Some Reformulations for the Quadratic Assignment Problem

Axel Nyberg

PhD Thesis in Process Design and Systems Engineering
Department of Chemical Engineering
Åbo Akademi University

Åbo, Finland 2014

ISBN 978-952-12-3019-6

Painosalama Oy
Åbo 2014

Preface

This thesis is the result of three years of working at the Center of Excellence in Optimization and Systems Engineering at Åbo Akademi University. I am very grateful to my adviser, Professor Tapio Westerlund for giving me this opportunity and for all the counseling as well as the interesting discussions both on and off topic. I also want to thank everybody at the Process Design and Systems Engineering Laboratory for a wonderful working environment. I especially want to thank Dr. Andreas Lundell for helping me with formatting and optimization related questions around twenty times a day. Without the help of Andreas, this thesis would have been written in Comic Sans and the illustrations would have been made using Microsoft Paint.

I am also very grateful to Professor Ignacio Grossmann for my stay at Carnegie Mellon University as well as all my colleagues and friends there.

A special thanks goes to the members of the exclusive "lunchklubben" for sharing the most important hour of the work-day with me. Finally, I want to thank Ina for all the happiness outside of work.

Åbo, January 2014
Axel Nyberg

Contribution of the author

The author is responsible for all the papers included in this thesis.

Paper I: This paper is based on my ideas for the quadratic assignment problem. The work was carried out together with Professor Tapio Westerlund.

Paper II: This paper continues on the previous paper and is based on a my ideas for improvement. The formulations were derived together with Dr. Andreas Lundell.

Paper III: This paper continues work on the two earlier papers.

Paper IV: The work for this paper was carried out during my stay at Carnegie Mellon University together with Professor Ignacio Grossmann.

Sammanfattning på svenska

Kombinatorisk optimering handlar om att hitta en bra eller rent av den bästa möjliga lösningen från ett känt antal lösningar eller kombinationer. Ofta är antalet lösningar så enormt att en genomgång av alla olika lösningar inte är möjlig. En av huvudorsakerna till att det forskas inom kombinatorisk optimering är att liknande frågeställningar eller problem uppkommer inom så många olika områden. Påståendet stämmer speciellt bra för kvadratiska tilldelningsproblem (*eng.* Quadratic Assignment Problem, QAP). Sådana problem uppstår då man försöker beskriva en stor mängd tillämpade frågeställningar. Vilken gate skall väljas för flygen på större flygplatser för att minimera den totala väg människorna behöver gå och bagaget förflyttas? Var skall olika avdelningar på en fabrik placeras för att minimera materialförflyttningar mellan avdelningarna? Hur ser ett optimalt tangentbord ut för olika språk? Var skall komponenterna placeras på ett kretskort? De här är alla frågor som kan besvaras genom att lösa kvadratiska tilldelningsproblem. Kvadratiska tilldelningsproblem är dock mycket svåra att lösa. Det beror på att problemet i den standardform det matematiskt formuleras i huvudsak består av produkter av binära variabler.

I denna avhandling har problemet omformulerats till en linjär diskret form som innehåller färre variabler. Med omformuleringen har bland annat flera tidigare olösta kvadratiska tilldelningsproblem kunnat lösas till globalt optimum, den bästa möjliga lösningen, för första gången någonsin. Avhandlingen börjar med en kort introduktion till optimering. Sedan en genomgång av kvadratiska tilldelningsproblem samt de nya lösningsmetoderna. Slutligen finns en kort sammanfattning samt fyra publicerade artiklar.

Contents

CHAPTER 1

Introduction

In a highly globalized world, optimization of the processes in a company is essential for keeping its competitiveness. As computers, solvers and formulation techniques improve, more and more complex problems can be solved to optimality. When I tell people I work with optimization the usual follow up question is “optimization of what?” Optimization of anything and everything is the only real answer to that question. Since my work is in the field of combinatorial optimization, finding real world examples is not very difficult. Explaining the fact that optimizing the design of a computer chip can mathematically be exactly the same as optimizing gate assignments at airports might, on the other hand, be more challenging. However, the versatility of the fields of application where some of the well known combinatorial problems are found is also the reason why they are important to study. The above statement holds particularly true for the quadratic assignment problem (QAP).

1.1 Optimization

There are different classes of optimization problems. Linear programming (LP) problems consist of a linear objective, linear constraints and only continuous variables. LPs are usually solved using the simplex method readily implemented in most commercial solvers. If any of the variables in the problem are integer (or binary) then it is a mixed integer linear programming (MILP) problem. Binary variables usually represent decisions in the problem, *e.g.*, $x_1 = 1$ if a machine is bought and $x_1 = 0$ otherwise. Integer variables can, for example, represent the number of units to manufacture. For solving MILP problems branch and bound or cutting plane methods can be used. A combination of the two methods, called branch and cut, is the most widely used by commercial software and is implemented in solvers such as Cplex, Mosek and Gurobi. In a branch and bound approach, all the integer requirements of the MILP problem are first discarded. This makes it an LP relaxation which is used for obtaining a valid lower bound (LB) of the

MILP problem. The problem is then branched into two subproblems. This is done on one of the integer variables with a fractional value in the solution of the LP relaxation. For a binary variable where $0 < x_1 < 1$ in the LP relaxation, two new subproblems are created, one where $x_1 = 1$ and one where $x_1 = 0$. These new LP subproblems are solved and branched again. This gives rise to a tree structure where the branches represent different subproblems. After a number of iterations all integer variables have integer values and a valid solution, which is also an upper bound (UB) for the problem, is found. If any open subproblem has a lower bound greater than (if minimizing) the best found upper bound, the node and all subproblems created from that node can be discarded. This is called pruning and is essentially the step that inhibits the search tree from becoming too large. When the best found upper bound coincides with the lower bound the optimal solution has been found.

In a nonlinear programming (NLP) problem, the objective or some of the constraints are nonlinear, but all variables are continuous. These classes are all subclasses of mixed integer nonlinear programming (MINLP) problems where some of the variables are integers. One example of MINLP is a problem with a bilinear term $x_1 x_2$ with two integer or binary variables. Bilinear terms with binary variables stem from if and only if decisions, *e.g.* if both machine 1 ($x_1 = 1$) and machine 2 ($x_2 = 1$) are built, a warehouse has to be built as well ($w = 1$). Then the warehouse can be modeled with $w = x_1 x_2$.

MINLP problems can, in some cases, be linearized to MILP problems by adding additional variables and constraints and reformulating the problem. Since MILP problems are generally easier to solve than MINLP problems, reformulations are often used if possible. A bilinear term $x_1 x_2$ with two binary variables can easily be reformulated with an additional positive continuous variable and a single constraint $w \geq x_1 + x_2 - 1$. Depending on the problem there might be many ways to reformulate the model. The way a model is formulated will impact the lower bound, the solution time as well as the optimality gap if the solution process is aborted before optimality is proven. For large problems choosing the right formulation may dictate if the problem can be solved or not. A common approach for solving MINLP problems with only a few nonlinear constraints is using piecewise linear relaxations. In this way, the MINLP can be solved as an MILP to a predefined optimality gap. In this thesis, the optimality gap is calculated with the following equation:

$$\text{gap} = \frac{\text{UB–LB}}{\text{UB}} \times 100\%. \tag{1.1}$$

The optimal solution of a problem lies somewhere between its UB and LB. A problem is therefore solved to optimum when the UB and LB reach the same value and the gap is zero.

1.2 Scope of work

This thesis discusses new solution methods for the QAP and reformulations for bilinear terms. It is divided into two parts; the first part consists of a short summary of the QAP, its applications, model formulations and some techniques for solving it. The second part consists of four peer-reviewed articles on the subject. Some of the methods presented in the articles have been used to solve certain instances of the QAP, from the QAP library (Burkard et al., 1997), that have remained unsolved since 1990. The same type of reformulation of bilinear terms are also used in a different setting for solving a multiechelon supply chain model.

1.3 List of publications

The work described in this thesis is based mainly on the following scientific papers:

Paper I: Nyberg, A., Westerlund, T., 2012. A new exact discrete linear reformulation of the quadratic assignment problem. European Journal of Operational Research 220 (2), 314–319.

Paper II: Nyberg, A., Westerlund, T., Lundell, A., 2013. Improved discrete reformulations for the quadratic assignment problem. In: Gomes, C., Sellmann, M. (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Vol. 7874 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 193–203.

Paper III: Nyberg A. and Westerlund T., 2013. Tightening a discrete formulation of the quadratic assignment problem, Chemical engineering transactions, 32, 1309–1314.

Paper IV: Nyberg, A., Grossmann, I.E., Westerlund, T., 2013. An efficient reformulation of the multiechelon stochastic inventory system with uncertain demands. AIChE Journal 59 (1), 23–28.

CHAPTER 2

The quadratic assignment problem

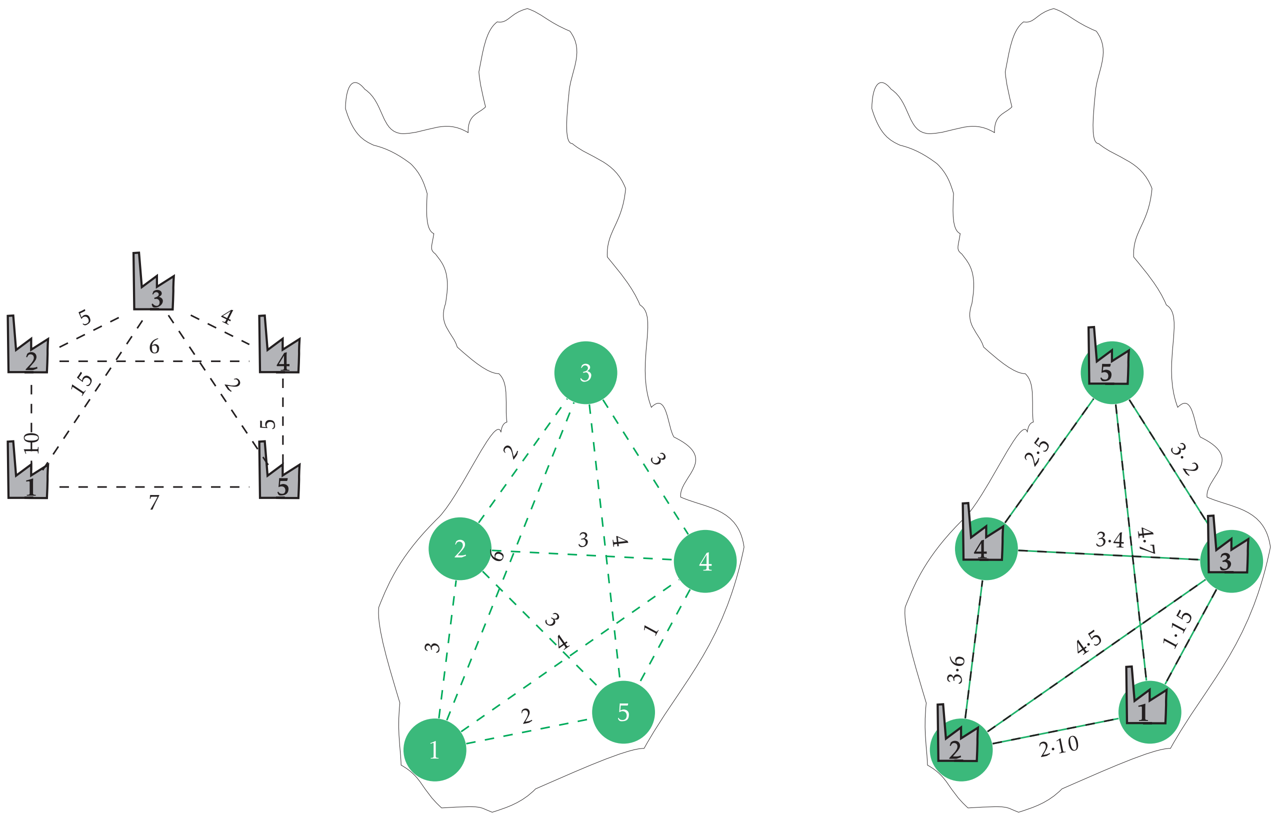
The QAP was presented in the literature by Koopmans and Beckmann (1957) as a mathematical model to allocate economic activities to specific locations. The QAP is often referred to as a facility location problem that determines the best placing of facilities on a predefined set of locations. The problem can be illustrated with two matrices, where one corresponds to the distances between the locations while the other corresponds to the flows between the facilities. The problem is then to find the optimal permutation vector $\tilde{p}$ minimizing total cost, *i.e.*

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\tilde{p}_i \tilde{p}_j}, \tag{2.1}$$

where $\tilde{p}_i$ and $\tilde{p}_j$ are elements of the vector $\tilde{p}$ while a_{ij} and $b_{\tilde{p}_i \tilde{p}_j}$ are elements in the distance and flow matrices respectively. In this form, the QAP is a permutation problem for which no direct efficient global optimization method is available today. Koopmans and Beckmann's approach is given in section 2.2, where the QAP is formulated as a zero-one mathematical programming problem. An illustration of the facilities, locations and the optimal assignment for a small example are shown in fig. 2.1 with their corresponding distance and flow matrices.

2.1 Applications

The number of applications found in the literature for the QAP is quite astonishing. Especially interesting is the fact that the same kinds of problems arise from completely different fields. Dickey and Hopkins (1972) assigned buildings on a university campus in order to minimize traffic on the campus area. Elshafei (1977) studied the Ahmed



$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 6 & 4 & 2 \\ 3 & 0 & 2 & 3 & 3 \\ 6 & 2 & 0 & 3 & 4 \\ 4 & 3 & 3 & 0 & 1 \\ 2 & 3 & 4 & 1 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 10 & 15 & 0 & 7 \\ 10 & 0 & 5 & 6 & 0 \\ 15 & 5 & 0 & 4 & 2 \\ 0 & 6 & 4 & 0 & 5 \\ 7 & 0 & 2 & 5 & 0 \end{bmatrix}$$

Figure 2.1: An illustration of a facility location problem and its optimal solution $\tilde{p} = [2,4,5,3,1]^T$. The factories to be allocated to specific locations are shown to the left. The picture to the right shows the optimal allocation of the factories. The matrices **A** and **B** are the distance and flow matrices.

Maher Hospital in Cairo, Egypt. The goal was to reorder the 17 clinics so that the transportation of patients between them was minimized. Krarup and Pruzan (1978) mention the design of the university hospital Klinikum Regensburg in Germany, also planned using QAPs. An interesting observation is that the instances were not solved to optimality until long after the hospitals were built. However, good solutions were probably used even though Hahn and Krarup (2001) mention that it was impossible for Krarup himself to know whether they had used the proposed solutions or not when he passed Klinikum Regensburg 20 years later.

Assigning gates for flights at airports in order to minimize the walking traffic for passengers can be modeled as QAPs. Haghani and Chen (1998) state that other versions of the same problems, for example, minimizing the amount of baggage that has to be moved or minimizing both passenger and baggage movement, can also be formulated as

QAPs.

One of the earliest applications of the QAP was the Steinberg wiring problem (Steinberg, 1961). The problem was to place computer components on a backboard to minimize the total wiring between components. In a newer approach to a similar problem, Rabak and Sichman (2003) minimize the amount of work the component placer has to do when inserting components on a printed circuit board. The same kinds of problems, addressing placement of electronic circuits in computer manufacturing, have been studied in Grötschel (1992) and Jünger et al. (1994). Also, the components themselves are usually microchips (or integrated circuits) consisting of numerous transistors. The creation of integrated circuits by placing transistors on a chip is called very large scale integrated design. Burkard et al. (1993) applied a generalized version of the QAP, called the biquadratic assignment problem, to minimize the number of transistors needed to model different states of the chip. This can also be done using finite state machines, but it too leads to an QAP (Eschermann and Wunderlich, 1990).

Laporte and Mercure (1988) studied turbine balancing in electricity generation. The idea is to arrange the blades in such a way that the center of gravity for the completed turbine becomes as close to the geometrical center of the cylinder as possible, thus maximizing stability of the turbine. The weight of the blades should be the same for each blade (around 17 000 kg) but due to manufacturing issues the weight can differ with around ± 5%. Therefore, QAPs are solved in order to determine which blade to place at what part of the cylindrical turbine. Fathi and Ginjupalli (1993) considered the same problem and proposed some better heuristics to solve it.

Forsberg et al. (1995) applied the QAP in chemistry while Phillips and Rosen (1994) optimized molecular conformations in biochemistry.

In order to optimize the layout of the typewriter keyboard for different languages, Burkard and Offermann (1977) used QAP. When writing running text in a specific language the letters appear with a known frequency. By checking the most common words in a language, the frequency of two letters appearing after each other can be gauged. Then, by solving a QAP the ordering of the letters on the keyboard can be optimized (w.r.t. writing speed). A more modern application of the same problem is the design of keyboards on touch screen devices (Dell'Amico et al., 2009). The main difference in this approach is that on a touch screen only one finger is used, and the letters can be placed anywhere on the screen instead of in a rectangle as with normal keyboards.

Ben-David and Malah (2005) looked into a special case of the QAP called index assignment in order to minimize channel errors in vector-quantization. Vector-quantization is used when mapping images or speech to digital signals. Yet another application in computer science is the mapping problem studied by Bokhari (1981). Here the objective is to place communicating modules to adjacent processors in a processor array. This can be written as a QAP where one of the matrices has only 0-1 entries. A similar mapping problem is also found when configuring the layout of microarrays, which is a problem in bioinformatics presented in Hannenhalli et al. (2002). This problem was presented as a QAP by de Carvalho Jr. and Rahmann (2006). In the production of the microarray chips,

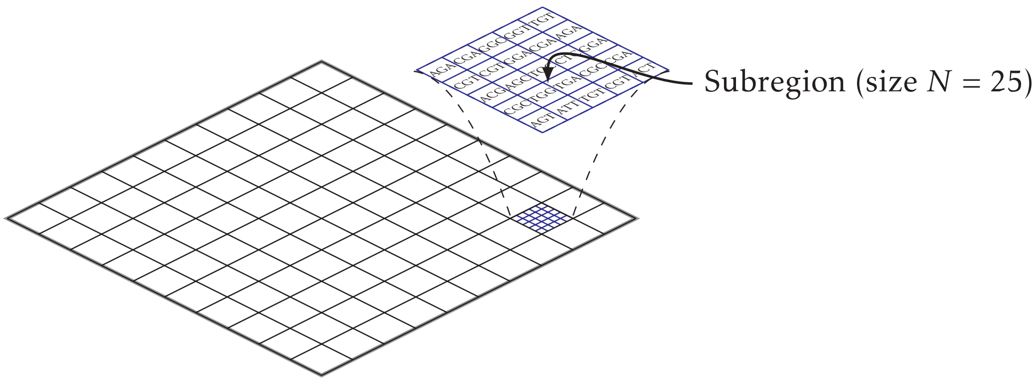


Figure 2.2: An illustration of a microarray chip with subsection

small DNA fragments on microarrays are arranged in such a way that the exposure to unwanted illumination is minimized. Since the chips themselves can contain over 10 million probes, only small segments of the chips can be optimized, as shown in fig. 2.2. Since calculating the optimal conformation for the whole chip is practically impossible with the solution approaches used today, a heuristic approach is proposed for solving the subproblems. However, these instances are also especially difficult for heuristic methods and therefore an effective exact method could be the best option for this class of problems.

Carlson and Nemhauser (1966) solved scheduling problems as quadratic programs very similar to the QAP. The assignment of classes at a university were scheduled in such a way that as few similar classes as possible would be in the same time slot. This study was extended in Davis et al. (1974) who also included conflicting assignments in the model. Geoffrion and Graves (1976) studied how to schedule production orders on production lines with many similar processes in parallel using QAP. They applied the model on a chemical reactor scheduling problem for Dart Industries Inc.

In addition to all the applications mentioned above, many of the well known difficult problems in combinatorial optimization, *e.g.* the traveling salesman problem and the maximum cut problem, can be expressed as QAPs (Çela, 1998).

2.1.1 The quadratic assignment problem library – QAPLib

Burkard et al. (1997) put together many of the real world, as well as, a few example instances to form a test library, QAPLib. This library is still updated and maintained online by Peter Hahn from University of Pennsylvania and Miguel F. Anjos from École Polytechnique de Montreal. The QAPLib lists lower bounds and best known solutions for 134 instances with sizes between $n = 12$ and $n = 256$. Optimality is proven for all instances where $n \leq 30$, except for the instance `tai30a`. However, only a handful of the instances where $n > 30$ have been solved to optimality. In the QAPLib, solver codes and links to articles on recent advances on the QAP are also listed.

2.2 Solution methods

The QAP was introduced by Koopmans and Beckmann (1957) in the basic form:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij} b_{kl} \cdot x_{ik} x_{jl} + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}, \tag{2.2}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,\dots,n, \tag{2.3}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\dots,n, \tag{2.4}$$

$$x_{ij} \in \{0,1\}, \qquad i,j = 1,\dots,n. \tag{2.5}$$

Here, a_{ij} are given distances between locations and b_{kl} flows between facilities defined in the matrices **A** and **B**. In this thesis we will assume that $c_{ij} = 0\ \forall i,j$ for brevity. Since this part is linear it can easily be added to any formulation. Also, we will write eqs. (2.3) to (2.5) as $x \in \mathbf{X_n}$ since these assignment constraints are found in all of the presented formulations.

2.2.1 Stochastic methods

Since the QAP is a very challenging combinatorial problem, a large number of different heuristics presented in the literature for combinatorial optimization have been implemented and tested also on the QAP. These approaches are often very useful when a close to optimal solution is sufficient. However, since the emphasis of the work in this thesis is on deterministic methods only a brief summary of heuristic approaches is given.

The most successful heuristics are tabu search approaches. Taillard (1991) applied robust tabu search on the QAP. Various improvements to this code has been proposed, *e.g.* Misevicius (2005) added some modifications, in order to avoid the stagnation of the search, with good results. Li et al. (1994) presented a greedy random adaptive search procedure, Connolly (1990) implemented simulated annealing and Ahuja et al. (2000) applied a genetic algorithm on the QAP, to name a few.

In practical applications where the values in the flow and distance matrices are approximations, optimality is not required or when many quick solutions are needed these methods are often the easiest to implement. Together with a good lower bounding method, heuristics can provide a reasonable optimality gap within short computational time.

2.2.2 Lower bounding techniques

Lower bound calculations are important for two different reasons. First, in order to have an idea of how good a solution a heuristic finds, a tight lower bound is important. Secondly, in a branch and bound environment a good lower bound is crucial for pruning nodes. However, in such a case emphasis should also be put on the computational speed of the lower bound calculation since it is performed in every node. Therefore the tightest lower bounding techniques might be too slow in a branch and bound code. Another critical drawback with many of the tighter lower bound techniques is the amount of RAM needed when the instances grow larger. One of the first and most widely used lower bounds for the QAP is the Gilmore-Lawler bound (GLB), presented in Gilmore (1962) and modified by Lawler (1963). The GLB is very fast to compute and relatively tight for small instances. However, as the instance size increases, the gap between the bound and the optimal solution increases quickly (Çela, 1998). Hahn and Grant (1998) derived a relaxation-linearization lower bounding technique based on the Hungarian method for solving linear assignment problems. The latest version of this code, the level-3 reformulation-linearization technique, is one of the tightest lower bounds for the QAP according to Hahn et al. (2012). However these methods have serious problems with RAM requirements when the problem sizes increases. The amount of memory required grows exponentially with instance size and, for example, an instance where $n = 25$ require 173 GB of RAM. Still, according to QAPLib, one of the really difficult problems, tai30b, was solved on the Palmetto Supercomputing Cluster at Clemson University using this code in a branch and bound environment.

Burer and Vandenbussche (2006) used a lift and project method to calculate good lower bounds on many quadratic programming problems including the QAP. The lift and project approach gives some very good bounds, but again, the solution times are extremely long if the problem is solved more than once.

A more approachable lower bound for calculations on a single computer is the bound arising from semidefinite programming (SDP) relaxations. The SDP relaxation of the QAP has been studied by numerous researchers, *e.g.*, de Klerk and Sotirov (2010), Peng et al. (2010) and Mittelmann and Peng (2010). These bounds are very tight and in fact most of the best known lower bounds on the unsolved instances in QAPLib are from SDP calculations by the mentioned authors. However, because of the huge amount of variables in the SDP relaxation of the QAP, the solution times are quite long when used as a bounding procedure in a branch and bound code. Anstreicher and Brixius (2001) derived a new bound using convex quadratic programming. This formulation was used in a parallel branch and bound environment to solve a few large instances to proven optimality for the first time on a huge computational grid (Anstreicher et al., 2002). Among these instances was the infamous nug30 that had remained unsolved since it was presented in Nugent et al. (1968). Around 1000 computers were used in parallel to solve nug30 and in total, the solution required seven days with an average of 653 computers connected to the grid.

2.2.3 Exact methods

Most of the exact methods for solving the QAP are based on MILP formulations and branch and bound. In general, compact formulations with fewer variables are fast to calculate, but provide a weak lower bound from the LP relaxation. Some of the tighter formulations on the other hand use so many variables that the LP relaxation takes a very long time to calculate. Finding a good balance between bound quality and problem size is a challenge when working with MILP formulations. The linearization resulting in one of the smallest model sizes is the one presented by Kaufman and Broeckx (1978). Their formulation has n^2 binary variables, n^2 continuous variables and n^2 constraints. However, the linear relaxation of this formulation is extremely weak with a lower bound equal to zero in the root node in every case (Zhang et al., 2013). Xia and Yuan (2006) propose two formulations similar to Kaufman and Broeckx that use the GLB to tighten the formulations. Both Zhang et al. (2013) and Fischetti et al. (2012) use models derived from Kaufman and Broeckx. Clearly, this formulation is well studied thanks to its small model size.

Frieze and Yadegar (1983) proposed a formulation replacing every bilinear term in the Koopmans-Beckmann formulation with a new variable. This formulation is huge, with n^4 continuous variables, n^2 binary variables and $n^4 + 4n^3 + n^2 + 2n$ constraints (Çela, 1998). Since solving such a huge MILP can be considered intractable (especially in 1983) a lower bounding technique based on Lagrangian relaxation was derived.

As can be seen from the applications in section 2.1, the QAP is an extremely important problem class since it arises in so many different fields. As mentioned in Hahn and Krarup (2001), there might not be a need for solving these real world QAPs to global optimum since the values in the instances are approximations themselves. However, since the applications where QAPs arise are so diverse, it can be argued that there is need for both exact methods and heuristics. Drezner et al. (2005) present instances that are really difficult for heuristics, while relatively easy to solve with exact methods. For example the `dre` instances presented in that paper are solved to optimality in a fraction of a second even for the largest instances using the GLB. Still the heuristics fail to find the optimal solution in a much longer time. Also, as modern QAP applications tend to stem from computer science (*e.g.* bandwidth minimization) the flow and distance matrices give an exact description of the underlying problem and therefore the exact optimum might be of interest.

CHAPTER 3

Formulations

In Paper I we show how to rewrite the objective function of the QAP to a previously unpublished form in order to decrease the number of bilinear terms. Consider two permutation vectors $\mathbf{p}$ and $\tilde{\mathbf{p}}$ where $p_i = k$ if facility i is at location k and $\tilde{p}_i = k$ if facility k is at location i. The objective function in eq. (2.2) can then be written as:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{p_i p_j} b_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{\tilde{p}_i \tilde{p}_j}. \tag{3.1}$$

In matrix form eqs. (2.3) to (2.4) can be written as $\mathbf{Xe} = \mathbf{X}^T\mathbf{e} = \mathbf{e}$ where $\mathbf{e}$ is a vector with all elements equal to 1 and $\mathbf{X}$ is a so called permutation matrix containing the binary variables x_{ij}. The permutation vectors $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are given by $\mathbf{p} = \mathbf{Xq}$ and $\tilde{\mathbf{p}} = \mathbf{X}^T\mathbf{q}$ where $\mathbf{q}^T = (1, 2, \ldots, n)$. Using the properties of the permutation vectors, eq. (3.1) can be rewritten in the form:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{p_i j} b_{i \tilde{p}_j} = \sum_{i=1}^{n}\sum_{j=1}^{n} a'_{ij} b'_{ij}. \tag{3.2}$$

By algebraic manipulation we obtain a new MINLP formulation of the QAP:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} a'_{ij} b'_{ij}, \tag{3.3}$$

subject to

$$a'_{ij} = \sum_{k=1}^{n} a_{kj} x_{ik} \qquad \forall i, j, \tag{3.4}$$

$$b'_{ij} = \sum_{k=1}^{n} b_{ik} x_{kj} \qquad \forall i, j, \tag{3.5}$$

$$x \in \mathbf{X_n}. \tag{3.6}$$

This formulation can be illustrated by moving the columns in one of the matrices while moving the rows in the other matrix. Therefore, each term $a'_{ij}b'_{ij}$ is equal to one specific element in column j of matrix **A** times an element in row i of the matrix **B**.

The above QAP form has only n^2 bilinear terms instead of $n^2(n-1)^2$ as in the original form by Koopmanns and Beckmann. The variables a'_{ij} and b'_{ij} are continuous and therefore the bilinear terms are now products of two continuous variables instead of two binary variables. Continuous bilinear terms are generally difficult to linearize. However, we take advantage of the fact that the variables can only assume values from the matrices **A** and **B** and are therefore essentially discrete.

3.1 Formulating the discrete bilinearities

Figure 3.1 shows the possible combinations for the term $a'_{11}b'_{11}$ where each dot equals a specific value. As can be seen, both a'_{11} and b'_{11} share one variable, x_{11}. The combinations in gray are therefore not possible because of eqs. (2.3) and (2.4). The variable x_{ij} is always found in both a'_{ij} and b'_{ij}. This is the variable that corresponds to the diagonals of both **A** and **B**. The values of the diagonal elements are usually zero. If not, then the diagonals can always be set to zero by adding a linear part $\sum_{i=1}^{n}\sum_{j=1}^{n} a_{jj}b_{ii}x_{ij}$ to the objective function. In the following formulations it is assumed that the diagonal elements are zero. Therefore, if $x_{ij} = 1$, then a'_{ij} and b'_{ij} are both zero. The diagonal elements are therefore excluded in every example and sum where the number of unique values is calculated. By taking advantage of the discrete values of the variables, the bilinear term can be written in linear form and the QAP reformulated to an MILP problem. Next we will present a few different ways of linearizing bilinear terms starting with the most widely used, the McCormick underestimators.

3.1.1 McCormick envelopes

McCormick (1976) presented the following over- and underestimators w for a bilinear term xy:

$$\begin{aligned}
w &\geq x^{LB}y + y^{LB}x - x^{LB}y^{LB}, \\
w &\geq x^{UB}y + y^{UB}x - x^{UB}y^{UB}, \\
w &\leq x^{UB}y + y^{LB}x - x^{UB}y^{LB}, \\
w &\leq x^{LB}y + y^{UB}x - x^{LB}y^{UB}.
\end{aligned} \tag{3.7}$$

Here uppercase LB and UB denote the lower and upper bounds of the variables respectively. These so called McCormick envelopes are exact in the cases when either x or y are at one of their bounds. These underestimators are therefore an exact reformulation of a bilinear term as long as one of the variables can only obtain two different values, for example when one of the variables is binary. When minimizing over w, the two latter constraints are redundant. The McCormick underestimators can easily be applied on the QAP in standard form (eq. (2.2)) where all variables are binary. Since the lower bound is zero for a binary variable, the first underestimator will take the form $w \geq 0$ and can be

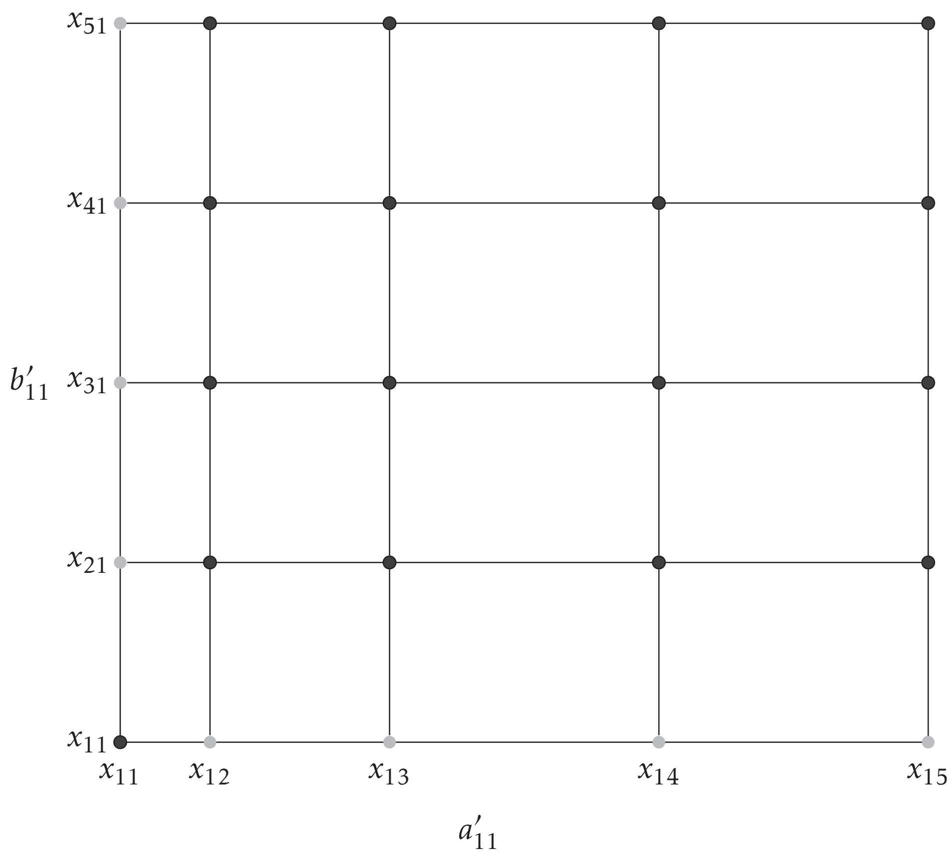


Figure 3.1: Possible nodes for the bilinear term $a'_{11}b'_{11}$. Since both a'_{11} and b'_{11} share one binary variable and exactly one of the binary variables in each direction equals one, the dots in lighter gray are not possible.

left out since all the variables w are defined as positive in the MILP solver. However, this formulation will always yield a LP solution at the root node where all binaries have an equally large fractional value and a lower bound of zero.

An exact reformulation can also be obtained for discrete bilinear terms using multiple McCormick envelopes. This can be done by extending the expressions such that the envelopes are active in decreasing regions and relaxed outside using a big-M formulation. When rewriting the objective function (eq. (3.3)) with these formulations we add one new variable w_{ij} and two constraints per McCormick envelope c, for every bilinear term $a'_{ij}b'_{ij}$.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \tag{3.8}$$

$$\left.\begin{aligned} w_{ij} &\geq \underline{A}_j^c b'_{ij} + \underline{B}_i^c a'_{ij} - \underline{A}_j^c \underline{B}_i^c - \overline{A}_j^1 \overline{B}_i^1 \left(2 - \sum_{k \in \mathbf{C}_j^c} x_{ik} - \sum_{k \in \mathbf{D}_i^c} x_{kj}\right) \\ w_{ij} &\geq \overline{A}_j^c b'_{ij} + \overline{B}_i^c a'_{ij} - \overline{A}_j^c \overline{B}_i^c - \overline{A}_j^1 \overline{B}_i^1 \left(2 - \sum_{k \in \mathbf{C}_j^c} x_{ik} - \sum_{k \in \mathbf{D}_i^c} x_{kj}\right) \end{aligned}\right\} \forall c, i, j, \tag{3.9}$$

where

$$c = 1, \ldots, \min\left\{\left\lceil \frac{N_j}{2} \right\rceil, \left\lceil \frac{M_i}{2} \right\rceil\right\}.$$

N_j is the number of unique elements in column j of the matrix $\mathbf{A}$ and M_i is the number of unique elements of row i of matrix $\mathbf{B}$. $\underline{A}_j^c$, $\overline{A}_j^c$ and $\underline{B}_i^c$, $\overline{B}_i^c$ are the lower and upper bounds of the variables a'_{ij} and b'_{ij} respectively in the corresponding envelope c.

The lower and upper bounds for the variables in each envelope are given by the unique elements of the corresponding column and row of the $\mathbf{A}$ and $\mathbf{B}$ matrix. For the first envelope the lower and upper bounds are given by the lower and upper bound of all the unique elements of the corresponding column and row:

$$\underline{A}_j^1 = \min_{i, i \neq j} a'_{ij}, \quad \overline{A}_j^1 = \max_{i, i \neq j} a'_{ij} \quad \forall j, \tag{3.10}$$

$$\underline{B}_i^1 = \min_{j, i \neq j} b'_{ij}, \quad \overline{B}_i^1 = \min_{j, i \neq j} b'_{ij} \quad \forall i. \tag{3.11}$$

$\mathbf{C}_j^c$ and $\mathbf{D}_i^c$ are the sets of indices, connected to the binary variables x_{ij}, in eq. (3.9) for which the variables a'_{ij} and b'_{ij} will obtain values within the bounds specified by $\underline{A}_j^c$, $\overline{A}_j^c$ and $\underline{B}_i^c$, $\overline{B}_i^c$ according to,

$$\mathbf{C}_j^c = \left\{ i \mid \underline{A}_j^c \leq a'_{ij} \leq \overline{A}_j^c \wedge i \neq j \right\} \quad \forall j, \tag{3.12}$$

$$\mathbf{D}_i^c = \left\{ j \mid \underline{B}_i^c \leq b'_{ij} \leq \overline{B}_i^c \wedge i \neq j \right\} \quad \forall i. \tag{3.13}$$

The lower and upper bounds in eq. (3.12) and eq. (3.13) for each envelope, c, are specified in increasing and decreasing order. An underestimator composed of multiple McCormick envelopes for a term $a'_{ij} b'_{ij}$ with $c = 3$ is shown in fig. 3.2.

The multiple McCormick formulation does not need additional variables (except for the w_{ij}) and thus result in a compact reformulation. However, due to the nature of the big-M constraints the continuous relaxation of a discrete bilinear term is weak. In the next section stronger continuous relaxations are derived, by utilizing the discrete nature of only one of the variables.

3.2 Discrete linear reformulation

In Paper I a discrete linear reformulation (DLR) was presented where auxiliary continuous variables were added in order to linearize the bilinear terms. The following notations

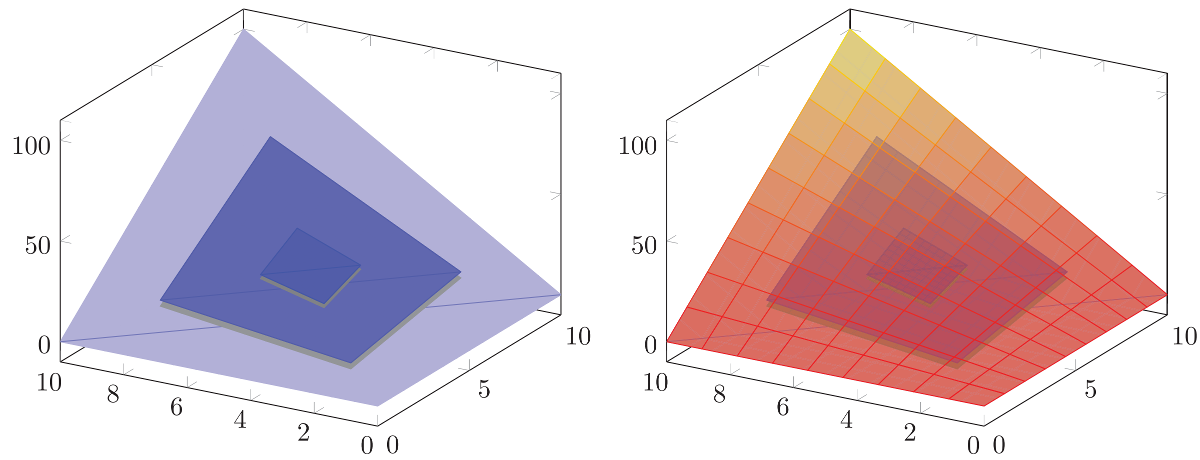


Figure 3.2: Underestimators for a term $a'_{ij}b'_{ij}$ composed of three McCormick envelopes.

are used below. The upper bound of the a'_{ij} variable is written as $\overline{A}_j$ and is the value of the largest element in column j of matrix $\mathbf{A}$, *i.e.*

$$\overline{A}_j = \max_i a'_{ij} \quad \forall j. \tag{3.14}$$

Observe that $\overline{A}_j$ corresponds to $\overline{A}_j^1$ in eq. (3.10) but in this formulation only the largest element is needed. M_i is the number of unique elements in row i of the discretized matrix and therefore also the number of auxiliary variables added for the specific term. Finally, $\mathbf{K}_i^m$ is the set of indices for which the elements are equal to the current value B_i^m:

$$\mathbf{K}_i^m = \left\{ j | b_{ij} = B_i^m \right\} \quad \forall i,j,i \neq j \wedge m = 1,...,M_i, \tag{3.15}$$

where

$$b_{ij} \in \left\{ B_i^1, B_i^2, ..., B_i^{M_i} \right\} \quad \forall i,j,i \neq j. \tag{3.16}$$

Here, a positive variable w_{ij} is again added for every bilinear term $a'_{ij}b'_{ij}$ in the same manner as earlier. When modeling, the w_{ij} variable can be omitted and the right hand side of eq. (3.17) added straight to the objective function. The discrete linear reformulation is written as follows:

$$w_{ij} \geq \sum_{m=1}^{M_i} B_i^m z_{ij}^m, \tag{3.17}$$

$$\sum_{m=1}^{M_i} z_{ij}^m = a'_{ij} \quad \forall i,j, \tag{3.18}$$

$$z_{ij}^m \leq \overline{A}_j \sum_{k \in K_i^m} x_{kj} \quad \forall i,j \wedge m = 1,\ldots,M_i, \tag{3.19}$$

$$z_{ij}^m \in [0, \overline{A}_j] \quad \forall i,j \wedge m = 1,\ldots,M_i. \tag{3.20}$$

B_i^m are the constant unique values of the elements b_{ij} in row i of the matrix $\mathbf{B}$ and M_i the number of unique elements in the row, where $M_i \leq n$. The variables z_{ij}^m are nonnegative where one will be equal to a'_{ij} while the others will be zero.

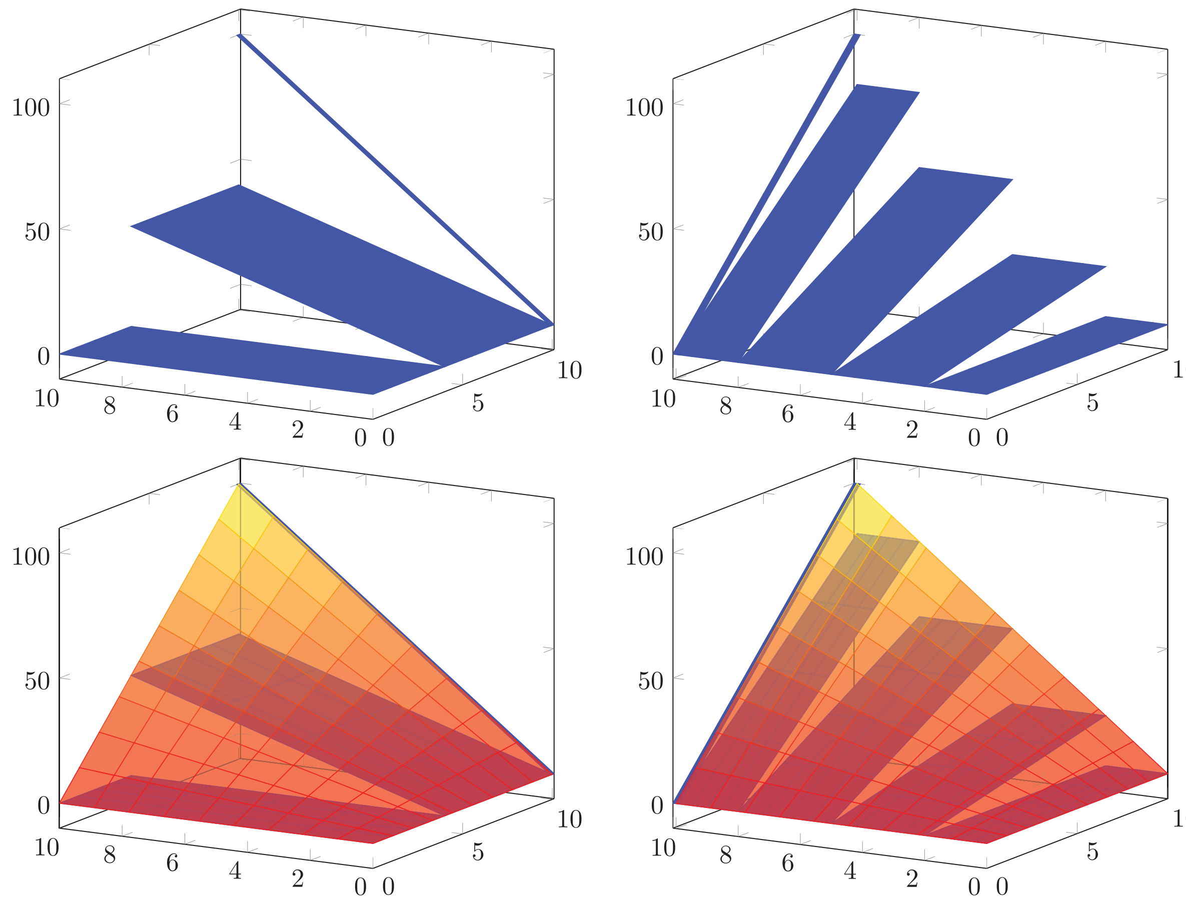


Figure 3.3: Bilinear term $a'_{ij}b'_{ij}$ discretized in a'_{ij} (to the left) and in b'_{ij} (to the right). The underestimators are shown in blue and each level corresponds to a z^m_{ij} variable in eq. (3.17).

3.3 DLR-version 2

In Paper II a different formulation for the bilinear terms is proposed. This formulation handles the auxiliary variables differently than the first one. For any solution in the DLR, only one auxiliary variable is active per bilinear term. In this formulation however, the auxiliary variables are of the incremental type meaning that the larger the bilinear term is the more of the auxiliary variables are active:

$$w_{ij} \geq B_i^1 a'_{ij} + \sum_{m=2}^{M_i} (B_i^m - B_i^{(m-1)}) z_{ij}^m, \quad \forall i, j, \tag{3.21}$$

$$z_{ij}^m \geq a'_{ij} - \overline{A}_j + \overline{A}_j \sum_{\substack{k \in \mathbf{K}_i^{m'} \\ m' \geq m}} x_{kj} \quad \forall i, j \wedge m = 1, \ldots, M_i. \tag{3.22}$$

3.4 Examples of the formulations

In this section all of the above reformulations are shown for a single bilinear term where $i = 2$ and $j = 3$. Bilinear terms for all i, j in the QAP can be handled in a similar way. The matrices in the examples are:

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & \mathbf{5} & 9 & 6 \\ 3 & 0 & \mathbf{2} & 6 & 9 \\ 5 & 2 & \mathbf{0} & 8 & 10 \\ 9 & 6 & \mathbf{8} & 0 & 2 \\ 6 & 9 & \mathbf{10} & 2 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 4 & 3 & 7 & 7 \\ \mathbf{4} & \mathbf{0} & \mathbf{4} & \mathbf{10} & \mathbf{6} \\ 3 & 4 & 0 & 2 & 3 \\ 7 & 10 & 2 & 0 & 4 \\ 7 & 6 & 3 & 4 & 0 \end{bmatrix}$$

where a'_{23} and b'_{23} are defined as:

$$a'_{23} = 5x_{21} + 2x_{22} + 0x_{23} + 8x_{24} + 10x_{25},$$

$$b'_{23} = 4x_{13} + 0x_{23} + 4x_{33} + 10x_{43} + 6x_{53},$$

while the assignment constraints for the specific variables are:

$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1,$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1.$$

The McCormick formulation (eq. (3.9))

From the unique elements in column $j = 3$ of $\mathbf{A}$ and row $i = 2$ of $\mathbf{B}$ we get $N_3 = 4$, $M_2 = 3$ and $c = 1,\ldots,\min\left\{\lceil \frac{4}{2} \rceil, \lceil \frac{3}{2} \rceil\right\} = 1,\ldots,2$. For $c = 1$, the upper and lower bounds are: $\underline{A}_3^1 = 2$, $\overline{A}_3^1 = 10$, $\underline{B}_2^1 = 4$ and $\overline{B}_2^1 = 10$. For $c = 2$, $\underline{A}_3^2 = 5$, $\overline{A}_3^2 = 8$, $\underline{B}_2^2 = 6$ and $\overline{B}_2^2 = 6$. The index sets in eq. (3.12) and eq. (3.13) are $\mathbf{C}_3^1 = \{1,2,4,5\}$, $\mathbf{D}_2^1 = \{1,3,4,5\}$, $\mathbf{C}_3^2 = \{1,4\}$ and $\mathbf{D}_2^2 = \{5\}$. Then, according to eq. (3.9) we get the following constraints:

$$w_{23} \geq 10a'_{23} + 10b'_{23} - 100 - 100(2 - x_{21} + x_{22} + x_{24} + x_{25} + x_{13} + x_{33} + x_{43} + x_{53}),$$

$$w_{23} \geq 4a'_{23} + 2b'_{23} - 8 - 100(2 - x_{21} + x_{22} + x_{24} + x_{25} + x_{13} + x_{33} + x_{43} + x_{53}),$$

$$w_{23} \geq 6a'_{23} + 5b'_{23} - 30 - 100(2 - x_{21} + x_{24} + x_{53}),$$

$$w_{23} \geq 6a'_{23} + 8b'_{23} - 30 - 100(2 - x_{21} + x_{24} + x_{53}).$$

The DLR formulation (eqs. (3.17) to (3.19))

For this term ($i = 2, j = 3$), $\overline{A}_3 = 10$ (*i.e.* the largest value in column three of the $\mathbf{A}$ matrix), according to eq. (3.14). Then $M_2 = 3$ according to eq. (3.16) since row two ($i = 2$) in the discretized matrix $\mathbf{B}$ has three unique values. The B_i^m values in eq. (3.16) are the unique values (of row $i = 2$ in the $\mathbf{B}$ matrix), *i.e.* $b'_{23} \in \{4,6,10\} = \left\{B_2^1, B_2^2, B_2^3\right\}$. The index sets K_2^m are, according to eq. (3.15), in this case $K_2^1 = \{1,3\}, K_2^2 = \{5\}$ and $K_2^3 = \{4\}$.

$$w_{23} \geq 4z_{23}^1 + 6z_{23}^2 + 10z_{23}^3,$$

$$z_{23}^1 + z_{23}^2 + z_{23}^3 = 5x_{21} + 2x_{22} + 8x_{24} + 10x_{25},$$

$$z_{23}^1 \leq 10(x_{13} + x_{33}),$$

$$z_{23}^2 \leq 10x_{53},$$

$$z_{23}^3 \leq 10x_{43}.$$

The DLR-version 2 formulation (eqs. (3.21) to (3.22))

In this formulation the same index sets are used as in the previous DLR formulation.

$$w_{23} \geq 4a'_{23} + 2z^1_{23} + 4z^2_{23},$$

$$z^1_{23} \geq a'_{23} - 10 + 10x_{53},$$

$$z^1_{23} \geq a'_{23} - 10 + 10x_{43}.$$

3.5 Special structures

Some of the instances in the QAPLib have rows and columns where every element is equal to zero. This is a result from adding so called dummy locations or facilities. For example if a facility location problem has 30 possible locations but only 20 facilities to place among them, this can still be written as a QAP instance of size $n = 30$ by simply adding 10 rows and columns with all values equal to zero in the flow matrix. Then solving the QAP determines both which locations to leave empty as well as where to place the facilities. Every location that is assigned a dummy facility is left empty. Clearly, changing the locations between two dummy facilities will give the same solution value since all the same locations are still unused and nothing else is changed in the model. In other words, instances containing many dummy facilities result in models with a lot of symmetries that can really destroy the efficiency of the branching. Especially the esc instances contain a lot of zero rows. Paper I shows how to remove these dummy facilities from the models in order to avoid an excessive number of equal symmetric solutions and therefore speed up the branch and bound solver.

3.5.1 Matrix modification

The number of variables in the presented linearizations are dependent on how many unique values each row of the discretized matrix has. Therefore the size of the model can be reduced by decreasing the number of unique values in one of the matrices. In most QAP instances at least one of the matrices is symmetric. The facility location problem can be used as an example. From there it is easy to understand that the distance between two locations should be the same regardless of the direction. This is not always the case, for example there might be one way roads or construction work making one of the directions longer. The modification scheme takes advantage of the fact that it is mathematically exactly the same if one truck drives between two facilities in both directions as if two trucks drive only in one direction. When modifying a QAP instance the elements a_{ij} in one of the matrices can therefore be changed (as long as the other matrix is symmetric) to any new element $\tilde{a}_{ij}$ as long as $a_{ij} + a_{ji} = \tilde{a}_{ij} + \tilde{a}_{ji}$ holds. When $\mathbf{B}$ is symmetric (*i.e.* $\mathbf{B} = \mathbf{B}^T$):

$$\mathbf{XA} \bullet \mathbf{BX} = \mathbf{A} \bullet \mathbf{X}^T\mathbf{BX} = \mathbf{A}^T \bullet \mathbf{X}^T\mathbf{B}^T\mathbf{X} = \mathbf{A}^T \bullet \mathbf{X}^T\mathbf{BX}. \tag{3.23}$$

In eq. (3.23) $\bullet$ represents the scalar product of the matrices, defined as the sum of the product of the corresponding elements. Then, assuming $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$, apply eq. (3.23) to

$\mathbf{A}_2$ and obtain:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a'_{ij} b'_{ij} = (\mathbf{A_1} + \mathbf{A}_2) \bullet \mathbf{X}^T \mathbf{B}\mathbf{X} = (\mathbf{A_1} + \mathbf{A_2^T}) \bullet \mathbf{X}^T \mathbf{B}\mathbf{X} = \tilde{\mathbf{A}} \bullet \mathbf{X}^T \mathbf{B}\mathbf{X}. \tag{3.24}$$

Paper II shows two different LP models that can be solved *a priori*. One modification scheme minimizes the maximum difference between the elements in the rows in order to give tighter formulations. The other LP model minimizes the difference between every element of a row in order to decrease the number of unique elements and thus reducing the model size. The distance matrix of an example instance shown below can then be rewritten to a new matrix $\tilde{\mathbf{A}}$, *e.g.* as:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 & 4 & 4 & 5 \\ 1 & 0 & 1 & 1 & 2 & 3 & 3 & 4 \\ 2 & 1 & 0 & 2 & 1 & 2 & 2 & 3 \\ 2 & 1 & 2 & 0 & 1 & 2 & 2 & 3 \\ 3 & 2 & 1 & 1 & 0 & 1 & 1 & 2 \\ 4 & 3 & 2 & 2 & 1 & 0 & 2 & 3 \\ 4 & 3 & 2 & 2 & 1 & 2 & 0 & 1 \\ 5 & 4 & 3 & 3 & 2 & 3 & 1 & 0 \end{bmatrix} \quad \tilde{\mathbf{A}} = \begin{bmatrix} 0 & 2 & 2 & 2 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \\ 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 0 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 0 & 0 \end{bmatrix}.$$

These modifications do not break the symmetric structure of the QAP itself. Therefore, techniques such as orbital branching (Ostrowski et al., 2011) and orbital shrinking (Fischetti and Liberti, 2012) can still be applied on these models. Simple alternative versions of the modifications can also be used with other formulations where a certain attribute of one of the matrices is desired. Paper II shows that the modifications also tighten the GLB in its normal form.

The QAPLib classifies instances as asymmetric for which one of the matrices is not symmetric. However using the scheme shown here, every instance where one of the matrices is symmetric can be rewritten as an instance where both matrices are symmetric without changing the objective value. The elements in the asymmetric matrix are then changed to $a_{ij} = \frac{a_{ij}+a_{ji}}{2}\ \forall i,j$. Every instance in the QAPLib is in fact symmetric since there is not a single instance listed where both of the matrices are asymmetric.

CHAPTER 4

Results and notes on the papers

The work in this thesis is mostly derived from four papers. A short summary as well as some results from the papers are presented below.

Paper I: A new exact discrete linear reformulation of the quadratic assignment problem

Paper I introduces the new MINLP form of the QAP. Also, the first linearization method to write the MINLP as a discrete MILP is presented. The formulation turned out to be one of the best available in the literature on sparse QAP instances. In this paper the instances `esc32a`, `esc32c`, `esc32d` and `esc64a` were solved for the first time to proven optimality. These instances had remained unsolved for over 20 years since their introduction in Eschermann and Wunderlich (1990). Paper I also shows how to remove auxiliary rows and columns where all elements are equal to zero. For example, the instance `esc64a` has 42 rows out of 64 where all the elements are zero. The number of possible permutations can in the case of `esc64a` be reduced from $64! = 1.3 \cdot 10^{89}$ to $\frac{64!}{42!} = 0.9 \cdot 10^{38}$.

Paper II: Improved discrete reformulations for the quadratic assignment problem

In paper II the original DLR formulation is improved. The most important contribution of this paper is to remove the diagonal elements from the equations, drastically tightening the DLR model as well as reducing the solution times. An alternative linearization, equally tight as the improved formulation, is also presented. In addition to this, two *ad hoc* matrix modification schemes are derived.

With these methods the instance `esc32b` is solved for the first time ever to proven optimality. The instances solved for the first time in Paper I are solved again in much shorter time. The solution times for both methods can be seen in table 4.1. Figure 4.1

Table 4.1: Solution times, in seconds, for some of the esc instances from the QAPLIB

Instance	Paper I (s)	Paper II (s)
esc32a	1 618 580	117 850
esc32b	***	210 045
esc32c	24 365	7 801
esc32d	36 256	610
esc64a	16 370	2 899

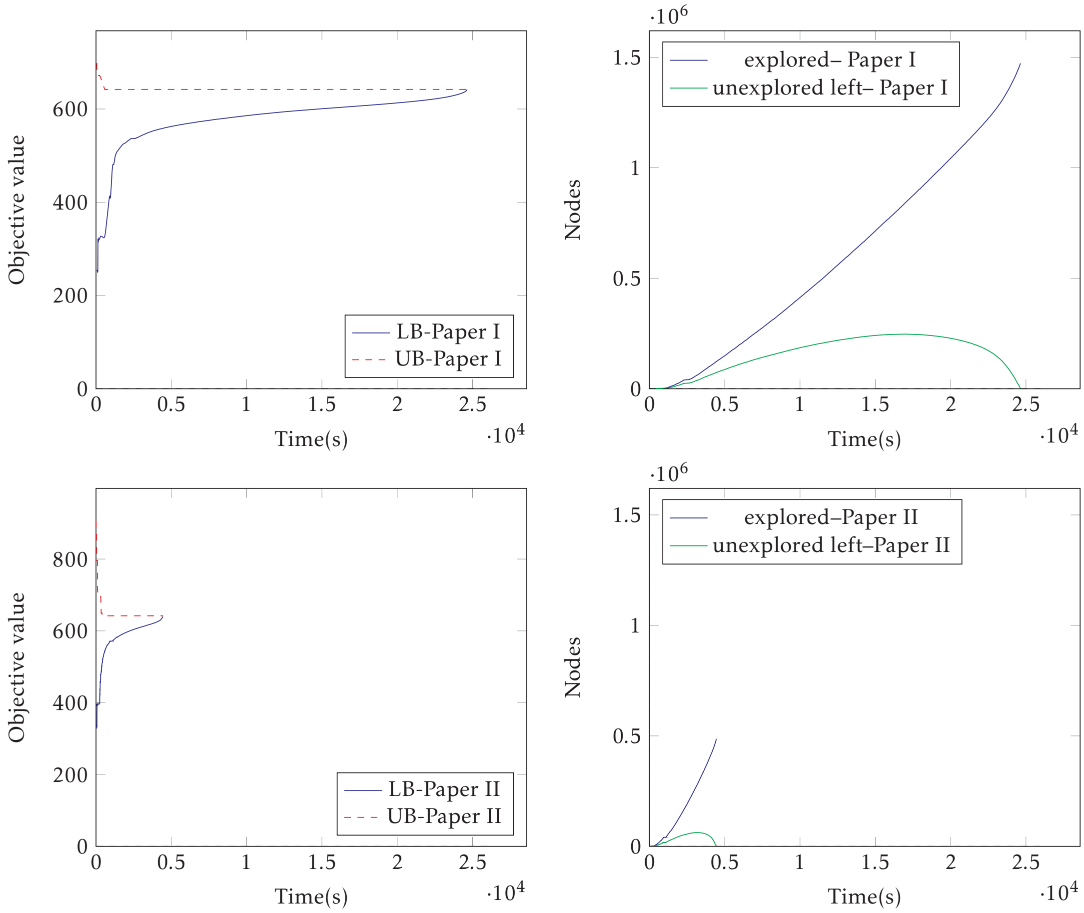


Figure 4.1: Solution progress for the instance esc32c, comparing the methods from Paper I and Paper II

shows a comparison between the results when solving the models from Paper I and Paper II for the instance esc32c. Clearly, the improved formulation is significantly better than the original.

Paper III: Tightening a discrete formulation of the quadratic assignment problem

One of the biggest drawbacks with most linearizations of bilinear terms is that the binary variables take as small fractional values as possible in the LP relaxations. This is also the case with the formulations presented in Paper I and Paper II. Because of this the auxiliary variables that correspond to the largest elements in the discretized matrix will in almost all cases be equal to zero. Paper III deals with an idea of solving QAPs containing only the largest elements of each column in order to obtain lower bounds on these variables. By changing every element to zero except the largest element of each column in the discretized matrix, a new QAP is obtained:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 4 & 1 \\ 2 & 4 & 0 & 5 \\ 3 & 1 & 5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 2 & 5 & 8 \\ 2 & 0 & 6 & 9 \\ 5 & 6 & 0 & 3 \\ 8 & 9 & 3 & 0 \end{bmatrix} \quad \longrightarrow$$

$$\mathbf{A}^{\mathbf{top}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 2 & 5 & 8 \\ 2 & 0 & 6 & 9 \\ 5 & 6 & 0 & 3 \\ 8 & 9 & 3 & 0 \end{bmatrix}$$

A lower bound, or the optimal solution for the new QAP, is also a lower bound for the sum of all the variables preceding the largest elements of each column in the objective function. Unfortunately the subproblems are QAPs of the same size as the original model and are therefore difficult to solve. The proposed cuts require too much computational time in order to be effective. However, the optimal solution of the subproblem is not needed for the cuts to work. Therefore a quick and tight lower bounding technique could be very useful in this setting.

Paper IV: An efficient reformulation of the multiechelon stochastic inventory system with uncertain demands

In this paper the formulations from Paper I are applied on a large scale supply chain problem presented in You and Grossmann (2010). One of the objectives for the considered supply chain is to decide where to place warehouses so that all customer demands (which are known *a priori*) are satisfied. This results in bilinear terms very similar to the objective function in a QAP. By reordering these terms in the objective function on a multiechelon supply chain model, a similar MINLP formulation is achieved as in eq. (3.3). Then, the same linearization techniques are used as in Paper I and Paper II. This results in a model both tighter and significantly smaller in size than the original one. In You and Grossmann (2010) a Lagrangian decomposition is applied in order to solve the MINLP problem. After reformulating the problem as in Paper IV these problems can be solved to a predefined gap without decomposing, using only an MILP solver in significantly shorter time. The original model contains many bilinear terms which are reformulated to MILP form before decomposing the problem. However, after rewriting the terms some of the bilinear reformulations can be left out altogether. For a supply chain with I plants, J distribution centers and K customer demand zones, the number of auxiliary continuous variables needed to linearize the bilinearities in the

			You and Grossmann (2010)			Paper IV		
$\|I\|$	$\|J\|$	$\|K\|$	Bin. vars.	Cont. vars.	Const.	Bin. vars.	Cont. vars.	Const.
2	20	20	460	2 480	5 300	460	940	1 800
5	30	50	1 680	13 640	33 190	1 680	3 410	6 520
20	50	100	6 050	120 250	335 350	6 050	12 200	22 400
15	100	200	21 600	300 300	1 040 700	21 600	43 400	83 800

Table 4.2: Sizes of the old and reformulated models

objective function decreases from $I \cdot J \cdot K$ to only $I \cdot J$ when using the new reformulations. In table 4.2 the number of variables needed for the different formulations is shown. Clearly, the newer formulations are much more compact with tighter models resulting in remarkably shorter solution times.

CHAPTER 5

Conclusion

In the work presented in this thesis some new effective techniques for solving the quadratic assignment problem have been derived. Since the first version of the DLR already solved a few unsolved instances from the QAPLib, developing these methods further should yield nice results. In particular the matrix modification schemes presented in Paper II could prove to be very useful for other solution strategies as well. This thesis has focused mainly on reformulations of bilinear terms. A good branching strategy would probably improve these formulations a lot. Also, symmetries could be considered in much greater detail, especially since many instances, such as the border length minimization problems, are highly symmetrical.

Future work should also address how to apply similar formulations as in Paper IV to multiechelon supply chains where single sourcing is not a requirement.

5.1 A remark on solution times

When solving MILP problems using commercial solvers, the objective and the constraints are written into problem files that are read in by the solver. With problems of this size, a change in the branching order can have a significant effect on the solution times. Since both Cplex and Gurobi determine the branching order of the variables from the ordering of the variables in the problem files, changing the order of the constraints in the file affects the solution times as well.

Solvers normally have many tunable parameters and some custom parameter settings might reduce the solution time drastically on one instance while increasing it on another. Also, choice of and version of solver all impact the solution times. Therefore, all the models in this work have been solved using Gurobi with default parameters without putting any effort into changing solver, parameters or branching priorities. All computations have been conducted on a single PC with an Intel i7 quadcore 2.8 GHz processor and 6 GB RAM. Finally, it should be mentioned that it is difficult to do a completely fair comparison between different MILP formulations since there are too many factors to be considered.

Bibliography

Ahuja, R. K., Orlin, J. B., Tiwari, A., 2000. A greedy genetic algorithm for the quadratic assignment problem. Comput. Oper. Res. 27, 917–934. [9]

Anstreicher, K., Brixius, N., Goux, J.-P., Linderoth, J., 2002. Solving large quadratic assignment problems on computational grids. Mathematical Programming 91, 563–588. [10]

Anstreicher, K. M., Brixius, N. W., 2001. A new bound for the quadratic assignment problem based on convex quadratic programming. Mathematical Programming 89, 341–357. [10]

Ben-David, G., Malah, D., 2005. Bounds on the performance of vector-quantizers under channel errors. Information Theory, IEEE Transactions on 51 (6), 2227–2235. [7]

Bokhari, S. H., 1981. On the mapping problem. IEEE Transactions on Computers 30 (3), 207–214. [7]

Burer, S., Vandenbussche, D., 2006. Solving lift-and-project relaxations of binary integer programs. SIAM Journal on Optimization 16, 726–750. [10]

Burkard, R., Çela, D. E., Karisch, S., Rendl, F., 1997. QAPLIB - A quadratic assignment problem library. Journal of Global Optimization 10, 391–403. [3, 8]

Burkard, R., Offermann, J., 1977. Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungsprobleme. Zeitschrift für Operations Research 21 (4), 121–132. [7]

Burkard, R. E., Cela, E., Klinz, B., 1993. On the biquadratic assignment problem. In: Quadratic Assignment and Related Problems. pp. 117–146. [7]

Carlson, R. C., Nemhauser, G. L., 1966. Scheduling to minimize interaction cost. Operations Research 14 (1), 52–58. [8]

Connolly, D. T., 1990. An improved annealing scheme for the QAP. European Journal of Operational Research 46 (1), 93–100. [9]

Davis, F., Devine, M., Lutz, R., 1974. Scheduling activities among conflicting facilities to minimize conflict cost. Mathematical Programming 6 (1), 224–228. [8]

de Carvalho Jr., S., Rahmann, S., 2006. Microarray layout as a quadratic assignment problem. Proceedings of the German Conference on Bioinformatics (GCB) P-83 of Lecture Notes in Informatics, 11–20. [7]

de Klerk, E., Sotirov, R., 2010. Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem. Mathematical Programming 122, 225–246. [10]

Dell'Amico, M., Díaz Díaz, J., Iori, M., Montanari, R., 2009. The single finger keyboard layout problem. Computers & Operations Research 36, 3002–3012. [7]

Dickey, J. W., Hopkins, J. W., 1972. Campus building arrangement using TOPAZ. Transportation Research 6, 59–68. [5]

Drezner, Z., Hahn, P., Taillard, E., 2005. Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. Annals of Operations Research 139, 65–94. [11]

Çela, E., 1998. The Quadratic Assignment Problem Theory and Algorithms. Kluwer Academic Publishers. [8, 10, 11]

Elshafei, A. N., 1977. Hospital layout as a quadratic assignment problem. Operational Research Quarterly 28 (1), 167–179. [5]

Eschermann, B., Wunderlich, H.-J., 1990. Optimized synthesis of self-testable finite state machines. In: Fault-Tolerant Computing, 1990. FTCS-20. Digest of Papers, 20th International Symposium. pp. 390–397. [7, 23]

Fathi, Y., Ginjupalli, K. K., 1993. A mathematical model and a heuristic procedure for the turbine balancing problem. European Journal of Operational Research 65 (3), 336–342. [7]

Fischetti, M., Liberti, L., 2012. Orbital shrinking. In: Mahjoub, A., Markakis, V., Milis, I., Paschos, V. (Eds.), Combinatorial Optimization. Vol. 7422 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 48–58. [21]

Fischetti, M., Monaci, M., Salvagnin, D., 2012. Three ideas for the quadratic assignment problem. Operations Research July/August 2012 60, 954–964. [11]

Forsberg, J. H., Delaney, R. M., Zhao, Q., Harakas, G., Chandran, R., 1995. Analyzing lanthanide-induced shifts in the nmr spectra of lanthanide(iii) complexes derived from 1,4,7,10-tetrakis(n,n-diethylacetamido)-1,4,7,10-tetraazacyclododecane. Inorganic Chemistry 34 (14), 3705–3715. [7]

Frieze, A., Yadegar, J., 1983. On the quadratic assignment problem. Discrete Applied Mathematics 5, 89–98. [11]

Geoffrion, A. M., Graves, G. W., 1976. Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach. Operations Research 24 (4), 595–610. [8]

Gilmore, P. C., 1962. Optimal and suboptimal algorithms for the quadratic assignment problem. Journal of the Society for Industrial and Applied Mathematics 10, 305–313. [10]

Grötschel, M., 1992. Discrete mathematics in manufacturing. Tech. Rep. SC-92-03, ZIB, Takustr.7, 14195 Berlin. [7]

Haghani, A., Chen, M.-C., 1998. Optimizing gate assignments at airport terminals. Transportation Research Part A: Policy and Practice 32 (6), 437–454. [6]

Hahn, P., Grant, T., 1998. Lower bounds for the quadratic assignment problem based upon a dual formulation. Operations Research 46 (6), pp. 912–922. [10]

Hahn, P. M., Krarup, J., 2001. A hospital facility layout problem finally solved. Journal of Intelligent Manufacturing 12 (5-6), 487–496. [6, 11]

Hahn, P. M., Zhu, Y.-R., Guignard, M., Hightower, W. L., Saltzman, M. J., 2012. A level-3 reformulation-linearization technique-based bound for the quadratic assignment problem. INFORMS Journal on Computing 24 (2), 202–209. [10]

Hannenhalli, S., Hubbell, E., Lipshutz, R., Pevzner, P., 2002. Combinatorial algorithms for design of dna arrays. In: Chip Technology. Vol. 77 of Advances in Biochemical Engineering/Biotechnology. Springer Berlin Heidelberg, pp. 1–19. [7]

Jünger, M., Martin, A., Reinelt, G., Weismantel, R., 1994. Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits. Mathematical Programming 63, 257–279. [7]

Kaufman, L., Broeckx, F., 1978. An algorithm for the quadratic assignment problem using Bender's decomposition. European Journal of Operational Research 2 (3), 207–211. [11]

Koopmans, T., Beckmann, M., 1957. Assignment problems and location of economic activities. Econometrica 25, 53–76. [5, 9]

Krarup, J., Pruzan, P. M., 1978. Computer-aided layout design. In: Balinski, M., Lemarechal, C. (Eds.), Mathematical Programming in Use. Vol. 9 of Mathematical Programming Studies. Springer Berlin Heidelberg, pp. 75–94. [6]

Laporte, G., Mercure, H., 1988. Balancing hydraulic turbine runners: A quadratic assignment problem. European Journal of Operational Research 35 (3), 378–381. [7]

Lawler, E. L., 1963. The quadratic assignment problem. Management Science 9 (4), 586–599. [10]

Li, Y., Pardalos, P. M., Resende, M. G., 1994. A greedy randomized adaptive search procedure for the quadratic assignment problem. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16, 237–261. [9]

McCormick, G. P., 1976. Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. Mathematical Programming 10, 147–175. [14]

Misevicius, A., 2005. A tabu search algorithm for the quadratic assignment problem. Computational Optimization and Applications 30 (1), 95–111. [9]

Mittelmann, H., Peng, J., 2010. Estimating bounds for quadratic assignment problems associated with hamming and manhattan distance matrices based on semidefinite programming. SIAM J. on Optimization 20 (6), 3408–3426. [10]

Nugent, C. E., Vollmann, T. E., Ruml, J., 1968. An experimental comparison of techniques for the assignment of facilities to locations. Operations Research 16 (1), 150–173. [10]

Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S., 2011. Orbital branching. Mathematical Programming 126, 147–178. [21]

Peng, J., Mittelmann, H., Li, X., 2010. A new relaxation framework for quadratic assignment problems based on matrix splitting. Mathematical Programming Computation 2 (1), 59–77. [10]

Phillips, A., Rosen, J., 1994. A quadratic assignment formulation of the molecular conformation problem. Journal of Global Optimization 4 (2), 229–241. [7]

Rabak, C., Sichman, J., 2003. Using a-teams to optimize automatic insertion of electronic components. Advanced Engineering Informatics 17 (2), 95–106. [7]

Steinberg, L., Jan. 1961. The backboard wiring problem: A placement algorithm. SIAM Review 3 (1), 37–50. [7]

Taillard, E., 1991. Robust taboo search for the quadratic assignment problem. Parallel Computing 17 (4-5), 443–455. [9]

Xia, Y., Yuan, Y.-X., 2006. A new linearization method for quadratic assignment problems. Optimization Methods and Software 21 (5), 805–818. [11]

You, F., Grossmann, I. E., 2010. Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies. AIChE Journal 56 (2), 419–440. [25, 26]

Zhang, H., Beltran-Royo, C., Ma, L., 2013. Solving the quadratic assignment problem by means of general purpose mixed integer linear programming solvers. Annals of Operations Research 207 (1), 261–278. [11]