A POLYHEDRAL APPROACH TO QUADRATIC ASSIGNMENT PROBLEM

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Ahmet Sertaç Murat Köksaldı
September, 1994

A POLYHEDRAL APPROACH TO QUADRATIC ASSIGNMENT PROBLEM

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREEOF
MASTER OF SCIENCE

By
Ahmet Sertaç Murat Köksaldı
September, 1994

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Mustafa Akgül(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Osman Oğuz

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Ömer Benli

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray, Director of Institute of Engineering and Science

ABSTRACT

A POLYHEDRAL APPROACH TO QUADRATIC ASSIGNMENT PROBLEM

Ahmet Sertaç Murat Köksaldı
M.S. in Industrial Engineering
Supervisor: Assoc. Prof. Mustafa Akgül
September, 1994

In this thesis, Quadratic Assignment Problem is considered. Since Quadratic Assignment Problem is NP-hard, no polynomial time exact solution method exists. Proving optimality of solutions to Quadratic Assignment Problems has been limited to instances of small dimension. In this study, Quadratic Assignment Problem is handled from a polyhedral point of view. A graph theoretic formulation of the problem is presented. Later, Quadratic Assignment Polytope is defined and subsets of valid equalities and inequalities for Quadratic Assignment Polytope is given. Finally, results of the experiments with a polyhedral cutting plane algorithm using the new formulation is also presented.

Keywords: Quadratic Assignment Problem, Quadratic Assignment Polytope, polyhedral cutting plane algorithm

ÖZET

KARESEL ATAMA PROBLEMİNE POLYHEDRAL BİR YAKLAŞIM

Ahmet Sertaç Murat Köksaldı
Endüstri Mühendisliği, Yüksek Lisans
Tez Yöneticisi : Doç. Mustafa Akgül
Eylül, 1994

Bu tez çalışmasında, Karesel Atama Problemi ele alınmıştır. Karesel Atama Problemi NP-zorlukta olduğu için, polinom zamanlı bir çözüm yöntemi mevcut değildir. Olabilir çözümlerin en iyiliğinin ispatı ancak küçük boyutlu problemlerde mümkündür. Çalışmamızda, Karesel Atama Problemi polyhedral bir açıdan ele alınmıştır. Karesel Atama Probleminin graf teorik bir ifadesi tanımlanmıştır. Daha sonra, Karesel Atama Poytopu ve, geçerli bazı eşitsizlik ve eşitlik alt kümeleri tanımlanmıştır. Son olarak da, Karesel Atama Probleminin yeni ifadesinin kullanıldığı bir polyhedral kesen düzlem yöntemi ile yapılan testlerin sonuçları verilmiştir.

Anahtar Sözcükler: Karesel Atama Problemi, Karesel Atama Poytopu, polyhedral kesen düzlem yöntemi

To my family and people who added value to my life

ACKNOWLEDGMENT

I would like to thank my advisor Mustafa Akgül who has suggested such an interesting topic for this study. I would also like to thank Assoc. Prof. Osman Oğuz and Assoc. Prof. Ömer Benli for accepting to read and review this thesis.

I would like to extend my thanks to my parents, my sister Ebru and my love Emine for their support and encouragement.

I greatly appreciate Levent Kandiller, İradj Ouveysi and Fatih Yılmaz for their valuable remarks, guidance and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Koopmans and Beckmann [40] are the first to state Quadratic Assignment Problem (QAP). While dealing with the allocation of plants to locations they realized that "The assumption that the benefit from an economic activity at some location does not depend on the uses of other locations is quite inadequate to the complexities of locational decisions." This conclusion is the result of experimentation with the linear assignment models of location theory. Therefore they attempted to incorporate the cost of transportation between plants to the model. With the motivation of considering the interactions between plants, they introduced the QAP:

They consider n plants and n locations. Given

c_{ij} : gross revenue obtained by assigning plant i to location j

a_{ik} : required commodity flow between plants i and k

b_{jl} : unit transportation cost between locations j and l

Given n x n real matrices A, B and C, let S_N be the set of permutations over $\mathcal{N}$ where $\mathcal{N} = \{1, 2, ..., n\}$ and $\varphi \in S_N$. Then QAP is

$$z = \min_{\varphi \in S_N} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{N}} a_{i,k} b_{\varphi(i),\varphi(k)} - \sum_{i \in \mathcal{N}} c_{i,\varphi(i)}$$

According to Koopmans and Beckmann, "QAP seems to be close to the core of location theory."

Following Koopmans and Beckmann, many real life problems are stated as QAP:

In the wiring problem of Steinberg [68], n modules have to be placed on a board and be connected by wires. Given

a_{ik} : distance between positions i and k

b_{jl} : number of connections between modules j and l

the aim is to minimize the total wire length.

Design of control panels and typewriter keyboards is another problem modeled as a QAP (Burkard and Offermann [13], Pollatschek, Gershoni and Radday [59]). Given

a_{ik} : the mean frequency of a pair of letters i and k in language L

b_{jl} : the time needed to press key l after pressing key j

their aim is to arrange the keys of a typewriter on the keyboard such that the time needed to write a certain text in language L is minimized.

For other fields of application see Burkard [9].

There is a strong interest in the exact solution algorithms for QAP. Since QAP is $\mathcal{NP}$- hard, no polynomial time algorithm is known. Existing exact solution algorithms, branch and bound, and cutting plane algorithms, are so ineffective that they can solve problems up to size $n \leq 17$ where as explicit enumeration can solve problems of size $n \leq 10$.

We will handle the QAP from the polyhedral point of view. In the late 70'ies a new method, Polyhedral Cutting Plane Method, is introduced for combinatorial optimization problems. Its roots lie in the seminal paper of Dantzig, Fulkerson and Johnson [19]. The idea is to define the feasible set of the combinatorial problem by linear inequalities and equalities, and algorithmically add those to the linear programming relaxation. Since it is impossible to know all the linear inequalities and equalities defining the underlying polytope, one works with a subset of facets. After solving the linear programming relaxation, violated facets are identified and added to the linear programming relaxation. One resorts to branching if no violated inequalities are found. This is what branch and cut is, use of cutting planes and branching sequentially until finding the optimal solution.

In chapter 2 we will give a summary of solution methods, lower bounds, heuristics and complexity results regarding QAP. Chapter 3 will be devoted to our studies. We mainly concentrate on symmetric problems, i.e. A and B are symmetric matrices. This is not merely a restriction because any QAP instance can be transformed to a symmetric QAP (SQAP) instance by the technique of Hadley, Rendl and Wolkowicz [34]. A graph theoretic formulation of QAP will be presented. Later quadratic assignment polytope will be defined and sets of valid equalities and inequalities for quadratic assignment polytope will be given. Results of a polyhedral cutting plane algorithm will also be presented. Chapter 4 consists of the discussion of our study and further areas of research.

Chapter 2

Literature Review

Koopmans-Beckmann formulation of the first chapter is a special type of QAP. Lawler [43] stated general QAP as:

Problem 2.1 (QAP) *Given n^4 cost coefficients c_{ijkl} where $i, j, k, l \in \mathcal{N}$ and $\mathcal{N} = \{1, 2, ..., n\}$, let S_N be the set of permutations over $\mathcal{N}$ and $\varphi \in S_N$. Then QAP is*

$$z = \min_{\varphi \in S_N} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{N}} c_{i,\varphi(i),k,\varphi(k)} + \sum_{i \in \mathcal{N}} d_{i,\varphi(i)}$$

In Koopmans-Beckmann formulation cost coefficients c_{ijkl} equal to $a_{ik}.b_{jl}$. Researchers concentrated on Koopmans-Beckmann problems rather than general QAPs. Although some of the works are applicable to general QAPs, it is at least customary to use Koopmans-Beckmann type test problems. Burkard, Karish and Rendl [12] collected QAP instances used in the literature. All of these instances are Koopmans-Beckmann problems.

2.1 Mathematical Programming Formulations of QAP

Like other combinatorial problems QAP can be formulated as a mathematical programming problem. Mainly there are three classes of formulations:

1.) Nonlinear Integer Programming formulations,

2.) Integer Programming formulations,

3.) Mixed-Integer Programming formulations.

2.1.1 Nonlinear Integer Programming Formulation

This formulation immediately follows from the problem statement. A permutation of the set $\mathcal{N}$ can be uniquely represented by a vector $x \in \mathcal{R}^{n^2}$ such that $x = (x_{11}, x_{12}, ..., x_{1n}, ..., x_{n1}, x_{n2}, ...x_{nn})$ where

$$\forall i, j \in \mathcal{N} \;\; x_{ij} = \begin{cases} 1 & \text{if} \quad \varphi(i) = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and x_{ij}s obey the following assignment (multiple choice) constraints :

$$\sum_{i \in \mathcal{N}} x_{ij} = 1 \quad \forall j \in \mathcal{N} \tag{2}$$

$$\sum_{j \in \mathcal{N}} x_{ij} = 1 \quad \forall i \in \mathcal{N} \tag{3}$$

Defining Assignment Polytope AP^n to be:

$$AP^n = \{ \, x \;\in \mathcal{R}^{n^2} \;:\; x \text{ satisfies equations 2 and 3 } \}$$

Henceforth, QAP can be reformulated as:

Problem 2.2

$$\min_{x \in AP^n} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{N}} c_{ijkl} x_{ij} x_{kl} + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} d_{ij} x_{ij}$$

Since it is usually more difficult to deal with nonlinearities many researchers, in the hope of tractability, try to linearize the objective function. Next sections are devoted to these works.

2.1.2 Integer Programming Formulations

The first linearization technique was proposed by Lawler [43]. It simply replaces each quadratic term ,i.e $x_{ij}x_{kl}$, by a new binary variable y_{ijkl} through imposing some constraints and transforms problem 2.2 to

Problem 2.3

$$\begin{array}{lll}
\min & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} c_{ijkl}y_{ijkl} & +\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} d_{ij}x_{ij} \\
s.t. & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} y_{ijkl} \;=\; n^2 & \\
& x_{ij} + x_{kl} - 2y_{ijkl} \geq 0 & \forall i,j,k,l \;\in \mathcal{N} \\
& x \;\in\; AP^n & \\
& y_{ijkl} \;\in\; \{0,1\} &
\end{array}$$

Theorem 2.1 (Lawler) *The feasible solutions of problems 2.2 and 2.3 can be placed in one-to-one correspondence with equal values of the objective functions. A feasible solution $\overline{x}$ of problem 2.2 corresponds to a feasible solution (x', y') of problem 2.3 if and only if $\overline{x} = x'$.*

proof:
Let $\overline{x}$ describe a feasible solution of problem 2.2. By letting $y'_{ijkl} = \overline{x}_{ij}\,\overline{x}_{kl}$, all constrains of problem 2.3 is satisfied and both problems yield the same objective function value.
Conversely, let (x', y') be a feasible solution of problem 2.3. Letting $x'_{i\varphi(i)} = 1$ in a given solution, from assignment constraints

$$x_{ij} \;=\; 0 \;\; \forall j \;\in \mathcal{N} \setminus \varphi(i) \tag{4}$$

Then, $y_{ijkl} = 0$ unless $j = \varphi(i)$ and $l = \varphi(k)$. It follows that

$$\sum_{j\in\mathcal{N}}\sum_{l\in\mathcal{N}} y'_{ijkl} \;\leq\; 1 \tag{5}$$

and this inequality is strict unless $y'_{i\varphi(i),k\varphi(k)} = 1$. Now summing over all i, k we have

$$\sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{N}}(\sum_{j\in\mathcal{N}}\sum_{l\in\mathcal{N}} y'_{ijkl}) \leq n^2 \tag{6}$$

unless $y'_{ijkl} = 1$ whenever $x'_{ij} = x'_{kl} = 1.\square$

Lawler's formulation has n^4 additional binary variables and n^4+1 additional constraints. Hence, this linearization besides giving some insight does not make the world easy. If you try to solve a problem of size, say 10, you will fed into your integer programming solver a problem with 10100 binary variables and 10021 constraints which is hopeless. This frustrating result caused researchers to look for smaller sized, more tractable linearizations.

2.1.3 Mixed-Integer Programming Formulations

As a remedy to the computational burden of introducing binary variables, continuous variables are used in linearization. Bazaara and Sherali [7] used additional $\frac{1}{2}n^2(n-1)^2$ continuous variables and $2n^2$ additional linear constraints. Frieze and Yadegar [25] proposed a method which uses the same number of continuous variables as Bazaara and Sherali [7] but introduces only n^2 constraints. Kaufman and Broeckx's [39] linearization, which utilizes the method of Glover [29], is applicable only in the case of nonnegative objective function coefficients. It introduces n^2 new continuous variables and n^2 additional linear constraints. Another linearization technique which is of the same effectiveness, in terms of additional variables and constraints, is suggested by Oral and Kettani [48] [49]. Their method is applicable to general QAPs. We will discuss their method and further reduction techniques they employed for decreasing the number of binary variables. Defining a lower bound and an upper bound, respectively D^-_{ij} and D^+_{ij} on the terms

$$D^-_{ij} \leq \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} \leq D^+_{ij} \tag{7}$$

problem 2.4 is

Problem 2.4

$$\begin{aligned}
\min \quad & \textstyle\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} \left(D_{ij}^{-}\, x_{ij} + \xi_{ij} \right) \\
s.t. \quad & \xi_{ij} \geq \textstyle\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{-}x_{ij} - D_{ij}^{+}\left(1 - x_{ij} \right) \quad \forall\ i,j \in \mathcal{N} \\
& x \in AP^{n} \\
& \xi_{ij} \geq 0
\end{aligned}$$

Theorem 2.2 (Oral and Kettani) *problem 2.2 is equivalent to problem 2.4 in the sense that they have the same optimal solution.*

Proof:

Observe that each ξ_{ij} is subject to two constraints:

$$\text{if} \quad x_{ij} = 0 \quad \text{then} \quad \xi_{ij} \geq \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{+} \quad \text{and} \quad \xi_{ij} \geq 0 \tag{8}$$

$$\text{if} \quad x_{ij} = 1 \quad \text{then} \quad \xi_{ij} \geq \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{-} \quad \text{and} \quad \xi_{ij} \geq 0 \tag{9}$$

Given the definitions of D_{ij}^{-} and D_{ij}^{+}, which imply that

$$\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{+} \leq 0 \tag{10}$$

$$\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{-} \geq 0 \tag{11}$$

the constraints on ξ_{ij} reduce to:

$$\xi_{ij} \geq \begin{cases} 0 & \text{if} \quad x_{ij} = 0 \\ \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{-} & \text{if} \quad x_{ij} = 1 \end{cases} \tag{12}$$

Since ξ_{ij} appears in the objective function as a linear term with a coefficient $+1$, there could never be an optimal solution to problem 2.4 in which ξ_{ij} is greater than the minimum possible value. Thus the requirements on ξ_{ij} in problem 2.4 can just as well be written

$$\xi_{ij} = \begin{cases} 0 & \text{if} \quad x_{ij} = 0 \\ \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}} d_{ijkl}x_{kl} - D_{ij}^{-} & \text{if} \quad x_{ij} = 1 \end{cases} \tag{13}$$

Substituting for ξ_{ij} in the objective function of problem 2.4 we get the original problem.□

Thus Oral and Kettani [48] [49] introduces n^2 new continuous variables and n^2 additional constraints to linearize QAP. This is the smallest and most general linearization as of today. Bounds D^-_{ij} and D^+_{ij} can be computed simply by the method of Kaufman and Broeckx [39]:

$$D^+_{ij} = \min_{x \in AP^n} \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{N}} d_{ijkl} x_{kl} \tag{14}$$

$$D^-_{ij} = \max_{x \in AP^n} \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{N}} d_{ijkl} x_{kl} \tag{15}$$

Both are simple Linear Assignment Problems.

They further reduce the number of binary variables at the cost of additional constraints. They managed to reduce the number of binary variables from n^2 to $n \log_2 n$ while adding $n(n+2)$ constraints. This reduction enables the use of mixed integer programming codes for moderate sized problems.

2.2 Computational Complexity

It is known that traveling salesman problem is a special case of Koopmans-Beckmann problem where one of the matrices A or B is a permutation matrix [9]. Therefore QAPs are $\mathcal{NP}$- hard. Sahni and Gonzales [65] showed that QAPs belong even to the hard core of this complexity class. That is, they proved that even the full approximation problem is $\mathcal{NP}$ - complete: Let an arbitrarily small ε be given, for all problem instances find a permutation $\overline{\varphi}$ with objective function value $z(\overline{\varphi})$ such that

$$\left| \frac{z^* - z(\overline{\varphi})}{z^*} \right| < \varepsilon \tag{16}$$

2.3 Lower Bounds for QAP

Since QAP is $\mathcal{NP}$-complete implicit enumeration methods are mostly used as exact solution techniques. Especially these are special branch-and-bound methods. Good lower bounds (LBs) increase the effectiveness of branch-and-bound procedures. Therefore a great amount of work is done on finding good LBs.

The first LB proposed is due to Gilmore [28] and Lawler [43] independently. Computation of Gilmore-Lawler (GL) bound is as follows:
Given problem 2.1, compute for each pair $i, j \in \mathcal{N}$,

$$e_{ij} = \min_{\varphi} \sum_{k \in \mathcal{N}} c_{ij,k\varphi(k)} \tag{17}$$

e_{ij} is a lower bound on the cost of interactions between assignment $i \rightsquigarrow j$ and remaining assignments. In computing e_{ij}, only the pairwise interactions between assignment $i \rightsquigarrow j$ and the remaining assignments is considered.
If we let $f_{ij} = e_{ij} + d_{ij}$, f_{ij} represents a LB on the assignment of i to j. f_{ij}'s form an nxn matrix. Then,

$$z \geq GL = \min_{\varphi} \sum_{i \in \mathcal{N}} f_{i,\varphi_i} \tag{18}$$

is a LB on problem 2.1. e_{ij}'s can be computed by solving a Linear Assignment Problem of size $n-1$. We have to compute e_{ij} for n^2 pairs. Minimization given in 18 is also a Linear Assignment Problem of size n. Hence, GL bound can be computed in $\mathcal{O}(n^5)$ time. In case of Koopmans-Beckmann problems solution of Linear Assignment Problems of size $n-1$ drops to a simple ordering scheme which further reduces the computation time.

Other lower bounding techniques are more elegant applications of GL method. The quality of the bound can be improved if as much information as possible is shifted from the quadratic term of the objective function to the linear term. This idea, named as *reduction* is originally stated by Lawler [43]. The reduction procedure is effective because by decreasing the significance of quadratic term, they lessen the bias caused by ignoring the interaction between certain pairs.

Burkard and Stratmann [15], Edwards [20], Roucairol [63] have utilized reduction method for Koopmans-Beckmann problems. They decomposed the problem in an attempt to reduce the quadratic coefficients and then applied the GL method. This decomposition can be carried out through:

a.) adding a constant to A or B row or column-wise and appropriately modifying C,or, b.) changing the main diagonal of A or B, and appropriately modifying C.

which keeps the equivalence of the original and modified problems.

Frieze and Yadegar [25] linked *the reduction* method to a Lagrangian relaxation approach. They introduced a mixed-integer programming formulation of QAP and computed two LBs through the approximate solutions of the Lagrangian relaxation of this formulation.

Assad and Xu's [2] method generates a monotonic sequence of LBs and may be interpreted as a Lagrangian dual ascent procedure.

Carraresi and Malucelli's [16] approach is also a consequence of the reduction method.

The last family of LBs is lately introduced by Finke et.al. [23], Rendl et.al. [62] and Hadley et.al. [33]. The main approach is instead of minimizing over the set of permutation matrices, to minimize over orthogonal matrices. The results are applicable to symmetric Koopmans-Beckmann problems only. We will state their main result excluding its proof:

Theorem 2.3 (Finke, Burkard and Rendl) *Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ be the eigenvalues of A and $\mu_1, \mu_2, \cdots, \mu_n$ be the eigenvalues of B. Since A and B are symmetric, λ_is and μ_is are real. Assume λ_is and μ_is are in nondecreasing order. Then for all permutations φ*

$$\sum_{i \in \mathcal{N}} \lambda_i \; \mu_{n-i} \;\; \leq \;\; \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{N}} a_{ik} \; b_{\varphi(i)\varphi(k)} \;\; \leq \;\; \sum_{i \in \mathcal{N}} \lambda_i \; \mu_i \tag{19}$$

This is a tool for bounding quadratic part only. The lower bound for the objective function is

$$\sum_{i \in \mathcal{N}} \lambda_i \mu_{n-i} \; + \; \min_{x \in AP^n} \; \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} d_{ij} x_{ij} \tag{20}$$

But this lower bound has no practical value since it is generally dominated by the trivial lower bound zero (Hadley et.al. [33]). But it can be improved if the quadratic part can be modified. This can be achieved through the reduction methods. Finke et.al. [23] and Rendl et.al [62] proposed two different strategies:

- Finke et.al. [23] tried to minimize the spreads of A and B while keeping them symmetric. The LB obtained by this transformation is compatible with GL bounds.

- Rendl et.al. [62] proposed an iterative improvement technique for the determination of transformation parameters. The bounds obtained are the best ones available so far but computational effort is enormous.

Lately Hadley et.al. [33] tried to minimize over orthogonal matrices having constant row and column sums. This resulted in strong and easily computable LBs.

2.4 Exact Solution Methods

There are mainly two types of procedures:

1.) **Branch-and-Bound** (Implicit Enumeration) **Methods:**

a.) **Single-assignment algorithms :**
At each node of the branch-and-bound tree a facility is assigned to a location and a lower bound is computed for the resulting subproblem. Gilmore [28], Lawler [43], Graves and Whiston [30], Bazaara and Elshafei [5], Burkard and Stratmann [15], Kaku and Thompson [38], and Edwards [20] are some to be stated. The crucial part is the previously studied lower bounding techniques.

b.) **Pair-assignment algorithms :**
At each node of the branch-and-bound tree a pair of facilities is assigned to a pair of locations and a lower bound is computed for the resulting subproblem. This technique is only used by Land [41], and Gavett and Plyter [27]. The reason of low reputation is that it is out performed by single-assignment methods.

c.) **Others:**
In the *relative positioning algorithm* of Mirchandani and Obata [46] the levels in the branch-and-bound tree do not corresponds to the assignment of facilities to locations. The partial placements at each

level are in terms of distances between facilities,that is, their relative positions. Pierce and Crowston's [58] *pair-exclusion* algorithm proceeds on the basis of a stage-by-stage exclusion of assignments from a solution to the problem. Also Roucairol [64] proposed a *parallel* branch-and-bound algorithm.

2.) **Cutting-plane methods:**
Bazaara and Sherali [8] utilized several cutting planes. They first transform the QAP to the minimization of a concave quadratic function over the assignment polytope AP^n. Then they investigate several cutting planes, such as, intersection cuts and disjunctive cuts. Although they obtained stronger cuts using the special structure of QAP, they realized that either the cuts obtained are useless for higher dimensions or number of cuts needed for termination is $2(n-2)!$. Also they stated that they have worked on several other cuts but *"they all in vain"*.

Another solution methodology is to use some linearization scheme followed by the solution of a mixed-integer programming problem. Kaufman and Broeckx [39] and Bazaara and Sherali [7] solved the resulting MIP problem through Benders' decomposition. But computational experience with this method is not satisfactory too. Oral and Kettani [49] solved MIP problem 2.4 with a MIP solver. Solution times are reasonable for sizes up to $n = 15$.

All the exact solution methodologies utilized up to now, both branch-and-bound techniques and cutting plane methods, are not capable of solving problems of size $n > 17$. When you realize that complete enumeration can solve problems of size $n < 11$, the weakness of exact solution techniques is apparent.

2.5 Heuristics

Heuristics are designed for finding good feasible solutions which are not necessarily optimal. A good heuristic will require reasonable CPU time, will yield good quality solutions and will be easy to implement. Several heuristics are proposed for QAP:

a.) Construction methods:

As the name suggests they start from scratch and construct a complete assignment by locating one or more facilities at each iteration. Gilmore [28] is the first of this type. It depends on the Gilmore's lower bound,i.e. GL bound. After constructing the F matrix (section 2.3), starting from the null assignment, a complete assignment is reached by successively applying:

- Select through
  - a maximin criterion or
  - the solution of Linear Assignment Problem
- Make the new assignment
- Update F

The difference between various construction procedures is the selection of next assignment. Graves and Whiston's [30] choice of the following assignment is done so as to minimize the value of the associated *mean completion* of the partial permutation thus formed.

Edwards et.al.s [21] *pair linking* procedure considers, in each iteration, a pair of facilities which has the highest *traffic intensity*, a_{ij}. Then $\varphi(i)$ is set to the closest location to $\varphi(j)$, if j is already assigned. If j is also unmatched, two unoccupied locations with smallest inter-distance are chosen.

Hillier and Connors [37] and Gaschutz and Ahrens [26] are among other construction procedures.

b.) Improvement Methods:

They start from a feasible solution, possibly obtained by a construction method, and try to get better solutions with lower objective function values. The search for better solutions is carried out by applying some exchange routines to the existing solution. There are some decision rules involved:

a.) How many assignments to interchange at a time, that is, pairwise, triple-size etc.

b.) When to restart, i.e. when to update the current solution.

c.) In which order to consider exchanges.

d.) Whether to update this order after a restart or not.

Hillier [36] and Hillier and Connors [37] consider a certain set of pairwise exchanges among $\binom{n}{2}$ ones. Armour and Buffa [1] considers all pairwise exchanges and restarts with the one having minimum cost. Parker [57] proposed several improvement procedures which differs with respect to rules above. His comparison yields that Armour and Buffa is superior to others. Heider [35] makes the restart at the first improvement. Steinberg [68] is another pairwise exchange algorithm.

Lashkari and Jaisingh [42] proposed a different improvement scheme. It is an improvement over the Gilmore's [28]. Remember that final matrix F (section 2.3) is the basis of decisions in Gilmore's construction method. The entries f_{ij} are lower bounds on the assignment $i \rightsquigarrow j$. But these take into account only the interaction between ij and other assignments, while ignoring the cost between the remaining ones. Their motivation is to update f_{ij}s in the hope of decreasing this bias. Reeves [61] improved this procedure on the basis of computational time.

c.) Modified enumeration methods:

They use an exact solution scheme in conjunction with some heuristic techniques. Hence,in some sense, exact solution method is used over a subset of the feasible region which is obtained by the use of heuristic techniques throughout the course.

Three different exact solution methods are incorporated: Benders' decomposition, cutting planes and branch-and-bound.

Bazaara and Sherali [7] implements Benders' partitioning method to a MIP formulation of the QAP and before adding the cut apply several heuristic improvement procedures to the solution found throughout the course of partitioning.

Burkard and Bonniger [10] realized that cutting planes can be directly obtained without Benders' decomposition by applying a linearization technique proposed by Balas and Mazzola [3]. They heuristically obtained cuts, in a sense a relaxation of the master problem is solved, and pairwise exchange algorithms are applied to the resulting solution.

Bazaara and Sherali [8] used the cutting planes they found heuristically. In the heuristic procedure number of cuts to be generated is bounded. This is because of the diminishing returns. Through an observation only a certain portion of the solution space is considered. After performing pairwise exchanges new cuts are generated.

In their heuristic branch-and-bound method, Bazaara and Kirca [6] used several techniques to decrease the search effort made. They eliminate *mirror image* branches, use improvement heuristics, impose variable upper bounds and ignore some *presumably bad* branches.

Burkard and Stratmann's [15] is a similar one. It alternates between a branch-and-bound routine and an exchange routine.

d.) Probabilistic local search methods:

All previously stated improvement procedures are deterministic. Deterministic in this case means that, given a particular starting solution, a particular sequence of improved solutions is generated, leading to a certain solution. Repeated applications of the complete procedure using the same starting solution would yield the identical trial to the identical final solution.

Nugent et.al. [47] are first to propose a probabilistic search heuristic, namely *biased sampling*. It is a pair exchange method. But in order to escape from local optimum it assigns certain acceptance probabilities to each pair exchange with positive cost reduction. Those probabilities is related to percentage cost reduction. Then chooses among those.

This idea, with a different perspective, is used in Burkard and Rendl [14], Wilhelm and Ward [70] and Skorin-Kapov [67]. The main difference of those methods from that of Nugent et.al.s is that increases in the objective function can occur. Burkard and Rendl [14],and Wilhelm and Ward [70] used simulated annealing. Wilhelm and Ward also made an experimental study on the parameters of simulated annealing. Skorin-Kapov [67] applied tabu search.

2.6 Probabilistic Asymptotic Behavior of QAP

Burkard and Finke [11] showed that a rather strange property holds for general QAPs. They showed that under certain assumptions the ratio between the objective function values of the best and the worst solutions is arbitrarily close to one with probability tending to one as the size of the instance approaches to infinity. Namely their result is:

Theorem 2.4 (Burkard and Finke) *For $n \in N$ let $c_{ijkl} \forall i,j,k,l \in \mathcal{N}$ be uniform random variables, independently distributed in [0,1]. Then*

$$\mathcal{P}\left\{ \frac{\max_{\varphi} \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{N}} c_{i\varphi(i),k\varphi(k)}}{\min_{\varphi} \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{N}} c_{i\varphi(i),k\varphi(k)}} \leq 1+\varepsilon \right\} \overset{n\rightarrow\infty}{\longrightarrow} 1 \qquad (21)$$

Frenk et.al. [24] strengthen the result and showed that convergence is *almost everywhere.*

These results imply that almost any method, even random choice, would yield good solutions for large QAPs. Hence, in generating QAP instances one has to be extremely careful.

Burkard and Finke [11] showed that the probabilistic asymptotic properties stated are also valid for certain discrete optimization problems. Asymptotic behavior of such problems are determined by the number of feasible solutions and the number of coefficients in the objective function. Whenever the number of coefficients in the objective function increases faster then the logarithm of the number of feasible solutions, a behavior like this can be expected. But these do not hold for Linear Assignment Problems and Traveling Salesman Problems.

Chapter 3

Quadratic Assignment Polytope

Polyhedral combinatorics is the use of the polyhedral theory in the solution of combinatorial problems. In the past two decades it has been a rapidly growing field. Successful applications to the Traveling Salesman Problem (Padberg and Rinaldi [55]), max-cut problem (Barahona et.al. [4]), set covering and set packing problems (Padberg [50]) made the area to be promising. This fact encouraged us to explore the polyhedral aspects of QAP. In the subsequent sections basic definitions related to graph theory and polyhedral theory will be given. In section 3.2 we will try to give the general methodology of polyhedral combinatorics for 'hard' problems. The material in 3.2 is based on papers by Pulleyblank [60], Padberg and Grötschel [52], Grötschel and Padberg [32], Padberg and Rinaldi [55] and Lovasz and Schrijver [44]. The remaining parts of the chapter will be devoted to our preliminary work towards understanding the polyhedral structure of QAP. Three families of equality sets, partition equalities, layer equalities and leaf equalities, will be introduced. Later, two families of valid inequalities, triangle and chordless cycle inequalities, will be given.

3.1 Definitions

A *graph* $G = (V, E)$ consists of a finite, nonempty set $V = \{1, 2, \cdots, n\}$ and a set $E = \{e_1, e_2, \cdots, e_m\}$ whose elements are subsets of V of size 2, that is, $e_i = (u, v)$ where $u, v \in V$. The elements of V are called *nodes*, and the elements of E are called *edges*. We say $e_i \in E$ is *incident to* $v \in V$ or that v is an *endpoint* of e_i, if $v \in e_i$.

If $H = (W, F)$ is a graph with $W \subseteq V$ and $F \subseteq E$, then H is called a *subgraph* of G. Subsets of a set can be represented by *incidence* vectors. Thus $W \subset V$ can be given by the vector $w \in \mathcal{R}^n$ where $n = \mid E \mid$, $w = (w_1, w_2, \cdots, w_n)$ and w_i is one if $i \in V$, zero otherwise.

Given $W, S \subseteq V$ we define the following edge sets:

$$\delta(W) = \{i \in E : \text{one end of i is in W}\}$$

$$\gamma(W) = \{i \in E : \text{both ends of i are in W}\}$$

$$(W : S) = \{i \in E : \text{one end of i is in W and the other one is in S}\}$$

If $W = \{v\}$, $\mid \delta(v) \mid$ is called the *degree* of node v.

A graph $G = (V, E)$ is called *complete* if $\forall i, j \in V \; e_{ij} \in E$, i.e. there is an edge between every pair of nodes. The complete graph with n nodes is denoted by K_n. A *clique* is a complete subgraph.

A graph $G = (V, E)$ is called *n-partite* if V can be decomposed into disjoint node sets $S_1, S_2, \cdots, S_n$ such that $\gamma(S_i) = \emptyset \; i = 1, 2, \cdots, n$.

A graph $G = (V, E)$ is called *k-regular* if $\forall v \in V \mid \delta(v) \mid = k$.

A node set $W \subseteq V$ is said to induce a *chordless cycle* if the nodes of W can be ordered as $(v_1, v_2, \cdots, v_p)$ such that

$$(v_r, v_s) \in E \iff s = r + 1 \text{ or } s = 1 \text{ and } r = p$$

If $x_1, \cdots, x_k \in \mathcal{R}^n$ and $\lambda_1, \cdots, \lambda_k \in \mathcal{R}$, then the vector $x = \lambda_1 x_1 + \cdots + \lambda_k x_k$ is called a *linear combination* of the vectors $x_1, \cdots, x_k$. If λ_i in addition satisfy $\lambda_1 + \cdots + \lambda_k = 1$, then x is called an *affine combination* of vectors $x_1, \cdots, x_k$. If

$x = \lambda_1 x_1 + \cdots + \lambda_k x_k$ is an affine combination such that $\lambda_i \geq 0 \; i = 1, \cdots, k$, then x is called a *convex combination* of the vectors $x_1, \cdots, x_k$. Vectors $x_1, \cdots, x_k$ are called *linearly independent* if linear combinations equal to zero only if $\lambda_i = 0 \; i = 1, \cdots, k$; otherwise *linearly dependent.*

If $\emptyset \neq S \subseteq \mathcal{R}^n$, then the set of all linear (affine, convex) combinations of finitely many vectors in S is called the *linear* (*affine, convex*) *hull* of S and is denoted by lin(S) (aff(S),conv(S)). A set $S \subseteq \mathcal{R}^n$ with S = lin(S) (S = aff(S), S = conv(S)) is called a *linear subspace* (*affine subspace, convex set*). Affine subspaces of the form $H = \{x \in \mathcal{R}^n \mid a^T x = a_0\}$ where $a \in \mathcal{R}^n - \{0\}$ and $a_0 \in \mathcal{R}$ is called a *hyperplane.* Hyperplane H divides the whole space into two halfspaces such that, $H_1 = \{x \in \mathcal{R}^n \mid a^T x \leq a_0\}$ and $H_2 = \{x \in \mathcal{R}^n \mid a^T x \geq a_0\}$. An inequality $a^T x \leq a_0$ is called *valid* with respect to S if S is contained in the halfspace H_1.

A *polyhedron* is the intersection of finitely many halfspaces, i.e. every polyhedron P can be represented in the form $P = \{x \in \mathcal{R}^n \mid Ax \leq b\}$. A bounded polyhedron is called a *polytope.*

A subset F of a polyhedron P is called a *face* of P if there exists an inequality (a, a_0) valid with respect to P such that $F = \{x \in P \mid a^T x = a_0\}$. A face with only one element is called a *vertex.* A *facet* F of P is a proper, nonempty face, (i.e. a face satisfying $\emptyset \neq F \neq P$) which is maximal with respect to set inclusion. The *dimension* of the set $S \subseteq \mathcal{R}^n$, denoted by dim($S$), is the cardinality of largest affinely independent subset of S minus one. S is called *full-dimensional* if dim(S)= n.

3.2 General Methodology of Polyhedral Combinatoric

Assume that we are given an instance Q of a hard combinatorial optimization problem. Let P be the convex hull of the feasible solutions of Q. By a theorem of Weyl [69] there exists a finite set of linear inequalities which define P and whose vertices are precisely the solutions of Q. That is :

$$P = \{y \in \mathcal{R}^J \mid ly \leq l_0 \;\forall (l, l_0) \in \mathcal{L}\} \qquad (1)$$

where $\mathcal{L}$ is a finite family of linear inequalities. If this system is minimal and non-redundant, (l, l_0)'s are facets. So, if we can characterize $\mathcal{L}$, we can solve Q by linear programming based algorithms. The optimal solution of Q can be found by solving the following Linear Programming Problem :

Problem 3.1

$$\begin{aligned} \min \quad & c^T y \\ s.t. \quad & ly \leq l_0 \quad \forall (l, l_0) \in \mathcal{L} \\ & y \geq 0 \end{aligned}$$

The number of facets of P is exponentially large in the length of original combinatorial structure, therefore there is little hope that complete and nonredundant systems of linear inequalities describing P will ever be found for 'hard' combinatorial problems. Therefore, if a sub-family $\mathcal{L}'$ of the family of defining inequalities $\mathcal{L}$ is known, following relaxation is solved:

Problem 3.2

$$\begin{aligned} \min \quad & c^T y \\ s.t. \quad & ly \leq l_0 \quad \forall (l, l_0) \in \mathcal{L}' \\ & y \geq 0 \end{aligned}$$

Its solution y^* is either the incidence vector of a feasible solution or it violates some unknown inequality contained in $\mathcal{L} - \mathcal{L}'$. In the first case we have solved the problem 3.1 In the second case, we have a lower bound on the optimal value of problem 3.1 and we can resort to a Branch and Cut algorithm. In Branch and Cut, the cuts used in each node of the search tree is globally valid inequalities, namely the elements of the $\mathcal{L}'$, for the polytope P. Details of Branch and Cut approach is given in section 3.6.

The cardinality of the subfamily $\mathcal{L}'$ can be super-exponentially large and hence it is impossible to solve problem 3.2 by giving an explicit list of all inequalities in $\mathcal{L}'$. Yet problem 3.2 can be solved by the following Polyhedral Cutting Plane Algorithm (PCPA):

Algorithm 3.1 (Polyhedral Cutting Plane Algorithm) *Set* $\mathcal{L}_0 = \mathcal{L}'$
STEP1: Set $\mathcal{L}' = \emptyset$
STEP2: Solve problem 3.2 and let $\bar{y}$ *be its optimal solution.*
STEP3: Find one or more inequalities in $\mathcal{L}_0$ *violated by* $\bar{y}$
STEP4: If none is found stop. Otherwise add the violated inequalities to $\mathcal{L}'$ *and go to STEP2.*

Algorithm 3.1 stops after a finite number of steps because $\mathcal{L}_0$ is finite. The core of the procedure is Step 3, which is called the **identification problem** (or **separation problem**) and which is stated as follows:

Problem 3.3 *Given a point* $y \in \mathcal{R}^J$ *and a family* $\mathcal{L}'$ *of inequalities, identify one or more inequalities in* $\mathcal{L}'$ *violated by* y *or prove that no such inequality exists.*

An identification procedure accepts as input the weighted support graph of a point (or current vector $\bar{y}$) which is not contained in the related polytope and returns as output some of the inequalities violated by the point. Given a family of inequalities $\mathcal{L}'$, we call a procedure that solves problem 3.3 **exact**, and we call a procedure that sometimes identifies violated inequalities, **heuristic**. Above results imply that we can optimize in polynomial time over the related partial description of the polytope P if we can solve the separation problems of the families of valid inequalities in $\mathcal{L}'$. This sequential procedure has its roots in the seminal paper of Dantzig, Fulkerson and Johnson [19]. They applied PCPA to the traveling salesman problem. They used subtour elimination constraints to obtain the partial description. After each iteration they visually identify the violated subtour elimination constraints and add those by hand to the relaxation. This approach is disregarded for a long time. In the late seventies the findings on the facial structure of traveling salesman problem and Ellipsoid algorithm initiated new research. A good review of research on traveling salesman problem can be found in Grötschel and Padberg [32] and Padberg and Grötschel [52]. Hong [?], Padberg and Hong [53], Crowder and Padberg [18], Padberg and Rao [54], and Padberg and Rinaldi [55] [56] tried to automize the PCPA. The largest problem solved before PCPAs is on 120 nodes (Grötschel [31]) where as Padberg and Rinaldi [56] reports the solution of a problem on 2392 nodes.

While dealing with an $\mathcal{NP}$-hard problem the best you can hope for is to obtain a partial description of the underlying polytope and to use PCPA. When PCPA stops you can go to branch-and-bound. This in fact what branch-and-cut is. We adopt this methodology in the context of QAP. Basic steps are:

i.) Represent the feasible objects by vectors (usually by incidence vectors).

ii.) Consider these vectors as points in $\mathcal{R}^J$ for suitable J and let P be their convex hull.

iii.) Obtain families of valid inequalities, preferably facets, which gives a partial but presumably strong definition of P.

iv.) Use those families of valid inequalities in a branch-and-cut algorithm.

We will try to follow the above steps.

3.3 Graphical Representation and a New Formulation of QAP

Given an instance QAP^n of QAP of size n, we associate the following graph $G = (V, E)$ with the feasible solutions of QAP^n:
Each node x_{ij} corresponds to the variable x_{ij}. Clearly, $| V |= n^2$. An edge y_{ijkl} between nodes x_{ij} and x_{kl} exists if and only if the multiplicative term $x_{ij}x_{kl}$ equals one in at least one of the feasible solutions of the QAP^n. Since x is a solution of QAP^n if and only if it is in assignment polytope, i.e.

$$AP^n = \{x \in \mathcal{R}^{n^2} \ : \ \sum_{i \in \mathcal{N}} x_{ij} = 1 \ \forall j \in \mathcal{N}; \ \sum_{j \in \mathcal{N}} x_{ij} = 1 \ \forall i \in \mathcal{N}\} \qquad (2)$$

Now consider following pairs of nodes:

$$x_{ij} \text{ and } x_{il} \ j \neq l \ \Rightarrow x_{ij}x_{il} = 0 \text{ in all feasible solutions} \qquad (3)$$

$$x_{ij} \text{ and } x_{kj} \ i \neq k \ \Rightarrow x_{ij}x_{kj} = 0 \text{ in all feasible solutions} \qquad (4)$$

Above facts follow from $x \in AP^n$. Therefore,

$$y_{ijkl} \in E \iff i \neq k \text{ and } j \neq l \qquad (5)$$

From a simple counting scheme

$$\delta(x_{ij}) = (n-1)^2 \ \ \forall x_{ij} \in V \tag{6}$$

which implies that $\mid E \mid = \frac{1}{2}n^2(n-1)^2$. Therefore the graph G is $(n-1)^2$-regular.

Consider the following node sets $S_1, \cdots, S_n \subseteq V$ such that

$$S_i = \{x_{ij} : \ j \in \mathcal{N}\} \ \ \forall i \in \mathcal{N} \tag{7}$$

Since $S_i \cap S_j = \emptyset \ \ i \neq j \ \forall i, j \ \in \mathcal{N}$ we have a disjoint node partition of G. The facts $\gamma(S_i) = \emptyset \ \ \forall \ i \in \mathcal{N}$ and $(S_i : S_j) \neq \emptyset \ \ \forall i, j \in \mathcal{N}$ imply that G is an n-partite graph.

Theorem 3.1 *For each feasible solution $\bar{x}$ of QAP^n there corresponds the following subgraph $H = (W, F)$ of $G = (V, E)$ where*

$$W = \{x_{ij} \ : \bar{x}_{ij} = 1\} \tag{8}$$

$$F = \{y_{ijkl} \ : x_{ij}, \ x_{kl} \ \in W\} \tag{9}$$

H is a complete graph on n vertices and this correspondence is one to one.

proof:
Assume we are given a feasible solution $\bar{x} \in AP^n$. By assignment constraints 2, $\bar{x}$ contains exactly one node from each partition and layer. By the definition of G, there exist an edge between every pair of nodes of $\bar{x}$. Therefore, H is a complete graph on n vertices.
Now, assume we are given a subgraph $H = (W, F)$ of G which is a K_n. $\mid W \mid = n$. Since G is an n-partite graph $\mid W \cap S_i \mid = 1 \ i = 1, \cdots, n$. So, the incidence vector x_H satisfies the first set of assignment constraints.
Letting

$$W = \{x_{1j_1}, \cdots, x_{nj_n}\}$$

$\forall \ i, k \ i \neq k \ j_i \neq j_k$ because the edge y_{ij_i,kj_k} exists. So, the second set of assignment constraints are also satisfied. Therefore, $x_H \in AP^n$ and solves QAP^n.

We define Quadratic Assignment Polytope, QA^n, to be the convex hull of the incidence vectors of maximal cliques of G,

$QA^n = conv\{y \in \mathcal{R}^{\frac{1}{2}n^2(n-1)^2} :$ y is the incidence vector of a maximal clique in G$\}$

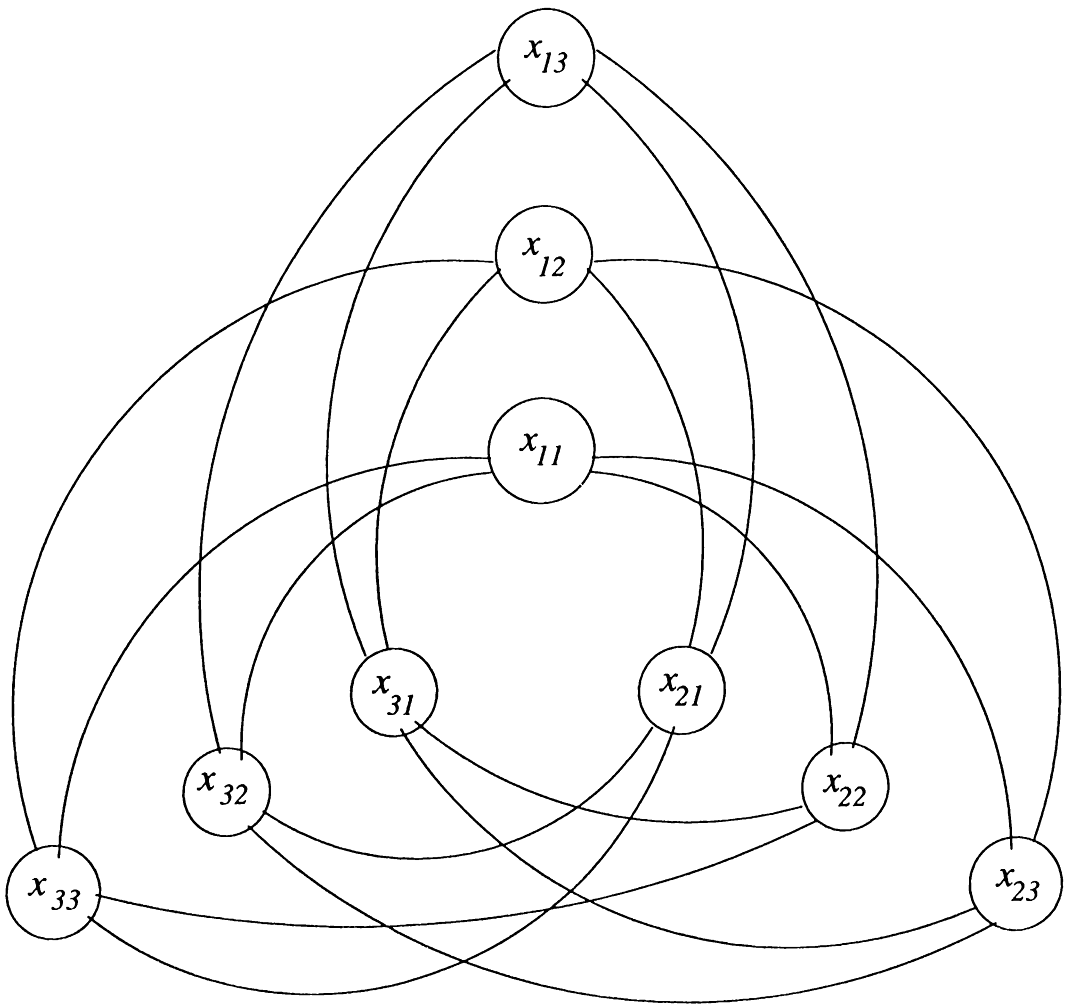

Figure 3.1: Graph G for n = 3

If we associate the weights c_{ijkl} with each edge y_{ijkl} of G, we can define a different problem, Minimum Weight Clique Problem (MWCP), over G:

Problem 3.4 (Minimum Weight Clique Problem (MWCP))

$$\min_{y \in QA^n} c^T y \qquad\qquad MWCP$$

Corollary 3.1 *The feasible solutions of problem 2.2 and MWCP can be placed in one-to-one correspondence with equal objective function values.*

QAP has been transformed to finding a maximal clique in a graph. This is a different conceptualization. Analyzing the structure of the graph G may yield substantial information about the solution space of QAP.

3.4 Partition and Layer Equalities for QA^n

QAP is originally a nonlinear integer problem in the x space of dimension n^2. The equivalent problem MWCP is a linear problem in y space of dimension $m = \frac{1}{2}n^2(n-1)^2$. We linearize the problem at the cost of increasing the dimension and a more structured formulation. Since there is a direct relationship, one-to-oneness, between the feasible solutions in x and y spaces, we expect that assignment constraints of x space, in some way, will be carried to the higher dimensional y space. We will state two equality sets which imitates the role of assignment constraints in x space. But those are not enough for obtaining a clique in the y space. The first set, **partition constraints**, tries to get each facility to be assigned to a single location and the second set, **layer constraints** tries to match each location with a single facility.

Theorem 3.2 *Each $y \in QA^n$ satisfies following partition constraints*

$$\sum_{y_{ijkl} \in (S_i : S_k)} y_{ijkl} = 1 \qquad \forall\, i,k \in \mathcal{N} \qquad (10)$$

proof:
Let y be the incidence vector of a K_n and $H_y = (W, F)$ be the support graph of y. Hence

$$| W \cap S_i |= 1 \; \forall \; S_i \; i \in \mathcal{N}$$

Therefore, exactly one node from each partition exists and only one of the edges going between a pair of partitions can be positive.

Theorem 3.3 *Each* $y \in QA^n$ *satisfies following layer constraints*

$$\sum_{y_{ijkl} \in (L_j : L_l)} y_{ijkl} = 1 \qquad \forall \, j, l \in \mathcal{N} \tag{11}$$

where $L_j = \{x_{ij} \; : \; i = 1, \cdots, n\}$ *is called to be a layer.*

proof:
Let y be the incidence vector of a K_n and $H_y = (W, F)$ be the support graph of y. Consider layers $L_1, ..., L_n$

$$L_j \cap L_l = \emptyset \quad \forall j, l \in \mathcal{N} \quad j \neq l \quad \Rightarrow \quad L_1, ..., L_n \quad \text{is an n-partition of} \quad G = (V, E) \tag{12}$$

Since $| W |= n$ and H_y is a clique, $| W \cap L_j |= 1$, which implies that only one of the edges going between a pair of layers can be positive.

3.5 Special Structures in $G = (V, E)$

It is time to see what we can gain from binary relations and the structure of the graph. Previous work on Polyhedral Combinatorics has shown that some structures like cliques, cycles and their unions can be fruitful to analyze. The rational behind the approach is that the whole graph is the union of such simple structures. If one can find valid inequalities for those special structures, one can use them in some lifting procedure and use in the solution of higher dimensional instances. The first structure we will investigate is the *triangles.* A triangle is a clique of three nodes. Valid inequalities for triangles are studied in the context of max-cut problem (Barahona and Mahjoup [4]), quadratic boolean programming (Padberg [51]), partitioning problem (Chopra and Rao [17]) and various other problems.

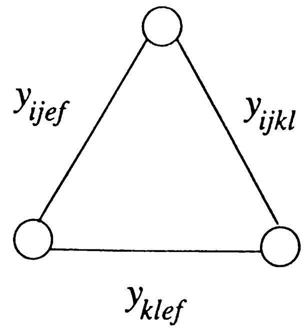

Figure 3.2: A triangle

Theorem 3.4 *Let $F \subset E$ such that the graph induced by F is a triangle. Letting $F = \{y_{ijkl},\ y_{ijst},\ y_{klst}\}$, following* **triangle inequalities** *are valid for QA^n.*

$$y_{ijkl} + y_{ijst} - y_{klst} \leq 1 \tag{13}$$

$$y_{ijkl} + y_{klst} - y_{ijst} \leq 1 \tag{14}$$

$$y_{klst} + y_{ijst} - y_{ijkl} \leq 1 \tag{15}$$

proof:
Let y be the incidence vector of a K_n and $H_y = (W, D)$ be the support graph of y. Inequalities 13, 14 and 15 say that $\mid F \cap D \mid \neq 2$ which is true for any clique, hence extreme point of QA^n. Therefore it is valid for QA^n.

The second structure we will examine in G is *chordless cycles.* At first we shall prove that G contains only chordless cycles of length four and six. Then we will introduce a valid inequality for chordless cycles of length four.

Theorem 3.5 *The graph $G = (V, E)$ does not contain chordless cycles of length 7.*

proof:
Suppose there exist a chordless cycle $C \in \mathcal{C}_7$ and without loss of generality $C = \{y_{ijkl}, y_{klrp}, y_{rpst}, y_{stuv}, y_{uvcd}, y_{cdef}, y_{ijef}\}$. The existence of C depends on

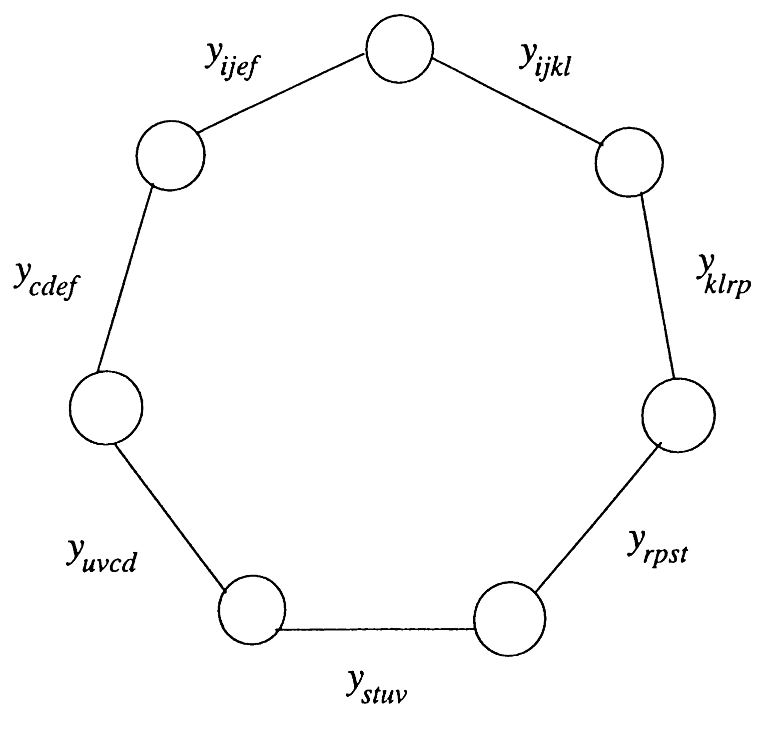

Figure 3.3: Chordless cycle of length 7

some *binary relations* on the indices of the nodes. These binary relations state the conditions for the existence or the absence of the edges, i.e. the indices must be so that the edges in C must exist and the edges which will constitute a chord must not exist. We will call them *edge existence* (EER) and *chordless cycle* relations (CCR) respectively. If $C \in \mathcal{C}_7$, following CCR must be simultaneously satisfied

$$y_{ijrp} \notin E \quad \Rightarrow \quad (i = r) \lor (j = p) \tag{16}$$

$$y_{ijst} \notin E \quad \Rightarrow \quad (i = s) \lor (j = t) \tag{17}$$

$$y_{ijuv} \notin E \quad \Rightarrow \quad (i = u) \lor (j = v) \tag{18}$$

$$y_{ijcd} \notin E \quad \Rightarrow \quad (i = c) \lor (j = d) \tag{19}$$

$$y_{klst} \notin E \quad \Rightarrow \quad (k = s) \lor (l = t) \tag{20}$$

$$y_{kluv} \notin E \quad \Rightarrow \quad (k = u) \lor (l = v) \tag{21}$$

$$y_{klcd} \notin E \quad \Rightarrow \quad (k = c) \lor (l = d) \tag{22}$$

$$y_{klef} \notin E \quad \Rightarrow \quad (k = e) \lor (l = f) \tag{23}$$

$$y_{rpuv} \notin E \quad \Rightarrow \quad (r = u) \lor (p = v) \tag{24}$$

$$y_{rpcd} \notin E \quad \Rightarrow \quad (r = c) \lor (p = d) \tag{25}$$

$$y_{rpef} \notin E \quad \Rightarrow \quad (r = e) \lor (p = f) \tag{26}$$

$$y_{stcd} \notin E \quad \Rightarrow \quad (s = c) \lor (t = d) \tag{27}$$

$$y_{stef} \notin E \quad \Rightarrow \quad (s = e) \lor (t = f) \tag{28}$$

$$y_{uvef} \notin E \quad \Rightarrow \quad (u = e) \lor (v = f) \tag{29}$$

and the corresponding (EER) are

$$y_{ijkl} \in E \quad \Rightarrow \quad (i \neq k) \land (j \neq l) \tag{30}$$

$$y_{klrp} \in E \quad \Rightarrow \quad (k \neq r) \land (l \neq p) \tag{31}$$

$$y_{rpst} \in E \quad \Rightarrow \quad (r \neq s) \land (p \neq t) \tag{32}$$

$$y_{stuv} \in E \quad \Rightarrow \quad (s \neq u) \land (t \neq v) \tag{33}$$

$$y_{uvcd} \in E \quad \Rightarrow \quad (u \neq c) \land (v \neq d) \tag{34}$$

$$y_{cdef} \in E \quad \Rightarrow \quad (c \neq e) \land (d \neq f) \tag{35}$$

$$y_{ijef} \in E \quad \Rightarrow \quad (i \neq e) \land (j \neq f) \tag{36}$$

All the relations listed above must be satisfied simultaneously, i.e. they are connected by an "and" $\wedge$ operator. Using this fact we will step by step combine the relations into a single binary relation.

Let us start with 16 $\wedge$ 17

$$(i = r = s) \vee [(i = r) \wedge (j = t)] \vee [(i = s) \wedge (j = p)] \vee (j = p = t) \quad (37)$$

By 32, 37 reduces to

$$[(i = r) \wedge (j = t)] \vee [(i = s) \wedge (j = p)] \quad (38)$$

Now, take 18 $\wedge$ 19

$$(i = u = c) \vee [(i = c) \wedge (j = v)] \vee [(i = u) \wedge (j = d)] \vee (j = v = d) \quad (39)$$

By 34, 39 reduces to

$$[(i = c) \wedge (j = v)] \vee [(i = u) \wedge (j = d)] \quad (40)$$

Taking 38 $\wedge$ 40 yields

$$[(i = r = c) \wedge (j = t = v)] \vee [(i = r = u) \wedge (j = t = d)] \vee$$

$$[(i = s = c) \wedge (j = p = v)] \vee [(i = s = u) \wedge (j = p = d)] \quad (41)$$

By 33, 41 reduces to

$$[(i = r = u) \wedge (j = t = d)] \vee [(i = s = c) \wedge (j = p = v)] \quad (42)$$

Realize that 42 is equivalent to 16 $\wedge$ 17 $\wedge$ 18 $\wedge$ 19. In a similar way the relations 20, 21, 22 and 23 reduces to

$$[(k = e = u) \wedge (l = t = d)] \vee [(k = s = c) \wedge (l = f = v)] \quad (43)$$

and the relations 24, 25 and 26 reduces to

$$[(r = e = u) \wedge (p = d)] \vee [(r = c) \wedge (p = f = v)] \quad (44)$$

Now, evaluating 42 $\wedge$ 43 yields

$$[(i = r = u = k = e) \wedge (j = t = d = l)] \vee$$

$$[(i = r = u) \wedge (j = t = d) \wedge (k = c = s) \wedge (l = f = v)] \vee$$
$$[(i = s = c = k) \wedge (j = l = p = v = f)] \vee$$
$$[(i = s = c) \wedge (j = p = v) \wedge (k = e = u) \wedge (l = d = t)] \qquad (45)$$

By 30, 45 reduces to

$$[(i = r = u) \wedge (j = t = d) \wedge (k = c = s) \wedge (l = f = v)] \vee$$
$$[(i = s = c) \wedge (j = p = v) \wedge (k = e = u) \wedge (l = d = t)] \qquad (46)$$

Finally we evaluate 44 $\wedge$ 46. Realize that 44 $\wedge$ 46 is equivalent to 16 $\wedge \cdots \wedge$ 26.

$$[(i = r = u = e) \wedge (j = t = d = p) \wedge (k = c = s) \wedge (l = f = v)] \vee$$
$$[(i = s = c) \wedge (j = p = v = l = d = t) \wedge (k = e = u = r)] \vee$$
$$[(i = r = u = k = c = s) \wedge (j = t = d) \wedge (l = f = v = p)] \vee$$
$$[(i = s = c = r) \wedge (j = p = v = f) \wedge (k = e = u) \wedge (l = d = t)] \qquad (47)$$

By 36, 34, 32 47 cannot be realized. Hence, G does not contain chordless cycles of length 7, i.e. $\mathcal{C}_7 = \emptyset$.

Theorem 3.6 *The graph G does not contain chordless cycles of length $m > 7$.*

proof:
We will follow a similar method to the previous one. Suppose there exist a $C \in \mathcal{C}_m$ where $m > 7$ and without loss of generality C is shown in figure . We will concentrate on the binary relations over a subset of edges of the C, over $C' = \{y_{ijkl}, y_{klrp}, y_{rpst}, y_{uvcd}, y_{cdef}, y_{ijef}\}$. All the previously stated binary relations, 16, $\cdots$, 36 are still valid with one exception, 33 we do not have edge y_{stuv} any more. There is only one additional requirement

$$y_{stuv} \notin E \qquad \Rightarrow \qquad (s = u) \;\vee\; (t = v) \qquad (48)$$

which guarantees the absence of y_{stuv}. Again all the binary relations must be satisfied simultaneously, that is, all the binary relations are connected by $\wedge$ operator.

Since 16 $\wedge \cdots \wedge$ 19 is not affected by the changes, it is equivalent to 42. Also 24 $\wedge \cdots \wedge$ 26 is still equivalent to 44.

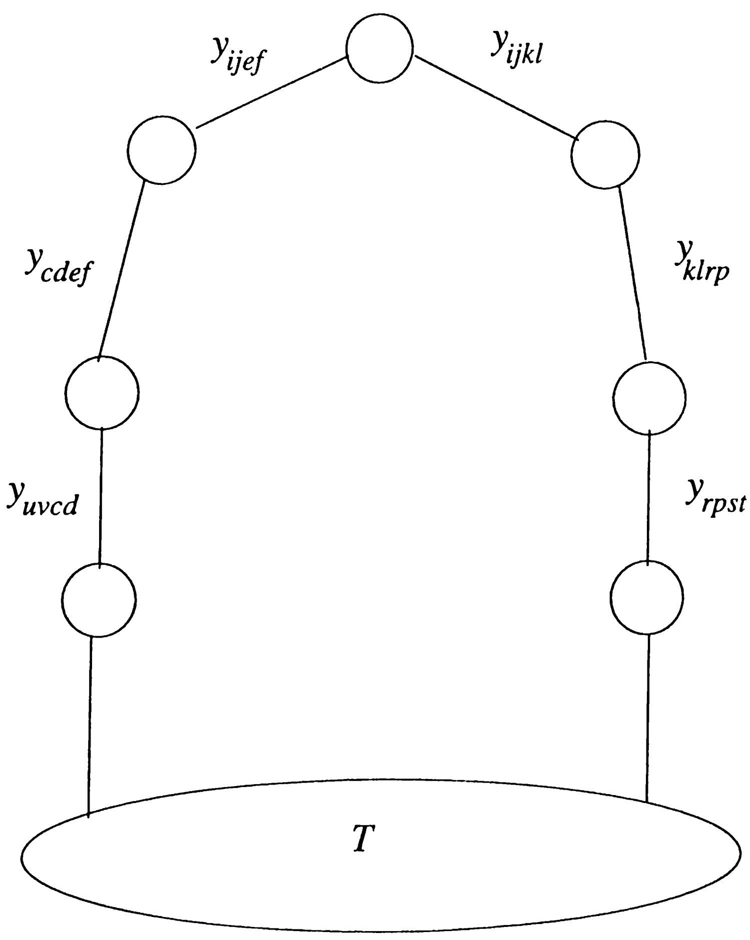

Figure 3.4: Chordless cycle of length m

Let us reevaluate 20 $\wedge \cdots \wedge$ 23. It is

$$[(k=s=u=c)\wedge(l=f)]\vee[(k=s=u=e)\wedge(l=d)]\vee$$

$$[(k=s=c)\wedge(l=v=f)]\vee[(k=s=e)\wedge(l=v=d)]\vee$$

$$[(k=u=c)\wedge(l=t=f)]\vee[(k=u=e)\wedge(l=t=d)]\vee$$

$$[(k=c)\wedge(l=t=v=f)]\vee[(k=e)\wedge(l=t=v=d)] \qquad (49)$$

By 34, 49 reduces to

$$[(k=s=u=c)\wedge(l=f)]\vee[(k=s=e)\wedge(l=v=d)]\vee$$

$$[(k=u=c)\wedge(l=t=f)]\vee[(k=e)\wedge(l=t=v=d)] \qquad (50)$$

Combining 42 with 50 yields

$$[(i=r=u=k=s=e)\wedge(j=t=d=l)]\vee$$

$$[(i=r=u)\wedge(j=t=d)\wedge(k=s=c)\wedge(l=v=f)]\vee$$

$$[(i=r=u=k=e)\wedge(j=t=d=l)]\vee$$

$$[(i=r=u)\wedge(j=t=d=l=v=f)\wedge(k=c)]\vee$$

$$[(i=u=k=s=e=c)\wedge(j=p=v)\wedge(l=d)]\vee$$

$$[(i=s=c=k)\wedge(j=p=v=l=f)]\vee$$

$$[(i=s=c)\wedge(j=p=v)\wedge(k=u=e)\wedge(l=t=d)]\vee$$

$$[(i=s=c=k)\wedge(j=t=p=l=v=f)] \qquad (51)$$

which reduces to

$$[(i=r=u)\wedge(j=t=d)\wedge(k=s=c)\wedge(l=v=f)]\vee$$

$$[(i=s=c)\wedge(j=p=v)\wedge(k=u=e)\wedge(l=t=d)] \qquad (52)$$

by 30, 31, 32, 34, 35 and 36, i.e. *edge existence relations.* When we combine 52 with 44 we get

$$[(i=r=u=e)\wedge(j=t=d=p)\wedge(k=s=c)\wedge(l=v=f)]\vee$$

$$[(i=r=u=k=s)\wedge(j=t=d)\wedge(l=v=f=p)]\vee$$

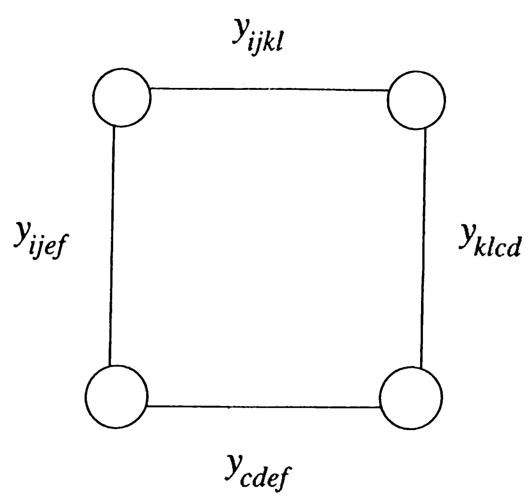

Figure 3.5: Chordless cycle of length 4

$$[(i = s = c) \wedge (j = p = v = l = t = d) \wedge (k = u = e)]$$
$$[(i = s = c = r) \wedge (j = p = v = f) \wedge (k = u = e) \wedge (l = t = d)] \tag{53}$$

30 and 36 tells us that 53 cannot be realized. Therefore $\mathcal{C}_m = \emptyset \; m = 8, 9, 10, \cdots$.

In a similar way it can be shown that G does not contain chordless cycles of length 5 either.

Theorem 3.7 *The graph G contains chordless cycles of length 4 and furthermore for any $C \in \mathcal{C}_4$ following* **chordless cycle** *inequalities are valid for QA^n.*

$$\sum_{y_{ijkl} \in C} y_{ijkl} \leq 1 \tag{54}$$

proof :
Suppose $C = \{y_{ijkl}, y_{klcd}, y_{cdef}, y_{ijef}\}$.

Edge existence relations (EER) are

$$y_{ijkl} \in E \quad \Rightarrow \quad (i \neq k) \;\wedge\; (j \neq l) \tag{55}$$

$$y_{klef} \in E \quad \Rightarrow \quad (k \neq e) \;\wedge\; (l \neq f) \tag{56}$$

$$y_{cdef} \in E \quad \Rightarrow \quad (c \neq e) \ \wedge \ (d \neq f) \tag{57}$$

$$y_{ijef} \in E \quad \Rightarrow \quad (i \neq e) \ \wedge \ (j \neq f) \tag{58}$$

and respective chordless cycle relations (CCR) are

$$y_{ijcd} \notin E \quad \Rightarrow \quad (i = c) \ \vee \ (j = d) \tag{59}$$

$$y_{klef} \notin E \quad \Rightarrow \quad (k = e) \ \vee \ (l = f) \tag{60}$$

Evaluating 59 $\wedge$ 60 yields

$$[(i = c) \wedge (k = e)] \vee [(i = c) \wedge (j = f)] \vee [(j = d) \wedge (k = e)] \vee [(j = d) \wedge (l = f)] \tag{61}$$

61 is not contradictory with EER. Hence, $\mathcal{C}_4 \neq \emptyset$ and moreover there are four ways of constructing a $C \in \mathcal{C}_4$. Validity of chordless cycle inequality 54 follows immediately from the fact that a clique can contain at most one arc from a chordless cycle.

We can show, in a similar way, that also $\mathcal{C}_6 \neq \emptyset$. An inequality of the form 54 is dominated by layer constraints 11 in the case of $\mathcal{C}_6$.

3.6 Branch and Cut Experimentation

In this section we solve the following linear programming relaxation of QAP^n :

Problem 3.5

$$\begin{array}{ll} \min & c^T y \\ s.t. & Ay = 1 \\ & ly \leq l_0 \quad \forall (l, l_0) \in \mathcal{L}' \\ & y \geq 0 \end{array}$$

where $\mathcal{L}'$ is the set of triangle and chordless cycle inequalities for QA^n.

We adopt the Branch and Cut algorithm given by Padberg and Rinaldi [56]. Given a set $\mathcal{L}'$, a subset of valid inequalities for QA^n and two disjoint edge sets $\mathcal{F}_0, \mathcal{F}_1 \subset \mathcal{E}$, we denote by $\mathcal{P}(\mathcal{L}', \mathcal{F}_0, \mathcal{F}_1)$ the linear program

Problem 3.6 $\mathcal{P}(\mathcal{L}', \mathcal{F}_0, \mathcal{F}_1)$

$$
\begin{aligned}
\min \quad & c^T y \\
s.t. \quad & Ay = 1 \\
& ly \le l_0 \quad \forall (l, l_0) \in \mathcal{L}' \\
& y_e = 0 \quad \forall e \in \mathcal{F}_0 \\
& y_e = 1 \quad \forall e \in \mathcal{F}_1 \\
& y \ge 0
\end{aligned}
$$

where A is the constraint matrix of partition and layer equalities. By $\mathcal{S}$ we denote a family of ordered pairs of disjoint edge sets, $(\mathcal{F}_0, \mathcal{F}_1)$ denotes an ordered pair in $\mathcal{S}$, and y^* is the incidence vector of some clique of G.

Algorithm 3.2 (Branch and Cut) *STEP 0. Set* $\mathcal{S} = \{(\emptyset, \emptyset)\}, \mathcal{L} = \emptyset$
STEP 1. If $\mathcal{S} = \emptyset$*, then stop. Otherwise pick an ordered pair* $(\mathcal{F}_0, \mathcal{F}_1)$ *from* $\mathcal{S}$ *and replace* $\mathcal{S}$ *by* $\mathcal{S} - (\mathcal{F}_0, \mathcal{F}_1)$.
STEP 2. Solve the linear program $\mathcal{P}(\mathcal{L}', \mathcal{F}_0, \mathcal{F}_1)$*. If the program is inconsistent, go to Step 1. Otherwise let* $\bar{y}$ *be its optimal solution.*
STEP 3. If $c\bar{y} \ge cy^*$*, go to Step 1.*
STEP 4. Find one or more inequalities of $\mathcal{L}'$ *that are violated by* $\bar{y}$*.*
STEP 5. If none is found , go to Step 6. Otherwise add the violated inequalities to $\mathcal{L}$ *and go to Step 2.*
STEP 6. If $\bar{y}$ *is integer, then replace* y^* *by* $\bar{y}$ *and go to Step 1.*
STEP 7. Pick an edge $e \in \mathcal{E}$ *such that* $0 \le \bar{y}_e \le 1$*. Replace* $\mathcal{S}$ *by* $\mathcal{S} + (\mathcal{F}_0 + \{e\}, \mathcal{F}_1) + (\mathcal{F}_0, \mathcal{F}$ *and go to Step 1.*

Once an ordered pair is removed from $\mathcal{S}$ it is never generated again in Step 7. Since $\mathcal{L}'$ is finite, algorithm terminates in a finite number of steps. When the algorithm stops, y^* is the optimal solution to problem 3.5.

A C code which utilizes the the steps 2 to 6 of the algorithm 3.2 is written. CPLEX Version 2.1 primal simplex algorithm is used as the linear programming solver. Following algorithm is used for solving the separation problem, Step 4, of triangle and chordless cycle inequalities.

problem	SIZE	z^*	z_i	z_f	CPU time	REOPT	TRI	CCI
nug05	5	50	50	50	1	7	18	17
nug06	6	86	82	82	5	14	75	75
nug07	7	148	134	134	53	37	389	404
nug08	8	214	182	182	96	35	520	588
esc08a	8	2	0	0	350	77	1041	1088
esc08b	8	8	0	0	124	40	712	783
esc08c	8	32	4	4	492	85	1242	1364
esc08d	8	6	0	0	572	88	1421	1565
esc08e	8	2	0	0	349	80	947	1022
esc08f	8	18	8	8	21	13	268	265

Table 3.1: Branch and Cut Experimentation

Algorithm 3.3 (Separation Algorithm) *INPUT.* $\mathcal{L}'$ *and the support graph* $G_s = (V_s, E_s)$ *of* $\bar{y}$ *where* $|E_s| = r$.
STEP 0. Put nonzero edges of G_s *into ascending order, i.e.*

$$y_{(1)},\ y_{(2)}, \cdots, y_{(r)}$$

STEP 1.
For $i = 1$ *to* $r - 1$ *do*
For $k = i + 1$ *to* r *do*
Check whether $y_{(i)}$ *and* $y_{(k)}$ *form a chordless cycle or a triangle*
If yes, does respective inequality violated
If so, add violated inequality to $\mathcal{L}$

Results of the experiments are given in table 3.1. Experiments are carried out on a SUN SPARCsystem 300 series computer (20 MIPS 64 Kb RAM). Problems nug* are taken from Nugent et.al. [47]. Problems esc* are taken from Eschermann and Wunderlich [22]. All problem data is supplied by Rendl [12]. z^* stands for the optimal objective function value. z_i is the optimal objective function value of the initial linear programming relaxation and z_f is the optimal objective function value after the cutting plane phase. CPU time is given in seconds. REOPT is the number of re-optimizations. TRI and CCI stands for the number of triangle and chordless cycle inequalities respectively. As it can be observed from the table 3.1, although violated valid inequalities are found and added, y^* has never improved. This made us to think that triangle and chordless cycle inequalities are very weak and are not useful in a branch and cut algorithm. Since no improvement is obtained in the cutting plane phase, we did not resort to branching.

3.7 Leaf Equalities for QA^n

Support graphs of the optimal solutions of the problems 3.5 show us that the graph contains leafs. The idea that a clique cannot contain leaves made us to find the following leaf equalities. They are in a sense balance equalities for nodes of G.

Theorem 3.8 *Each $y \in QA^n$ satisfies following leaf equalities*

$$\sum_{y_e \in (S_k : x_{ij})} y_e - \sum_{y_e \in (S_l : x_{ij})} y_e = 0 \qquad \forall\, x_{ij} \in V\ \forall k, l \in \mathcal{N}\ k \neq l \neq i \quad (62)$$

$$\sum_{y_e \in (L_k : x_{ij})} y_e - \sum_{y_e \in (L_l : x_{ij})} y_e = 0 \qquad \forall\, x_{ij} \in V\ \forall k, l \in \mathcal{N}\ k \neq l \neq j \quad (63)$$

proof:
Let y be the incidence vector of a clique K_n in G. By theorems 3.2 and 3.3 y contains a node from each partition and layer. Therefore, each node in y is *adjacent* to exactly a node from each partition and layer. Since all the edges of y has value one, equations 62 and 63 are satisfied by y.

Leaf-1 equalities 62 state that flow to a node x_{ij} from partition k is equal to flow from partition l. Leaf-2 equalities 63 state that flow to a node x_{ij} from layer k is equal to flow from layer l. For each node x_{ij}, there are $2\binom{n-1}{2}$ leaf equalities. Therefore, number of leaf equalities is $n^2(n-1)(n-2)$. Since leaf equalities 62 for node x_{ij} implies that flow to x_{ij} from partitions is equal to each other and leaf equalities 63 imply that flow to x_{ij} from layers is equal to each other, only $n-2$ leaf-1 62 and $n-2$ leaf-2 63 equalities are needed for x_{ij}. Hence, number of leaf equalities is $2n^2(n-2)$.

Leaf equalities 62 and 63, and, partition equalities 10 and layer equalities 11 are closely related. In the following theorem we will show that only a partition and a layer equality used with leaf equalities are enough.

Theorem 3.9 *Any partition 10 and layer 11 equality can be expressed as a linear combination of an arbitrary partition 10 or layer 11 equality respectively and leaf equalities 62 and 63.*

proof:
Take the following sum over leaf-1 equalities 62

$$\sum_j \Big[\sum_{y_e \in (S_k : x_{ij})} y_e - \sum_{y_e \in (S_l : x_{ij})} y_e \Big] = 0$$

which yields to

$$\sum_{y_e \in (S_k : S_i)} y_e - \sum_{y_e \in (S_l : S_i)} y_e = 0$$

By this way we can show that

$$\sum_{y_e \in (S_k : S_i)} y_e = \sum_{y_e \in (S_l : S_j)} y_e \qquad \forall\, i, j, k, l \in \mathcal{N}\ i \neq k\ , j \neq l$$

Therefore, only an arbitrary partition equality is enough.

For layer equalities, take the following sum over leaf-2 equalities 63

$$\sum_i \Big[\sum_{y_e \in (L_k : x_{ij})} y_e - \sum_{y_e \in (L_l : x_{ij})} y_e \Big] = 0$$

which yields to

$$\sum_{y_e \in (L_k : L_j)} y_e - \sum_{y_e \in (L_l : L_j)} y_e = 0$$

By this way we can show that

$$\sum_{y_e \in (L_k : L_i)} y_e = \sum_{y_e \in (L_l : L_j)} y_e \qquad \forall\, i, j, k, l \in \mathcal{N}\ i \neq k\ , j \neq l$$

Therefore, only an arbitrary layer equality is enough.

We then solved the following linear programming relaxation by CPLEX optimizer:

Problem 3.7

$$\begin{aligned} \min\ & c^T y \\ s.t.\ & By = 0 \\ & \textstyle\sum_{y_{1j2l} \in (S_1 : S_2)} y_{1j2l} = 1 \\ & \textstyle\sum_{y_{i1k2} \in (L_1 : L_2)} y_{i1k2} = 1 \\ & y \geq 0 \end{aligned}$$

problem	size	z^*	z_r	$1 - z_r/z^*$	CPU time
nug05	5	50	50	0	2
nug06	6	86	86	0	8
nug07	7	148	148	0	49
nug08	8	214	203.5	5	163
esc08a	8	2	0	100	148
esc08b	8	8	2	75	127
esc08c	8	32	22	31	121
esc08d	8	6	2	67	131
esc08e	8	2	0	100	139
esc08f	8	18	18	0	172
rou10	10	174220	170400	2	2473
scr10	10	26992	26873.05	0.5	3273

Table 3.2: Results of Final linear programming Relaxation

where B is the constraint matrix of leaf equalities.

In table 3.2 problem rou10 is from Roucairol [64] and problem scr10 is from Scriabin and Vergin [66]. z^* is the optimal/best known objective function value. z_r is the optimal objective function value of the problem 3.7. Experiments are carried out on a SUN SPARCsystem 2 computer (22 MIPS 64 Kb RAM). CPLEX Version 2.1 barrier (interior point) algorithm is used. Solutions to problems nug05, nug06 and nug07 are optimal and integer valued. Also optimality gaps (column $1 - z_r/z^*$) of the remaining problems are reasonable. If CPU times are reduced, current linear programming relaxation can be a good starting point in branch and bound algorithms. CPU times are highly affected by degeneracy and redundant constraints.

Chapter 4

Conclusion

We have tried to solve Quadratic Assignment Problem using the general methodology of Polyhedral Combinatorics. There were four basic steps of Polyhedral Combinatorics where obtaining a complete description is hopeless:

i.) Represent the feasible objects by vectors (usually by incidence vectors).

ii.) Consider these vectors as points in $\mathcal{R}^J$ for suitable J and let P be their convex hull.

iii.) Obtain families of valid inequalities, preferably facets, which gives a partial but presumably strong definition of P.

iv.) Use those families of valid inequalities in a branch-and-cut algorithm.

At first, we stated QAP as a graph theoretic problem. We associate a graph $G = (V, E)$ with QAP and showed that QAP is equivalent to finding minimum weight maximal clique of G. Then we defined QA^n Polytope to be the convex hull of the incidence vectors of maximal cliques of G. This forms the basis of the polyhedral approach. Next step is to find a good description of QA^n. Three equality sets, partition, layer and leaf equalities, are found. The intuition behind these equalities either comes from the structure of the graph or from the new problem definition. Then two sets of special structures, triangles and chordless cycles are investigated and two sets of valid inequalities are found. Although we could not prove triangle and chordless cycle inequalities are facets or not, with the hope of they are strong we tried to implement those findings in a branch and cut algorithm. The initial relaxation contained the partition and layer equalities. An exact separation algorithm is designed for triangle and chordless cycle inequalities. Then we iteratively added violated valid inequalities to the linear programming relaxation. It was the first phase, cutting plane phase, of a branch and cut algorithm. Since there was no improvement in the objective function value, we unfortunately realized that the valid inequalities on hand are so weak. Since the core of branch and cut is how strong the definition of the underlying polytope is, we did not resort to branching with weak valid inequalities on hand.

In our study, third and fourth steps of Polyhedral approach remained weak. Although QA^n is defined, we only have a set of valid equalities for QA^n. This equality set can be used as the starting point of further research. One can study the equality set of QA^n. Later more complex structures which may yield facets can be examined. y's of QA^n have strong interconnections and these are reflected in the current equality set, hence only a few family of facets will ease the application of branch and cut algorithm. Since setting a variable y_{ijkl} to one means setting all the edges coming to the nodes of partitions i and k and layers j and l to zero, except the ones coming to the nodes x_{ij} and x_{kl}, number of variables decrease extremely fast in branching. Low optimality gap and decreasing number of variables may result in good solutions in the early stages of branching.

Finally, the structure of graph $G = (V, E)$ can be used for designing good heuristics. In a similar construction for placement of electronic circuits (Junger et.al. [45]) good heuristics are developed using the underlying structure of graph $G = (V, E)$.

Bibliography

[1] G.C. Armour and E.S. Buffa. A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9:294–, 1963.

[2] A.A. Assad and W. Xu. On lower bounds for a class of quadratic 0-1 programs. *Operations Research Letters*, 4:175–180, 1985.

[3] E. Balas and J. B. Mazzola. Nonlinear 0-1 programming: Linearization techniques. *Mathemathical Programming*, 30:1–, 1984.

[4] F. Barahona and A.R. Mahjoup. On the cut polytope. *Mathemathical Programming*, 36:157–173, 1986.

[5] M.S. Bazaara and A.N. Elshafei. An exact branch-and-bound procedure for quadratic assignment problems. *Naval Research Logistics Quarterly*, 26:109–121, 1979.

[6] M.S. Bazaara and O. Kirca. A branch-and-bound based heuristic for solving the quadratic assignment problem. *Naval Research Logistics Quarterly*, 30:287–304, 1983.

[7] M.S. Bazaara and M.D. Sherali. Benders' partitioning scheme applied to a new formulation of quadratic assignment problem. *Naval Research Logistics Quarterly*, 27:29–41, 1980.

[8] M.S. Bazaara and M.D. Sherali. On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. *Journal of Operations Research Society*, 33:991–1003, 1982.

[9] R.E. Burkard. Locations with spatial interactions: The quadratic assignment problem. In R. B. Mirchandani and R.L. Francis, editors, *Discrete Location Theory*. Wiley, 1990.

[10] R.E. Burkard and T. Bonniger. A heuristic for quadratic boolean programs with applications to quadratic assignment problems. *Eurepean Journal of Operations Research*, 13:374–386, 1983.

[11] R.E. Burkard and U. Fincke. The asymptotic properties of some combinatorial optimization problems. *Discrete Applied Mathemathics*, 12:21–79, 1985.

[12] R.E. Burkard S. Karisch and F. Rendl. QAPLIB- A quadratic assignment problem library. *Eurepean Journal of Operations Research*, 55:115–119, 1991.

[13] R.E. Burkard and J. Offermann. Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungs probleme. *Zeitschrift fur operations research*, 27:73–81, 1977.

[14] R.E. Burkard and F. Rendl. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *Eurepean Journal of Operations Research*, 17:169–174, 1984.

[15] R.E. Burkard and K. H. Stratmann. Numerical investigations on quadratic assignment problem. *Naval Research Logistics Quarterly*, 25:129–148, 1978.

[16] P. Carraresi and F. Malucelli. A new lower bound for the quadratic assignment problem. *Operations Research*, 40:S22–S27, 1992.

[17] S. Chopra and M.R. Rao. The partition problem. *Mathemathical Programming*, 59:87–116, 1993.

[18] H. Crowder and M. Padberg. Solving large-scale symmetric traveling salesman problem: to optimality. *Management Scince*, 26:495–509, 1980.

[19] G.B. Dantzig D.R. Fulkerson and S.M. Johnson. Solution of a large scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.

[20] C.S. Edwards. A branch-and-bound algorithm for the Koopmans-Beckmann quadratic assignment problem. *Mathemathical Programming Study*, 13:35–52, 1980.

[21] H.K. Edwards B.E. Gillett and M.E. Hale. The modular allocation technique (MAT). *Management Science*, 17:161–169, 1970.

[22] B. Eschermann and H.J. Wunderlich. Optimized synthesis of self-testable finite state machines. 1990.

[23] G. Finke R.E. Burkard and F. Rendl. Quadratic assignment problems. *Annals of Discrete Mathemathics*, 31:61–82, 1984.

[24] J.C.B. Frenk M. van Houweninge and A.H.G. Rinnoy Kan. Asymptotic properties of assignment problems. *Mathemathics of Operations Research*, 10:100–116, 1985.

[25] A.M. Frieze and J. Yadegar. On the quadratic assignment problems. *Discrete Applied Mathemathics*, 5:89–98, 1983.

[26] G.K. Gaschutz and J.H. Ahrens. Suboptimal algorithms for quadratic assignment problem. *Naval Research Logistics Quarterly*, 15:49–, 1968.

[27] J.W. Gavett and N.V. Playter. The optimal assignment of facilities to locations by branch-and-bound. *Operations Research*, 14:210–232, 1966.

[28] P.C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal on Applied Mathemathics*, 10:305–313, 1962.

[29] F. Glover. Improved linear integer programming formulations of nonlinear integer programs. *Management Science*, 22:455–460, 1975.

[30] G.W. Graves and A.B. Whinston. An algorithm for the quadratic assignment problem. *Management Science*, 17:453–471, 1970.

[31] M. Grötschel. On the symmetric traveling salesman problem: solution of a 120-city problem. *Mathemathical Programming Studies*, 12:61–77, 1980.

[32] M. Grötschel and M. Padberg. Polyhedral theory. In A.H.G. Rinooy Kan E.L. Lawler, J.K. Lenstra and D.B. Shmoys, editors, *The traveling salesman problem*. John Wiley and Sons, 1985.

[33] S.W. Hadley F. Rendl and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Mathemathics of Operations Research*, 17:727–739, 1992.

[34] S.W. Hadley F. Rendl and H. Wolkowicz. Symmetrization of nonsymmetric quadratic assignment problems and the hoffman-weilandt inequality. *Linear Algebra and Its Applications*, 167:53–64, 1992.

[35] C.H. Heider. A computationally simlified pair exchange algorithm for the quadratic assignment problem. Pap.no.101, Center for Naval Analyses, 1972.

[36] F.S. Hillier. Quantative tools for plant layout analysis. *Journal of Industrial Engineering*, 14:33–, 1963.

[37] F.S. Hillier and M.M. Connors. Quadratic assignment problems and the location of indivisable facilities. *Management Science*, 13:42–, 1966.

[38] B.K. Kaku and G.L. Thompson. An exact algorithm for the general quadratic assignment problem. *Eurepean Journal of Operations Research*, 23:382–390, 1986.

[39] L. Kaufman and F. Broeckx. An algorithm for the quadratic assignment problem using Benders' decomposition. *Eurepean Journal of Operations Research*, 2:204–211, 1978.

[40] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[41] A.M. Land. A problem of assignment with interrelated costs. *Operations Research Quarterly*, 14:185–198, 1963.

[42] R.S. Lashkari and S.C. Jaisingh. A heuristic approach to quadratic assignment problem. *Journal of Operations Research Society*, 31:845–850, 1980.

[43] E. L. Lawler. The quadratic assingment problem. *Management Science*, 9:586–599, 1963.

[44] L. Lovasz and A. Schrijver. Some combinatorial appplications of the new linear programming algorithm. In S.B. Rao, editor, *Combinatorics and Graph theory*. Springer-verlag, 1981.

[45] M. Junger, G. Reinelt, A. Martin and R. Weismantel. Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits. *Mathemathical Programming*, 63:257–280, 1994.

[46] P.B. Mirchandani and T. Obata. Locational decisions with interactions between facilities: The quadratic assignment problem - a review. Ps-79-1, Rensselear Polytechnic Institute, 1979.

[47] C.E. Nugent T.E. Vollmann and J. Ruml. An experimental comparision of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[48] M. Oral and O. Kettani. A linearization procedure for quadratic and qubic mixed integer programs. *Operations Research*, 40:S109–S116, 1992.

[49] M. Oral and O. Kettani. Reformulating quadratic assignment problems for efficient optimization. *IIE Transactions*, 25:97–107, 1993.

[50] M. Padberg. Covering, packing and knapsack problems. *Annals of Discrete Mathemathics*, 4:265–287, 1979.

[51] M. Padberg. The boolean quadric poytope: some characteristics, facets and relatives. *Mathemathical Programming*, 45:139–172, 1989.

[52] M. Padberg and M. Grötschel. Polyhedral computations. In A.H.G. Rinooy Kan E.L. Lawler, J.K. Lenstra and D.B. Shmoys, editors, *The traveling salesman problem.* John Wiley and Sons, 1985.

[53] M. Padberg and S. Hong. On the symmetric traveling salesman problem: a computational study. *Mathemathical Programming Studies*, 12:78–107, 1980.

[54] M. Padberg and M.R. Rao. Odd minimum cut-sets and b-matchings. *Mathemathics of Operations Research*, 7:67–80, 1982.

[55] M. Padberg and G. Rinaldi. Optimization of a 512-city symmetric traveling salesman problem by branch-and-cut. *Operations Research Letters*, 6:1–7, 1987.

[56] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33:60–100, 1991.

[57] C.S. Parker. An experimental investigation of some heuristic strategies for component replacement. *Operations Research Quarterly*, 27:71–81, 1976.

[58] J.F. Pierce and W.B. Crowston. Tree search algorithms for quadratic assignment problems. *Naval Research Logistics Quarterly*, 18:1–36, 1971.

[59] M.A. Pollatschek H. Gershoni and Y.T. Radday. Optimization of the typewriter keyboard by computer simulation. *Angewandte Informatik*, 10:438–439, 1976.

[60] W.R. Pulleyblank. Polyhedral combinatorics. In M. Grötschel A. Bachem and B. Korte, editors, *Mathemathical Programming: The state of the art.* Springer, 1983.

[61] C.R. Reeves. An improved heuristic for quadratic assignment problem. *Journal of Operations Research Society*, 36:163–167, 1985.

[62] F. Rendl and H. Wolkowicz. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming*, 53:63–67, 1992.

[63] C. Roucairol. A reduction method for quadratic assignment problems. *Operations Research Verfahren*, 32:183–187, 1979.

[64] C. Roucairol. A parallel branch-and-bound algorithm for quadratic assignment problem. *Discrete Applied Mathemathics*, 18:211–225, 1987.

[65] S. Sahni and T. Gonzales. P-complete approximation problems. *Journal of the ACM*, 23:555–565, 1976.

[66] M. Scriabin and R.C. Vergin. Comparision of computer algorithms and visual based methods for plant layout. *Management Science*, 22:172–187, 1975.

[67] J. Skorin-Kapov. Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, 2:33–45, 1990.

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

[69] H. Weyl. Elementare theorieder konvexen polyeder. *Comment. Math. Helv.*, 7:290–306, 1935.

[70] M.R. Wilhelm and T.L. Ward. Solving quadratic assignment problem by simulated annealing. *IIE Transactions*, 19:107–119, 1987.