Assignment Problems*

Eranda Çela†

Abstract

Assignment problems arise in different situations where we have to find an optimal way to assign n objects to m other objects in an injective fashion. Depending on the objective we want to optimize, we obtain different problems ranging from linear assignment problems to quadratic and higher dimensional assignment problems.

The assignment problems are a well studied topic in combinatorial optimization. These problems find numerous application in production planning, telecommunication, VLSI design, economics, etc.

We introduce the basic problems classified into three groups: linear assignment problems, three and higher dimensional assignment problems, and quadratic assignment problems and problems related to it. For each group of problems we mention some applications, show some basic properties and describe briefly some of the most successful algorithms used to solve these problems.

1 Introduction

Assignment problems deal with the question how to assign n objects to m other objects in an injective fashion in the best possible way. An assignment problem is completely specified by its two components: the assignments - which represent the underlying combinatorial structure, and the objective function to be optimized which models "the best possible way".
In the classical assignment problem one has $m = n$ and most of the problems with $m > n$ can be transformed or are strongly related to analogous problems with $m = n$. Therefore, we will consider $m = n$ through the rest of this chapter, unless otherwise specified.

*This research has been supported by the Spezialforschungsbereich F 003 "Optimierung und Kontrolle", Projektbereich Diskrete Optimierung.

†Technical University Graz, Institute of Mathematics, Steyrergasse 30, A-8010 Graz, Austria, E-mail: cela@opt.math.tu-graz.ac.at

From the mathematical point of view an *assignment* is a bijective mapping of a finite set $N = \{1, 2, \ldots, n\}$ into itself, i.e., a *permutation* ϕ assigning some $j = \phi(i)$ to each $i \in N$. The set of all permutations (assignments) of n items will be denoted by $\mathcal{S}_n$ and has $n!$ elements. Every permutation ϕ of the set $N = \{1, \ldots, n\}$ corresponds uniquely to a *permutation matrix* $X_\phi = (x_{ij})$ with $x_{ij} = 1$ for $j = \phi(i)$ and $x_{ij} = 0$ for $j \neq \phi(i)$. Thus a permutation matrix $X = (x_{ij})$ can be defined as a matrix which fulfills the following conditions, so-called *assignment constraints*

$$\begin{aligned} \sum_{i=1}^{n} x_{ij} &= 1 \qquad \text{for all} \quad j = 1, \ldots, n \\ \sum_{j=1}^{n} x_{ij} &= 1 \qquad \text{for all} \quad i = 1, \ldots, n \\ x_{ij} &\in \{0, 1\} \qquad \text{for all} \quad i, j = 1, \ldots, n \end{aligned} \tag{1}$$

By replacing the conditions $x_{ij} \in \{0, 1\}$ by $x_{ij} \geq 0$ in (1), we get a *doubly stochastic matrix*. The set of all doubly stochastic matrices forms *the assignment polytope* P_A. Due to a famous result of (Birkhoff, 1946), the assignment polytope P_A is the convex hull of all assignments, or equivalently, every doubly stochastic matrix can be written as convex combination of permutation matrices.

The concept of an assignment is strongly related to another well known concept in graph theory and in combinatorial optimization, *matching in bipartite graphs*. A bipartite graph G is a triple $(V, W; E)$, where the vertex sets V and W have no vertices in common and the edge set E is a set of pairs (i, j) where $i \in V$ and $j \in W$. A subset M of E is called a *matching*, if every vertex of G is incident with at most one edge from M. The cardinality of M is called cardinality of the matching. The *maximum matching problem* asks for a matching with as many edges as possible. A matching M is called a *perfect matching*, if every vertex of G is incident with exactly one edge from M. Evidently, every perfect matching is a maximum matching. A perfect matching in a bipartite graph $G = (V, W; E)$ with $V = \{v_1, v_2, \ldots, v_n\}$, $W = \{w_1, w_2, \ldots, w_n\}$ can be represented by a permutation ϕ_M of $\{1, 2, \ldots, n\}$ such that $\phi_M(i) = j$ if and only if $(v_i, w_j) \in M$. Hence a perfect matching in a bipartite graph is an assignment.
(Hopcroft and Karp, 1973) gave an $O(|E|\sqrt{|V|})$-algorithm which constructs a perfect matching if it exists. (Even and Tarjan, 1975) gave an $O(\sqrt{|V|}|E|)$ algorithm for the maximum flow problem on unit capacity simple networks, algorithm which can also be applied to find a matching of maximum cardinality in a bipartite graph. (Alt et al., 1991) gave an $O(|V|^{1.5}\sqrt{|E|/\log|V|})$ implementation for the Hopcroft-Karp algorithm. Based on ideas similar to those in (Hopcroft and Karp, 1973), a fast randomized Monte-Carlo algorithm is given by (Mulmuley et al., 1087). This

algorithm finds a perfect matching at costs of a single matrix inversion. The reader is referred to the bibliography in (Burkard and Çela, 1999) for further reference pointers related to algorithms for cardinality matching problems.

2 Linear Assignment Problems

The linear assignment problem (LAP) is one of the oldest and most studies problems in combinatorial optimization. Many different algorithms have been developed to solve this problem. Also other aspects of the problem as the asymptotic behavior or special cases have been thoroughly investigated. The reader is referred to (Dell'Amico and Martello, 1997) for a comprehensive annotated bibliography and to (Burkard and Çela, 1999) for a recent review on assignment problems.

2.1 Problem definition and applications

Recall the original model where n items are to be assigned to n other objects in the best possible way. Let c_{ij} be the cost incurred by the assignment of object i to object j. We are looking for an assignment ϕ which minimizes the overall cast $\sum_{i=1}^{n} c_{i\phi(i)}$. Thus, the *linear assignment problem* (LAP) is given as follows

$$\min_{\phi \in \mathcal{S}_n} \sum_{i=1}^{n} c_{i\phi(i)}\,, \tag{2}$$

where $\mathcal{S}_n$ is the set of permutations of $\{1, 2, \ldots, n\}$. Based on the description (1) of the set of all assignments (see Section 1), the LAP can also be formulated as follows:

$$\min \textstyle\sum_{ij} c_{ij}x_{ij} \text{ over all matrices } X = (x_{ij}) \text{ which fulfill (1).}$$

Due to Birkhoff's result we can relax the conditions $x_{ij} \in \{0, 1\}$ to $x_{ij} \geq 0$ and obtain the linear programming formulation of the LAP. Any basic solution of this linear program corresponds to a permutation matrix.

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_{ij}x_{ij} & & \\
& \sum_{i=1}^{n} x_{ij} = 1 & & j = 1, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} = 1 & & i = 1, \ldots, n \\
& x_{ij} \geq 0 & & i, j = 1, \ldots, n.
\end{aligned} \tag{LP}$$

As we will mention in the next section many algorithms for the LAP are based on linear programming techniques and consider often the dual linear program:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j & \\ & u_i + v_j \leq c_{ij} & i,j = 1,\ldots,n \\ & u_i, v_j \ \in \mathbb{R} & i,j = 1,\ldots,n, \end{aligned} \tag{3}$$

where u_i and v_j, $1 \leq i,j \leq n$, are dual variables.

Among the numerous applications of the LAP the so-called personnel assignments are the most typical. In the personnel assignment we want to assign people to objects, e.g. jobs, machines, rooms, to other people etc. Each assignment has a "cost" and we want to make the assignment so as to minimize the overall sum of the costs. For example one company might want to assign graduates to vacant jobs In this case the cost c_{ij} is given by $c_{ij} = -p_{ij}$ where p_{ij} is the proficiency index for placing candidate i to job j, and the goal is to assign each candidate i to some vacancy $\phi(i)$ such that the overall cost $\sum_i c_{i\phi(i)}$ is minimized, or equivalently, the overall proficiency $\sum_i p_{i\phi(i)}$ is maximized.

There are many other applications of the linear assignment problem e.g. in locating and tracing objects in space, scheduling on parallel machines, inventory planning, vehicle and crew scheduling, wiring of typewriters etc. The reader is referred to (Ahuja et al., 1995) and (Burkard and Çela, 1999) for a detailed description of some applications of the LAP and literature pointers to other applications.

2.2 Algorithms for the LAP

The LAP can be solved efficiently, and the design of efficient solution methods for this problem has been an object of research for many years. There exists an amazing amount of algorithms, sequential and parallel, for the LAP, ranging from primal-dual combinatorial algorithms, to simplex-like methods. The worst-case complexity of the best sequential algorithms for the LAP is $O(n^3)$, where n is the size of the problem. From the computational point of view very large scale dense assignment problems with about 10^6 nodes can be solved within a couple of minutes by sequential algorithms, see (Lee and Orlin, 1994).

There is a number of survey papers and books on algorithms, among others (Derigs, 1985), (Dell'Amico and Toth, 2000) and the book on the first DIMACS challenge edited by (Johnson and McGeoch, 1993). Among papers reporting on computational experience we mention (Carpaneto et al., 1988; Lee and Orlin, 1994; Dell'Amico and Toth, 2000) and some of the papers in (Johnson and McGeoch, 1993).

Most sequential algorithms for the LAP can be classified into *primal-dual algorithms* and *simplex-based algorithms*. Primal-dual algorithms work with a pair consisting of an infeasible solution x_{ij}, $1 \leq i,j \leq n$, of LP (called primal solution), and a feasible solution u_i, v_j, $1 \leq i,j \leq n$ of the dual (3) (called dual solution). These solutions fulfill the complementarity slackness conditions:

$$x_{ij}(c_{ij} - u_i - v_j) = 0\,, \quad \text{for } 1 \leq i,j \leq n \tag{4}$$

These solutions are updated iteratively until the primal solution becomes feasible, while keeping the complementary slackness conditions fulfilled and the dual solution feasible. At this point the primal solution would be optimal, according to duality theory.

Different primal-dual algorithms differ on 1) the way they obtain a starting pair of a primal and a dual solution fulfilling the conditions described above, and 2) the way the solutions are updated. A starting dual solution can be obtained as in the Hungarian method by setting $u_i := \min\{c_{ij} : 1 \leq j \leq n\}$, for $1 \leq i \leq n$, and then $v_j := \min\{c_{ij} - u_i : 1 \leq i \leq n\}$, for $1 \leq j \leq n$. An infeasible primal starting solution could be given by a matching of maximal cardinality in the bipartite graph $\bar{G} = (V, W; \bar{E})$, where $V = W = \{1, 2, \ldots, n\}$, and $\bar{E} = \{(i,j) : \bar{c}_{ij} := c_{ij} - u_i - v_j = 0\}$. Then, set $x_{ij} := 1$ if (i,j) is an edge of the matching and $x_{ij} = 0$, otherwise. Clearly, the pair of solutions obtained in this way fulfill the complementarity slackness conditions. One way of updating the pair of solutions is the shortest augmenting path method. This method gives raise to a whole class of algorithms which meet the best known time complexity bound for the LAP, namely $O(n^3)$. For a given pair of solutions as above construct a weighted directed bipartite graph $\tilde{G} = (V, W; \tilde{E})$ with arc set $\tilde{E} = D \cup R$ with set of forward arcs $D = \{(i,j) : (i,j) \in E,\ x_{ij} = 0\}$ and set of *backward arcs* $R = \{(j,i) : (i,j) \in \bar{E},\ x_{ij} = 1\}$. The weights of the backward arcs are set equal to 0, whereas the weights of the forward arcs are set equal to the corresponding *reduced costs* $\bar{c}_{ij}$. Then select a node r in V which has not been assigned yet, and solve the single-source shortest path problem, i.e., compute the shortest paths from r to all nodes of $\tilde{G}$. The shortest among all paths from r to some free node in W is used to augment the current primal solution by swapping the free and matched edges. The dual solution and the reduced costs are then accordingly updated. It can be shown that after n augmentations an optimal primal solution results, see (Derigs, 1985).

There are various shortest augmenting path algorithms for the LAP. Basically they differ in the way they determine a starting pair of primal and dual solutions, and by the subroutine they use for computing the shortest paths. Most of the existing algorithms use the Dijkstra algorithm for the shortest path computations.

Simplex-based algorithms are special implementations of the primal or the dual simplex algorithm for linear programming applied to LP. Simplex-based algorithms

for the LAP are specific implementations of the network simplex algorithm. The latter is a specialization of the simplex method for linear programming to network problems. The specialization relies on exploiting the combinatorial structure of network problems to perform efficient pivots acting on trees rather than on the coefficient matrix.
It is well known that there is a one-to-one correspondence between primal (integer) basic solutions of the LAP and spanning trees of the bipartite graph G related to assignment problems as described in Section 1. Moreover, given a spanning tree, one can uniquely determine the values of the corresponding dual variables so as to fulfill the complementarity slackness conditions, as soon as the value of one of those variables is fixed (arbitrarily). Every integer primal feasible basic solution is highly degenerate because it contains $2n-1$ variables and $n-1$ of them are equal to 0. Hence degeneracy poses a problem, and the first simplex-based algorithms for the LAP were exponential. The first steps towards the design of polynomial-time simplex-based algorithms were made by introducing the concept of so-called *strongly feasible trees*, introduced by (Cunningham, 1976). There are implementations of simplex-based algorithm for the LAP which match the best known time complexity bound of $O(n^3)$. The reader is referred to (Burkard and Çela, 1999) for references and further details.
More recently (Ramakrishnan et al., 1993) applied an interior point algorithm to the LAP and got promising results, in particular for large size instances, see (Johnson and McGeoch, 1993).

Since the late 1980s a number of parallel algorithms for the LAP has been proposed. The speed-up achieved by such algorithms is limited by the sparsity of the cost matrices and/or the decreasing load across the iterations. For a good review on parallel algorithms for the LAP and network flow problems in general the reader is referred to (Bertsekas et al., 1995).

2.3 Asymptotic behavior and probabilistic analysis

When dealing with the asymptotic behavior of the LAP, it is always assumed that the cost coefficients c_{ij} are independent random variables (i.r.v.) with a common prespecified distribution. The main question concerns the behavior of the expected optimal value of the problem as its size tends to infinity.
For cost coefficients c_{ij} being i.r.v. with a uniform distribution on $[0,1]$ it has been shown that the optimal value of the LAP remains within constant bounds as the size n of the problem tends to infinity. The best upper bound equals 2 and is due to (Karp, 1987). The best lower bound equals 1.51 and is due to (Olin, 1992). Although the gap between the current lower and upper bounds on the expected optimal value of the LAP is large, it is believed that the expected value is close to 1.6 or more

exactly $\frac{\pi^2}{6}$ in the case of independent cost coefficients c_{ij} uniformly distributed on $[0,1]$. For a discussion in some details and for more references see (Burkard and Çela, 1999).
A more general scenario where the coefficients c_{ij} are i.r.v. with a common arbitrary general distribution has been investigated by (Frenk et al., 1987) and (Olin, 1992). Under mild assumption on the probability distribution of the coefficients the authors derive constant upper and lower bounds on the expected optimal value of the LAP.

In the case of coefficients c_{ij} being i.r.v. uniformly distributed on $[0,1]$, the LAP can be solved in expected $O(n^2 \log n)$ time by a randomized algorithm proposed by (Karp, 1980). Faster randomized algorithms which produce a solution whose objective function value is within a constant factor of the optimal objective function value have been proposed by several authors. There is for instance an algorithm of (Karp et al., 1994) which runs in expected linear time ($O(n)$) and provides a solution within a factor of $3 + O(n^{-a})$) of the optimum with probability $1 - O(n^{-a})$, where a is some fixed positive number. The best known linear assignment problem with an objective function different from that of the LAP is the *bottleneck linear assignment problem* (BLAP)

$$\min_{\phi} \max_{1 \le i \le n} c_{i\phi(i)} \,. \tag{5}$$

considered originally by (Fulkerson et al., 1953). This problem occurs e.g. in the assignment of jobs to parallel machines so as to minimize the latest completion time. One of the first algorithms proposed for the BLAP is the so-called *threshold algorithm*. The threshold algorithm chooses a cost element $c^*_{ij} =: K$ - the *threshold value* - and constructs a matrix $\bar{C}$ defined as follows

$$\bar{c}_{ij} := \begin{cases} 1 & \text{if} \quad c_{ij} > K \\ 0 & \text{if} \quad c_{ij} \le K \end{cases}$$

Then the algorithm checks whether the bipartite graph with adjacency matrix $\bar{C}$ contains a perfect matching or not. The algorithms repeats than this procedure for a new (and eventually smaller) value of the threshold K. The smallest value K for which the corresponding bipartite graph contains a perfect matching, is the optimum value of the BLAP. The best time complexity known today amounts to $O(n\sqrt{nm})$ and is due to (Punnen and Nair, 1994). (Here m is the number of finite elements in the coefficient matrix (c_{ij}) which would correspond to the number of edges of the bipartite graph in the graph theoretical setting.)
A randomized algorithm with quadratic expected running time ($O(n^2)$) has been given by (Pferschy, 1996). A computational study on the comparison of different deterministic algorithms for the BLAP has been given by (Pferschy, 1997).

Another linear assignment problem introduced by (Burkard and Rendl, 1991) is the so-called *lexicographic bottleneck assignment problem* (LexBAP). In the LexLAP we want to find a permutation (assignment) ϕ^* which lexicographically minimizes c_ϕ over all permutation ϕ, where c_ϕ is the vector of costs $c_{i\phi(i)}$, $1 \leq i \leq n$, sorted non-increasingly.
(Martello et al., 1984) have considered the so-called *balanced assignment problem* (BalAP). Given a real $n \times n$ matrix $C = (c_{ij})$, the balanced assignment problem can be formulated as

$$\min_{\phi} \left[\max_{i} c_{i\phi(i)} - \min_{i} c_{i\phi(i)} \right] .$$

The problem can be solved efficiently in $O(n^4)$ time.

A more general linear assignment problem which includes as special cases the linear assignment problem (2) and the bottleneck assignment problem (5) is the *algebraic assignment problem* (AAP) introduced by (Burkard et al., 1977). In the AAP the coefficients c_{ij} are elements of a totally ordered semigroup $(H, *, \prec)$ with composition $*$ and order relation $\prec$. The AAP can then be formulated as follows:

$$\min_{\phi \in \mathcal{S}_n} \; c_{1\phi(1)} * c_{2\phi(2)} * \cdots * c_{n\phi(n)} \, . \tag{6}$$

The AAP can be solved efficiently if the order relation and the composition fulfill some natural algebraic properties. For further results and reference pointers consult the survey on algebraic optimization by (Burkard and Zimmermann, 1982).

2.4 Available computer codes and test instances

FORTRAN listings of codes for the LAP and the LBAP can be found in the book by (Burkard and Derigs, 1980). The code for the LAP is a primal-dual algorithm based on shortest path computations done by a version of Dijkstra's algorithm.
Source codes of another primal-dual algorithm for the LAP and the LBAP can be downloaded from `http://207.158.230.188/assignment.html`. One can choose among a C++, a PASCAL, and a FORTRAN implementations of an algorithm of (Jonker and Volgenant, 1986).
A (compressed) FORTRAN source file - called `548.Z` - of an implementation of the Hungarian algorithm, due to (Carpaneto and Toth, 1980), can be downloaded from `ftp://netlib.att.com` in `/netlib/toms`. Other listings of FORTRAN codes for the LAP can be found in (Carpaneto et al., 1988). The codes are available from the floppy disk included in the book.
The C code of an efficient implementation of the scaling push-relabel algorithm of (Goldberg and Kennedy, 1995) for the LAP can be downloaded from Goldberg's

network optimization library at

`http://www.neci.nj.nec.com/homepages/avg/soft.html`.

Finally, listings of 5 FORTRAN codes of auction algorithms for the LAP can be found in Bertsekas' homepage at

`http://web.mit.edu/dimitrib/www/auction.txt`.

Test instances of the LAP can be downloaded as `ascii` files from the homepage of the OR-Library maintained by J. Beasley at

`http://mscmga.ms.ic.ac.uk/pub`.

Other test instances can be obtained from the ELIB library at

`ftp://ftp.zib.de/pub/Packages/mp-testdata/assign/index.html`.

Clearly, since the LAP can be formulated as as a minimum cost flow problem, algorithms developed for the later can also be applied to the LAP. However, such algorithms are not supposed to exploit the specific features of the LAP, and hence may not be competitive with algorithms developed especially for the LAP. Besides Goldberg's network optimization library and Bertsekas' homepage, other codes for network optimization can be found in `Netlib` at

`http://www.OpsResearch.com/OR-Links/index.html`.

Furthermore, `C` codes of implementations of the primal and the dual network simplex algorithm, due to Löbel, can be obtained through

`http://www.zib.de/Optimization/index.de.html`.

3 Multidimensional Assignment Problems

3.1 General Remarks and Applications

Multi-dimensional (sometimes referred as *multi-index*) assignment problems (MAP) are natural extensions of the linear assignment problem. They have been considered for the first time by (Pierskalla, 1967). The most prominent representatives of this class are axial and planar 3-dimensional assignment problems to be considered in the next section. The MAP asks for $d-1$ permutations $\phi_1, \phi_2, \ldots, \phi_{d-1}$ which minimize the following objective function:

$$\min_{\phi_1,\phi_2,\ldots,\phi_{d-1}} \sum_{i=1}^{n} c_{i\phi_1(i)\phi_2(i)\ldots\phi_{d-1}(i)} \cdot$$

In terms of graphs a multidimensional assignment problem can be described as follows: Let a complete d-partite graph $G = (V_1, V_2, \ldots, V_d; E)$ with vertex sets V_i, $|V_i| = n$, $i = 1, 2, \ldots, d$, and edge set E be given. A subset X of $V = \bigcup_{i=1}^{d} V_i$ is a

clique, if it meets every set V_i in exactly one vertex. A d-dimensional assignment is a partition of V into n pairwise disjoint cliques. If c is a real valued cost function defined on the set of cliques of $G = (V_1, V_2, \ldots, V_d; E)$, the d-dimensional assignment problem asks for a d-dimensional assignment of minimum cost. Special cases where the costs c of a clique are not arbitrary, but given as a function of elementary costs attached to the edges of the complete d-partite graph (eg. sum costs, star costs, tour costs or tree costs) have been investigated and the performance of simple heuristics has been analyzed in these cases. For more information see e.g. (Burkard and Çela, 1999) and the references therein.

Multidimensional assignment problems in their general form have found some applications as a means to solve data association problems in in multi-target tracking and multi-sensor surveillance. The data association problem consists in partitioning the observations into tracks and false alarms in real time. General classes of these problems can be formulated as multidimensional assignment problems. Other applications of MAPs are related to track initiation, track maintenance, and multi-sensor tracking. Another interesting MAP arises in the context of tracking elementary particles. For more information on these two applications see (Burkard and Çela, 1999) and the references therein.

3.2 Axial 3-Dimensional Assignment Problems

Consider n^3 cost coefficients c_{ijk}. The *axial 3-dimensional assignment problem* (3-DAP) can then be stated as

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}x_{ijk} \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1\,, \quad i = 1,2,\ldots,n\,, \\
& \sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1, \quad i = 1,2,\ldots,n\,, \\
& \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} = 1, \quad i = 1,2,\ldots,n\,, \\
& x_{ijk} \in \{0,1\} \ \forall \quad 1 \le i,j,k \le n\,.
\end{aligned}
\tag{7}
$$

We can think of c_{ijk} as the cost of assigning job j to be performed by worker i in machine k. It follows that $x_{ijk} = 1$, if job j is assigned to worker i in machine k, and $x_{ijk} = 0$, otherwise.

Equivalently, a 3-DAP can be described with the help of two permutations ϕ and ψ

$$\min_{\phi,\psi\in\mathcal{S}_n} \sum_{i=1}^{n} c_{i\phi(i)\psi(i)}. \tag{8}$$

Thus this problem has $(n!)^2$ feasible solutions. It has been shown that the dimension of the axial 3-index assignment polytope (i.e., the convex hull of feasible solutions to problem (7)) is n^3-3n+2. Furthermore several classes of facet defining inequalities have been identified and efficient algorithms for the separation of these facets have been developed. For more information on these topic see the description in (Burkard and Çela, 1999) and consult the references cited therein.
In contrast to the linear assignment problem LAP the 3-DAP cannot be solved efficiently and branch and bound algorithms are the mostly used algorithms to solve this problem. The lower bounds are usually computed by solving some Lagrangean relaxation of the 3-DAP by subgradient optimization approaches, see e.g. (Balas and Saltzman, 1991). The authors introduce in (Balas and Saltzman, 1991) also a non-trivial branching strategy which exploits the structure of the problem and allows to fix several variables at each branching node.
A heuristic for solving the 3-DAP has been proposed by (Pierskalla, 1967).

Finally, there exists a number of efficiently solvable special cases of the 3-DAP, e.g. if the cost coefficients are taken from a 3-dimensional Monge array or if the cost coefficients are decomposable, i.e., $c_{ijk}=u_iv_jw_k$ and u_i, v_j, and w_k are nonnegative. The reader is referred to (Burkard and Çela, 1999) and the references therein for more information on this topic.

3.3 Planar 3-Dimensional Assignment Problems

Let c_{ijk}, $1\le i,j,k\le n$, be n^3 cost coefficients. The *planar 3-dimensional assignment problems* (3-PAP) is stated as follows:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}x_{ijk} \\
\text{s.t.} \quad & \\
& \sum_{i=1}^{n} x_{ijk} = 1\,, \quad j,k=1,2,\ldots,n\,, \\
& \sum_{j=1}^{n} x_{ijk} = 1\,, \quad i,k=1,2,\ldots,n\,, \\
& \sum_{k=1}^{n} x_{ijk} = 1\,, \quad i,j=1,2,\ldots,n\,,
\end{aligned} \tag{9}$$

$$x_{ijk} \in \{0,1\} \quad i,j,k = 1,2,\ldots,n\,.$$

The 3-PAP has interesting applications in time tabling problems. See (Euler and Le Verge, 1996) for a recent study on time tables and related polyhedra.
It is easily seen that the feasible solutions of the 3-PAP correspond to *Latin squares*. Thus, the number of feasible solutions of a 3-PAP of size n equals the number of Latin squares of order n, and hence increases very fast. Similarly to the 3-DAP also the 3-PAP is a hard problem and cannot be solved efficiently. There are not many algorithms known for the 3-PAP. Besides two branch and bound algorithms due to (Vlach, 1967) and (Magos and Miliotis, 1994) there is also a tabu search algorithm for the 3-PAP due to (Magos, 1996). It uses a neighborhood structure based on the relationship between the 3-PAP and the latin squares. See e.g. (Burkard and Çela, 1999) for more information on algorithms for the 3-PAP.

4 The Quadratic Assignment Problem

The quadratic assignment problem (QAP) was introduced in 1957 by Koopmans and Beckmann as a model for a plant location problem. Since then the QAP has been object of intensive investigations concerning different aspects of the problem ranging from algorithms to asymptotic behavior and special cases. In contrast to its linear counterpart the QAP can not be solved efficiently. However there are some restricted cases of the QAP which can be solved in polynomial time. Such special cases are QAPs whose coefficient matrices show special combinatorial properties, e.g. Monge and Monge-like properties. The interested reader is referred to (Çela, 1998) for a detailed discussion on efficiently solvable special cases of the QAP.
Also from the practical point of view the QAP is widely considered as one of the hardest problems in combinatorial optimization and there are instances of size 24 which can not yet be solved to optimality in reasonable computational time, e.g. the Nugent instance of size [24], see QAPLIB (Burkard et al., 1997)
The reader is referred to (Burkard and Çela, 1997) for a comprehensive annotated bibliography, to (Çela, 1998) for a recent monograph, and to (Burkard et al., 1998) for a recent review on quadratic assignment problems and for reference pointers.

4.1 Problem definition and applications

Among the most common and best known applications of the QAP are those which arise in a facility location context. This is one reason for choosing the facility location terminology to introduce the problem.
Consider the problem of allocating n facilities to n locations, with costs depending on the distance between the locations and the flow between the facilities, plus costs

associated with a facility being placed at a certain location. The objective is to assign each facility to a location such that the total cost is minimized. More specifically, we are given three $n \times n$ real matrices $A = (a_{ij})$, $B = (b_{kl})$ and $C = (c_{ik})$, where a_{ij} is the flow between the facility i and facility j, b_{kl} is the distance between the location k and location l, and c_{ik} is the cost of placing facility i at location k. The Koopmans-Beckmann version of the QAP can be then formulated as follows:

$$\min_{\phi \in \mathcal{S}_n} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\phi(i)\phi(j)} + \sum_{i=1}^{n} a_{i\phi(i)} \right) \tag{10}$$

where $\mathcal{S}_n$ is the set of all permutations of N. A product $a_{ij}b_{\phi(i)\phi(j)}$ is the cost of assigning facility i to location $\phi(i)$ and facility j to location $\phi(j)$.
A more general version of the QAP was introduced by Lawler in 1963. In this version we are given a four-dimensional array $D = (d_{ijkl})$ of coefficients instead of the two matrices A and B, and the problem can be stated as

$$\min_{\phi \in \mathcal{S}_n} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij\phi(i)\phi(j)} + \sum_{i=1}^{n} c_{i\phi(i)} \right) . \tag{11}$$

Besides applications in facility location the QAP finds applications in VLSI design, computer manufacturing, scheduling, process communications, backboard wiring and typewriter keybord design. We describe here a less known application on turbine balancing. Hydraulic turbine runners as used in electricity generation consist of a cylinder around which a number of blades are welded at regular spacings. Due to inaccuracies in the manufacturing process, the weights of these blades differ slightly, and it is desirable to locate the blades around the cylinder in such a way that the distance between the center of mass of the blades and the axis of the cylinder is minimized. This problem was introduced by Mosevich in 1986 and has been formulated as a QAP by (Laporte and Mercure, 1988). The places at regular spacings on the cylinder are modeled by the vertices $v_1, \ldots, v_n$ of a regular n-gon on the unit circle in the Euclidean plane. Thus

$$v_i = \left(\sin\left(\frac{2i\pi}{n}\right), \cos\left(\frac{2i\pi}{n}\right) \right), \quad 1 \leq i \leq n.$$

The masses of the n blades are given by the positive reals $0 < m_1 \leq m_2 \leq \cdots \leq m_n$. The goal is to assign each mass m_i to some vertex $v_{\phi(i)}$ in such a way that the center of gravity

$$\sum_{i=1}^{n} m_{\phi(i)} \begin{pmatrix} \sin(\frac{2i\pi}{n}) \\ \cos(\frac{2i\pi}{n}) \end{pmatrix},$$

of the resulting mass system is as close to the origin as possible. Minimizing the Euclidean norm of the above vector is equivalent to minimizing the expression

$$\sum_{i=1}^{n}\sum_{j=1}^{n} m_{\phi(i)}m_{\phi(j)} \cos\left(\frac{2(i-j)\pi}{n}\right). \tag{12}$$

Thus we have a quadratic assignment problem, where the coefficient matrices A and B have a very special structure: $a_{ij} = \cos\left(\frac{2(i-j)\pi}{n}\right)$ and $b_{ij} = m_i \cdot m_j$.
Finally, notice that there are a number of other well known combinatorial optimization problems which can be formulated as QAPs, e.g. graph partitioning, maximum clique, the traveling salesman problem, the linear arrangement problem, and the minimum weight feedback arc set problem.

To conclude this section notice that the QAP is a "very hard" problem from the theoretical point of view. Not only that the QAP cannot be solved efficiently but it even cannot be approximated efficiently within some constant approximation ratio. Furthermore, also finding local optima is in general not a trivial task. For information and references concerning the computational complexity of the QAP see (Çela, 1998).

4.2 Alternative formulations of the QAP

There exist several equivalent formulation of the QAP. Different formulations stress different characteristics of the problem and lead to different solution approaches. Recall that there is a one-to-one correspondence between the permutations of $N = \{1, 2, \ldots, n\}$ and the $n \times n$ permutation matrices defined in Section 1. Let $\mathbf{X}_n$ be the set of $n \times n$ permutation matrices. In terms of permutation matrices QAP(A,B,C) can be formulated as the following *quadratic integer program*:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij}b_{kl}x_{ik}x_{jl} + \sum_{i,j=1}^{n} c_{ij}x_{ij} \tag{13}$$

$$\text{s.t.} \quad (x_{ij}) \in \mathbf{X}_n \tag{14}$$

Let us define an *inner product* between $n \times n$ matrices as follows

$$\langle A, B\rangle := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij},$$

Clearly, we have $XAX^T = (a_{\phi(i)\phi(j)})$, for some $n \times n$ matrix A, a permutation $\phi \in \mathcal{S}_n$ and the associated permutation matrix $X \in \mathbf{X}_n$. Thus the QAP (13)-(14) can be

formulated alternatively as

$$\begin{array}{ll} \min & \langle A, XBX^T \rangle + \langle C, X \rangle \\ \text{s.t.} & X \in \mathbf{X}_n. \end{array} \tag{15}$$

Finally consider the *trace formulation* of the QAP. The trace $\mathrm{tr}(A)$ of an $n \times n$ matrix $A = (a_{ij})$ is defined as sum of its diagonal elements: $tr(A) = \sum_{i=1}^{n} a_{ii}$. Let us denote $B^{\phi} := XB^tX^t$, where X is the permutation matrix corresponding to ϕ. We get

$$\mathrm{tr}(AB^{\phi}) = \sum_{i,j=1}^{n} a_{ij} b_{ji}^{\phi} = \sum_{i,j=1}^{n} a_{ij} b_{\phi(i)\phi(j)} \,,$$

since $b_{ij}^{\phi} = b_{\phi(i)\phi(j)}$, for $i, j \in \{1, 2, \ldots, n\}$. Since $tr(CX^t) = \sum_{i=1}^{n} c_{i\phi(i)}$, the QAP in (15) can be formulated as

$$\begin{array}{ll} \min & tr(AXB^T + C)X^T \\ \text{s.t.} & X \in \mathbf{X}_n. \end{array} \tag{16}$$

4.3 Linearizations

A first attempt to solve the QAP would be to eliminate the quadratic term in the objective function (13), in order to transform the problem into a (mixed) 0-1 linear program (MILP). The linearization of the objective function is usually achieved by introducing new variables and new linear (and binary) constraints. The very large number of new variables and constraints, however, poses an obstacle for efficiently solving the resulting linear integer programs. The optimal value of an LP relaxation of some MILP formulation is a lower bound for the QAP. In this context the "tightness" of the continuous relaxation of the linear integer program mentioned above is a desirable property. Several linearizations of the QAP have been proposed in the literature, e.g. by (Lawler, 1963), by (Kaufmann and Broeckx, 1978) (this linearization has the smallest number of variables and constraints), by (Frieze and Yadegar, 1983) and by (Adams and Johnson, 1994). The last linearization unifies most of the previous linearizations and is important for getting lower bounds. The QAP with array of coefficients $C = (d_{ijkl})$ is proved to be equivalent to the following mixed 0-1 linear program

$$\begin{array}{ll} \min & \displaystyle\sum_{i,j=1}^{n} \sum_{k,l=1}^{n} d_{ijkl} y_{ijkl} \\ \text{s.t.} & (x_{ij}) \in \mathbf{X}_n, \end{array}$$

$$\sum_{i=1}^{n} y_{ijkl} = x_{jl}, \quad j,k,l = 1,\ldots,n,$$
$$\sum_{k=1}^{n} y_{ijkl} = x_{jl}, \quad i,j,l = 1,2,\ldots,n,$$
$$y_{ijkl} = y_{jilk}, \quad i,j,k,l = 1,\ldots,n,$$
$$y_{ijkl} \geq 0, \quad i,j,k,l = 1,2,\ldots,n,$$

where each y_{ijkl} represents the product $x_{ik}x_{jl}$. Although as noted by (Adams and Johnson, 1994) a significant smaller formulation in terms of both the variables and constraints could be obtained, the structure of the continuous relaxation of the above formulation is favorable for solving it approximately by Lagrangean dual methods. The theoretical strength of the linearization (17) relies on the fact that the constraints of the continuous relaxations of previous linearizations can be expressed as linear combinations of the constraints of the continuous relaxation of (17). Moreover, many of the previously published lower-bounding techniques can be explained based on the Lagrangean dual of this relaxation. For more details on this topic we refer to Section 4.4.2.

4.4 Lower bounds

Lower bounding techniques are used within implicit enumeration algorithms, such as branch and bound, to perform a *limited* search of the set of feasible solution, until an optimal solution is found. Numerous bounding techniques have been developed for the QAP so far. The most successful bounding techniques for the QAP can be classified in 3 groups: Gilmore-Lawler type lower bounds, bounds based on LP relaxation, eigenvalue related bound

4.4.1 Gilmore-Lawler type lower bounds

The basic idea of these type of lower bounds goes back to the early 60s when (Gilmore, 1962) and (Lawler, 1963) developed the so-called Gilmore-Lawler bound (GLB) for the QAP. Nowadays the Gilmore-Lawler type lower bounds and especially the GLB are the most frequently used bounds within branch and bound algorithms for the QAP. The most advantageous property of these bounds is that they can be computed efficiently. The main drawback is the fast deterioration of their quality with increasing problem size.

To compute the GLB for a given QAP of size n one has to solve $n^2 + 1$ LAPs, n^2 of them of size $n-1$ and the last one of size n, as described below.

Consider an instance of the Lawler QAP (11) with coefficients $D = (d_{ijkl})$. For each ordered pair of indices (i,k), $1 \leq i,k \leq n$, solve an LAP with coefficient matrix

$D^{(i,k)} = (d_{ijkl})$ under the additional constraint that $\phi(i) = k$. Let us denoted by l_{ik} the optimal solutions of the above mentioned LAP. Finally, solve an LAP with coefficient matrix $(l_{ij} + c_{ij})$; its optimal value is the GLB. Since the LAP can be solved efficiently, also the GLB can be computed efficiently.

Several Gilmore-Lawler type lower bounds have been developed aiming at improving the quality of the GLB. One of the ideas on which such bounds are based are the so-called *reduction methods*. These methods decompose each quadratic cost coefficient into several terms so as to guarantee that some of them end up in being linear cost coefficients and can be moved to the linear term of the objective function. This would yield a tighter lower bound because the LAP can be solved exactly.
More recently another bounding procedure which shares the basic idea of the GLB has been proposed by (Hahn and Grant, 1998). This procedure combines GLB ideas with reduction steps in a general framework. The resulting bound (HGB) shows a good trade off between computation time and bound quality when tested in instances from QAPLIB, see (Burkard et al., 1997).

4.4.2 Bounds based on linear programming relaxations

Consider a mixed integer linear programming (MILP) formulations of the QAP. Clearly, the optimal solution of the continuous relaxation of an MILP formulation is a lower bound for the optimal value of the corresponding QAP. Moreover, each feasible solution of the dual of this relaxation is also a lower bound. Even for QAPs of moderate size it is practically impossible to solve to optimality the LP relaxations of any MILP formulation. Generally, such formulations are highly degenerated and already for QAPs of moderate size the memory requirements become prohibitive as shown by (Resende et al., 1995). The LP relaxations of the MILPs are therefore approximately solved by applying subgradient optimization based techniques or Lagrangean relaxations. Adams et al. have shown that a particular Lagrangean relaxation of the MILP (17) can be solved efficiently for each fixed set of Lagrangean multipliers, see (Adams and Johnson, 1994). The Lagrangean multipliers are then updated iteratively in the fashion of a dual ascent procedure. The strength of the method of Adams and Johnson relies on the fact that it can produce all Gilmore-Lawler-like bounds described in Section 4.4.1 and others, but the HGB, for different settings of the Lagrangean multipliers.
Recently (Karisch et al., 1999) have shown that although HGB cannot be obtained by applying the algorithm of Adams and Johnson, both AJB and HGB can be obtained as feasible solutions of the dual of the continuous relaxation of the MILP formulation (17). Karisch et al. propose an iterative algorithm to approximately solve this dual, and show that AJB, HGB, and all other Gilmore-Lawler-like bounds can be obtained by applying this algorithm with specific settings for the control

parameters.

4.4.3 Eigenvalue based lower bounds

These bounds were introduced by (Finke et al., 1987), and can be applied to the Koopmans-Beckmann QAP in (10). They are based on the relationship between the objective function value of the QAP in the trace formulation (16) and the eigenvalues of its coefficient matrices. When designed and implemented carefully, these techniques produce bounds of good quality in comparison with Gilmore-Lawler-like bounds or, more generally, with bounds based on linear relaxations. However, these bounds are quite expensive in terms of computation time requirements and are, therefore, not appropriate for use within branch and bound algorithms.

4.5 Exact solution methods

Since QAP is a hard problem from the theoretical (and also from the practical) point of view and no efficient algorithms are known for this problem. All existing exact algorithms are in principle enumeration methods like branch and bound, cutting plane algorithms and branch and cut algorithms.

4.5.1 Branch and bound algorithms

Nowadays branch and bound algorithms appear to be the most efficient exact algorithms for solving the QAP.
The most efficient branch and bound algorithms for the QAP employ the Gilmore-Lawler bound (GLB). The reason is that other bounds which outperform GLB in terms of bound quality are simply too expensive in terms of computation time. More recently some efforts have been made to employ other Gilmore-Lawler-like bounds. The bound of HGB has been used in a branch and bound algorithm by (Hahn et al., 1998) and the results are promising.

Three types of branching strategies are mostly used for the QAP: *single assignment branching*, *pair assignment branching* and branching based on *relative positioning*. The most efficient strategy is the single assignment branching. It assigns a facility to a location in each branching step, i.e., each problem is divided into subproblems by fixing the location of one of the facilities which are not assigned yet. Several rules for the choice of the facility-location pair to determine the subproblems of a new level of the search tree have been proposed by different authors. The appropriate rule usually depends on the bounding technique. If the GLB is employed the branching rule is frequently formulated in terms of the reduced costs of the last assignment problem solved to bound the subproblem which is currently being branched.

As for the *selection rule* there seems to be no clear winner among different strategies tested for the QAP, ranging from problem-independent depth or breadth first search to instance dependent criteria related to the maximization of lower bounds or reduced costs.
More recently a number of parallel branch and bound algorithms have been developed for the QAP, e.g. by (Pardalos and Crouse, 1989), (Bruengger et al., 1997), and (Clausen and Perregaard, 1997).

4.5.2 Cutting plane methods

The cutting plane methods in general can be classified into *traditional cutting plane methods* and *polyhedral cutting plane methods*. Traditional cutting plane algorithms for the QAP have been developed by different authors, e.g. (Bazaraa and Sherali, 1982), (Balas and Mazzola, 1984a; Balas and Mazzola, 1984b)), and (Kaufmann and Broeckx, 1978). These algorithms make use of mixed integer linear programming (MILP) formulations for the QAP which are suitable for Benders' decomposition. Generally, the time needed for these methods to converge is too large, and hence these methods may solve to optimality only very small QAPs. However, heuristics derived from cutting plane approaches produce good suboptimal solutions in early stages of the search.

Also polyhedral cutting planes or *branch and cut algorithms* make use of MILP formulations of the QAP. Additionally, polyhedral cutting plane methods make use of a class of (nontrivial) valid or facet defining inequalities known to be fulfilled by *all* feasible solutions of the original problem. Some properties and few facet defining inequalities of the QAP polytope are already known, but still polyhedral cutting plane methods for the QAP are not yet backed by a strong theory. Some efforts to design branch and cut algorithms for the QAP have been made by (Padberg and Rijal, 1996) and (Kaibel, 1997). The numerical results are encouraging, although the developed software is of preliminary nature, as claimed by the authors.

4.6 Heuristics

The QAP is a very difficult problem from the practical point of view, instances of dimension $n > 20$ being still not practical to solve because of very high computation time requirements. This is probably the reason why the literature abounds in heuristics which are the only available algorithms to provide good quality solutions for the QAP in a reasonable computational time. The reader is referred to (Burkard et al., 1998) for numerous literature pointers to numerous heuristics applied to the QAP.

The numerous heuristic approaches developed for the QAP can be classified in construction methods, limited enumeration methods, local search algorithms including improvement methods, tabu search and simulated annealing, genetic algorithms, greedy randomized adaptive search procedures (GRASP), and ant systems. In the case of the QAP there is no widely accepted winner among these strategies. The construction methods, the limited enumeration methods, and the improvement methods seem, however, to be outperformed by the other heuristic approaches.

We refer to other chapters of this handbook for a detailed description of the general metaheuristic approaches mentioned above. In the following we briefly mention the most frequently used neighborhood structures included in local search algorithms for the QAP.

Frequently used neighborhoods for the QAP are the *pair-exchange* neighborhood and the *cyclic triple-exchange* neighborhood. In the case of pair-exchanges the neighborhood of a given solution (permutation) consists of all permutations which can be obtained from the given one by applying a transposition to it. The size of this neighborhood is $O(\binom{n}{2})$. In the case of cyclic triple-exchanges, the neighborhood of a solution (permutation) π consists of all permutations obtained from π by a cyclic exchange of some triple of indices. The size of this neighborhood is $O(\binom{n}{3})$. In general cyclic triple-exchanges do not lead to better results when compared with pair-exchanges.

4.7 Available computer codes for the QAP

(Burkard et al., 1997) have compiled a library of QAP instances (QAPLIB) which is widely used to test bounds, exact algorithms, and heuristics for the QAP. Many of these instances have not been solved to optimality yet, the most celebrated among them being the instances of (Nugent et al., 1969) of size larger than 25. QAPLIB can be found at

`http://www.opt.math.tu-graz.ac.at/~karisch/qaplib`.

Two codes for computing lower bounds are also available from the QAPLIB web page: a FORTRAN code due to (Burkard and Derigs, 1980) to compute the GLB for instances of size up to 256, and another FORTRAN code to compute the elimination bound (ELI) for symmetric QAP instances of size up to 256. Finally a FORTRAN code of the branch and bound algorithm developed by (Burkard and Derigs, 1980) can also be downloaded from the QAPLIB web page.

Recently, (Espersen et al.,) have developed `QAPpack` which is a JAVA package containing a branch and bound algorithm to solve the QAP. In `QAPpack` a number of bounds based on linearization are implemented: the Gilmore-Lawler bound, the bound of (Carraresi and Malucelli, 1994), the bound of (Adams and Johnson, 1994), the bound of (Hahn and Grant, 1998), and the bound of (Karisch et al., 1999). The

implementation is based on the dual framework provided by (Karisch et al., 1999). `QAPpack` can be found at `http://www.imm.dtu.dk/~te/QAPpack`.

There are also some codes of heuristics available. The (compressed) FORTRAN source file - `608.Z` - of a heuristic due to (West, 1983), can be downloaded at `ftp://netlib.att.com` in `/netlib/toms`.
The source files (compressed tar-files) of two FORTRAN implementations of GRASP for dense QAPs by (Resende et al., 1996) and sparse QAPs by (Pardalos et al., 1997) can be downloaded from Resende's web page at `http://www.research.att.com/~mgcr/src/index.h`
The source file of a FORTRAN implementation of the simulated annealing algorithm of (Burkard and Rendl, 1984) can be downloaded from the QAPLIB web page.
The source file of a C++ implementation of the simulated annealing algorithm of (Connolly, 1990), due to Taillard, can be downloaded from Taillard's web page at `http://www.idsia.ch/~eric/codes.dir/sa_qap.c`. Also a source file of a PASCAL implementation of the robust tabu search algorithm by (Taillard, 1991) can be found at Taillard's web page.

4.8 Asymptotic behavior

While being a very difficult problem both from the theoretical and from the practical point of view, the QAP shows an interesting asymptotic behavior, which suggests that under certain probabilistic conditions on the problem data, QAPs which are large enough are trivial to solve. Namely, it can be shown that the ratio between the "best" and "worst" values of the objective function approaches 1, as the size of the QAP approaches infinity. Thus the relative error of every heuristic method vanishes as the size of the problem tends to infinity, i.e., every heuristic finds almost always an almost optimal solution when applied to QAP instances which are large enough. A number of authors have investigated the asymptotic behavior of the QAP. (Burkard and Fincke, 1983; Burkard and Fincke, 1985) and (Frenk et al., 1985) have shown the convergence of the above mentioned ratio to 1 in probability. Later (Szpankowski, 1995) improved the convergence to almost surely. Summarizing we get the following theorem.

Theorem 4.1 *Consider a sequence of QAPs P_n, for $n \in \mathbb{N}$, with $n \times n$ coefficient matrices $A^{(n)} = \left(a_{ij}^{(n)}\right)$ and $B = \left(b_{ij}^{(n)}\right)$. Assume that $a_{ij}^{(n)}$ and $b_{ij}^{(n)}$, $n \in \mathbb{N}$, $1 \leq i, j \leq n$, are independently distributed random variables on $[0, M]$, where M is a positive constant. Moreover, assume that entries $a_{ij}^{(n)}$, $n \in \mathbb{N}$, $1 \leq i, j \leq n$, have a common distribution, and entries $b_{ij}^{(n)}$, $n \in \mathbb{N}$, $1 \leq i, j \leq n$, have also a common distribution (which does not necessarily coincide with that of $a_{ij}^{(n)}$). Furthermore, assume that these random variables have finite expected values, variances and third*

moments.
Let $\pi_{opt}^{(n)}$ *and* $\pi_{wor}^{(n)}$ *denote an optimal and a worst solution of* P_n *with their objective function values* $z\left(\pi_{opt}^{(n)}\right)$ *and* $z\left(\pi_{wor}^{(n)}\right)$*, respectively. Then the following equality holds almost surely* $\lim_{n\to\infty} z\left(\pi_{opt}^{(n)}\right)/z\left(\pi_{wor}^{(n)}\right) = 1$

The asymptotic behavior of the QAP has been exploited by (Dyer et al., 1986) to analyze the performance of branch and bound algorithms for QAPs with coefficients generated randomly as described above. Dyer et al. have shown that any branch and bound algorithm that uses single assignment branching and employs a bound obtained by solving the continuous relaxation of the linearization of Frieze and Yadegar would branch on at least $n^{(1-o(1))n/4}$ nodes with probability tending to 1 as the size n of the problem tends to infinity.

4.9 The biquadratic assignment problem

A generalization of the QAP arises if we consider objective functions of higher degree and obtain in this way *cubic*, *biquadratic* and generally N*-adic* assignment problems as introduced by (Lawler, 1963). In this way we get among other also *the biquadratic assignment problem*, denoted by BiQAP and stated as follows:

$$\begin{aligned} \min \quad & \sum_{i,j=1}^{n}\sum_{k,l=1}^{n}\sum_{m,p=1}^{n}\sum_{s,t=1}^{n} a_{ijkl}b_{mpst}x_{im}x_{jp}x_{ks}x_{lt} \\ \text{s.t.} \quad & X = (x_{ij}) \in \mathbf{X}_n, \end{aligned}$$

where $A = (a_{ijkl})$ and $B = (b_{mpst})$ are two $n^4 \times n^4$ arrays.
An application of the BiQAP arises in Very Large Scale Integrated (VLSI) circuit design. A detailed description of the mathematical modeling of the VLSI problem as a BiQAP is given by (Burkard et al., 1994). Similarly to the QAP also the BiQAP is a hard problem and cannot be solved efficiently. Gilmore-Lawler-like lower bounds, branch and bound algorithms and some local search heuristics as well as a GRASP implementation for the BiQAP can be found in the literature. See (Burkard and Çela, 1997) or (Çela, 1998) for more information and reference pointers.

4.10 The Bottleneck QAP

Another problem related to the QAP is the *bottleneck quadratic assignment problem* (BQAP), obtained by substituting the sum by a *max* operation in the objective function of the QAP:

$$\min_{\phi\in\mathcal{S}_n} \max\{a_{ij}b_{\phi(i)\phi(j)} : 1 \le i,j \le n\}\,.$$

The first occurrence of the BQAP in the literature is due to (Steinberg, 1961) and arises as an application in backboard wiring while trying to minimize the maximum length of the involved wires. A well studied problem in graph theory which can be modeled as a BQAP is the *bandwidth problem*. In the bandwidth problem we are given an undirected graph $G = (V, E)$ with vertex set V and edge set E, and seek a labeling of the vertices of G by the numbers $1, 2, \ldots, n$, where $|V| = n$, such that the maximum distance of 1-entries of the resulting adjacency matrix from the diagonal is minimized, i.e., the bandwidth of the adjacency matrix is minimized.
Analogously to the QAP also the BQAP is a hard problem which cannot be solved efficiently. Some enumeration algorithms to solve BQAP to optimality have been proposed by (Burkard, 1974). Those algorithms employ a Gilmore-Lawler-like bound for the BQAP.
(Burkard and Fincke, 1982) investigated the asymptotic behavior of the BQAP and proved results analogous to those obtained for the QAP: Under certain probabilistic constraints on the problem data, the relative difference between the worst and the best value of the objective function approaches 0 with probability tending to 0 as the size of the problem approaches infinity.

References

Adams, W. P. and Johnson, T. A. (1994). Improved linear programming-based lower bounds for the quadratic assignment problem. In Pardalos, P. M. and Wolkowicz, H., editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 43–75. AMS, (Providence, RI).

Ahuja, R. K., Magnanti, T. L., Orlin, J. B., and Reddy, M. R. (1995). Applications of network optimization. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, *Network Models - Handbooks of Operations Research and Management Science 7*, pages 1–83. Elsevier, Amsterdam.

Alt, H., Blum, N., Mehlhorn, K., and Paul, M. (1991). Computing maximum cardinality matching in time $o(n^{1.5}\sqrt{m/\log n})$. *Information Process. Letters*, 37:237–240.

Balas, E. and Mazzola, J. B. (1984a). Nonlinear programming: I. linearization techniques. *Mathematical Programming*, pages 1–21.

Balas, E. and Mazzola, J. B. (1984b). Nonlinear programming: Ii. dominance relations and algorithms. *Mathematical Programming*, 30:22–45.

Balas, E. and Saltzman, M. J. (1991). An algorithm for the three-index assignment problem. *Operations Research*, 39:150–161.

Bazaraa, M. S. and Sherali, H. D. (1982). On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. *Journal of Operations Research Society*, 33:991–1003.

Bertsekas, D. P., non, D. A. C., Eckstein, J., and Zenios, S. (1995). Parallel computing in network optimization. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, *Network Models - Handbooks in Operations Research and Management Science 7*, pages 330–399. Elsevier, Amsterdam, The Netherlands.

Birkhoff, G. (1946). Tres observaciones sobre el algebra lineal. *Rev. univ. nac. Tucumán (A)*, 5:147–151.

Bruengger, A., Clausen, J., Marzetta, A., and Perregaard, M. (1997). Joining forces in solving large-scale quadratic assignment problems in parallel. In *Proceedings of the 11-th IEEE International Parallel Processing Symposium (IPPS)*, pages 418–427.

Burkard, R. E. (1974). Quadratische bottleneckprobleme. *Operations Research Verfahren*, 18:26–41.

Burkard, R. E. and Çela, E. (1997). Quadratic and three-dimensional assignments. In Dell'Amico, M., Maffioli, F., and Martello, S., editors, *Annotated Bibliographies in combinatorial Optimization*, pages 373–392. Wiley, Chichester.

Burkard, R. E. and Çela, E. (1999). Linear assignment problems and extensions. In Pardalos, P. and Du, D.-Z., editors, *Handbook of Combinatorial Optimization 4*. Kluwer Academic Publishers. to appear.

Burkard, R. E., Çela, E., and Klinz, B. (1994). On the biquadratic assignment problem. In Pardalos, P. M. and Wolkowicz, H., editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 117–147. AMS, (Providence, RI).

Burkard, R. E., Çela, E., Pardalos, P., and Pitsoulis, L. (1998). The quadratic assignment problem. In Pardalos, P. and Du, D.-Z., editors, *Handbook of Combinatorial Optimization 3*, pages 241–338. Kluwer Academic Publishers.

Burkard, R. E. and Derigs, U. (1980). *Assignment and Matching Problems: Solution Methods with FORTRAN Programs*. Springer, Berlin.

Burkard, R. E. and Fincke, U. (1982). On random quadratic bottleneck assignment problems. *Mathematical Programming*, 23:227–232.

Burkard, R. E. and Fincke, U. (1983). The asymptotic probabilistic behavior of the quadratic sum assignment problem. *Zeitschrift für Operations Research*, 27:73–81.

Burkard, R. E. and Fincke, U. (1985). Probabilistic asymptotic properties of some combinatorial optimization problems. *Discrete Applied Mathematics*, 12:21–29.

Burkard, R. E., Hahn, W., and Zimmermann, U. (1977). An algebraic approach to assignment problems. *Mathematical Programming*, 12:318–327.

Burkard, R. E., Karisch, S. E., and Rendl, F. (1997). Qaplib - a quadratic assignment problem library. *Journal of Global Optimization*, 10:391–403. An on-line version is available via World Wide Web at the following URL: `http://www.opt.math.tu-graz.ac.at/~karisch/qaplib/`.

Burkard, R. E. and Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17:169–174.

Burkard, R. E. and Rendl, F. (1991). Lexicographic bottleneck problems. *Operations Research Letters*, 10:303–308.

Burkard, R. E. and Zimmermann, U. (1982). Combinatorial optimization in linearly ordered semimodules: a survey. In Korte, B., editor, *Modern Applied Mathematics*, pages 392–436. North Holland, Amsterdam.

Carpaneto, G., Martello, S., and Toth, P. (1988). Algorithms and codes for the assignment problem. *Annals of Operations Research*, 13:193–223.

Carpaneto, G. and Toth, P. (1980). Solution of the assignment problem. *ACM Transactions on Mathematical Software*, 6:104–111.

Carraresi, P. and Malucelli, F. (1994). A reformulation scheme and new lower bounds for the qap. In Pardalos, P. and Wolkowicz, H., editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 147–160. AMS, Providence, RI.

Çela, E. (1998). *The Quadratic Assignment Problem: Theory and Algorithms.* Kluwer Academic Publishers, Dordrecht.

Clausen, J. and Perregaard, M. (1997). Solving large quadratic assignment problems in parallel. *Computational Optimization and Applications*, 8:111–127.

Connolly, D. T. (1990). An improved annealing scheme for the qap. *European Journal of Operational Research*, 46:93–100.

Cunningham, W. H. (1976). A network simplex method. *Mathematical Programming*, 11:105–116.

Dell'Amico, M. and Martello, S. (1997). Linear assignment. In Dell'Amico, M., Maffioli, F., and Martello, S., editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 355–372. Wiley, Chichester.

Dell'Amico, M. and Toth, P. (2000). Algorithms and codes for dense assignment problems. To appear.

Derigs, U. (1985). The shortest augmenting path for solving assignment problems - motivation and computational experience.

Dyer, M. E., Frieze, A. M., and McDiarmid, C. J. H. (1986). On linear programs with random costs. *Mathematical Programming*, 35:3–16.

Espersen, T., Karisch, S. E., Çela, E., and Clausen, J. QAPPACK- a java package for solving quadratic assignment problems. working paper, Department of Mathematical Modelling, Technical University of Denmark, Denmark, and Institute of Mathematics, Technical University Graz, Austria.

Euler, R. and Le Verge, H. (1996). Time-tables, polyhedra and the greedy algorithm. *Discrete Applied Mathematics*, 65:207–221.

Even, S. and Tarjan, R. E. (1975). Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4:507–518.

Finke, G., Burkard, R. E., and Rendl, F. (1987). Quadratic assignment problems. *Annals of Discrete Mathematics*, 31:61–82.

Frenk, J. B. G., van Houweninge, M., and Kan, A. H. G. R. (1985). Asymptotic properties of the quadratic assignment problem. *Mathematics of Operations Research*, 10:100–116.

Frenk, J. B. G., van Houweninge, M., and Rinnooy Kan, A. H. G. (1987). Order statistics and the linear assignment problem. *Computing*, bf 39:165–174.

Frieze, A. M. and Yadegar, J. (1983). On the quadratic assignment problem. *Discrete Applied Mathematics*, 5:89–98.

Fulkerson, R., Glicksberg, I., and Gross, O. (1953). A production line assignment problem. Technical Report RM–1102, The Rand Corporation, Sta. Monica, CA.

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

Goldberg, A. V. and Kennedy, R. (1995). An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 75:153–177.

Hahn, P. and Grant, T. (1998). Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research*, 46:912–922.

Hahn, P., Grant, T., and Hall, N. (1998). Solution of the quadratic assignment problem using the hungarian method. *European Journal of Operational Research*, 108:629–640.

Hopcroft, J. E. and Karp, R. M. (1973). An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231.

Johnson, D. S. and McGeoch, C. C., editors (1993). *Network Flows and Matching - First DIMACS Implementation Challenge*, volume 12 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS, Providence, RI.

Jonker, R. and Volgenant, A. (1986). Improving the hungarian assignment algorithm. *Operations Research Letters*, 5:171–175.

Kaibel, V. (1997). *Polyhedral Combinatorics of the Quadratic Assignment Problem*. PhD thesis, Universität zu Köln, Germany.

Karisch, S. E., Çela, E., Clausen, J., and Espersen, T. (1999). A dual framework for lower bounds of the quadratic assignment problem based on linearization. *Computing*, 63:351–403.

Karp, R. M. (1980). An algorithm to solve the $m \times n$ assignment problem in expected time $o(mn \log n)$. *Networks*, 10:143–152.

Karp, R. M. (1987). An upper bound on the expected cost of an optimal assignment. In *Discrete Algorithms and Complexity*, pages 1–4. Academic Press, Boston.

Karp, R. M., Kan, A. H. G. R., and Vohra, R. V. (1994). Average case analysis of a heuristic for the assignment problem. *Mathematics of Operations Research*, 19:513–522.

Kaufmann, L. and Broeckx, F. (1978). An algorithm for the quadratic assignment problem using benders' decomposition. *European Journal of Operational Research*, 2:204–211.

Laporte, G. and Mercure, H. (1988). Balancing hydraulic turbine runners: A quadratic assignment problem. *European Journal of Operational Research*, 35:378–382.

Lawler, E. L. (1963). The quadratic assignment problem. *Management Science*, 9:586–599.

Lee, Y. and Orlin, J. B. (1994). On very large scale assignment problems. In Hager, W. W., Hearn, D. W., and Pardalos, P. M., editors, *Large Scale Optimization: State of the Art*, pages 206–244. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Magos, D. (1996). Tabu search for the planar three-index assignment problem. *Journal of Global Optimization*, 8:35–48.

Magos, D. and Miliotis, P. (1994). An algorithm for the planar three-index assignment problem. *European Journal of Operational Research*, 77:141–153.

Martello, S., Pulleyblank, W. R., Toth, P., and de Werra, D. (1984). Balanced optimization problems. *Operations Research Letters*, 3:275–278.

Mulmuley, K., Vazirani, U. V., and Vazirani, V. V. (1087). Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113.

Nugent, C. E., Vollmann, T. E., and Ruml, J. (1969). An experimental comparison of techniques for the assignment of facilities to locations. *Journal of Operations Research*, 16:150–173.

Olin, B. (1992). *Asymptotic Properties of Random Assignment Problems*. PhD thesis, Division of Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, Stockholm.

Padberg, M. W. and Rijal, M. P. (1996). *Location, Scheduling, Design and Integer Programming*. Kluwer Academic Publishers, Boston.

Pardalos, P. and Crouse, J. (1989). A parallel algorithm for the quadratic assignment problem. In *Proceedings of the Supercomputing Conference 1989*, pages 351–360. ACM Press.

Pardalos, P. M., Pitsoulis, L. S., and Resende, M. G. C. (1997). Fortran subroutines for approximate solution of sparse quadratic assignment problems using grasp. *ACM Transcations on Mathematical Software*, 23:196–208.

Pferschy, U. (1996). The random linear bottleneck assignment problem. *RAIRO Operations Research*, 30:127–142.

Pferschy, U. (1997). Solution methods and computational investigations for the linear bottleneck assignment problem. *Computing*, 59:237–258.

Pierskalla, W. P. (1967). The tri-substitution method for the three-multidimensional assignment problem. *Canadian ORS Journal*, 5:71–81.

Punnen, A. P. and Nair, K. P. K. (1994). Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem. *Discrete Applied Mathematics*, 55:91–93.

Ramakrishnan, K. G., Karmarkar, N. K., and Kamath, A. P. (1993). An approximate dual projective algorithm for solving assignment problems. In Johnson, D. S.

and McGeoch, C. C., editors, *Network flows and matching– First DIMACS Implementation Challenge*, volume 12 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 431–449. AMS, Providence, RI.

Resende, M. G. C., Pardalos, P. M., and Li, Y. (1996). Fortran subroutines for approximate solution of dense quadratic assignment problems using grasp. *ACM Transcations on Mathematical Software*, 22:104–118.

Resende, M. G. C., Ramakrishnan, K. G., and Drezner, Z. (1995). Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming. *Operations Research*, 43:781–791.

Steinberg, L. (1961). The backboard wiring problem: A placement algorithm. *SIAM Review*, 3:37–50.

Szpankowski, W. (1995). Combinatorial optimization problems for which almost every algorithm is asymptotically optimal! *Optimization*, 33:359–367.

Taillard, E. (1991). Robust tabu search for the quadratic assignment problem. *Parallel Computing*, 17:443–455.

Vlach, M. (1967). Branch and bound method for the three-index assignment problem. *Ekonomicko-Matematicky Obzor*, 12:181–191.

West, D. H. (1983). Algorithm 608: Approximate solution of the quadratic assignment problem. *ACM Transactions on Mathematical Software*, 9:461–466.