Math. Program., Ser. B 97: 27–42 (2003)

Digital Object Identifier (DOI) 10.1007/s10107-003-0437-z

Kurt M. Anstreicher

Recent advances in the solution of quadratic assignment problems

Received: February 13, 2003 / Accepted: April 22, 2003
Published online: May 28, 2003 –

Abstract. The quadratic assignment problem (QAP) is notoriously difficult for exact solution methods. In the past few years a number of long-open QAPs, including those posed by Steinberg (1961), Nugent et al. (1968) and Krarup (1972) were solved to optimality for the first time. The solution of these problems has utilized both new algorithms and novel computing structures. We describe these developments, as well as recent work which is likely to result in the solution of even more difficult instances.

Key words. quadratic assignment problem – discrete optimization – branch and bound

1. Introduction

The quadratic assignment problem (QAP) is a well known problem in location theory. The most general form of the problem, introduced by Lawler [43], is

$$\begin{aligned} \text{QAP}: \quad & \min \sum_{i,j,k,l} d_{ijkl} x_{ij} x_{kl} \\ & \text{s.t.} \quad X \in \Pi, \end{aligned}$$

where Π denotes the set of $n \times n$ permutation matrices. In practice many QAPs have the more restricted "Koopmans-Beckmann" form, corresponding to $d_{ijkl} = a_{ik} b_{jl}$, $i \neq k$, $j \neq l$. In a facility location application $x_{ij} = 1$ corresponds to facility i being placed in location j, a_{ik} is the flow between facilities i and k, and b_{jl} is the distance between locations j and l. In addition to location theory, QAP has applications in areas as diverse as ergonometric design [16] and data classification [35]. Several recent surveys [12, 18, 54] give extensive references on applications and solution methods for QAP.

It is well known that QAP is NP-hard, since for example common graph problems such as TSP can be posed as QAPs. Although some "easy" cases are known [13], QAPs in general have proven to be extraordinarily difficult to solve to optimality. Several famous instances, including the problems of size $n = 36$ posed by Steinberg in 1961 [63], and problems of size $n = 30$ posed by Nugent et al. in 1968 [51] and Krarup in 1972 [34, 42], have only very recently been solved to optimality for the first time. The purpose of this paper is to describe the advancements in algorithms and computational platforms that have made the solution of these large instances possible.

K.M. Anstreicher: Department of Management Sciences, University of Iowa, Iowa City, IA 52242, e-mail: kurt-anstreicher@uiowa.edu

Mathematics Subject Classification (1991): 90C27, 90C09, 90C20

In the next section we briefly describe heuristic search methods for QAP, and in section 3 we describe a variety of lower-bounding techniques for QAP. Lower bounds are useful for providing worst-case measures for the quality of solutions obtained by heuristic search, and are an essential ingredient in the construction of Branch and Bound (B&B) methods for solving problems to optimality. In recent years a variety of new bounds have been derived, some of which have obtained the best known root bounds for difficult problems and/or been successfully incorporated into state-of-the-art B&B algorithms. In section 4 we describe B&B algorithms for solving QAPs to optimality. Most recent solutions of large problems have utilized advanced bounding techniques, but the ste36a/b/c problems from [63] were (rather surprisingly) solved using the well-known Gilmore-Lawler bound. (All problem names are taken from QAPLIB [15].) In addition to the use of new bounds, recent B&B implementations have employed *strong branching*, described in section 4.1, to substantially reduce the amount of enumeration required. The solutions of several large QAPs were accomplished using massively distributed computation, facilitated by recently-developed "metacomputing" systems. Some details concerning these metacomputing (also known as "grid computing") implementations are considered in section 4.2. In section 4.3 we describe symmetry that is present in many QAPs. The symmetry inherent in grid-based problems is well known, but some other benchmark problems have high degrees of symmetry that could be exploited to reduce redundant enumeration. Consideration of such problem-specific structure is likely to be important in attempts to solve some currently-unsolved large instances.

Notation. For a square matrix A, $\mathrm{tr}(A)$ denotes the trace of A, $\mathrm{diag}(A)$ is the vector consisting of the diagonal components of A, and $\mathbf{vec}(A)$ is the vector obtained by "stacking" the columns of A in the natural order. The Kronecker product of matrices A and B is denoted $A \otimes B$, and $A \bullet B = \mathrm{tr}(AB^T)$. For symmetric matrices A and B, $A \succeq B$ denotes that $A - B$ is positive semidefinite, and $\lambda(A) \in \Re^n$ denotes the vector of eigenvalues of A. The set of permutation matrices is denoted Π, and $\mathcal{O}$ denotes the set of orthogonal matrices; $\mathcal{O} = \{X \mid XX^T = X^TX = I\}$.

A vector of arbitrary dimension with all components equal to one is denoted e. For vectors u and v, $\langle u, v\rangle_-$ denotes the "minimal product"

$$\langle u, v\rangle_- := \min_{\pi} \sum_{i=1}^{n} u_i v_{\pi(i)},$$

where $\pi(\cdot)$ is a permutation of $1, 2, \ldots, n$. It is easy to show that $\langle u, v\rangle_-$ is obtained by putting the components of one of the vectors in nondecreasing order, and the components of the other in nonincreasing order, before taking the inner product.

For convenience we often use the name of an optimization problem, such as QAP, to also refer to its optimal value.

2. Heuristics

The extreme difficulty of QAP has made it an ideal problem for the development of heuristic search methods. Simulated annealing [17, 20], tabu search [61, 64], genetic algorithms [2, 21, 66], GRASP [44], ant systems [25] and other specialized methods

have all been applied to QAP. Recent implementations often give optimal or near-optimal solutions on most QAPLIB problems [21], although some problems (such as the larger "sko" instances [61]) are still challenging. The performance of different heuristics also tends to vary with certain problem characteristics [65].

It has recently been shown that the "typical" QAP behavior of being relatively easy for heuristic search and very difficult for exact methods does not universally hold. The authors of [22] construct new classes of test problems, up to size $n = 75$, which prove to be relatively difficult for state-of-the-art heuristics [21, 25], but solvable to optimality using the Dual-LP-based B&B algorithm of [33].

3. Lower bounds for QAP

In this section we consider a number of different procedures for obtaining a lower bound on the optimal value of QAP. When a search heuristic is applied to QAP, such a bound provides a "worst-case" gap to optimality for the best solution found. Lower bounds are also essential in the construction of B&B algorithms for solving QAP to optimality, considered in the next section.

3.1. Gilmore-Lawler bound

The best-known lower bound for the QAP is the Gilmore-Lawler bound (GLB) [26, 43]. For $i, j = 1, \ldots, n$, let f_{ij} denote the solution value in the linear assignment problem

$$\begin{aligned} \min \ & \sum_{k,l} d_{ijkl} x_{kl} \\ \text{s.t.} \ & X \in \Pi, \ x_{ij} = 1. \end{aligned}$$

It is then clear that GLB $:=$ LAP$(F) \leq$ QAP, where LAP(F) denotes the linear assignment problem with cost matrix F. For a general Lawler QAP the computation of GLB requires the solution of $n^2 + 1$ LAPs. However, for a problem in Koopmans-Beckmann form the LAP associated with each f_{ij} is trivial to solve, and as a result F can be obtained in a total of only $O(n^3)$ operations.

Until recently virtually all successful B&B algorithms for QAP were based on the GLB or closely related bounds; see for example [11, 14, 19, 49, 50, 55]. On the benchmark problems of Nugent et al. [51], GLB-based algorithms were effective for problems up to about size $n = 24$, but the growth rate of the B&B tree made the solution of larger instances such as nug30 appear to be impossible.

3.2. Eigenvalue bounds

A Koopmans-Beckmann QAP can be written in the matrix form

$$\min_{X \in \Pi} \ \operatorname{tr}(AXB + C)X^T. \tag{1}$$

When A and B are symmetric, a bound for the quadratic term can be based on the fact that $X \in \Pi \Rightarrow X \in \mathcal{O}$. In particular [24]

$$\min_{X \in \mathcal{O}} \operatorname{tr}(AXBX^T) = \langle \lambda(A), \lambda(B) \rangle_-,$$

and therefore

$$\langle \lambda(A), \lambda(B) \rangle_- + \mathrm{LAP}(C) \tag{2}$$

is a valid lower bound for (1). The simple eigenvalue bound (2) is too weak to be computationally useful, but several schemes for improving the bound have been considered [24, 28, 58]. One well-known variant is the projected eigenvalue bound (PB) of [28]. The construction of PB is based on enforcing the row and column sum constraints on X, in addition to orthogonality. Let V be an $n \times (n-1)$ matrix whose columns are an orthonormal basis for the nullspace of e^T, $\hat{A} = V^T A V$, $\hat{B} = V^T B V$ and $D = C + (2/n) A e e^T B$. Then

$$\mathrm{PB} := \langle \lambda(\hat{A}), \lambda(\hat{B}) \rangle_- + \mathrm{LAP}(D) - \frac{1}{n^2}(e^T A e)(e^T B e).$$

For many problems PB provides a good quality bound at modest computational cost.

3.3. Quadratic programming bounds

Another methodology for obtaining a lower bound on QAP is to somehow convexify the quadratic objective. A scheme of this type for Koopmans-Beckmann problems, based on modifying the diagonal elements of A and B, is well known [54]. Some computational results for this bound are given in [7, Section 6.1]. A different approach, based on the derivation of the projected eigenvalue bound PB, was introduced in [5]. The result is a bound

$$\begin{aligned} \mathrm{QPB}: \quad & \min \ \mathbf{vec}(X)^T Q \, \mathbf{vec}(X) + C \bullet X + \langle \lambda(\hat{A}), \lambda(\hat{B}) \rangle_- \\ & \text{s.t.} \ Xe = X^T e = e \\ & \quad X \geq 0, \end{aligned}$$

where Q is a matrix of the form $Q = (B \otimes A) - (I \otimes S) - (T \otimes I)$. The matrices S and T are of the form $S = V \hat{S} V^T$, $T = V \hat{T} V^T$, where $\hat{S}$ and $\hat{T}$ are optimal solutions of the semidefinite programming problem

$$\begin{aligned} \mathrm{SDD}(\hat{A}, \hat{B}): \quad & \max \ \operatorname{tr} \hat{S} + \operatorname{tr} \hat{T} \\ & \text{s.t.} \ (I \otimes \hat{S}) + (\hat{T} \otimes I) \preceq (\hat{B} \otimes \hat{A}). \end{aligned}$$

It can be shown [6] that $\mathrm{SDD}(\hat{A}, \hat{B}) = \langle \lambda(\hat{A}), \lambda(\hat{B}) \rangle_-$, and as shown in [5, Section 4] optimal $\hat{S}$, $\hat{T}$ can easily be obtained from the spectral decompositions of $\hat{A}$ and $\hat{B}$. The objective in QPB is convex on the nullspace of the equality constraints, so QPB is a convex quadratic program.

By construction QPB $\geq$ PB. On typical benchmark problems the improvement over PB is modest [5], but QPB has several significant advantages for implementation within B&B. For example, the value of QPB is associated with a doubly-stochastic matrix X.

This X, and dual information provided from the solution of QPB, are very valuable in devising branching strategies. The value of QPB also appears to rise much faster than PB as branching occurs.

Evaluating QPB requires the approximate solution of a convex quadratic program in the n^2 variables X. There are a variety of approaches to such a problem, but most would be too expensive for implementation within B&B. In [9] the Frank-Wolfe (FW) algorithm is used to approximately evaluate QPB. Although the asymptotic properties of the FW algorithm are poor, this scheme has potential for solution of QPB because the work on each iteration of the FW algorithm is dominated by the solution of a single, small LAP.

3.4. LP and dual–LP bounds

A large class of bounds for the QAP are related to linear programming (LP) relaxations of the problem. Defining new variables $y_{ijkl} = x_{ij}x_{kl}$, and dropping the integrality conditions, results in an LP relaxation [1, 59]

$$
\begin{aligned}
\text{LPQAP}: \quad \min \ & \sum_{i,j,k,l} y_{ijkl} d_{ijkl} \\
\text{s.t.} \ & \sum_{j} x_{ij} = 1, \quad i = 1, \ldots, n, \\
& \sum_{i} x_{ij} = 1, \quad j = 1, \ldots, n, \\
& \sum_{l} y_{ijkl} = x_{ij} \quad i, j, k = 1, \ldots, n, \\
& \sum_{k} y_{ijkl} = x_{ij} \quad i, j, l = 1, \ldots, n, \\
& y_{ijkl} = y_{klij} \quad i, j, k, l = 1, \ldots, n, \qquad (3) \\
& x_{ij} \geq 0, \quad y_{ijkl} \geq 0, \quad i, j, k, l = 1, \ldots, n.
\end{aligned}
$$

The symmetry constraints (3) imply that LPQAP can be formulated using variables $y_{ijkl}, i \leq k$. Additional variables can be eliminated using the facts that for any X feasible in QAP, $y_{ijij} = x_{ij}$ for all i and j, $y_{ijil} = 0$ for all i and $j \neq l$, and $y_{ijkj} = 0$ for all $i \neq k$ and j. Taken together, these observations allow for a reformulation of LPQAP as an LP problem with $n^2 + n^2(n-1)^2/2$ variables. Further analysis [53, Section 7.1] can be used to reduce the number of equality constraints required in LPQAP to $2n(n-1)^2 - (n-1)(n-2)$, $n \geq 3$. For problems with symmetric data ($d_{ijkl} = d_{ilkj}$) the number of variables and equality constraints can be approximately halved [36, 53].

For QAPs of size $n \geq 25$ the solution of LPQAP can be computationally quite challenging. The solution of LPQAP using an interior-point method was investigated in [59]. This approach produces excellent bounds for many problems, but appears to be prohibitively costly for implementation in a B&B algorithm.

It is known [1] that if the symmetry conditions (3) are dropped then the solution value in LPQAP is exactly GLB. It can also be shown [40] that many bounding schemes

for QAP can be viewed as Lagrangian procedures that attempt to approximately solve the dual of LPQAP. Computationally the most successful of these is a method motivated by the Hungarian algorithm for LAP, due to P. Hahn and co-workers [30, 31, 33].

3.5. Polyhedral bounds

Polyhedral bounds for QAP are based on considering the convex hull of integer solutions to the problem LPQAP. There has recently been considerable progress in the development of such methods. Preliminary results characterizing the affine hull of solutions, and simple facets, were obtained in [36, 37, 53]. A nontrivial class of facets (the "box-inequalities") is described in [37]. Computational results in [37] show considerable improvement over the bound from LPQAP on many QAPLIB problems up to size $n = 20$. The results are particulary good on the "esc16" problems [23], for which the optimal solution is obtained in all but one case. Further refinements [38, 39] lead to the solution (without branching) of three of the esc32 instances, and the highest known bounds for the remaining esc32 problems.

3.6. Second-order RLT bounds

The bound for QAP based on LPQAP, described above, can be viewed as an application of the "Reformulation-Linearization Technique" (RLT) of [60]. In fact a hierarchy of RLT bounds is possible, beginning with the first-order bound from LPQAP. A second-order RLT bound can be computed from an LP problem with variables z_{ijklpq}, corresponding to products of the form $x_{ij}x_{kl}x_{pq}$. The possibility of using these bounds was first suggested in [57]. More extensive computations performed in [56] obtained the optimal solutions for QAPs of size up to $n = 12$ without branching. The major difficulty with these RLT-2 bounds is that the size of the LP that must be solved grows extremely rapidly with n.

In [32] the dual-LP approach of Hahn et al. [30, 31, 33] is extended to apply to the LP arising from the second-order RLT. Computational results suggest that this dual-LP approach is much more efficient than direct solution of the level-2 LP, although storage requirements are still a non-trivial issue for larger problems.

3.7. Semidefinite programming bounds

In recent years there has been considerable interest in obtaining semidefinite programming (SDP) bounds for discrete optimization problems. SDP bounds for QAP were first described in [45] and [68]. For an $n^2 \times n^2$ matrix Y, let $Y_{[ij]}$ denote the $n \times n$ matrix which is the ij "block" of Y, $i, j = 1, \ldots, n$. In other words,

$$Y = \begin{pmatrix} Y_{[11]} & \cdots & Y_{[1n]} \\ \vdots & \ddots & \vdots \\ Y_{[n1]} & \cdots & Y_{[nn]} \end{pmatrix}.$$

Define [68] the linear operators from $\Re^{n^2 \times n^2}$ to $\Re^{n \times n}$:

$$\begin{aligned} \operatorname{bdiag}(Y) &= \sum_{i=1}^{n} Y_{[ii]} \\ (\operatorname{odiag}(Y))_{ij} &= \operatorname{tr} Y_{[ij]}, \quad i, j = 1, \ldots, n. \end{aligned}$$

Finally let F be the $2n \times n^2$ matrix

$$F = \begin{pmatrix} e^T \otimes I \\ I \otimes e^T \end{pmatrix}.$$

The matrix F arises naturally in the representation of the assignment constraints of QAP when the matrix X is written as a vector $\mathbf{vec}(X)$; $Xe = X^T e = e$ is equivalent to $F\,\mathbf{vec}(X) = e$. For a homogeneous ($C = 0$) Koopmans-Beckmann QAP, the basic SDP bound of [68] is

$$\begin{aligned} \text{SDPB1}: \quad \min\ & (B \otimes A) \bullet Y \\ \text{s.t.}\ & \operatorname{bdiag}(Y) = I \\ & \operatorname{odiag}(Y) = I \\ & F \bullet Y = 2n \\ & Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0. \end{aligned}$$

In SDPB1 the matrix Y is a relaxation of the rank-one matrix $\mathbf{vec}(X)\,\mathbf{vec}(X)^T$. The constraints involving $\operatorname{odiag}(Y)$ and $\operatorname{bdiag}(Y)$ are generalizations of the orthogonality conditions $XX^T = XX^T = I$, and the constraint on $F \bullet Y$ generalizes the constraints $Xe = X^T e = e$. It is shown in [4] that SDPB1 and PB are closely related, and SDPB1 $\geq$ PB always holds in the computational results of [68]. Results in [5] show that it is generally the case that SDPB1 $\geq$ QPB, although QPB $>$ SDPB1 is possible.

Evaluation of SDPB1 is complicated by the fact that all feasible solutions are singular, so standard SDP algorithms cannot be applied without first obtaining a lower-dimensional representation of the problem [68]. SDPB1 can be strengthened by imposing non-negativity constraints on the components of Y, as well as constraints that certain components of Y must be zero. (The latter are identical to the constraints that certain variables y_{ijkl} of LPQAP should be zero, as described in Section 3.4, although in the case of SDP bounds the components of Y cannot be simply "removed.") Imposing both types of constraints can result in very good bounds, but the resulting mixed SDP/LP problem is computationally difficult to solve. Recent work [62] has considered the application of bundle methods to handle the large number of constraints that appear in such strong SDP relaxations for QAP. This methodology obtains the strongest known root bounds for some benchmark problems, including nug30. Whether the cost of obtaining these bounds is low enough to permit their use within a B&B algorithm is still an open question.

3.8. Comparison of bounds

When comparing bounds for QAP the most important issues are the strength of a bound and the computational expense required to compute it. Some methodologies (LP, SDP,

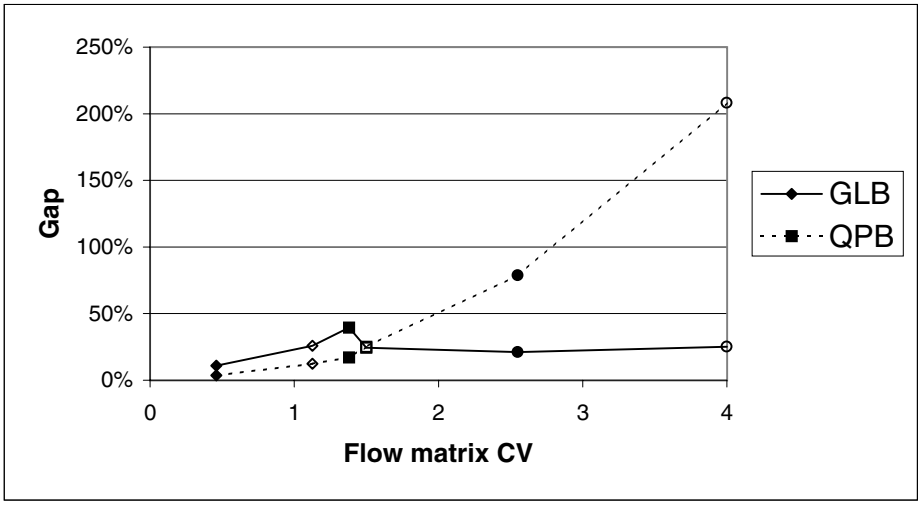

Fig. 1. Gaps for bounds on grid-based QAPs

RLT-2) give excellent root bounds but may be too expensive to implement in B&B algorithms. Parametric strengthenings, such as those described in [41, 58], can provide substantial improvements of root bounds but also seem ill-suited for implementation within B&B.

The behavior of some bounds appears to vary considerably with certain problem characteristics. This is particulary true of eigenvalue bounds, which have been noted to often behave poorly on sparse problems. Figure 1 [10] compares the root gap for GLB and QPB on a number of grid-based QAPLIB problems, plotted against the coefficient of variation (CV, equal to the standard deviation divided by the mean) of the flow matrix components. Recall that QPB is closely related to the projected eigenvalue bound PB, and note that the CV naturally increases with sparsity of the flow matrix. The problems considered, in order of increasing CV, are had20, nug30, tho30, kra30b, scr20 and ste36a. For dense problems with a low CV, such as had20, nug30 and tho30, the gap for QPB is substantially less than for GLB. However, for the very sparse problem ste36a the value of QPB is negative, and the relative gap exceeds 200%. Poor behavior of SDPB1, which is also related to PB, has been noted on some sparse problems [68]. In the literature on heuristics for QAP, CV (often termed "flow dominance") is commonly used to classify problems and/or control algorithms; see for example [65].

In Table 1 we compare a number of different bounds on the kra30a, nug30 and ste36a instances. The bounds are listed in approximate order of computational expense – n.a. denotes that a bound is not available. TDB is the "triangle decomposition bound" [41], a parametric strengthening of PB that applies to grid-based problems, and SDPB3 is a strong SDP bound computed using a bundle method [62]. The Dual-LP bounds, taken from [40], are based on a methodology very similar to that used in [30]. Underlined values correspond to the bounds used in algorithms that first solved these instances to optimality, as described in the next section. In addition to the differences in PB and QPB versus GLB noted above, the general trends that GLB $<$ Dual-LP, and PB $<$ QPB $<$

Table 1. Bounds for kra30a, nug30 and ste36a QAPs

	kra30a		nug30		ste36a	
Bound	Value	Gap	Value	Gap	Value	Gap
GLB	68360	23.1%	4539	25.9%	7124	25.2%
PB	63717	28.3%	5266	14.0%	−11700	222.8%
QPB	68572	22.9%	5365	12.4%	−10294	208.1%
TDB	n.a.	n.a.	5772	5.7%	6997	26.5%
Dual-LP	75566	15.0%	4785	21.9%	7860	17.5%
SDPB1	69736	21.6%	5413	11.6%	n.a.	n.a.
SDPB3	77647	12.7%	5803	5.2%	n.a.	n.a.
OPT	88900	0.0%	6124	0.0%	9526	0.0%

SDPB1 < SDPB3, are evident. The SDPB3 bounds for kra30a and nug30 are the best known bounds for these problems obtained without branching.

4. Exact solution algorithms

Exact algorithms for QAP are typically of the B&B, or "implicit enumeration" type. The major characteristic distinguishing methods from one another is the choice of lower bound used. To date, complete B&B algorithms have been implemented using the GLB, QPB, and dual-LP bounds described in Section 3.

In recent years a number of long-open large QAPs have been solved to optimality for the first time. In Table 2 we summarize these results. In the table, "CPU days" reports the total amount of unnormalized CPU time expended on the solution of a problem. The "ste36" problems arose from a backboard wiring application in 1961 [63]. Nug30, perhaps the best-known problem in the QAP literature, is the largest of the problems posed by Nugent et al. [51] in 1968. The "kra" problems come from a hospital planning application in 1972 [42]; see [34] for an interesting discussion of the history of these problems. The kra32 problem is based on kra30a, but adds two "dummy" facilites to allow for the use of the full 4 by 4 by 2 grid of locations which forms the basis for the distances in kra30a[1]. The tai25a problem has randomly generated flows and distances [64]; the related tai30a problem is now the smallest unsolved problem in QAPLIB with symmetric data. The tho30 problem, which is more recent [67], may be the most difficult QAP solved to date. All of the problems listed in Table 2 except tai25a have distance matrices that are based on two-dimensional or three-dimensional rectangular grids.

It is interesting to note the variety of bounds used to solve the problems listed in Table 2, in particular the use of GLB to solve the ste36a/b/c problems. In the 1990s a great deal of computational effort was focused on the solution of the nug problems, from [51]. The fact that solution of the largest instances (nug27/28/30) using GLB is impractical, if not impossible, spurred the development of the alternative bounding techniques described in Section 3. Given the difficulty of nug30, most researchers probably viewed the ste36a/b/c problems as being hopelessly beyond the reach of exact algorithms. However, as described above, the ste36 problems are structurally very different from nug30.

[1] The optimal values for kra30a and kra32 are 88900 and 88700, respectively. Due to a clerical error the optimal value of kra32 was incorrectly reported in [3], but the optimal permutation given there is correct.

Table 2. Recently-solved large QAPs

Problem	Reference	Bound	Platform	CPU days
kra30a	[33]	Dual-LP	Serial	99
kra30b/32	[3]	QPB	Distributed	1527/5536
nug27/28/30	[3]	QPB	Distributed	113/722/3999
ste36a	[10]	GLB	Serial	18
ste36b/c	[52]	GLB	Distributed	60/200
tai25a	[29]	Dual-LP	Serial	394
tho30	[3]	QPB	Distributed	8997

As shown in Table 1 the root gaps on nug30 and ste36a using GLB are similar. However the gaps using GLB decrease much faster as branching occurs on ste36a than on nug30, so that in the end the solution of ste36a using GLB is relatively practical.

In addition to the "first solution" results reported in Table 2, the problems kra30b, nug27 and nug28 were subsequently solved using Dual-LP methods [33] based on RLT-1 and RLT-2 relaxations of QAP in less time than required for the QPB-based algorithm of [3], after adjusting for differences in machine speed.

4.1. Strong branching

Branch and bound algorithms for QAP typically use "polytomic" branching, where at any node children are created by either *(row branching)* assigning a chosen facility to each available location, or *(column branching)* fixing a location and assigning to it all available facilities. In early B&B implementations the facility or location on which to branch was typically chosen arbitrarily. The use of dual information associated with GLB as the basis for branching decisions was first suggested in [50]. This methodology is very effective on small problems, but as n increases the quality of information near the root deteriorates. However, the first few branching decisions can have a very large effect on the eventual evolution of the B&B tree. The idea of *strong branching* is to compute (partially or fully) bounds for prospective children, and use this additional information to make a final branching decision. It is now recognized that strong branching can be very effective in reducing the computation for difficult discrete optimization problems [46], but until recently the methodology had not been applied to QAP. All of the results described in Table 2, with the exception of [52], used strong branching techniques to reduce the size of the B&B tree. In the solution of nug30, based on QPB, over 40% of total worker CPU time was spent executing strong branching strategies on selected nodes in levels 0–7 of the tree.

Table 3 illustrates the effect of strong branching in the solution of the ste36a problem, using a GLB-based algorithm [10]. The results without strong branching use the technique introduced in [50] throughout. The second set of results, from the final solution run, use a strong branching strategy applied to selected nodes on levels 0-6. In the table "Gap" is the average gap between the optimal value and the lower bounds of the nodes at each level, "Fthm" is the fraction of nodes fathomed at each level, and "Elim" is the fraction of potential children of unfathomed nodes eliminated. The effect of strong branching is very clear. Compared to the solution run, the algorithm without strong

Table 3. Effect of strong branching applied to ste36a

	Without strong branching				With strong branching			
Level	Nodes	Gap	Fthm	Elim	Nodes	Gap	Fthm	Elim
0	1	2402.0	0.000	0.000	1	2402.0	0.000	0.000
1	10	2280.5	0.000	0.000	10	1953.7	0.000	0.054
2	318	1770.3	0.003	0.069	301	1218.4	0.000	0.421
3	9,941	1239.8	0.080	0.549	5,869	697.1	0.070	0.787
4	136,112	727.7	0.209	0.580	38,263	542.5	0.320	0.441
5	1,445,612	594.7	0.336	0.535	465,182	354.3	0.404	0.569
6	13,832,243	438.4	0.445	0.629	3,703,103	260.8	0.579	0.752
7	85,562,934	322.4	0.546	0.613	11,627,541	183.0	0.641	0.806
8	436,142,577	266.5			23,549,921	132.2	0.730	0.821

branching has over 18 times as many nodes on level 8, and the average gap for these nodes is doubled. (The algorithm without strong branching was only run down to level 8 due to the excessive growth in the tree.) The time expended on strong branching in the solution run was a negligible fraction ($< 4\%$) of the total CPU time.

4.2. Distributed computation

The solution of many of the problems listed in Table 2 involved distributed computation. In the past, landmark results on large QAPs have often utilized parallel processing hardware [11, 19, 49]. The capability of such supercomputers has continued to increase, but expense and lack of availability limits their usefullness for applications like QAP. Distributed computation, an alternative paradigm for high-throughput computing, utilizes multiple machines interfaced with some form of communication network. The latter could be as small as a local area network (LAN) connecting a few machines physically located near one another, or as large as the Internet. The terms *metacomputing* and *grid computing* refer to very-large-scale distributed computation, where machines are geographically dispersed and beyond the control of any one party.

In distributed computation the availability of CPUs is typically dynamic, and a "master" process utilizes "worker" (or "slave") resources as they become available. Fault tolerance is of critical importance since worker processes are inherently unreliable, and can abort before completing their assigned tasks. Fault tolerance for the master process is also essential during long runs.

The solution of the ste36b/c problems by M. Nyström [52] was based on a distributed computing network consisting of 22 Pentium Pro processors. Nyström's work was all-but-unknown until the solution of ste36a was announced in [10]. The problems solved in [3] utilized the MW (for "Master-Worker") metacomputing system developed as part of the metaNEOS project at Argonne National Lab [27]. MW uses the Condor system [47] to manage a potentially very large number of worker machines. The solution of nug30 utilized an average of about 650 CPUs over a one week period, providing the equivalent of almost 7 years of computation on a single HP9000 C3000 workstation. The computations to solve the tho30 problem were even more extensive.

With further development of metacomputing systems like MW, it should become easier to use highly distributed computation to solve large QAPs. It is worth noting,

however, that inherent characteristics of a metacomputing environment may be more compatible with some solution techniques than with others. For QAP, methods based on relatively inexpensive bounds such as GLB and QPB are easily adapted to grid computing. On the other hand the use of very expensive bounds (LP, SDP, RLT-2) would raise some serious issues. For example, the resources in a large grid computing environment are typically very heterogenous, so requiring high performance (fast CPU speed and/or large memory) could rule out many available processors. In addition, if worker tasks become very large then checkpointing and migration of tasks becomes important to avoid the potential loss of computation when workers become unavailable. The use of some bounds could also substantially increase the amount of information that must be transmitted to workers, and/or stored on the master processor. The repeated transfer of large amounts of data is undesirable in a metacomputing environment because the communication speed between the master and worker machines can be very slow. For example, the child problems created when using some dual-LP bounds [30, 31, 33] inherit a transformed version of the problem data from the parent node. This $O(n^4)$ information would need to be transferred when a worker is sent a node in the B&B tree to explore, and the worker would have to return the information associated with any unfathomed nodes that it sends back to the master process. The management of the master queue in such an implementation could also become nontrivial.

4.3. Symmetry

Many QAPs have distance matrices that are based on rectangular grids. It is well known that the symmetry in such problems can be exploited in the branching process to avoid the creation of redundant children. The algorithms used to solve the problems in Table 2 all use such logic, at least at the root of the B&B tree.

Besides the obvious symmetry present in grid-based problems, some other benchmark QAPs have high degrees of symmetry that do not appear to have been previously elucidated. For example, the esc problems [23] of size $n = 2^k$ have distance matrices that arise from Hamming distances between k-bit binary words. In the data for these problems, location $j = 1, \ldots, n$ is associated with the k-bit representation of the integer $j - 1$. Using this association, it is clear that distances are preserved by arbitrary complementing and/or permutations of bits. It follows that any assignment is a member of an equivalence class having $k!2^k$ members, all with the same objective value.

In addition to this high degree of distance symmetry, many of the esc problems have groups of equivalent facilities that can be interchanged without changing the value of the objective. Each group of size m contributes an additional factor of $m!$ to the number of assignments equivalent to any given assignment. An obvious example of equivalent facilities is a group with no flows whatsoever, corresponding to blocks of zero rows and columns in the flow matrix. Many of the esc problems have such blocks, a property that was exploited in the polyhedral method described in [38]. However there are other instances, for example esc16b/h, that have high degrees of flow symmetry *not* corresponding to zero rows and columns.

In Table 4 we illustrate the effect of flow and distance symmetry on the esc16 problems solved in [55]. The combinatorial branching rule used to solve these problems

Table 4. Degree of symmetry in esc16 problems

Problem	Optimal Solutions	Flow Factor	Distance Factor	Distinct Solutions
esc16a	13,271,040	2,880	384	12
esc16c	2,064,384	24	384	224
esc16e	30,965,760	20,160	384	4
esc16g	46,448,640	40,320	384	3
esc16i	710,277,120	5,040	384	367

Table 5. Solution of esc16a using flow and distance symmetries

Level	Nodes	Fthm	Elim	Secs
0	1	0.0000	0.0000	0.01
1	1	0.0000	0.0000	0.04
2	4	0.0000	0.1818	0.67
3	18	0.0000	0.7203	3.64
4	31	0.1613	0.7892	4.42
5	39	0.0256	0.5123	0.13
6	119	0.5210	0.4522	0.35
7	195	0.6667	0.4468	0.41
8	213	0.6197	0.5356	0.44
9	215	0.6605	0.3871	0.39
10	247	0.7166	0.6378	0.21
11	117	0.7778	0.6804	0.11
12	31	0.8710	0.7857	0.01
13	3			0.00
Total	1,234			10.85

generated all optimal permutations. The number of such permutations found is listed under "Optimal Solutions." "Flow Factor" and "Distance Factor" are the factors for the number of equivalent assignments that can be generated from any given assignment using the flow and distance symmetries described above. For all of the esc16 problems the distance factor is $4!2^4 = 384$. "Distinct Solutions" is the number of optimal solutions divided by the flow and distance factors, representing the number of optimal solutions that *cannot* be obtained from one another using using the flow and distance symmetries. It is notable that in all cases the number of optimal solutions found in [55] is divisible by the flow and distance factors, as should be the case.

It is obvious that the high degreee of symmetry in the esc problems could be exploited within a B&B algorithm to avoid the creation of redundant children. In Table 5 we illustrate the solution of esc16a using a prototype algorithm [8] that considers both the flow and distance symmetries present in these problems. The solution run shown is based on QPB, but the consideration of symmetry applies to any bounding methodology. The algorithm considers both row branching using distance symmetry and column branching using flow symmetry, with priority given to the former. The "Fthm" and "Elim" statistics are as described for Table 3, and "Secs" is total CPU seconds expended at each level of the tree, running on an 800 MHz Pentium III. It is worth noting that the first solution of esc16a, reported in 1997 [19], required substantially *more* nodes (4,187,636) when attempting to use problem symmetry than when symmetry was ignored (594,177). Careful consideration of the structure present in the larger esc instances is likely to be

important in attempts to solve these problems to optimality. (The esc32a/b/c/d/h QAPs are now, after tai30a, the smallest unsolved QAPLIB problems with symmetric data.) A general methodology for exploiting high degrees of symmetry in discrete optimization problems, including the distance symmetry present in the esc problems, is described in [48].

5. Conclusion

The QAP remains very challenging, but advances in algorithms and computing platforms have facilitated the exact solution of long-open instances that were until quite recently thought to be beyond reach. The use of more complex bounds, strong branching, and distributed processing implementation have all played essential roles in the recent solutions of large problems. Further advances in these areas, as well as consideration of problem-specific structure, will likely lead to the solution of even more difficult instances in the near future.

References

1. Adams, W.P., Johnson, T.: Improved linear programming based lower bounds for the quadratic assignment problem quadratic assignment and related problems. In: Pardalos P. and Wolkowicz H., eds., Quadratic Assignment and Related Problems, Volume **16** of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS. **16**, 43–77 (1994)
2. Ahuja, R.K., Orlin, J.B., Tiwari, A.: A descent genetic algorithm for the quadratic assignment problem. Com. Oper. Res. **27**, 917–934 (2000)
3. Anstreicher, K., Brixius, N., Goux J-P., Linderoth, J.: Solving large quadratic assignment problems on computational grids. Math. Prog. **91**, 563–588 (2002)
4. Anstreicher, K.M.: Eigenvalue bounds versus semidefinite relaxations for the quadratic assignment problem. SIAM J. Optim. **11**, 254–265 (2001)
5. Anstreicher, K.M., Brixius, N.W.: A new bound for the quadratic assignment problem based on convex quadratic programming. Math. Prog. **89**, 341–357 (2001)
6. Anstreicher, K.M., Wolkowicz, H.: On Lagrangian relaxation of quadratic matrix constraints. SIAM J. Matrix Anal. App. **22**, 41–55 2000
7. Brixius, N.W.: Solving large-scale quadratic assignment problems. Ph.D. thesis, Department of Computer Science, University of Iowa, 2000
8. Brixius, NW.: Private communications, 2002
9. Brixius, N.W., Anstreicher, K.M.: Solving quadratic assignment problems using convex quadratic programming relaxations. Optim. Methods and Software **16**, 49–68 (2001)
10. Brixius, N.W., Anstreicher, K.M.: The Steinberg wiring problem. In: Grötschel, M. ed., The Sharpest Cut, Fest-Schrift in Honor of Manfred Padberg's 60th Birthday, SIAM, 2003, pp. 331–348
11. Brüngger, A., Marzetta, A., Clausen, J., Perregaard, M.: Solving large-scale QAP problems in parallel with the search library ZRAM. J. Parallel and Distributed Com. **50**, 157–169 (1998)
12. Burkard, R.E., Çela, E., Pardalos, P.M., Pitsoulis, L.S.: The quadratic assignment problem. In: Du, D.-Z. and Pardalos, P.M. eds., Handbook of Combinatorial Optimization. volume **3**, Kluwer, 1998, pp. 241–337
13. Burkard, R.E., Çela, E., Rote, G., Woeginger, G.J.: The quadratic assignment problem with a monotone anti-Monge and a symmetric Toeplitz matrix: Easy and hard cases. Math. Prog. **82**, 125–158 (1998)
14. Burkard, R.E., Derigs, U.: Assignment and Matching Problems: Solution Methods with Fortran Programs. Volume **184** of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, 1980
15. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB – a quadratic assignment problem library. J. Global Optim. **10**, 391–403 (1997) See also `www.opt.math.tu-graz.ac.at/~qaplib`
16. Burkard, R.E., Offermann, J.: Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme. Zeifschrift für Oper. Res. **21**, B121–B132 (1977)
17. Burkard, R.E., Rendl, F.A.: Thermodynamically motivated simulation procedure for combinatorial optimization problems. Eur. J. Oper. Res. **17**, 169–174 (1984)

18. Çela, E.: The Quadratic Assignment Problem: Theory and Algorithms. Kluwer, 1998
19. Clausen, J., Perregaard, M.: Solving Large quadratic assignment problems in parallel. Computational Optim. App. **8**, 111–127 (1997)
20. Connolly, D.T.: An improved annealing scheme for the QAP. Eur. J. Oper. Res. **46**, 93–100 (1990)
21. Drezner, Z.: A new genetic algorithm for the quadratic assignment problem. INFORMS J. Com. to appear 2003
22. Drezner, Z., Hahn, P.M., Taillard, É.D.: A study of quadratic assignment problem instances that are difficult for meta-heuristic methods. Technical report, Systems Engineering, University of Pennsylvania, 2002
23. Eschermann, B., Wunderlich, H.-J.: Optimized synthesis of self-testable finite state machines. In: Proceedings of the 20th International Symposium on Fault-Tolerant Computing, Newcastle upon Tyne, 1990
24. Finke, G., Burkard, R.E., Rendl, F.: Quadratic assignment problems. Ann. Disc. Math. **31**, 61–82 (1987)
25. Gambardella, L.M., Taillard, É.D., Dorigo M.: Ant colonies for the quadratic assignment problem. J. Oper. Res. Soc. **50**, 167–176 (1999)
26. Gilmore, P.C.: Optimal and suboptimal algorithms for the quadratic assignment problem. SIAM J. Appl. Math. **10**, 305–313 (1962)
27. Goux, J.-P., Kulkarni, S., Linderoth, J., Yoder, M.: An enabling framework for master-worker applications on the computational grid. In: Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, 2000
28. Hadley, S.W., Rendl, F., Wolkowicz, H.: A new lower bound via projection for the quadratic assignment problem. Math. Oper. Res. **17**, 727–739 (1992)
29. Hahn, PM.: Private communication, 2003
30. Hahn, P.M., Grant, T.: Lower Bounds for the quadratic assignment problem based upon a dual formulation. Oper. Res. **46**, 912–922 (1998)
31. Hahn, P.M., Grant, T., Hall, N.: A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method. Eur. J. Oper. Res. **108**, 629–640 (1998)
32. Hahn, P.M., Hightower, W.L., Johnson, T.A., Guignard-Spielberg, M., Roucairol, C.: A level-2 reformulation-linearization technique bound for the quadratic assignment problem. Working paper, Systems Engineering, University of Pennsylvania, 2001
33. Hahn, P.M., Hightower, W.L., Johnson, T.A., Guignard-Spielberg, M., Roucairol, C.: Tree elaboration strategies in branch and bound algorithms for solving the quadratic assignment problem. Yugoslav J. Oper. Res. **11**, 41–60 (2001)
34. Hahn, P.M., Krarup, J.: A hospital facility location problem finally solved. J. Intelligent Manufacturing **5**(6), 487–496 (2001)
35. Hubert, L.J.: Assignment Methods in Combinatorial Data Analysis. Marcel Dekker, 1987
36. Jünger, M., Kaibel, V.: On the SQAP-polytope. SIAM J. Optim. **11**, 444–468 (2001)
37. Jünger, M., Kaibel, V.: The QAP-polytope and the star-transformation. Disc. Appl. Math. **111**, 283–306 (2001)
38. Kaibel, V.: Polyhedral combinatorics of QAPs with less objects than locations In: Bixby, RE., Boyd, EA., Ríos-Mercado RZ., eds. Integer Programming and Combinatorial Optimization. Volume **1412** of Lecture Notes in Computer Science. Springer, 1998, pp. 409–422
39. Kaibel, V.: Polyhedral methods for the QAP. In: Pardalos, PM., Pitsoulis, L. eds., Nonlinear Assignment Problems: Algorithms and Applications. Kluwer, 2000, pp. 109–141
40. Karisch, S.E., Çela, E., Clausen, J., Espersen, T.: A dual framework for lower bounds of the quadratic assignment problem based on linearization. Com. **63**, 351–403 (1999)
41. Karisch, S.E., Rendl, F.: Lower bounds for the quadratic assignment problem via triangle decompositions. Math. Prog. **71**, 137–152 (1995)
42. Krarup, J., Pruzan, P.M.: Computer-aided layout design. Math. Prog. Study **9**, 75–94 (1978)
43. Lawler, E.L.: The quadratic assignment problem. Mgmt. Sci. **9**, 586–599 (1963)
44. Li, Y., Pardalos, P.M., Resende, M.G.C.: A greedy randomized adaptive search procedure for the quadratic assignment problem. In: Pardalos, PM., Wolkowicz, H. eds., Quadratic Assignment and Related Problems. Volume **16** of DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1994, pp. 237–261
45. Lin, C.-J., Saigal, R.: On solving large-scale semidefinite programming problems – a case study of quadratic assignment problem. Technical report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor MI, 1997
46. Linderoth, J.T., Savelsbergh, M.W.P.: A computational study of branch and bound search strategies for mixed integer programming INFORMS J. Com. **11**, 173–187 (1999)
47. Livny, M., Basney, J., Raman, R., Tannenbaum, T.: Mechanisms for high throughput computing. SPEEDUP J. **11**(1) (1997)
48. Margot, F.: Exploiting orbits in symmetric ILP. Math. Prog. to appear 2003

49. Marzetta, A., Brüngger, A.: A dynamic-programming bound for the quadratic assignment problem. In: Computing and Combinatorics 5th Annual International Conference-COCOON'99 Volume **1627** of Lecture Notes in Computer Science. Springer, 1999, pp. 339–348
50. Mautor, T., Roucairol, CA.: New exact algorithm for the solution of quadratic assignment problems. Dis. App. Math. **55**, 281–293 (1994)
51. Nugent, C.E., Vollman, T.E., Ruml, J.: An experimental comparison of techniques for the assignment of facilities to locations. Oper. Res. **16**, 150–173 (1968)
52. Nyström, M.: Solving certain large instances of the quadratic assignment problem: Steinberg's examples. Department of Computer Science, California Institute of Technology Pasadena CA, 1999
53. Padberg, M.W., Rijal, M.P.: Location, Scheduling, Design and Integer Programming. Kluwer, 1996
54. Pardalos, P.M., Rendl, F., Wolkowicz, H.: The Quadratic assignment problem: A survey and recent developments. In: Pardalos, PM., Wolkowicz, H. eds., Quadratic Assignment and Related Problems, Volume **16** of DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1994, pp. 1–42
55. Pardalos, P.M., Ramakrishnan, K.G., Resende, M.G.C., Li, Y.: Implementation of a variance reduction-based lower bound in a branch-and-bound algorithm for the quadratic assignment problem. SIAM J. Optim. **7**, 280–294 (1997)
56. Ramachandran, K.G., Resende, M.G.C., Ramachandran, B., Pekny, J.F.: Tight QAP bounds via linear programming. In: Migdalas, A., Pardalos, PM., eds., Combinatorial and Global Optimization. Kluwer, 2001
57. Ramachandran, R., Pekny, J.F.: Dynamic factorization methods for using formulations derived from higher order lifting techniques in the solution of the quadratic assignment problem. In: State of the Art in Global Optimization: Computational Methods and Applications. Kluwer, 1996, pp. 75–92
58. Rendl, F., Wolkowicz, H.: Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. Math. Prog. **53**, 63–78 (1992)
59. Resende, M.G.C., Ramakrishnan, K.G., Drezner Z.: Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming. Oper. Res. **43**, 781–791 (1995)
60. Sherali, H.D., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM. J. Disc. Math. **3**, 411–430 (1990)
61. Skorin-Kapov, J.: Tabu search applied to the quadratic assignment problem. ORSA. J. Com. **2**, 33–45 (1990)
62. Sotirov, R., Rendl, F.: Semidefinite relaxations for the quadratic assignment problem. Presentation, SIAM Optimization Conference, Toronto, Canada, May 2002
63. Steinberg, L.: The backboard wiring problem: A placement algorithm. SIAM Rev. **3**, 37–50 (1961)
64. Taillard, É.D.: Robust tabu search for the quadratic assignment problem. Parallel Com. **17**, 443–455 (1991)
65. Taillard, É.D.: Comparison of iterative searches for the quadratic assignment problem. Location Sci. **3**, 87–105 (1995)
66. Tate, D.D., Smith, A.E.: A genetic approach to the quadratic assignment problem. Com. Oper. Res. **1**, 855–865 (1995)
67. Thonemann, U.W., Bölte, A.: An improved simulating annealing algorithm for the quadratic assignment problem. Working paper, School of Business, Dept. of Production and Operations Research, University of Paderborn, Germany, 1994
68. Zhao, Q., Karisch, S.E., Rendl, F., Wolkowicz H.: Semidefinite programming relaxations for the quadratic assignment problem. J. Combin. Opt. **2**, 71–109 (1998)