Exact Solution of Emerging Quadratic Assignment Problems

Peter M. Hahn*, Yi-Rong Zhu† Monique Guignard‡ J. MacGregor Smith§

hahn@seas.upenn.edu yrzhu@seas.upenn.edu guignard_monique@yahoo.fr jmsmith@ecs.umass.edu

July 31, 2009

Abstract — We report on a growing class of assignment problems that are increasingly of interest and very challenging in terms of the difficulty they pose to attempts at exact solution. These problems address economic issues in the location and design of factories, hospitals, depots, transportation hubs and military bases. Others involve improvements in communication network design. In this article we survey the latest and best methods available for solving exactly these difficult problems and suggest a taxonomy that provides a framework for combining existing solution methods and sets of computer tools that can be modified and extended to make inroads in solving this growing class of optimization problems.

Keywords — quadratic assignment, integer programming, reformulation linearization

Acknowledgements

This material is based upon work supported by the U.S. National Science Foundation under grant No. DMI-0400155. The authors are grateful to Professor Miguel Anjos of Waterloo University for reviewing this survey article, thus adding to its clarity and completeness.

1 INTRODUCTION

The Quadratic Assignment Problem (QAP) is one in which $N$ units have to be assigned to $N$ sites in such a way that the cost of the assignment, depending on the distances between the sites and the flows between the units, is minimal.

It can be formulated as follows: Given two $N \times N$ matrices, $F = [f_{ik}]$ with $f_{ik}$ the flow between units $i$ and $k$, and $D = [d_{jn}]$ with $d_{jn}$ the distance between sites $j$ and $n$, find a permutation $p$ of the set $S = \{1, 2, ..., N\}$ which minimizes the global cost function, $\text{Cost} (p) = \sum_{i=1}^{N} \sum_{k=1}^{N} f_{ik} d_{p(i)p(k)}$. The QAP is one of the most difficult $NP$-hard combinatorial optimization problems. Solving general problems of size greater than 30 (i.e., with more than 900$(0-1)$ variables) is still computationally challenging. If among exact algorithms, branch-and-bound are the most successful ones, the lack of a sharp lower bound technique in these algorithms is one of the major difficulties. The fact that the QAP is $NP$-hard is not sufficient to explain its difficulty, as we can now solve exactly very large instances of a great number of $NP$-hard problems. The homogeneity of the values of the solutions for most of the applications, due to the structure of the problem (scalar product of the two matrices) is a more convincing explanation. Indeed, first, we have many solutions whose value is close to the optimum. So, even when the best solution is obtained, it is very hard to prove its optimality. Then, fixing one assignment has a low influence on the average value of the solutions. Even when traversing the branch-and-bound tree, the problem remains very hard. Moreover, it is difficult to prune branches that contain significantly large numbers of non-optimal feasible solutions. A recent paper [8] by Barvinok and Stephen gives some insights into to the difficulty of solving the QAP. They obtain a number of interesting results regarding the distribution of objective function values on typical and specific QAP instances.

2 CURRENT STATUS OF QAP SOLVERS

While great progress has been made on generating good solutions to large and difficult QAP instances, this has not been the case for finding exact solutions. In the late 1960s, it was an achievement to find the optimum solution to a difficult instance of size $n = 8$. In the 1970’s and 80’s, one could only expect to solve difficult instances for $n < 16$. It was not until the mid-1990s that Clausen and Perregaard [16] were able to enumerate a difficult size 20 instance. Much progress has been made since then. In the mid 1960’s, Nugent, Vollmann and Ruml [46] posed a set of problem instances of size $\{5, 6, 7, 8, 12, 15, 20, \mbox{ and } 30\}$, noted for...
their difficulty. In these instances, the distance matrix stems from an $n_1 \times n_2$ grid with Manhattan distance between grid points. Most of the resulting QAP instances have multiple global optima (at least four if $n_1 \neq n_2$ and at least eight if $n_1 = n_2$). Even worse, these globally optimal solutions are at the maximally possible distance from other globally optimal solutions. The Nugent instances have been the benchmark, against which exact and heuristic solution algorithms have been measured. Figure 1 shows the rapid progress made in exact solution QAP algorithms from 1995 until early in the 21st century.

Other forms of the QAP do not have that flow/distance cost structure. One example is in balancing hydraulic turbine runners ([35], [44] and [51], for instance). A jet engine consists of several turbines, and the objective of this engine maintenance problem is to remove unwanted vibrations. This can be formulated as a QAP, where the 0/1 decision variable $x_{ij}$ is 1 if blade $i$ is allocated to position $j$, 0 otherwise, and the quadratic objective function corresponds to minimizing the distance between the center of gravity of the turbine shaft and its center of rotation.

The QAP, while still of great interest to researchers, is only one of a growing class of assignment problems that are increasingly of interest and even more challenging in terms of the difficulty they pose to attempts at exact solution. It is this class of problems that we address in this survey.

Figure 2 shows the relationships between exact solution methods for several assignment problems that have appeared in the operations research literature. These problems include the Generalized Quadratic Assignment Problem (GQAP), the 3-dimensional Assignment Problem (3AP), the Quadratic 3-dimensional Assignment Problem (Q3AP), the Generalized 3AP (G3AP) and the Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP) and Stochastic Quadratic Assignment Problem (SQAP). A short discussion of these problems is given in the ensuing paragraphs.

3 The Generalized Quadratic Assignment Problem (GQAP)

The GQAP covers a much broader class of problems than the QAP. Problems in this class involve the minimization of a total pairwise interaction cost among $M$ departments, equipment, tasks or other entities, and where placement of these entities into $N$ possible destinations is dependent upon existing resource capacities at each destination. These problems include finding the assignment of departments to fixed locations given limited area capacities at each possible location. The Lee and Ma [37] version of the problem can be stated with reference to a practical situation where it is desired to locate $M$ departments among $N$ fixed locations, where for each pair of departments $i, k$ a certain traffic flow of commodities $f_{ik}$ is known and for each pair of locations $j, n$ a corresponding distance $d_{jn}$ is known. The two-way transportation costs between departments $i$ and $k$, given that $i$ is assigned to location $j$ and $k$ is assigned to location $n$, are $f_{ik}d_{jn} + f_{kj}d_{nj}$. The objective is to find an assignment minimizing the sum of all such transportation costs given that the capacity or resource constraints are met. In the general case of the GQAP, the cost of transportation between departments is known but is not decomposable into a product of a flow and a distance matrix.

Lee and Ma [37] only recently formulated the GQAP. However, problems that are special cases, including the QAP, have long been of interest to researchers in various fields, both because of their wide applicability and their resistance to reliable computer solution. Problems which come under the class of GQAP include the Process Allocation Problem of Sofianopoulos ([63] and [64]), the Constrained Module Allocation Problem of Eloumi et al. [19], the Quadratic Semi-Assignment Prob-
problem covered by Billionnet and Elloumi [9], the Multiprocessor Assignment Problem by Magirou and Milis [43], the Task Assignment and Multiway Cut Problems of Magirou [42], the Memory Constrained Allocation problem of Roupin [55] and the constrained Task Assignment Problem of Billionnet and Elloumi [10].

Exact solution strategies for GQAP type problems have been successful for only small instances (approximately $M = 30$). As a result, researchers have put forth a significant amount of effort in developing exact, or heuristic methods that obtain good suboptimal assignments, using a reasonable amount of CPU time. Cordeau et al. [17], discusses a memetic heuristic for the GQAP. They do not report using their method to find exact solutions. Lee and Ma [37] were the first to devise an exact solution method for the GQAP. They introduced three linearization approaches together with a branch-and-bound algorithm. Their lower bounding strategy involves solving $M \times N$ plus 1 GAP sub-problems. This they did using calls to CPLEX. Using this bound in a branch-and-bound algorithm, they were able to pose and solve exactly 27 problem instances, the largest of which was of size 16x7.

Hahn et al. [29] improved upon the exact solution method of Lee and Ma by introducing a Lagrangean dual for the GQAP based on a Level-1 Reformulation Linearization Technique (RLT-1) Dual Ascent Procedure similar to one they successfully used for solving the Quadratic Assignment Problem (QAP). A unique and very important aspect of the RLT-1 Dual Ascent Procedure is that at each stage, the GQAP is restructured as fully equivalent to the original QAP in a manner that brings it closer to solution. Their RLT-1 Dual Ascent Procedure was embedded in a branch-and-bound algorithm that is unique in many respects. A number of test instances, selected from a web site dedicated to the Task Assignment Problem (TAP) and the Constrained Task Assignment Problem (CTAP) set up by Sourour Elloumi at the Centre de Recherche en Informatique du CNAM [20], were solved in record time. Comparisons were also made with instances of the QAP devised by Cordeau, et al. [17] and Lee and Ma [37]. The B-and-B of Hahn et al. is generally faster than the method of Lee and Ma and is about 20 times faster than the Lee and Ma runtime for the difficult 16x7 instance.

Pessoa et al. [47] provide the most recent and most promising solution methods for the GQAP. They developed a hybrid branch-and-bound exact solution method. Their lower bound calculation is based on a Lagrangean relaxation that makes efficient use of the integer linearization property in its modeling phase and of the volume algorithm in its solution phase. Pessoa et al.’s branch and bound is as fast or faster than other exact solution methods on easy GQAP problem instances, and is remarkable in that it solves difficult instances that could not be solved exactly with any previous solvers. Their look-ahead branching strategy is based on the same techniques as found in [29]. Thus, the improved performance reported is due entirely to the effectiveness of the their new lower bound.

4 THREE-DIMENSIONAL ASSIGNMENT PROBLEMS

4.1 The 3-dimensional Assignment Problem

The 3-dimensional Assignment Problem (3AP), also known as the Three Index Assignment Problem, involves the optimization of the assignment of $N$ type-1 entities and simultaneously $N$ type-2 entities to $N$ destinations. The 3AP is applied to find the minimal idling time of a rolling mill, optimal location of production plants in regions, optimal number of satellites in different directions and orbits for maximization of the scanned regions [50], teaching schedules [21], in statistical processing of measurement results [32], etc. The 3AP is shown in [66] to be a special case of the QAP and thus is difficult and also $NP$-hard.

Branch and bound is the preferred method for solving the 3AP exactly. One of the first 3AP branch-and-bound algorithms was proposed by Pierskalla [49]. Bound- ing techniques using Lagrangian and subgradient optimization were proposed by Burkard and Rudolf [15]. Balas and Saltzman [7] improved on existing bounding techniques by using dual heuristics. Among exact algorithms, branch-and-bound schemes using the Lagrangian dual and subgradient optimization are the most successful, but the computation time for the subgradient procedure has been one of the major difficulties. An average 65 percent of the total computation time for branch-and-bound enumeration is spent in the subgradient solution procedure [7].

Kim et al. [34] describe new bounding methods for the axial three-index assignment problem (3AP). For calculating 3AP lower bounds, they use a projection method followed by a Hungarian algorithm, based on a new Lagrangian relaxation. They also use a cost transformation scheme, which iteratively transforms 3AP costs in a series of equivalent 3APs, which provides the possibility of improving the 3AP lower bound. Their methods produce efficiently computed relatively tight lower bounds on standard test instances.

4.2 The Quadratic 3-dimensional Assignment Problem

Pierskalla [48] introduced the Quadratic 3-dimensional Assignment Problem (Q3AP) in a technical memorandum. Since then, little on the subject has appeared. Hahn et al. [28] re-discovered the Q3AP while working on a problem arising in data transmission system design. The Q3AP is an extension of two $NP$-hard problems, the QAP and the 3AP. Thus, it is easy to see that the Q3AP is also $NP$-hard. The interest in the Q3AP stems from the fact that it is applied to problems where the objective is to minimize linear and quadratic costs associated with a pair of independent simultaneous one-to-one assignments. Such a problem arises in the design of wireless communication systems, wherein
a digital message is repeated two times. During each of the repeats, the assignment of data word to transmitted symbols is modified. The Q3AP models the problem of optimizing the two assignments in such a way that the transmission errors are minimized. See [56] and [57].

Hahn et al. [28] are the first to have solved Q3AP instances. They developed a branch-and-bound algorithm based upon one of the best techniques available for solving the QAP exactly, as well as four different heuristic solution methods whose genesis came from previous work applied to solving the QAP. Implementing the exact algorithm required the development of new lower bounds for the 3AP. Although the computational results are encouraging, they also illustrate the level of difficulty associated with the Q3AP. Recently, Galea et al. [22] developed a parallel version of the exact solution algorithm of Hahn et al. [28]. This parallel code is not only an instrument for solving exactly large instances, but will also enable experimentation for improving the runtime of Q3AP exact solution algorithms.

Presently, the exact solutions have been demonstrated only for Q3AP instances of size 14 or smaller. Parallel solution experiments are planned for larger instances. Stochastic local search (SLS) techniques [28] are essential for reaching high quality solutions to Q3AP instances of practical interest. Clearly, much more work is needed on this challenging and yet important new combinatorial optimization problem.

4.3 The Generalized Quadratic 3-dimensional Assignment Problem

The Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP) is a generalization of the Q3AP and the QAP. This problem arises in two very important situations. One is the assignment of spaces within multi-story buildings or within multi-deck naval vessels, so that the movement of people and materials between spaces is efficient and that the time to escape from the structure is simultaneously minimized. This problem is known as the Multi-story Space Assignment Problem [30]. MSAP test instances are currently available at http://www.seas.upenn.edu/~msaplib. MSAP test instances are also available at the Facility Layout Problem Library (FLPlib) http://FLPlib.uwaterloo.ca/. FLPlib was developed at Waterloo University by Professor Miguel Anjos and student Christie Kong. This web site serves as a resource of data for developing facility layout problems and solution methods. In addition to being an MSAP resource, FLPlib contains information and problem instances on the QGAP, the Single-row Facility Layout problem (SRFL) and the One-Dimensional Space Allocation Problem (ODSAP), also known as the linear single-row facility layout problem, which consists of finding an optimal linear placement of facilities with varying dimensions on a straight line.

Another application of the GQ3AP is in the design of cross-dock facilities in the less-than-full load (LTL) trucking business. In this situation, the GQ3AP is used to assign incoming trucks to unloading docks (strip doors) and simultaneously assigning outgoing trucks to shipping docks (stack doors) so that the cost of moving goods from strip doors to stack doors is minimized. This problem is known as the Cross-dock Door Assignment Problem (CDAP). Zhu, et al. [67] recently report on an exact algorithm for solving the CDAP as a GQ3AP.

5 Other Problems

5.1 Cubic Assignment Problem

In the Quadratic Assignment class, the Cubic Assignment Problem (CAP) is described in a newly published SIAM Monograph by Burkard, Dell’Amico and Martello [14]. Their book provides a comprehensive treatment of assignment problems from their conceptual beginnings in the 1920s, through present-day theoretical, algorithmic, and practical developments. The CAP is also mentioned, but not discussed in detail in the book by Du and Pardalos [18]. The CAP was first posed by Lawler in his seminal paper on the QAP [36]. The CAP optimizes the problem of placing $N$ entities at exactly $N$ destinations, where the cost of placement involves the interaction between triplets of entities, rather than the interaction between pairs of entities as is found in the QAP. We have searched, but have not found an exact solution method for the CAP. But, in [1], it is stated that the formulation RLT-2 of the QAP is exactly a CAP. Thus, the RLT-2 exact solution solver for the QAP is capable of solving the CAP exactly. To the best of our knowledge, no researchers have yet reported computational experience on solving the CAP using the RLT-2 form, though the RLT theory establishes the theoretical equivalence between these representations. Winter and Zimmermann [65] used a cubic assignment problem for optimizing the movement of materials in a storage yard. Burkard et. al in [14] point out that “the objective function (2.1.1) in [65] contains some typos in the indices, but is actually the objective function of a cubic assignment problem”.

5.2 Bi-Quadratic Assignment Problem

The BiQuadratic or Quartic Assignment Problem (BiQAP) is a generalization of the QAP. It was also posed in 1963 by Lawler [36]. It is a nonlinear integer-programming (IP) problem where the objective function is a fourth degree multivariable polynomial and the feasible domain is the assignment polytope. BiQAP problems have an application in VLSI synthesis, where programmable logic arrays have to be implemented. Due to the difficulty of this problem, only heuristic solution approaches have been proposed. For details of the VLSI application see Burkard, Çela and Klinz [13], who studied biquadratic assignment problems, derived
lower bounds and investigated the asymptotic probabilistic behavior of these problems. Burkard and Čela [12] developed metaheuristics for the BiQAP and compared their computational performance.

5.3 Generalized Cubic Assignment Problem

The Generalized Cubic Assignment Problem (GCAP) is a generalization of the Cubic Assignment Problem and the Generalized QAP. The GCAP optimizes a situation where \( M \) entities have to be placed at \( N \) destinations, such that the placement of entities at each possible location is limited by the capacity of the destination to accept entities, where the cost of placement involves the interaction between triplets of entities, rather than the interaction between pairs of entities as is found in the QAP. This problem is introduced in Zhu [66] and a solution method is suggested. However no papers have been published on this subject. For this problem, one would have to generate potential test instances, as none exist.

5.4 Stochastic Quadratic Assignment Problem

Li and Smith ([38], [39], [40], [62]) have developed a formulation and associated heuristic solution algorithms for the Stochastic Quadratic Assignment Problem (SQAP) which involves the examination of random flows in a facility layout. The random flows are modelled with an additional node in the layout to accommodate the dynamic flows of customers or products in the layout where congestion occurs. These models have many applications in facility planning, manufacturing systems and other QAP problems with dynamic flows. Traditionally, these models are solved with heuristics, so it would be worthwhile for someone to solve them exactly. While the objective function in the SQAP problems are nonlinear, there are lower bounds available from some of the queueing models that will effectuate the exact solution of these problems. See the latest Smith and Li paper for further details [62].

5.5 Summary

Table 1 and Table 2 summarize the achievements made in solving the various assignment problems considered in this survey. The tables give the applications for which the problems were originally posed and list the problem sizes that can be solved exactly, the method of solution and the publication describing the solution method.

6 Descriptions of Assignment Problems

Until now, assignment problems have been dealt with independently. Little attention was given to developing a unified approach. This survey emphasizes the common structures and suggests a common framework for dealing with these very difficult problems. Since the approach that has been most successful for solving assignment problems exactly is to develop tight lower bounds based upon Lagrangean relaxations, we divide the individual problems by the following descriptive characteristics:

- Dimensionality of the objective function: (2-dimensional, 3-dimensional, etc.)
- Degree of the objective function: (linear, quadratic, cubic, bi-quadratic, etc.)
- One-to-one versus many-to-one assignment: (AP versus GAP)
- Linearization and relaxation options (level of Reformulation Linearization)

6.1 2-dimensional Assignment Problems

6.1.1 The Linear Assignment Problem

The LAP is given by:

\[
\min \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij} x_{ij} \right\} \tag{1}
\]

subject to the following constraints on \( X \):

\[
\sum_{i=1}^{N} x_{ij} = 1 \ (j = 1, 2, \ldots, N), \tag{2}
\]

\[
\sum_{j=1}^{N} x_{ij} = 1 \ (i = 1, 2, \ldots, N), \tag{3}
\]

\[
x_{ij} = 0, 1 (i = 1, 2, \ldots, N; j = 1, 2, \ldots, N). \tag{4}
\]

The LAP is an easy problem, even though it has \( N! \) feasible solutions, as are found in the QAP. It is easy, not so much because the objective function is linear, but especially because the LP relaxation optimizes precisely over the convex hull of the feasible 0-1 integer points of the solution space. Figure 3 shows a typical LAP solution, superimposed on the square objective function cost matrix of a size 5 LAP. The solution shown is optimal.

\[
\begin{array}{cccc}
12 & 8 & 7 & 15 \quad 4 \\
7 & 9 & 17 & 14 \quad 10 \\
9 & 6 & 12 & 6 \quad 7 \\
7 & 6 & 14 & 6 \quad 10 \\
9 & 6 & 12 & 10 \quad 10
\end{array}
\]

Figure 3: Typical LAP Solution
### Table 1: 2-dimensional Exact Solution Methods

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Application</th>
<th>Size</th>
<th>Method</th>
<th>Who/When</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAP</td>
<td>Assign jobs to machines</td>
<td>&gt;2000</td>
<td>Sparse instances</td>
<td>Goldberg-Kennedy 1995 [23]</td>
</tr>
<tr>
<td>QAP</td>
<td>Facility loc./Ckt. layout</td>
<td>&gt;30</td>
<td>Quadratic Prog.</td>
<td>Jonker-Volgenant 1987 [33]</td>
</tr>
<tr>
<td>BQAP</td>
<td>Solve QAP-RLT-3</td>
<td>200x5</td>
<td>Branch and Bound</td>
<td>Haddadi-Ouzia 2004 [26]</td>
</tr>
<tr>
<td>GAP</td>
<td>Assign jobs to machines</td>
<td>100x5</td>
<td>Branch and Cut</td>
<td>Nauss 2003 [45]</td>
</tr>
<tr>
<td>GQAP</td>
<td>Assign tasks to processors</td>
<td>16x7</td>
<td>GLAP-LB</td>
<td>Lee-Ma 2004 [37]</td>
</tr>
<tr>
<td>GCAP</td>
<td>Solve QAP-RLT-2</td>
<td>30x20</td>
<td>GQAP-RLT-1</td>
<td>Hahn et al. 2006 [29]</td>
</tr>
<tr>
<td>SQAP</td>
<td>Random Flows in a layout</td>
<td>24x8</td>
<td>GQAP-RLT-2</td>
<td>Hahn et al., unpublished</td>
</tr>
</tbody>
</table>

### Table 2: 3-dimensional Exact Solution Methods

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Application</th>
<th>Size</th>
<th>Method</th>
<th>Who/When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3AP</td>
<td>Symbol Mapping Diversity</td>
<td>14</td>
<td>Q3AP Branch and Bound</td>
<td>Hahn et al., 2006 [28]</td>
</tr>
<tr>
<td>C3AP</td>
<td>Solving the Q3AP</td>
<td>8</td>
<td>RLT-2 Q3AP Branch and Bound</td>
<td>Hahn et al., 2007, unpublished</td>
</tr>
<tr>
<td>G3AP</td>
<td>Subproblem of GQ3AP</td>
<td>TBD</td>
<td>Volume Algorithm</td>
<td>Hahn et al. (in progress)</td>
</tr>
<tr>
<td>GQ3AP</td>
<td>Solve MSAP and CDAP</td>
<td>17x17x4</td>
<td>GQ3AP Branch and Bound</td>
<td>Hahn et al. [30]</td>
</tr>
</tbody>
</table>

### 6.1.2 The Quadratic Assignment Problem

The QAP is given by:

$$
\text{Min} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij}x_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{n=1}^{N} C_{ijn}x_{ij}x_{kn} \right\}
$$

subject to the following constraints on X:

$$
\sum_{i=1}^{N} x_{ij} = 1 \ (j = 1, 2, \ldots, N)
$$

$$
\sum_{j=1}^{N} x_{ij} = 1 \ (i = 1, 2, \ldots, N)
$$

$$
x_{ij} = 0, 1 (i = 1, 2, \ldots, N; j = 1, 2, \ldots, N),
$$

X are said to be a ‘solution’. Figure 4 shows a typical feasible QAP solution, superimposed on the 9x9 objective function cost matrix of a size 3 QAP. Certain elements of the objective function cost matrix are designated by asterisks, indicating that they cannot contribute to any feasible solution. There are $N!$ feasible solutions to the QAP.

### 6.1.3 The Cubic Assignment Problem

The CAP is given by:

$$
\text{Min} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij}x_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{n=1}^{N} C_{ijn}x_{ij}x_{kn} \\
+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} D_{ijklmn}x_{ij}x_{kl}x_{mn} \right\}
$$

subject to the following constraints on X:

$$
\sum_{i=1}^{N} x_{ij} = 1 \ (j = 1, 2, \ldots, N)
$$

$$
\sum_{j=1}^{N} x_{ij} = 1 \ (i = 1, 2, \ldots, N)
$$
\[ x_{ij} = 0, 1 (i = 1, 2, \ldots, N; j = 1, 2, \ldots, N), \quad (12) \]

Figure 5 shows a typical feasible CAP solution, superimposed on the 64x64 objective function cost matrix of a size 4 CAP. The objective function cost matrix has 16x16 submatrices, each with 4x4 cost elements. Only 16 of these submatrices are involved in a feasible solution. Within a solution submatrix only four of the sixteen elements are in the feasible solution. Again, certain elements in the objective function cost matrix are disallowed from any feasible solution. As before, these elements are designated by asterisks. As in the LAP and QAP, there are \( N! \) feasible solutions to the CAP.

RLT has been used to achieve significant advances in the solvability of classical and newly posed assignment problems (APs) and generalized assignment problems (GAPs). Applying RLT to Quadratic Assignment Problems was first done by Adams and Johnson [2]. QAP lower bounds for a level-1 RLT (RLT-1) formulation were first calculated by Adams and Johnson [2] and Resende et al. [54] and for a level-2 RLT (RLT-2) formulation were first calculated by Ramakrishnan et al. [53].

Appearing in the literature more recently are the RLT-1 QAP exact algorithm by Hahn et al. [27], the RLT-2 QAP exact algorithm by Adams et al. [1], the RLT-3 QAP lower bound calculations by Hahn et al. [31], the RLT-1 exact algorithm for the QAP by Hahn et al. [28] and the RLT-1 parallel exact QAP solver by Galea et al. [22] and the RLT-1 exact algorithms for the GQAP by Hahn et al. [29] and by Pessoa et al. [47].

Reformulation-Linearization Techniques achieve significant advances in the solvability of the QAP. Problem RLT-2 for the QAP, in particular, provides sharp lower bounds and consequently leads to very competitive exact solution approaches [1]. A striking outcome, documented in Table 2 of Loiola et al. [41], is the relatively few nodes considered in the binary search tree to verify optimality. This leads to marked success in solving difficult QAP instances of size 30 in record computational time. Hahn et al. [31] used the level-3 RLT in order to get even tighter bounds. The challenge was to take advantage of the additional strength, without being hurt by the substantial increment in problem dimensions.

In preparing this survey, it became clear to us that the RLT formulations connect all these problems into a taxonomy, wherein connections are made between problem types, objective function degree, problem dimension and whether mappings are one-to-one or many-to-one. We illustrate one aspect of this taxonomy by considering an artificial example, namely the RLT-2 formulation for the Linear Assignment Problem. There is, of course, no reason for applying RLT to the LAP, since it is a special case of a linear program and relatively easy to solve [14].
The RLT-2 for the LAP is given by:

\[
M \text{in } \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij} x_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{n=1}^{N} C_{ijkn} y_{ijkn} \right\} \quad (13)
\]

subject to the following constraints on \( X \) and \( Y \):

\[
\sum_{i=1}^{N} y_{ijkn} = x_{kn} (j, k, n = 1, 2, \ldots, N), j \neq n
\]

(14)

\[
\sum_{j=1}^{N} y_{ijkn} = x_{kn} (i, k, n = 1, 2, \ldots, N), i \neq k
\]

(15)

\[
y_{ijkn} = y_{knij} (i, j, k, n = 1, 2, \ldots, N), i < k
\]

(16)

\[
y_{ijkn} \geq 0 (i, j, k, n = 1, 2, \ldots, N), i \neq k
\]

(17)

\[
\sum_{i=1}^{N} x_{ij} = 1 (j = 1, 2, \ldots, N)
\]

(18)

\[
\sum_{j=1}^{N} x_{ij} = 1 (i = 1, 2, \ldots, N)
\]

(19)

\[
x_{ij} = 0, 1 (i = 1, 2, \ldots, N; j = 1, 2, \ldots, N),
\]

(20)

We recognize this formulation to be a linearization of the QAP. Specifically, it is identical to formulation LP from Hahn and Grant [27]. This is not an accident. It is an important fact about the relationship between RLT representations of assignment problems and the degree of the assignment problem objective function. Namely, each level of RLT representation of an assignment problem, results in a representation of a higher degree assignment problem. This fact is further illustrated later in Figure 13. A more detailed discussion of the advantages of RLT in solving QAPs is given in §7.

Thus, the connection between assignment problem degree and RLT level has been introduced. In the next section we introduce the generalized (many-to-one) versions of the assignment problems discussed above and illustrate the similarities and differences with the one-to-one versions by displaying their solution spaces.

6.3 Generalized Assignment Problems

6.3.1 Generalized (linear) Assignment Problem (GAP)

A GAP may be given by:

\[
M \text{in } \left\{ \sum_{i=1}^{M} \sum_{j=1}^{N} B_{ij} x_{ij} \right\}
\]

subject to the following constraints on \( X \):

\[
\sum_{i=1}^{M} a_{ij} x_{ij} \leq A_{j} (j = 1, 2, \ldots, N),
\]

(22)

\[
\sum_{j=1}^{N} x_{ij} = 1 (i = 1, 2, \ldots, M)
\]

(23)

\[
x_{ij} = 0, 1 (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N).
\]

(24)

where \( a_{ij} \) is a need associated with entity \( i \) and \( A_{j} \) is a resource associated with location \( j \), which limits the amount of \( i \) entities that can be assigned at location \( j \).Note that we have used a simplified definition of the GAP. In many representations the needs are more generally represented by \( a_{ij} \), which accommodates problem sets that involve needs that are dependent on location assignment.

Figure 6 shows a typical solution for the GAP, superimposed on the 7x4 objective function cost matrix of a 7x4 size GAP. In this problem, the feasible solution contains one element for every row of the objective function cost matrix. The amount of feasible solution elements in a column are dictated by the capacity constraints for that column. These are determined by the needs of each row and the resources allocated to each column. The number of feasible solutions is instance dependent.

![Figure 6: Typical GAP Solution](image)

6.3.2 Higher Degree Generalized Assignment Problems

The formulation of the GQAP is given in Lee and Ma [37], Hahn, et al. [29] and Pessoa et al. [47], so we do not re-introduce it here. But, we illustrate in Figure 7 the objective function cost matrix of a size 4x3 (4 facilities into 3 locations) GQAP superimposed with a typical feasible solution to this relatively easy to solve problem. We follow this in Figure 8 with an illustration of the objective function cost matrix of a 4x3 Generalized Cubic Assignment Problem (GCAP). We could detail the characteristics of the objective function cost matrices and superimposed solutions in the same detail as for the QAP and CAP. But, instead, we merely mention the fact that the
objective function cost matrices for the generalized assignment problems (GAPs) are similar to those for their related assignment problems (APs).

![Figure 7: Typical GQAP Solution](image1)

6.4 Three Dimensional Assignment Problem representations

6.4.1 3-dimensional Assignment Problems (3AP and Q3AP)

As mentioned earlier in §4.1 and §4.2, three-dimensional assignment problems involve the simultaneous and independent assignments of two different entities to a common location. It is important to be able to visualize the objective function space and the selection of those objective function coefficients that are summed for the objective function value of a specific problem instance. Thus, we have devised a means of drawing the three dimensional objective function cost space as an intersection of two two-dimensional objective function matrices. If the two two-dimensional objective function cost matrices are square, then the three-dimensional objective function cost matrix is a cube. This, in fact, is the case for the 3AP and the Q3AP. Superimposed on one side of the cube is the assignment pattern that constitutes a feasible solution for one of the independent assignments. Superimposed on another side is the assignment pattern that constitutes a feasible solution for the other independent assignment. The actual solution cost for the pair of feasible solution patterns is the sum of costs that reside inside the cube at the intersection of the two independent assignment patterns. The resulting representation of the three-dimensional objective function cost matrices and a typical feasible solution are shown in Figure 9 for the 3AP and in Figure 10 for the Q3AP. Figure 9 shows the inside of the 3AP cube, so that the actual feasible solution elements are identified. Whereas, in Figure 10 the inside of the cube is not shown. Instead, the two independent assignment patterns are projected on two faces of the cube. In the remaining description of objective function costs and problem solutions, we shall continue to use the latter illustrative technique.

![Figure 8: Typical GCAP Solution](image2)

6.4.2 Generalized 3-dimensional Assignment Problems

Here we introduce the Generalized 3-dimensional Assignment Problem. This problem arose as a subproblem of the GQ3AP. By studying exact solution methods for this linear problem, we hope to gain insights that will significantly improve our methods for solving the much more difficult quadratic version.

The Generalized 3-dimensional assignment problem is given by:

$$\min \left\{ \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{P} B_{ijp}x_{ij}y_{ip} \right\}$$  \hspace{1cm} (25)

subject to the following constraints on $X$:

$$\sum_{i=1}^{M} s_{ij}x_{ij} \leq S_j \hspace{0.5cm} (j = 1, 2, \cdots, N),$$  \hspace{1cm} (26)
Figure 9: Typical 3-dimensional Assignment Problem Solution

Figure 10: Typical Q3AP Solution

Figure 11: Typical G3AP solution

Figure 12: Typical GQ3AP solution

Now that we have introduced the family of quadratic assignment problems that are emerging, we point out the relationships between all these problem types. This defines the taxonomy of the models we have come across. Extrapolating from these relationships to include other models is intuitive. Consider Figure 13 which contains a table of all problem types reported on so far. The top half of the table contains 2-dimensional problem types and the bottom half 3-dimensional problem types. Problem types are connected with the subset symbol. The table tells us that the LAP may be solved by any algorithm for the following problems: LAP, QAP, CAP, BQAP, GAP, GQAP, GCAP or GBQAP. The table also tells us that the GQAP can be solved only by algorithms for the GQAP, GCAP and GBQAP, assuming that all of these algorithms exist. Of course, using an algo-
algorithm that is designed to solve a more complex problem, may or may not save computation time. Additionally, the more complex algorithms will certainly take more computational resources such as disk and RAM.

<table>
<thead>
<tr>
<th>2-Dim</th>
<th>Linear</th>
<th>Quadratic</th>
<th>Cubic</th>
<th>Bi-quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAP</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
</tr>
<tr>
<td>GAP</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
</tr>
<tr>
<td>3-Dim</td>
<td>Linear</td>
<td>Quadratic</td>
<td>Cubic</td>
<td>Bi-quadratic</td>
</tr>
<tr>
<td>3AP</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
</tr>
<tr>
<td>G3AP</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
<td>⊆</td>
</tr>
</tbody>
</table>

Shading indicates an exact solution method exists.

Figure 13: Relationships among various assignment problems

7 Promising Research

We describe here the promising areas of research that judiciously applied will not only allow significant advances in the art of solving the more complex versions of the Quadratic Assignment Problem, but will also improve the speed and efficiency with which the more familiar forms of the QAP can be solved.

7.1 Subgradient-Modification

The lower bound improvement provided by embedding a dual ascent method in a subgradient-modification algorithm, namely the volume algorithm, is a significant algorithmic progress for the GQAP (Pessoa, et al. [47]). Similar techniques should apply equally well to other difficult problems for which dual ascent steps permit the immediate improvement of the Lagrangean objective function value. Problems that could be addressed immediately are the 3-dimensional Assignment Problem (Zhu [66]), the Quadratic 3-dimensional Assignment Problem (Hahn et al. [28]) and the Generalized Quadratic 3-dimensional Assignment Problem (Zhu [66]).

7.2 Grid Computing

The University of Versailles PRISME Laboratory developed a parallel implementation of branch and bound solvers for the RLT-2 version of our QAP and the RLT-1 Q3AP algorithms. See [22]. These parallel implementations were built using the Bob++ library. See https://software.prism.uvsq.fr/bobpp/. Bob++ is an open-source framework for the implementation of parallel solvers based on tree-search algorithms such as branch-and-bound. Its supported parallel environments include POSIX threads for multi-core and multiprocessor SMP machines, and Kaapi for large scale computation grids.

For Bob++, PRISM implemented asynchronous fully distributed branch and bound algorithms. Their algorithms are asynchronous because they offer more efficient parallelization; allocation of work is not delayed while waiting for the next synchronization moment, and essential information, such as upper bound values, is distributed in a timely manner. The distributed approach for grid computing is essential in order to 1) reduce contention for the master processor and 2) reduce the impact of the computational grids poor and unpredictable communication properties. However, the distributed approach is complex, because it requires the implementation of a load balancing algorithm and a checkpoint mechanism to save not only the state of the computation of the master process, but of all the processes. There are also grid computing facilities in the U.S. These include the Condor Project at the University of Wisconsin (http://www.cs.wisc.edu/condor/) and the The NASA Advanced Supercomputing (NAS) Division located at the NASA Ames Research Center in Moffett Field, California. See http://www.nas.nasa.gov/.

7.3 Level-3 Reformulation-Linearization Technique

As mentioned in §6.2, Reformulation-Linearization Techniques achieve significant advances in the solvability of Quadratic Assignment Problems. Problem RLT-2, in particular, provides sharp lower bounds and consequently leads to very competitive exact solution approaches. A striking outcome, documented in Table 2 of Loiola et al. [41], is the relatively few nodes considered in the binary search tree to verify optimality. This leads to marked success in solving difficult QAP instances of size 30 in record computational time. Based on this success, Hahn et al. turned attention in [31] to the level-3 form in order to get even tighter bounds, knowing that we would have to pay a price for the increased model size. The challenge was to take advantage of the additional strength, without being hurt by the substantial increment in problem dimensions. This required novel computational steps, better adapted to the much larger increment size. They showed that, as for level 2, the level-3 RLT for the QAP can be handled via a Lagrangean approach to obtain subproblems with nested structure. This time, however, they had many more dualized constraints and decomposable subproblem blocks.

The work on RLT-3 for the QAP, shows promise but it also runs into severe limitations. The encouraging point made in [31] is that in all but three test instances, the RLT-3-based algorithm reached lower bounds sufficiently close to the optimal solution that the best known solution was confirmed as optimal. But, just because a lower bound is tight, does not mean that one can count...
on it to be useful in a branch-and-bound algorithm. The bound has to be calculated quickly. Fortunately, the dual ascent bounds that Hahn et al. developed for RLT-3 are calculated iteratively. The graph in Figure 1 of [31] shows the fraction of the Nug 22 optimum solution value that is reached by the RLT-3-based lower bound, as a function of runtime. Excellent lower bounds were reached in only a few minutes. To verify that RLT-3 bounds are indeed effective requires that they be tested in a branch-and-bound implementation.

A severe limitation is that the number of variables grows dramatically with RLT level. The RLT-2 branch-and-bound solver code already runs into memory limits of the current generation of computers for problem instances larger than \( N = 36 \). Memory limits of machines available to researchers today make it difficult, if not impossible to calculate RLT-3 lower bounds for problem instances larger than \( N = 25 \) using the current Fortran code. On the positive side, experiments have demonstrated promise for reducing the number of nodes that must be considered for proving optimality using branch-and-bound [1]. A potential approach for dealing with RLT-3 memory requirements is to distribute computation load to multiple CPUs, thus marshaling the RAM from several computers.

### 7.4 Convexification Methods

Another area of investigation tries convexification methods for quadratic optimization problems having at least some linear equality constraints. Convexification of nonconvex 0-1 integer programming problems has been done, for instance, using SDP [52] or techniques such as those proposed by Floudas and used by Zlobec [68]. One needs to identify at least one such technique that results in strong bounds for QAP-type problems, for the continuous or possibly some other relaxation. So far continuous bounds based on Plateau’s method have been disappointing for most nonconvex GQAPs. However the tools that have been developed jointly by our team and GAMS Dev Corp [58] can be transposed to other quadratic 0-1 problems, such as those studied by Plateau, and provide strong bounds.

Nonlinear convex 0-1 integer programs may have an objective function that is too convex, i.e., with a deep valley that produces lower than needed continuous lower bounds. As long as the alternate convex objective function chosen coincides with the original one at all integer feasible solutions, it produces a valid lower bound. Thus, this “de-”convexification can be applied to convex problems in order to improve the continuous relaxation bound, and this can be done using the tools mentioned above.

For convex models, the Convex Hull Relaxation (CHR) bound is the best among all Lagrangean-type bounds for a given objective function. One could investigate the possibility of generating strong lower bounds by combining some de-convexification of the objective function, and more general relaxation models than the continuous relaxation, for instance the CHR relaxation, for convex problems.

Current experiments ([6] and [25]) show that the CHR approach applied to nonconvex 0-1 quadratic problems is a very fast way of generating optimal or near-optimal solutions for several types of nonconvex 0-1 integer problems, such as quadratic knapsack problems and some assignment-type problems. The impact of adding more starting points to the CHR approach to increase the probability of reaching points close to the optimal 0-1 solution for these problems has been substantial.

### 8 Summary and Predictions

We have presented a survey of a rapidly growing class of 0-1 integer programming problems related to the Quadratic Assignment Problem. Not all of these problems have quadratic objective functions, but those that do not are closely related to those that do.

We reported the latest and best methods available for solving exactly these difficult problems and suggest a taxonomy that provides a framework for combining existing solution methods and sets of computer tools that can be modified and extended to make inroads in solving this growing class of optimization problems. The message we wish to impart is that the set of problem types we describe herein are closely related and what can be learned from the exact solution method for one of them carries over to most of the others. Moreover, a solution technique for the more complex of these problems automatically becomes a solution method for many of the simpler problems.

We predict that future research on this class of problems will include exact solution methods for the QAP and similar 0-1 integer problems that heretofore have been too large to solve exactly. For instance, solving a QAP of size 40 will be a practical everyday event. There is even hope that it will be possible to solve exactly problems of size 50 and larger with reasonable computational effort.

We predict that there will be increased interest in RLT techniques as a means of solving 0-1 mixed integer programs of all sorts, as well as in optimizing more general families of nonlinear, non-convex programs. The RLT process of recasting problems into higher-variable spaces can be generally applied when, for example, products of continuous variables arise. It has also been recently extended to nonlinear integer programs by Adams and Sherali [5] that can treat discrete variables that are not necessarily binary. Though the convex hull results differ in such cases, the basic constructs of reformulation and linearization continue to be applicable.

We hope that there will emerge a new trend in optimization that encourages the use of this heretofore underutilized technique.

Finally, we hope that there will be renewed interest in problems such as the QAP by the research community and its funding agencies. This is turn should help
to make it possible for developed as well as undeveloped countries to become more efficient in such areas as facilities planning, layout of communication and transportation networks and design of wireless communication devices. Similar benefits should accrue to various transportation networks and design of wireless communication facilities planning, layout of communication and transportation devices.

REFERENCES


