QAP with input data, the matrices F and D and the optimal cost, $C = w(\sum\sum d_{ij})$.

The proof is by induction on the number of iterations.

Base: To start with, let $F^{(0)} = (f_{ij} = w)$ be the flow matrix. Obviously, for this $F^{(0)}$ all permutations are optimal and the corresponding cost is C.

Hypothesis: At the end of the i-th iteration, for the flow matrix $F^{(i)}$, the identity permutation is optimal and the corresponding cost is C.

To prove: at the end of the $(i+1)$-th iteration, for the flow matrix $F^{(i+1)}$, the identity permutation is optimal and the corresponding cost is the same as at the end of the i-th iteration i.e., C.

During the $(i+1)$-th iteration, let $d_{lm} = \max\{d_{ij}\}$ over all i, j such that $g_{ij} \leq 0$. Also, let a grid point k be selected such that k is on the shortest path between l and m, i.e. $d_{lk} + d_{mk} = d_{lm}$ (Comment: k need not satisfy the condition $\mid d_{lk} - d_{mk} \mid \leq 1$ as stated in the algorithm. It is sufficient if $d_{lk} + d_{mk} = d_{lm}$). Then,

$$F^{(i+1)} = F^{(i)} + \hat{F}$$

where $\hat{F} = (\hat{f_{ij}})$ and

$$\hat{f_{ij}} = \begin{cases} w - \Delta & \text{if } i = l \text{ and } j = k, \\ w - \Delta & \text{if } i = m \text{ and } j = k, \\ \Delta - w & \text{if } i = l \text{ and } j = m, \\ 0 & \text{otherwise.} \end{cases}$$

It is obvious that

$$opt(F^{(i+1)}) \geq opt(F^{(i)}) + opt(\hat{F}).$$

From the hypothesis $opt(F^{(i)}) = C$ and it can be easily proved that $opt(\hat{F}) = 0$. Hence,

$$opt(F^{(i+1)}) \geq C.$$

But, again from the hypothesis, the identity permutation is optimal for $F^{(i)}$ and it can be checked easily to show that it is also optimal for $\hat{F}$. Hence, the identity permutation must be optimal for $F^{(i+1)}$ and the optimal cost must be the same as at the end of the i-th iteration, i.e. C.

Next, we provide two test problems constructed using the above generator.

Example 1: Here, $n = 10$ and the grid dimensions are: $r = 5$ and $s = 2$. Also, the input parameters to the generator are: $w = 9$ and $z = 3$. The generated matrices F and D are as follows:

$$F = \begin{bmatrix} 9 & 3 & 2 & 1 & 16 & 3 & 41 & 9 & 0 & 32 \\ 15 & 9 & 0 & 16 & 18 & 2 & 3 & 9 & 40 & 31 \\ 0 & 2 & 9 & 22 & 58 & 9 & 43 & 0 & 1 & 3 \\ 1 & 1 & 0 & 9 & 15 & 9 & 33 & 1 & 17 & 25 \\ 17 & 0 & 1 & 2 & 9 & 2 & 30 & 1 & 16 & 18 \\ 0 & 0 & 9 & 35 & 3 & 9 & 39 & 3 & 38 & 3 \\ 27 & 2 & 9 & 15 & 16 & 3 & 9 & 2 & 16 & 3 \\ 33 & 9 & 0 & 3 & 40 & 0 & 18 & 9 & 15 & 17 \\ 16 & 2 & 3 & 15 & 15 & 3 & 17 & 2 & 9 & 25 \\ 30 & 9 & 0 & 29 & 18 & 1 & 3 & 0 & 9 & 9 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 5 \\ 1 & 0 & 2 & 1 & 3 & 2 & 4 & 3 & 5 & 4 \\ 1 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 2 & 1 & 1 & 0 & 2 & 1 & 3 & 2 & 4 & 3 \\ 2 & 3 & 1 & 2 & 0 & 1 & 1 & 2 & 2 & 3 \\ 3 & 2 & 2 & 1 & 1 & 0 & 2 & 1 & 3 & 2 \\ 3 & 4 & 2 & 3 & 1 & 2 & 0 & 1 & 1 & 2 \\ 4 & 3 & 3 & 2 & 2 & 1 & 1 & 0 & 2 & 1 \\ 4 & 5 & 3 & 4 & 2 & 3 & 1 & 2 & 0 & 1 \\ 5 & 4 & 4 & 3 & 3 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}$$

For this problem, the known optimal cost is 1890 and the optimal permutation of the facilities in relation to the locations is (8 2 1 10 5 9 7 4 3 6). The results obtained by our algorithm are: cost due to our algorithm is 1890, which is same as the optimal value and the corresponding permutation of the facilities is (7 10 1 4 5 2 3 9 6 8).

Example 2: The grid for this example is $r = 2$, $s = 5$ and $n = 10$. The input parameters to the generator are: $w = 9$ and $z = 5$. The generated flow matrix F is given below.

$$F = \begin{bmatrix} 9 & 21 & 29 & 17 & 1 & 17 & 20 & 16 & 2 & 1 \\ 9 & 9 & 18 & 34 & 2 & 2 & 34 & 14 & 0 & 4 \\ 5 & 13 & 9 & 15 & 3 & 2 & 16 & 24 & 18 & 2 \\ 5 & 26 & 20 & 9 & 9 & 4 & 2 & 2 & 27 & 3 \\ 1 & 0 & 47 & 22 & 9 & 0 & 5 & 2 & 31 & 26 \\ 9 & 19 & 5 & 4 & 0 & 9 & 27 & 26 & 27 & 1 \\ 1 & 24 & 22 & 2 & 3 & 9 & 9 & 9 & 24 & 5 \\ 1 & 0 & 16 & 4 & 1 & 17 & 18 & 9 & 21 & 17 \\ 5 & 0 & 5 & 18 & 0 & 5 & 33 & 13 & 9 & 18 \\ 2 & 0 & 0 & 26 & 9 & 4 & 2 & 41 & 35 & 9 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 1 & 4 & 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 & 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 & 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 & 4 & 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 2 & 3 & 2 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 1 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

Optimal cost for this example is also 1890. Our algorithm cost is the same as the optimal cost and the corresponding permutation in both the cases is the identity permutation.

6.2. Li & Pardalos Generator for QAPs with a Known Solution. Li and Pardalos [**150**] have generalized the results of Palubetskis and constructed test problems for more general types of QAPs. Their generator includes Palubetskis' procedure [**175**] as a special case, in which the distance matrix is taken from a grid graph. The fortran generator described in [**150**] is available by e-mail from the authors (the fortran code can also be obtained by sending an e-mail message to "coap@math.ufl.edu", and in the body of the message put "send 92006").

6.3. QAP-LIB. Finally we point out that a collection of more than 130 instances of quadratic assignment problems is contained in a library called the "QAP-LIB", [**39**]. This library consists of two parts. The data part contains various instances given by the input matrices A, B and C, if $C \neq 0$. Then there is a documentation, corresponding to [**39**]. This documentation is updated regularly, the last update being from February 1994. The documentation contains the following information on each of the instances: the best known feasible solution value is given, along with some information on who found it and by which method. Secondly the best currently available lower bounds on the objective function is provided. The two parts are available via anonymous ftp from `ftp.tu-graz.ac.at` in the directory /pub/papers/qaplib.

6.4. OR-Library. QAP instances can be also obtained from the OR-Library (o.rlibrary@ic.ac.uk) - see the file qapinfo. For details see [**123**]. Information about test problems for quadratic assignment problems as well as other combinatorial problems can be obtained by sending email to o.rlibrary@ic.ac.uk with the email message being the file name for the problem areas you are interested in.

7. Concluding Remarks

In this paper we gave a survey regarding the most recent results and applications on QAP. In addition, an up-to-date bibliography is included which includes papers on QAP and realted problems as well as applications in diverse areas.

Appendix A. Notations

n	The size of an instance of the QAP
e	The column vector of ones
A	The flow matrix
B	The distance matrix
$QAP(A,B)$	An instance of the QAP with flow matrix A and distance matrix B
Π_m	The set of permutations
$f(p)$	The objective function value of $QAP(A,B)$ corresponding to p
$f'_{A,B}$	The maximum objective function value of $QAP(A,B)$
$f^*_{A,B}$	The minimum objective function value of $QAP(A,B)$
$GLB(A,B)$	A permutation yielding the maximum objective function value of $QAP(A,B)$
$EVB(A,B)$	The Eigenvalue bound for the $QAP(A,B)$
$\langle x,y\rangle$	The inner product of vectors x and y
$\langle x,y\rangle_+$	The maximum permuted inner product of vectors x and y
$\langle x,y\rangle_-$	The minimum permuted inner product of vectors x and y
x^+	The vector obtained by reordering the components of x ascendingly
x^-	The vector obtained by reordering the components of x descendingly
A^t	The transpose of the matrix A
$trace\, A$	The trace of the matrix A
$diag\,(A)$	The vector formed from the diagonal elements of A
$diag\,(v)$	The diagonal matrix formed from the vector v
Π	The set of permutation matrices
$\mathcal{P}$	The set of positive semidefinite symmetric matrices
$\mathcal{D}$	The set of doubly stochastic matrices
$\mathcal{O}$	The set of orthogonal matrices
$\mathcal{N}$	The set of nonnegative (elementwise) matrices
$\mathcal{E}$	The set of matrices with row and column sums 1
$\mathcal{S}$	The set of matrices with the sum of the squares of all elements equal to n

References

1. E. Aarts and J. Korst, *Simulated annealing and Boltzman machines: A stochastic approach to combinatorial optimization and neural computing*, John Wiley and Sons, 1989.
2. W.P. Adams and H.D. Sherali, *A tight linearization and an algorithm for zero-one quadratic programming problems*, Management Science **32** (1986), no. 10, 1274–1290.
3. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Some recent advances in network flows*, SIAM Review **33** (1991), no. 2, 175–219.
4. G.C. Armour and E.S. Buffa, *Heuristic algorithm and simulation approach to relative location of facilities*, Management Science **9** (1963), 294–309.
5. A.A. Assad and W. Xu, *On lower bounds for a class of quadratic* $\{0,1\}$ *programs*, Operations Research Letters **4** (1985), 175–180.
6. J.G. Augustin and J. Minker, *Analysis of some graph theoretical cluster techniques*, J. ACM **17** (1970), 571–588.
7. G. Avondo-Bodeno, *Economic applications of the theory of graphs*, Gordon and Breach Science Publishers, 1962.
8. E. Balas and J.B. Mazzola, *Quadratic 0-1 programming by a new linearization*, Proceedings of the TIMS/ORSA (Washington D.c.), TIMS/ORSA, May 1980.

9. E. Balas and L. Qi, *Linear-time separation algorithms for the three-index assignment polytope*, Discrete Applied Mathematics (1993), 1–12.
10. E. Balas and M.J. Saltzman, *Facets of the three-index assignment polytope*, Discrete Applied Mathematics (1989), 201–229.
11. ______, *An algorithm for the three-index assignment problem*, Operations Research **39** (1991), 150–161.
12. E. Balas and C.S. Yu, *Finding a maximum clique in an arbitrary graph*, SIAM Journal of Computing **15** (1986), 1054–1068.
13. H.-J. Bandelt, Y. Crama, and F.C.R. Spieksma, *Approximation algorithms for multidimensional assignment problems with decomposable costs*, Discrete Applied Mathematics (1991), to appear.
14. J.F. Bard and T.A. Feo, *Operations sequencing in discrete parts manufacturing*, Management Science **35** (1989), 249–255.
15. J.F. Bard and T.A. Feo, *An algorithm for the manufacturing equipment selection problem*, IIE Transactions **23** (1991), 83–92.
16. M.S. Bazaraa and A.N. Elshafei, *An exact branch-and-bound procedure for the quadratic-assignment problem*, Naval Research Logistics Quarterly **26** (1979), 109–121.
17. M.S. Bazaraa and O. Kirca, *A branch-and-bound heuristic for solving the quadratic assignment problem*, Naval Research Logistics Quarterly **30** (1983), 287–304.
18. M.S. Bazaraa and H.D. Sherali, *New approaches for solving the quadratic assignment problem*, Operations Research Verfahren **32** (1979), 29–46.
19. ______, *Bender's partitioning scheme applied to a new formulation of the quadratic assignment problem*, Naval Research Logistics Quaterly **27** (1980), 29–41.
20. ______, *On the use of exact and heuristic cutting plane methods for the quadratic assignment problem*, Journal of Operations Research Society **33** (1982), 991–1003.
21. G. Birkoff, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucuman Rev. **Ser. A** (1946), 147–151.
22. J. B. Blanks, *Near optimal quadratic-based placement for a class of ic layout problems*, IEEE Circuits and Devices (1985), 31–373.
23. S.H. Bokhari, *On the mapping problem*, IEEE Transactions on Computers **C-30** (1981), no. 3, 207–214.
24. ______, *Assignment problems in parallel and distributed computing*, Kluwer Academic Publishers, Boston, 1987.
25. V. Joseph Bowman and Fred Glover, *A note on zero-one integer and concave programming*, Operations Research **21** (1973), 182–183.
26. S.C. Boyd, *The subtour polytope of the travelling salesman problem*, Ph.D. thesis, University of Waterloo, 1986.
27. E. Donald Brown, L. Christopher Huntley, and R. Andrew Spillance, *A parallel genetic heuristic for the quadratic assignment problem*, Proc. 3rd Conf. on Genetic Algorithms, 406-415, 1989.
28. R.A. Brualdi and H.J. Ryser, *Combinatorial matrix theory*, Cambridge University Press, New York, 1991.
29. P.A. Bruijs, *On the quality of heuristic solutions to a 19 x 19 quadratic assignment problem*, European Journal of Operational Research **17** (1984), 21–30.
30. E.S. Buffa, G.C. Armour, and T.E. Vollmann, *Allocating facilities with CRAFT*, Harvard Business Review **42** (1962), 136–158.
31. R.E. Burkard, *Die störungsmethode zur lösung quadratischer zuordnungsprobleme*, Operations Research Verfahren **16** (1973), 84–108.
32. ______, *Some recent advances in quadratic assignment problems*, Mathematical Programming, Elsevier Publishers B.V. North-Holland, 1984, pp. 53–68.
33. R.E. Burkard, *Locations with spatial interactions: the quadratic assignment problem*, Discrete Location Theory (P.B. Mirchandani and R.L. Francis, eds.), John Wiley, 1991.
34. R.E. Burkard and T. Bonniger, *A heuristic for quadratic boolean programs with applications to quadratic assignment problems*, European Journal of Operational Research **13** (1983), 374–386.

35. R.E. Burkard, E. Cela, and B. Klinz, *On the biquadratic assignment problem*, Tech. Report 260, Institute of Mathematics, University of Technology Graz, 1993.
36. R.E. Burkard and U. Derigs, *Assignment and matching problems: Solution methods with fortran programs*, Lecture Notes in Economics and Mathematical Systems, vol. 184, Springer, Berlin, 1980.
37. R.E. Burkard and U. Fincke, *The asymptotic probabilistic behaviour of quadratic sum assignment problems*, Z. Operations Res. **27** (1983), 73–81.
38. R.E. Burkard and K. Fröhlich, *Some remarks on three-dimensional assignment problem*, Method of Oper. Res. **36** (1980), 31–36.
39. R.E. Burkard, S. Karisch, and F. Rendl, *QAPLIB – a quadratic assignment problem library*, European Journal of Operations Research **55** (1991), 115–119, Updated version – Feb. 1993.
40. R.E. Burkard and J. Offermann, *Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungsprobleme*, Z. Operations Res. **21** (1977), B121–B132.
41. R.E. Burkard and F. Rendl, *A thermodynamically motivated simulation procedure for combinatorial optimization problems*, European Journal of Operations Research **17** (1984), 169–174.
42. R.E. Burkard and R. Rudolf, *Computational investigations on 3-dimensional axial assignment problems*, Tech. report, Technische Universität Graz, Austria, 1993, to appear in: Belgian Journal of Operations Research.
43. R.E. Burkard, R. Rudolf, and G. Woeginer, *Three-dimensional axial assignment problems with decomposable cost-coefficients*, Tech. Report Report 238, Technische Universität Graz, Austria, 1992.
44. R.E. Burkard and W. Sandholzer, *Efficiently solvable special cases of bottleneck travelling salesman problems*, Discrete Applied Mathematics **32** (1991), 61–76.
45. R.E. Burkard and K.-H. Stratmann, *Numerical investigations on quadratic assignment problems*, Naval Research Logistics Quaterly **25** (1978), 129–148.
46. P. Camion, *Characterization of totally unimodular matrices*, Proc. Amer. Math. Soc. **16** (1965), 1068–73.
47. P. Carraresi and F. Malucelli, *A new lower bound for the quadratic assignment problem*, Operations Research **40** (1992), no. Supplement 1, S22–S27.
48. P. Carraresi and F. Malucelli, *A reformulation and new lower bounds for the quadratic assignment problem*, Tech. Report TR-34/92, Universita di Pisa, 1992.
49. J. Chakrapani and J. Skorin-Kapov, *A connectionist approaches to the quadratic assignment problem*, Computers and Operations Research **19** (1992), no. 3/4, 287–295.
50. ———, *A constructive method to improve lower bounds for the quadratic assignment problem*, Working paper, State University of New York at Stony Brook, 1992.
51. ———, *Massively parallel tabu search for the quadratic assignment problem*, Annals of Operations Research **forthcoming** (1992).
52. N. Christofides and E. Benavent, *An exact algorithm for the quadrtic assignment problem*, Operations Research **37** (1989), no. 5, 760–768.
53. N. Christofides and M. Gerrard, *Special cases of the quadratic assignment problem*, Management Science Research Report 391, Carnegie Mellon University, April 1976.
54. ———, *A graph theoretic analysis of bounds for the quadratic assignment problem*, Studies on graphs and discrete programming (P. Hansen, ed.), North-Holland, 1981, pp. 61–68.
55. N. Christofides, A. Mingozzi, and P. Toth, *Contributions to the quadratic assignment problem*, European Journal of Operations Research **4** (1980), 243–247.
56. A. Rogger C.N. Fiechter and D. de Werra, *Basic ideas of tabu search with an application to traveling salesman and quadratic assignment*, Ricerca Operativa **62** (1992), 5–28.
57. D.T. Connolly, *An improved annealing scheme fro the qap*, Journal of Operational Research **46** (1990), 93–100.
58. K. Conrad, *Das quadratische zuweisungsproblem und zwei seiner spezialfalle*, Mohr-Siebeck, Tubingen (1971).
59. Y. Crama and F.C.R. Spieksma, *Approximation algorithms for three-dimensional assignment problems with triangle inequalities*, European Journal of Operational Research **60**

(1992), 273–279.
60. P.J. Davis, *Circulant matrices*, John Wiley & Sons, New York, 1979.
61. N. Deo, *Graph theory with applications to engineering and computer science*, Prentice-Hall, 1974.
62. J.W. Dicky and J.W. Hopkins, *Campus building arrangement using TOPAZ*, Transportation Research **6** (1972), 59–68.
63. R.E. Echols and L. Cooper, *Solution of integer linear programming problems by direct search*, J. Assoc. Comput. Mach. **15** (1968), 75–84.
64. P.M. Pardalos (Editor), *Complexity in numerical optimization*, World Science, 1993.
65. C.S. Edwards, *The derivation of a greedy approximator for the koopmans-beckmann quadratic assignment problem*, Proc. CP77 Combinatorial Prog. Conf. (1977), 55–86.
66. ______, *A branch and bound algorithm for the Koopmans-Beckman quadratic assignment problem*, Mathematical Programming Study **13** (1980), 35–52.
67. A.N. Elshafei, *Hospital layout as a quadratic assignment problem*, Operations Research Quarterly **28** (1977), 167–179.
68. R. Euler, *Odd cycles and a class of facets of the axial 3-index assignment polytope*, Applicationes Mathematicae (Zastosowania Matematyki) **XIX** (1987), 375–386.
69. B.H. Faaland and F.S. Hillier, *Interior path methods for heuristic integer programming procedures*, Operations Research **27** (1979), 1069–1087.
70. K. Fan, *Maximum properties and inequalities for the eigenvalues of completely continuous operators*, Proc. Nat. Acad. Sci. U.S.A. **35** (1951), 1951.
71. T.A. Feo and J. Gonzalez-Velarde, *The intermodal trailer assignment problem*, Tech. report, Operations Research Group, The University of Texas at Austin, Austin, TX 78712-1063, 1992.
72. T.A. Feo and M.G.C. Resende, *A probabilistic heuristic for a computationally difficult set covering problem*, Operations Research Letters **8** (1989), 67–71.
73. T.A. Feo and M.G.C. Resende, *Greedy randomized adaptive search procedures*, Tech. report, AT&T Bell Laboratories, Murray Hill, NJ 07974-2070, 1994.
74. G. Finke, R.E. Burkard, and F. Rendl, *Quadratic assignment problems*, Annals of Discrete Mathematics **31** (1987), 61–82.
75. G. Finke and E.B. Medova-Dempster, *Approximation approach to combinatorial optimization problems*, Tech. report, Technical University of Nova Scotia, 1987.
76. C.A. Floudas and P.M. Pardalos, *A collection of test problems for constrained global optimization algorithms*, Springer-Verlag, Lecture Notes in Computer Science, No. 455, 1990.
77. R.L. Francis and J.A. White, *Facility layout and location*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
78. J.C.B. Frenk, M. van Houweninge, and A.H.G. Rinnooy Kan, *Asymptotic properties of assignment problems*, Tech. report, Erasmus University, Rotterdam, 1982.
79. C. Friden, A. Hertz, and D. de Werra, *STABULUS: A technique for finding stable sets in large graphs with tabu search*, Computing **42** (1989), 35–44.
80. A.M. Frieze, *Complexity of a 3-dimensional assignment problem*, European Journal of Operations Research **13** (1983), 161–164.
81. A.M. Frieze and J. Yadegar, *an algorithm for solving 3-dimensional assignment problem with application to scheduling a teaching practice*, Operations Research **32** (1981), 989–995.
82. ______, *On the quadratic assignment problem*, Discrete Applied Mathematics **5** (1983), 89–98.
83. K. Fröhlich, *Dreidimensionale zuordnungsprobleme*, Ph.D. thesis, Math. Institut, Universität Köln, 1979, Masters Thesis.
84. W. Gander, G. Golub, and U. von Matt, *A constrained eigenvalue problem*, Linear Algebra and its Applications **114/115** (1989), 815–839.
85. M.R. Garey and D.S. Johnson, *Computers and intractability - A guide to the theory of NP-completeness*, W.H. Freeman and Company, 1979.
86. J.W. Gavett and N.V. Plyter, *The optimal assignment of facilities to locations by branch*

and bound, Operations Research **14** (1966), 210–232.
87. A.M. Geoffrion, *An improved implicit enumeration approach for integer programming*, Operations Research **17** (1969), 437–454.
88. A.M. Geoffrion and G.W. Graves, *Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach*, Operations Research **24** (1976), 595–610.
89. P.C. Gilmore, *Optimal and suboptimal algorithms for the quadratic assignment problem*, J. SIAM **10** (1962), 305–313.
90. F. Glover, *Tabu search - Part 1*, ORSA Journal on Computing **1** (1989), no. 3, 190–206.
91. ______, *Tabu search - Part 2*, ORSA Journal on Computing **2** (1989), no. 1, 4–32.
92. Fred Glover and D. Klingman, *Concave programming applied to a special class of 0-1 integer programs*, Operations Research **21** (1973), 135–140.
93. E. David Goldburg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Publishing Company, Inc., 1989.
94. B. Gollan, *Eigenvalue perturbations and nonlinear parametric optimization*, Mathematical Programming Study **30** (1987), 67–81.
95. S. Goto and E.S. Kuh, *An approach to the two-dimensional placement problem in circuit layout*, IEEE Transactions on Circuits and Systems **25** (1978), no. 4, 208–214.
96. A. Graham, *Kronecker products and matrix calculus: with applications*, Halsted Press, Toronto, 1981.
97. G.W. Graves and A.B. Whinston, *An algorithm for the quadratic assignment problem*, Management Science **17** (1970), 453–471.
98. V.P. Gulati, S.K. Gupta, and A.K. Mittal, *Unconstrained bivalent programming problem*, European Journal of Operations Research **15** (1984), 121–125.
99. G. Gwan and L. Qi, *On facet of the three index assignment polytope*, Australasian Journal of Combinatorics **6** (1992), 67–87.
100. S.W. Hadley, *Continuous optimization approaches for the quadratic assignment problem*, Ph.D. thesis, University of Waterloo, 1989.
101. S.W. Hadley, F. Rendl, and H. Wolkowicz, *Bounds for the quadratic assignment problem using continuous optimization techniques*, Integer Programming and Combinatorial Optimization, University of Waterloo Press, 1990, pp. 237–248.
102. ______, *A new lower bound via projection for the quadratic assignment problem*, Mathematics of Operations Research **17** (1992), no. 3, 727–739.
103. S.W. Hadley, F. Rendl, and H. Wolkowicz, *Nonsymmetric quadratic assignment problems and the Hoffman-Wielandt inequality*, Linear Algebra and its Applications **58** (1992), 109–124.
104. M. Hall Jr., *Combinatorial theory*, Blaisdell Company, Waltham, MA, 1967.
105. M. Hanan and J.M. Kurtzberg, *A review of the placement and quadratic assignment problem*, SIAM Review **14** (1972), 324–342.
106. P. Hansen and L. Kaufman, *A primal-dual algorithm for the three dimensional assignment problem*, Cahiers Centre Étudess Rech. Opér **15** (1973), 327–336.
107. J.P. Hart and A.W. Shogan, *Semi-greedy heuristics: An empirical study*, Operations Research Letters **6** (1987), 107–114.
108. D.R. Heffley, *Assigning runners to a relay team*, Optimal Strategies in Sports (Amsterdam) (S.P. Ladany and R.E. Machol, eds.), North-Holland, Amsterdam, 1977, pp. 169–171.
109. C.H. Heider, *A computationally simplified pair exchange algorithm for the quadratic assignment problem*, Paper 101, Center for Naval Analysis, Arlington (Va), 1972.
110. C. Helmberg, B. Mohar, S. Poljak, and F. Rendl, *A spectral approach to bandwidth and separator problems in graphs*, Tech. Report CDLDO 32, Institute of Mathematics, University of Technology Graz, 1993.
111. P.S. Hiller and M.M. Connors, *Quadratic assignment algorithms and the location of indivisible facilities*, Management Science **13** (1966), 42–57.
112. F.S. Hillier, *Efficient heuristic procedures for integer linear programming with an interior*, Operations Research **17** (1969), 600–637.

113. ______, *A further investigation of efficient heuristic procedures for integer linear programming with an interior*, Tech. report, Dept. of Operations Research, Stanford University, Stanford, CA 94305, 1977.
114. J.B. Hiriart-Urruty, *From convex optimization to nonconvex optimization. part i: Necessary and sufficient conditions from global optimality*, Nonsmooth Optimization and Related Topics (F.H. Clarke, ed.), Plenum Press, 1989, pp. 219–239.
115. A.J. Hoffman and H.W. Wielandt, *The variation of the spectrum of a normal matrix*, Duke Mathematics **20** (1953), 37–39.
116. K.H. Hoffmann and P. Salamon, *Simulated annealing for single minimum optimization problems*, International J. Computer Math. **39** (1991), 193–204.
117. R. Horn and C. Johnson, *Matrix analysis*, Cambridge University Press, New York, 1985.
118. L.J. Hubert, *Assignment methods in combinatorial data analysis*, Marcel Dekker, Inc., New York, NY 10016, 1987.
119. T. Ibaraki, *Theoretical comparisons of search strategies in branch-and-bound algorithms*, International Journal of Computer and Information Sciences **5** (1976), no. 4, 315–344.
120. T. Ibaraki, T. Ohashi, and F. Mine, *A heuristic algorithm for mixed-integer programming problems*, Mathematical Programming Study **2** (1974), 115–136.
121. I.Gilar and M.A. Pollatschek, *Layout simulation for keyboard*, Behaviour and Information Technology **5** (1986), 273–281.
122. B. Jansen, *A note on "Lower bounds for the QAP"*, Tech. report, Delft University of Technology, Mathematics and Computer Science, December 1993.
123. J.E.Beasley, *Obtaining test problems using e-mail*, Journal of global optimization (1994), to appear.
124. Prashnna Jog, Jung Y. Suh, and Dirk Van Gucht, *Parallel genetic algorithms applied to the traveling salesman problem*, SIAM J. Optimization **1** (1991), no. 4, 515–529.
125. D.S. Johnson, C.A. Aragon, L.A. McGeoch, and C. Schevon, *Optimization by simulated annealing: An experimental evaluation; Part II, Graph coloring and number partitioning*, Operations Research **39** (1991), 378–406.
126. D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis, *How easy is local search?*, Journal of Computer and System Sciences **37** (1988), 79–100.
127. B. K. Kaku and G. L. Thompson, *An exact algorithm for the general quadratic assignment problem*, European Journal of Operational Research **23** (1986), 382–390.
128. B.K. Kaku, T.E. Morton, and G.L. Thompson, *A heuristic algorithm for the facilities layout problem*, Tech. report, Carnegie Mellon University, Pittsburgh, Pa., 1989.
129. S. E. Karisch, F. Rendl, and H. Wolkowicz, *Trust regions and the quadratic assignment problem*, Research Report CORR 93-15, DIMACS, Rutgers University, New Brunswick, NJ, 1993, To appear in the Proceedings of the Workshop on the Quadratic Programming Problem, DIMACS, 1993.
130. N. Karmarkar, *An interior-point approach to NP-complete problems – extended abstract*, Contemporary Mathematics **114** (1990), 297–308.
131. R. Karp, *Reducibility among combinatorial problems*, Proc. Complexity of Computer Computations (R.E. Miller and J.W. Thatcher, eds.), Plenum Press, 1972, pp. 85–104.
132. L. Kaufman and F. Broeckx, *An algorithm for the quadratic assignment problem using benders' decomposition*, European Journal of Operational Research **2** (1978), 204–211.
133. B. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, Bell Systems Journal **49** (1972), 291–307.
134. G.A.P. Kindervater and J.K. Lenstra, *An introduction to parallelism in combinatorial optimization*, Discrete Applied Mathematics **14** (1986), 135–156.
135. S. Kirpatrick, C.D. Gelatti, and M.P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), 671–680.
136. G.A. Kochenberger, B.A. McCarl, and F.P. Wyman, *A heuristic for general integer programming*, Decision Sci. **5** (1974), 36–44.
137. T.C. Koopmans and M.J. Beckmann, *Assignment problems and the location of economic activities*, Econometrica **25** (1957), 53–76.
138. J. Krarup and P.M. Pruzan, *Computer-aided layout design*, Mathematical Programming

Study **9** (1978), 75–94.
139. T.A. Lacksonen and Jr E.E. Enscore, *Quadratic assignment algorithms for the dynamic layout*, Int. J. Prod. Res. **31** (1993), 503–517.
140. M. Laguna and J.L. González-Velarde, *A search heuristic for just-in-time scheduling in parallel machines*, Journal of Intelligent Manufacturing **2** (1991), 253–260.
141. A. M. Land, *A problem of assignment with interrelated costs*, Operations Research Quarterly **14** (1963), 185–198.
142. R. S. Lashkari and S. C. Jaisingh, *An heuristic approach to quadratic assignment problem*, Journal of the Operational Research Society **31** (1980), no. 9, 845–850.
143. E.L. Lawler, *The quadratic assignment problem*, Management Science **9** (1963), 586–599.
144. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, *The traveling salesman problem: A guided tour of combinatorial optimization*, John Wiley & Sons, 1985.
145. O. Leue, *Methoden zur losung dreidimensionaler zuordnungsprobleme*, Angew. Inform. (1972), 154–162.
146. Tao Li, *Parallel imprecise iterative deepening for combinatorial optimization*, International J. of High Speed Computing **3** (1991), no. 1, 63–76.
147. Y. Li, *Heuristic and exact algorithms for the quadratic assignment problem*, Ph.D. thesis, The Pennsylvania State University, 1992.
148. Y. Li, P.M. Pardalos, K.G. Ramakrishnan, and M.G.C. Resende, *A branch-and-bound algorithm for the quadratic assignment problem*, Tech. report, AT&T Bell Laboratories, Murray Hill, NJ 07974-2070, December 1992.
149. ______, *Lower bounds for the quadratic assignment problem*, Tech. report, AT&T Bell Laboratories, Murray Hill, NJ 07974-2070, April 1992, To appear in *Annals of Operations Research*.
150. Yong Li and Panos M. Pardalos, *Generating quadratic assignment test problems with known optimal permutations*, Computational Optimization and Applications **1** (1992), no. 2, 163–184.
151. ______, Parallel algorithms for the quadratic assignment problem, in Recent Advances in Optimization and Parallel Computing, 177–189, Elsevier, Amsterdam, 1992, pp. 177–189.
152. S. Lin and B. Kernighan, *An effective heuristic algorithm for the travelling salesman problem*, Operations Research **21** (1973), 498–516.
153. L. Lovász, *On the ratio of optimal integral and fractional covers*, Discrete Mathematics **13** (1975), 391–398.
154. R. F. Love and J. Y. Wong, *On solving a one-dimensional space allocation problem with integer programming*, INFOR **14** (1976), 139–143.
155. ______, *Solving quadratic assignment problems with rectangular distances and integer programming*, Naval Research Logistics Quarterly **23** (1976), 623–627.
156. Robert F. Love, James G. Morris, and George O. Wesolowsky, *Facilities location: Models and methods*, North-Holland, 1988.
157. M. Grötschel and L. Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, 1988.
158. W.L. Maxwell, *The scheduling of economic lot sizes*, Naval Research Logistics Quaterly **11** (1964), 89–124.
159. E.J. McCormik, *Human factors engineering*, McGraw-Hill, New York, 1970.
160. P.B. Mirchandani and T. Obata, *Locational decisions with interactions between facilities: the quadratic assignment problem a review*, Working Paper Ps-79-1, Rensselaer Polytechnic Institute, Troy, New York, May 1979.
161. H. Muhlenbein, *Parallel genetic algorithms, population genetics and combinatorial optimization*, Proc. 3rd Conf. on Genetic Algorithms, 416-421, 1989.
162. ______, *Parallel genetic algorithms in optimization, to appear in informatik fachbenchle*, Springer-Verlag, 1992.
163. H. Muhlenbein, M. Gorges-Schleuter, and O. Kramer, *Evolution algorithms in combinatorial optimization*, Parallel Computing **7** (1988), 65–85.
164. H. Muller-Merbach, *Optimale reihenlorgen*, Springer, Berlin, 1970.
165. K.A. Murthy and P.M. Pardalos, *A polynomial-time approximation algorithm for the*

quadratic assignment problem, Tech. Report CS-33-90, The Pennsylvania State University, July 1990.

166. K.A. Murthy, P.M. Pardalos, and Y. Li, *A local search algorithm for the quadratic assignment problem*, Informatica **3** (1992), no. 4, 524–538.
167. L.D. Muu and W. Oettli, *Method for minimizing a convex-concave function over a convex set*, Journal of Optimization and Applications **70** (1991), no. 2, 377–384.
168. M.A. Pollatschek N. Gershoni and Y.T. Radday, *A better hebrew typewritter?*, Industrial Engineering **October** (1976), 35–37.
169. George L. Nemhauser and Laurence A. Wolsey, *Integer and combinatorial optimization*, John Wiley and Sons, 1988.
170. V. Nissen, *A new efficient evolutionary algorithm for the quadratic assignment problem*, Operations Research Proceedings (K.W. Hansmann, A. Bachem, M. Jarke, W.E. Katzenberg, and A. Marusev, eds.), Springer Berlin, 1992, pp. 259–267.
171. C.E. Nugent, T.E. Vollmann, and J. Ruml, *An experimental comparison of techniques for the assignment of facilities to locations*, Journal of Operations Research **16** (1969), 150–173.
172. M.L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optimization **2** (1992), 88–120.
173. M. Padberg, *On the facial structure of the set packing polyhedra*, Mathematical Programming **5** (1973), 199–216.
174. G. Palubetskis, *Quadratic 0-1 optimization*, Informatica **1** (1990), 89–106.
175. G.S. Palubetskis, *Generation of quadratic assignment test problems with known optimal solutions (in Russian)*, Zh. Vychisl. Mat. Mat. Fiz. **28** (1988), no. 11, 1740–1743.
176. C.H Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, USA, 1982.
177. C.H. Papadimitriou and D. Wolfe, *The complexity of facets resolved*, Proceedings of the Foundations Of Computer Science, 1985, pp. 74–78.
178. P.M. Pardalos, *Generation of large-scale quadratic programs for use as global optimization test problems*, ACM Transactions on Mathematical Software **13** (1987), no. 2, 133–137.
179. ______, *Construction of test problems in quadratic bivalent programming*, ACM Transactions on Mathematical Software **17** (1991), no. 1, 74–87.
180. P.M. Pardalos and J. Crouse, *A parallel algorithm for the quadratic assignment problem*, Proceedings of the Supercomputing 1989 Conference, ACM Press, 1989, pp. 351–360.
181. P.M. Pardalos and S. Jha, *Complexity of uniqueness and local search in quadratic 0-1 programming*, Operations Research Letters (1992), 119–123.
182. P.M. Pardalos and X. Li, *Parallel branch and bound algorithms for combinatorial optimization*, Supercomputer **39** (1990), 23–30.
183. P.M. Pardalos, K.A. Murthy, and T.P. Harrison, *A computational comparison of local search heuristics for solving quadratic assignment problems*, Informatica **4** (1993), no. 1-2, 172–187.
184. P.M. Pardalos, K.A. Murthy, and Y. Li, *Computational experience with parallel algorithms for solving the quadratic assignment problem*, to appear in Computer Science and Operations Research: New developments in their interfaces (Williamsburg, VA), ORSA CSTS, January 1992.
185. P.M. Pardalos, A. Phillips, and J.B. Rosen, *Topics in parallel computing in mathematical programming*, Science Press, 1993.
186. P.M. Pardalos and G.P. Rodgers, *Parallel branch and bound algorithms for unconstrained quadratic 0-1 programming*, Impact of Recent Advances on Operations Research ((R. Sharda et al.), ed.), North-Holland Press, 1989, pp. 131–143.
187. ______, *Computational aspects of a branch and bound algorithm for quadratic zero-one programming*, Computing **45** (1990), 131–144.
188. ______, *A branch and bound algorithm for the maximum clique problem*, Comp. and Oper. Research **19** (1992), no. 5, 363–375.
189. P.M. Pardalos and J.B. Rosen, *Methods for global concave minimization: A bibliographic survey*, Siam Review **28** (1986), 367–379.

190. ______, *Constrained global optimization: Algorithms and applications*, Lecture Notes in Computer Science, vol. 268, Springer-Verlag, Berlin, 1987.
191. P.M. Pardalos and G. Schnitger, *Checking local optimality in constrained quadratic programming is np-hard*, Operations Research Letters **7** (1988), 33–35.
192. P.M. Pardalos and S. Vavasis, *Quadratic programming with one negative eigenvalue is np-hard*, Journal of Global Optimization **1** (1991), 15–23.
193. P.M. Pardalos and Jue Xue, *The maximum clique problem*, Tech. report, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 1992.
194. C.C. Pegels, *Plant layout and discrete optimizing*, International Journal of Production Research **5** (1966), 81–92.
195. J.F. Pierce and W.B. Crowston, *Tree search algorithms for quadratic assignment problems*, Naval Research Logistics Quaterly **18** (1971), 1–36.
196. W.P. Pierskalla, *The tri-substitution method for the three-multidimensional assignment problem*, CORS J. **5** (1967), 71–81.
197. ______, *The multidimensional assignment problem*, Operations Research **16** (1968), 422–431.
198. S. Poljak, F. Rendl, and H. Wolkowicz, *General relaxations and concave extensions for (0,1)-quadratic programming*, Tech. report, University of Waterloo, 1994, in progress.
199. M.A. Pollatschek, N. Gershoni, and Y.T. Radday, *Optimization of the typewritter keyboard by computer simulation*, Angewandte Informatik **10** (1976), 438–439.
200. W.R. Pulleyblank, Mathematical Programming The State of the Art, 312–345, Springer-Verlag, 1982, pp. 312–345.
201. A.K. Dewdney P.Z. Chinn, J. Chvatalova and N.E. Gibbs, *The bandwidth problem for graphs and matrices: a survey*, Journal of Graph Theory **6** (1982), 223–254.
202. L. Qi, E. Balas, and G. Gwan, *A new facet class and a polyhedral method for the three-index assignment problem*, Advances in Optimization (D.-Z. Du, ed.), Kluwer Academic, 1994, pp. 256–274.
203. M.J. Quinn, *Designing efficient algorithms for the speedup of parallel best-bound branch-and-bound algorithms*, McGraw-Hill, New York, 1987.
204. M. Raghavachari, *On connections between zero-one integer programming and concave programming under linear constraints*, Operations Research **17** (1969), 680–684.
205. G. Reinelt, *The linear ordering problem: algorithms and applications*, Research and Exposition in Mathematics, vol. 8, Heldermann Verlag Berlin, Berlin, 1985.
206. S. Reiter and D.B. Rice, *Discrete optimization solution procedures for linear and nonlinear integer programming problems*, Management Science **12** (1966), 829–850.
207. F. Rendl, *Ranking scalar products to improve bounds for the quadratic assignment problem*, European Journal of Operations Research **20** (1985), 363–372.
208. F. Rendl, *Quadratic assignment problems on series-parallel digraphs*, Zeitschrift fur Operations Research A **30** (1986), 161–173.
209. F. Rendl and H. Wolkowicz, *A projection technique for partitioning the nodes of a graph*, Annals of Operations Research (1990), no. CORR 90-20, To appear in the special issue of APMOD93 conference Budapest 1993.
210. F. Rendl and H. Wolkowicz, *Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem*, Mathematical Programming **53** (1992), 63–78.
211. W.T. Rhee, *A note on asymptotic properties of teh quadratic assigment problem*, Operations Research Letters (1989), 197–200.
212. ______, *Stochastic analysis of the quadratic assignment problem*, Manuscript, Ohio State University, Columbus, Ohio 43210, 1990.
213. R.H. Roth, *An approach to solving linear discrete optimization problems*, J. Assoc. Comput. Mach. **17** (1970), 300–313.
214. C. Roucairol, *Affection quadratique*, Ph.D. thesis, Universite Pierre et Marie Curie, Paris VI, 1976.
215. ______, *A reduction method for quadratic assignment problems*, Operations Research Verfahren **32** (1979), 183–187.

216. ———, *Un nouvel algorithme pour le problème d'affectation quadratique*, R.A.I.R.O. **13** (1979), 275–301.
217. ———, *Du sequentiel au parallele: la recherche aborescente a la programmation quadratique en variables 0 et 1*, 1987, Thèse d'Etat.
218. ———, *A parallel branch and bound algorithm for the quadratic assignment problem*, Discrete Applied Mathematics **18** (1987), 211–225.
219. R. Rudolf, *Dreidimensionale axiale zuordnungsprobleme*, Ph.D. thesis, Technische Universität Graz, Austria, 1991, Masters Thesis.
220. S. Sahni, *Computationally related problems*, SIAM Journal of Computing **3** (1974), 262–279.
221. S. Sahni and T. Gonzalez, *P-complete approximation problems*, Journal of the Association of Computing Machinery **23** (1976), 555–565.
222. J.E. Savage and M.G. Wloka, *Parallelism in graph-partitioning*, Journal of Parallel and Distributed Computing **13** (1991), no. 3, 257–272.
223. A.A. Schäffer and M. Yannakakis, *Simple local search problems that are hard to solve*, Tech. report, AT&T Bell Laboratories, 1989.
224. L. Schrage, *A more portable fortran random number generator*, ACM Transactions on Mathematical Software **5** (1979), 132–138.
225. Alexander Schrijver, *Theory of linear and integer programming*, John Wiley and Sons, 1986.
226. M. Scriabin and R.C. Vergin, *Comparison of computer algorithms and visual based methods for plant layout*, Management Science **22** (1975), 172–187.
227. R. Sedgewick, *Permutation generation methods*, Computing Surveys **9** (1977), no. 2, 137–164.
228. S. Senju and Y. Toyoda, *An approach to linear programming with 0-1 variables*, Management Science **15** (1968), B196–B207.
229. S.I. Sergeyev, *A new lower bound for the quadratic assignment problem*, Zh. Vychisl. Mat. Mat. Fiz. **27** (1987), no. 12, 1802–1811.
230. C.E. Shannon, *The zero-error capacity of a noisy channel*, I.R.E. Transactions **3** (1956).
231. SHARE, *Computerized relative allocation of facilities technique*, Share Program Library, 1965, SDA3391.
232. H.D. Sherali, *The quadratic assignment problem: exact and heuristic algorithms*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, June 1979.
233. H.D. Sherali and P. Rajgopal, *A flexible polynomial time construction and improvement heuristic for the quadratic assignment problem*, Computers & Operations Research **13** (1986), no. 5, 587–600.
234. B. Simeone, *An asymptotically exact polynomial time algorithm for equipartition problems*, Discrete Applied Math. **14** (1986), 283–293.
235. ———, *Combinatorial optimization*, Lecture Notes in Mathematics, vol. 1403, Springer-Verlag, 1986.
236. C.C. Skiscim and B.L. Golden, *Optimization by simulated annealing: A preliminary computational study for the tsp*, Proceedings of the 1983 Winter Simulation Conference, 1983.
237. J. Skorin-Kapov, *Extensions of tabu search adaptation to the quadratic assignment problem*, Computers and Operations Research **forthcoming**.
238. ———, *Tabu search applied to the quadratic assignment problem*, ORSA Journal on Computing **2** (1990), no. 1, 33–45.
239. T.H.C. Smith, *A computational comparision of an improved pair assignment algorithm and a pair exchange algorithm for the quadratic assignment problem*, Man. Sci. Res. Rep. 383, Carnegie Mellon University, Pittsburgh, Pa., November 1975.
240. L. Steinberg, *The backboard wiring problem: A placement algorithm*, SIAM Review **3** (1961), 37–50.
241. R.J. Stern and H. Wolkowicz, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations*, Tech. Report CORR 92-38, University of Waterloo, Waterloo, Canada, 1992, to appear in SIOPT.

242. E. Taillard, *Robust tabu search for the quadratic assignment problem*, Parallel Computing **17** (1991), 443–455.
243. Y. Toyoda, *A simplified algorithm for obtaining approximate solutions to zero-one integer programming*, Management Science **21** (1975), 1417–1427.
244. J. Turner and W.H. Kautz, *A survey of progress in graph theory in the Soviet Union*, SIAM **12** (1970).
245. I. Ugi, J. Bauer, J. Friedrich, J. Gasteiger, C. Jochum, and W. Schubert, *Neue anwendungsgebiete fur computer in der chemie*, Angew. Chemie **91** (1979), 99–184.
246. A. Vannelli and G. S. Rowan, *An eigenvector based approach for efficient multi-stack layout in vlsi design*, 29th Midwest Symposium on Circuits and Systems (M. Ismail, ed.), Noth Holland, New York, 1986, pp. 435–439.
247. M. Vlach, *Branch and bound method for the three index assignment problem*, Ekonomicko-Matematicky Obzor (1967), 181–191.
248. T. E. Vollman and E. S. Buffa, *The facilities layout problem in perspective*, Management Science **12** (1966), 450–468.
249. M. Werman, *The relationship between integer and real solutions of constrained convex programming*, Mathematical Programming **51** (1991), 133–135.
250. D.H. West, *Algorithm 608: Approximate solution of the quadratic assignment problem*, ACM Transactions on Mathematical Software **9** (1983), 461–466.
251. D.J. White, *A parametric-based heuristic program for the quadratic assignment problem*, Naval Research Logistics Quaterly **40** (1993), no. 4, 553–568.
252. B. Whitehead and M. Z. Elders, *An approach to the optimum layout of single-story buildings*, Architect's Journal **139** (1964), 1373–1380.
253. M.R. Wilhelm and T.L. Ward, *Solving quadratic assignment problems by simulated annealing*, IEEE Transactions **19** (1987), no. 1, 107–119.
254. W.S. Wong and R.J.T. Morris, *A new appraoch to choosing initial points in local search*, Information Processing Letters **30** (1991), 67–72.

(P.M. Pardalos) Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 USA and Technical University of Crete, Greece
E-mail address, P.M. Pardalos: pardalos@math.ufl.edu

(F. Rendl) Technische Universitat Graz, Institut fur Mathematik, Kopernikusgassa 24, A-8010 Graz, Austria
E-mail address, F. Rendl: rendl@ftug.dnet.tu-graz.ac.at

(H. Wolkowicz) University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario, Canada
E-mail address, H. Wolkowicz: hwolkowicz@orion.uwaterloo.ca