

**Facility Location and Network Design
With Congestion Costs and
Interdependency**

Gilberto de Miranda Junior

05/25/2004

To my parents, my wife and my daughter.

Abstract

In this work we develop mathematical programming formulations for location models, congested network design models and the integration of both. Location and network design problems arise in several applications of Computer Science, Engineering and Economy. Nowadays, these problems can not be solved efficiently, what is our major motivation. Established the relevance of these problems, we try to expand their solution frontiers, rewriting them with the aid of flow formulations and using a Benders decomposition framework. Our main goal is to deal with large scale mixed integer programming problems as the *Quadratic Assignment Problem*, the *Uncapacitated Hub Location Problem* and large scale mixed integer nonlinear programming problems. Extensive computational experiments were carried out. The output data is analyzed and discussed, becoming possible to evaluate the quality of the proposed approach

Contents

Abstract	v
1 Problem Context and Motivation	1
1.1 The Location of Economic Activities	1
1.2 The Local Access Network Design	2
1.3 Facility Location and Network Design	3
2 Assignment Problems	5
2.1 Theory and Background	5
2.2 The Linear Assignment Problem	7
2.3 The Quadratic Assignment Problem	8
2.3.1 Alternative Problem Formulations, Linearizations and Bounds	9
2.3.2 Flow Formulations for <i>QAP</i>	11
2.3.3 A Brief Computational Experiment	12
2.4 Benders Decomposition of the Problem	16
2.4.1 Subproblems	23
2.4.2 Enhancing the Benders Decomposition Algorithm with Flow Equilibrium Constraints	24
2.5 Computational Experiments Using Enhanced Benders Decompo- sition	26
2.6 Concluding Remarks	28
3 The Placement of Electronics with Thermal Effects	33
3.1 Introduction	33
3.2 Thermal Modeling and Temperature Penalty Costs	34
3.2.1 Maximum Temperature and Penalty Costs	35
3.3 Computational Experiments	38
3.3.1 Experiment Description	38
3.3.2 Numerical Results	41
3.4 Concluding Remarks	42
4 The Local Access Network Design With Congestion Costs	47
4.1 Introduction	47
4.2 Multi-commodity Flow Formulation	49

4.2.1	Variables and Parameters	50
4.2.2	Mixed Integer Nonlinear Program	50
4.2.3	Theoretical Properties of the Linear and the Concave Versions	51
4.2.4	Convexification of Leasing and Congestion Costs	52
4.3	Benders Decomposition of the Problem	53
4.3.1	Problem Manipulations	53
4.3.2	Subproblems	56
4.3.3	Master Problem	59
4.3.4	Algorithm	60
4.3.5	Avoiding Cycles	61
4.4	Computational Results	62
4.5	Conclusions	66
5	Integrating Facility Location and Network Design	69
5.1	Problem Description	69
5.2	Mathematical Programming Formulations	71
5.2.1	Improving the Design of the Local Access Network	74
5.2.2	Generalized Model Including Hub Transshipment and Network Design	77
5.3	Computational Experiences	78
5.3.1	Benders Decomposition for the p-Hub Median Problem and the Uncapacitated Hub Location Problem	78
5.3.2	The Integrated Model: <i>QAP</i> + Local Access Network Design With Congestion Costs	81
5.3.3	Testing the Hub Transshipment Network Design Model	81
5.4	Concluding Remarks	82
6	Conclusions and Future Work	87
6.1	Contributions	87
6.1.1	The Quadratic Assignment Problem	87
6.1.2	The Placement of Electronics With Thermal Effects	88
6.1.3	The Local Access Network Design With Congestion Costs	88
6.1.4	Integrating Facility Location and Network Design	88
6.2	Hints for Future Work	89

List of Figures

2.1	Different representations of assignments.	5
2.2	Perfect matching in a bipartite graph and corresponding network flow model.	6
2.3	Comparison of linear programming bounds for both formulations.	17
2.4	Comparison of lp computing times for both formulations.	18
2.5	Comparison of mip computing times for both formulations.	19
2.6	Comparison of mip computing times for both formulations.	20
2.7	Comparison of mip computing times for both formulations.	21
2.8	Comparison of mip computing times for both formulations.	22
2.9	An example of automatic construction of a feasible solution for the dual subproblem.	23
2.10	Evolution of computing times with p/q ratio.	29
2.11	Evolution of computing times with p/q ratio.	30
2.12	Evolution of computing times with p/q ratio.	31
3.1	<i>QAP</i> instance and Finite Volume Grid representation.	36
3.2	The penalty overheating cost function, for a threshold temperature of 85 Celsius.	37
3.3	Evolution of bounds during the method execution.	39
3.4	Number of Benders iterations versus p/q cost reason.	44
3.5	Execution time [s] versus p/q cost reason.	44
3.6	Temperature field for <i>ste36a</i> placement solution without overheating penalty.	45
3.7	Temperature field for <i>ste36a</i> placement solution considering overheating penalty.	45
4.1	The tree network design problem.	48
4.2	An example of convexified integrated leasing and congestion cost function.	54
5.1	Hub-and-spoke system with different kinds of local access networks.	70
5.2	Possible routes for the commodity ij for a given $x = x^h$	73
5.3	A city partitioned in regions	84
5.4	A feasible solution	85

List of Tables

2.1	Linear programming bounds for both formulations under comparison and respective computing times.	13
2.2	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of integer mixed programming computing times.	14
2.3	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of mixed integer programming computing times.	15
2.4	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of mixed integer programming computing times.	16
2.5	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition.	26
2.6	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition.	27
2.7	Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition and flow formulation.	28
2.8	Evolution of computing times for Benders algorithm and the flow formulation	30
2.9	Evolution of computing times for Benders algorithm and the flow formulation	31
2.10	Solution of larger instances using the Benders algorithm.	32
3.1	Thermo-physical properties for the thermal model.	36
3.2	Test instances for computational experiments.	40
3.3	Results for computational experiments - first set.	41
3.4	Results for computational experiments - second set.	42
3.5	Results for computational experiments - third set.	43
4.1	Network Dimensions for Test Problems.	63
4.2	Average computing time, number of Benders iterations and number of cycle avoiding constraints for experiments 1 to 5.	65

4.3	Average computing time, number of Benders iterations, Nonlinear/linear gap and number of different arcs for experiments 1 to 5.	66
4.4	Computing time, number of Benders iterations, number of cycle avoiding constraints for experiment 6.	67
4.5	Computing time, number of Benders iterations, and Nonlinear/linear gap for experiment 6.	68
5.1	Benders decomposition for the p-Hub Median Problem.	79
5.2	Benders decomposition for the Uncapacitated Hub Location Problem.	80
5.3	Computational results for the integrated model.	81
5.4	Report for a brief experiment using the Hub Transshipment Network Design model.	82

Chapter 1

Problem Context and Motivation

1.1 The Location of Economic Activities

There are important areas of economic analysis in which progress depends of methods for solving or analyzing problems for efficient allocation of indivisible resources. There are practical decision problems that we can cite. For instance, to determine suitable numbers of machine tools of various kinds within a plant, or to define the number of channel capacities and frequencies in telecommunication networks.

Furthermore, indivisibilities in the more highly specialized human or material factors of production are always at the root of increasing returns to the scale of production. They can arise within the plant or firm, or in relation with a cluster of firms. Strong interest in the effects of indivisibilities comes from the fact that: if industry increasing returns to scale persist at a production level that sizes the total demand in the respective market, we do not have perfect competition and the efficiency of a price system in allocating resources is reduced. Summarizing, the location theory of economic activities is dependent on the indivisibilities of human and material resources to better explain the reality. This was preconceived by Koopmans and Beckmann [77]. These indivisibilities, by the way, are responsible for some of the greatest mathematical challenges that *Mathematical Programming* and *Computer Science* have been facing in the latest 45 years.

It is possible to describe a huge list of practical applications of location problems. Among them, the best location of producers in a multi-commodity transportation network, the best location of warehouses and plants, given the location of his customers and suppliers, the best location of points of inflow or outflow in transportation networks. On the highly competitive economic system created by globalization, it is unnecessary to point out that every single cost component is important: in one side the minimization of production (fixed

or operational) costs improves any organization survivability, at the other side improves the quality of the provided services, maximizing social benefits.

All the systems that have "hub-and-spoke" features are candidates to locational studies. In a major scale, we can think about the location of private and public facilities to improve the quality of service, and thus quality of life, for a given population. In this context, the facility that we are talking about can be generally called a *server*, an indivisible resource that is responsible for locally provide access to some kind of commodity for final customers and which is connected with other facilities by a transportation network. The commodity being transported from and to these servers can be water, fuel and other petroleum sub-products, electrical energy in power transmission networks, data and other signals in telecommunication and computer networks. The servers can be merely concentrators or complex base stations, depending on the commodity and the transportation technology associated.

Several researchers are dealing with location problems. The work and research conducted around *Assignment Problems*, which are some of the simplest approaches to give answers to location theorists, is intensively increasing. The work of Koopmans and Beckmann [77] is a landmark for location theory. For some traditional surveys on this subject, we suggest the work of Motzkin [99], Losch [82], Kuhn [79], Mills [98], Samuelson [121] and Heffley [61]. On the last years, deserve attention the work of Beasley [14], Christofides and Beasley [36], Franca and Luna [45], Mateus and Luna [92], Aikens [2] and Mateus and Thizy [94]. Assignment problems are being studied more recently by Burkard [23], Balas and Saltzman [11], Burkard and Cela [27], Burkard, Cela, Pardalos and Pitsoulis [28], Cela [26], Anstreicher [4], Anstreicher and Brixius [5], Anstreicher, Brixius, Goux and Linderoth [6].

1.2 The Local Access Network Design

Once defined the location of the servers, the question on how to design the local access network, that the final customer will use to access a server, naturally arises. The quality of the local access network, in some cases, is representative of a great amount of the total cost. Network design problems have found wide application in computer networks and telecommunication systems, exploring issues of topological design, routing and capacity assignment [21, 47, 93]. The hierarchical organization of telephone and computer networks plays a major role, inasmuch as optimized levels of customers concentration enables substantial economies of scale of increasing transmission bandwidth. Cost minimization is the objective of most of these operations research models, the main difference among the models being the hierarchical level of network design, typically concerning *backbones* or *local access* networks. In this work we have considered the *local access network design* problem with congestion costs. The local access network design problem consists of linking a supply node to its demand nodes satisfying their demand at minimal total cost. The problem presents heterogeneous terminals, and there are also *Steiner* or transshipment nodes. Each arc

of the network has three associated costs: a linear variable operational cost depending on the flow through the arc, a fixed cost associated with the installation of the arc and a non-linear congestion cost that penalize flows close to implicit capacities. This problem can be viewed as a generalization of the Steiner tree problem on a directed graph [85]. In fact, if we neglect variable and congestion costs at the arcs we will have basically the Steiner problem, and in this sense we are treating a \mathcal{NP} -Hard problem, for which some computational strategies have been devised [87, 128, 75, 83, 67, 68]. On the other hand, if we neglect fixed and congestion costs on the arcs we have the single source transshipment problem, which can be solved easily [40].

As one can see, it is possible to use this approach to deal with any kind of network flow problem that has a single source tree as optimal solution. The network design problem associated with centralized computer networks and the multiparty multicast tree construction problem are good examples. The last one has been treated with the aid of heuristics [69], but on the two versions pointed by the literature, the Single Source Tree Networks - where we have a true root of the multi-party multicasting tree - and the Core Based Tree Networks - where a single node, the core, is chosen to play a role as the tree root - it is possible to adjust the data to make the model treatment to accomplish the nature of the problem. The provision of multi-point connections is one of most important services that will be required in future broadband communication networks that support distributed multimedia applications. Multimedia video-conference applications, for instance, require that audio and video be transmitted to multiple conference participants simultaneously. This requires that an efficient multicast capability be provided by the underlying network. Beyond problems in telecommunications and centralized computer networks, this approach is useful to deal also with petrochemical products distribution networks, water distribution networks, and many other local access networks (distributing energy, material resources or signals and data) under mild assumptions.

1.3 Facility Location and Network Design

Network location models have been used extensively to analyze and determine the location of facilities. Classical network models include the location set covering problem [124], the maximum covering location problem [37] and p-median and p-center problems [60].

In addition to these, the uncapacitated facility location problem [78] has been treated in its own right and is also known as *simple plant location problem* and *warehouse location problem*. All these models locate facilities on a given network. However, the topology of the underlying network may have profound impact on facility location. Models that integrates the tasks of facility location and network design has being presented by Melkote and Daskin [97], [96], Berger et. al. [18], Berman, Ingco and Odoni [19] and Campbell [34]. In this kind of problems, a set of nodes is given that represents the demand nodes, as well as candidate facility locations, and a set of uncapacitated links. Each link has a fixed construction

cost as well as a per unit transportation cost, and each node is associated with a fixed charge for building an uncapacitated facility at that node. The objective is to find the network design and the set of facility locations that minimize the total system cost (fixed + operational). This model is reported to be used in the design of pipeline distribution systems, inter-modal transportation systems, power transmission networks and all the hub location problems (that arises in various transportation contexts, and simultaneously address where to locate the hubs and how to design the hub-level network and the access level network).

These models, however, deals with problems where the background assignment problem is always a linear one. This means firstly, that no distinction is made between the links used to interconnect facilities and links used to establish the local access network. Here, the location does not present *interdependency*: the definition of a site for one server does not influence the possible assignments for the others. As one can see, none of these two features is much realistic. In many problems involving location of facilities (or servers) and the design of the two underlying networks: the transportation network (established between facilities) and the local access network there are hierarchical considerations to be made. Usually, the technology used to implement the transportation network (larger link capacities and higher traffic velocity) is even different from that used for local access network problems (smaller link capacities and lower traffic velocity).

In this work we present a formulation that solves the integrated network design/facility location problem adding two features: hierarchical solution, which constitutes a coherent way to detach the two problems, ensuring mathematical consistency, enabling the use of parallel/distributed computing as a way to solve larger instances, and the accomplishment of interdependency in the assignment of servers to locations. To treat the local access network cost component we incorporate a non-linear effect: the congestion cost and capacity expansion cost trade-off, as suggested by Luna and Mahey [84]. The application framework selected here is the Tree Network Design for Centralized Computer Networks, Local Access Telecommunication Networks and Multicast Multiparty Tree Networks. For the location problem, each server can be viewed as a computational resource center, a switching center or the different multicast multiparty servers that are interconnecting the participants.

Chapter 2

Assignment Problems

2.1 Theory and Background

Assignment problems deal with the question of how to assign n items (jobs, students) to n other items (machines, tasks). Their underlying structure is an *assignment* which is nothing else than a bijective mapping ϕ between two finite sets of n elements. In the optimization problem where we are looking for a *best* possible assignment, we have to optimize some objective function which depends on the assignment ϕ . Assignments can be represented in different ways. The bijective mapping between two finite sets V and W can be represented in a straightforward way as a *perfect matching* in a bipartite graph $G = (V, W; E)$, where the vertex sets V and W have, each one, n vertices. Edge $(k, i) \in E$ is an edge of the perfect matching if, and only if, $i = \phi(k)$.

After characterizing the sets V and W we get a representation of an assignment as a *permutation*. Every permutation ϕ of the set $N = 1, \dots, n$ corresponds in an unique way to a *permutation matrix* $X_\phi = (x_{ki})$ with $x_{ki} = 1$ for $i = \phi(k)$ and $x_{ki} = 0$ for $i \neq \phi(k)$. This matrix X_ϕ can be viewed as adjacency matrix of the bipartite graph G representing the perfect matching, see Figure (2.1).

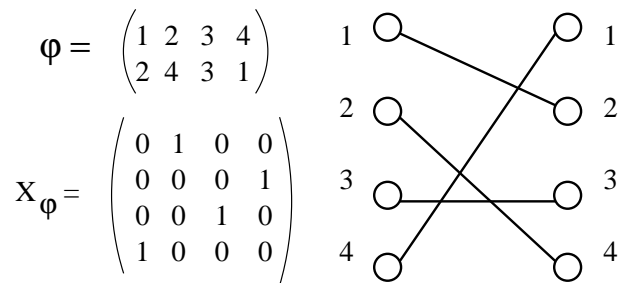


Figure 2.1: Different representations of assignments.

The set of all assignments (permutations) of n items will be denoted by S_n

and has $n!$ elements. This set can be described by the following constraints called *assignment constraints*.

$$\sum_{k=1}^n x_{ki} = 1, \quad \forall i = 1, \dots, n \quad (2.1)$$

$$\sum_{i=1}^n x_{ki} = 1, \quad \forall k = 1, \dots, n \quad (2.2)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k, i = 1, \dots, n. \quad (2.3)$$

The set of all matrices $X = (x_{ki})$ fulfilling the assignment constraints will be denoted by X_n . When we replace the conditions $x_{ki} \in \{0, 1\}$ in (2.3) by $x_{ki} \geq 0$, we get a *doubly stochastic matrix* [24]. The set of all doubly stochastic matrices forms the *assignment polytope* P_A . Birkhoff [20] showed that the assignments correspond uniquely to the vertices of P_A . Thus every doubly stochastic matrix can be written as convex combination of permutation matrices.

Theorem 2.1.1 (Birkhoff [20]) *The vertices of the assignment polytope corresponds uniquely to permutation matrices.*

Network flows offer another choice of modeling assignments. Let $G = (V, W; E)$ be a bipartite graph with $|V| = |W| = n$. We embed G in the network $\mathcal{N} = (N, A, c)$ with *node set* N , *arc set* A and *arc capacities* c . The node set N consists of a source s , a sink t and the vertices of $V \cup W$, see Figure (2.2).

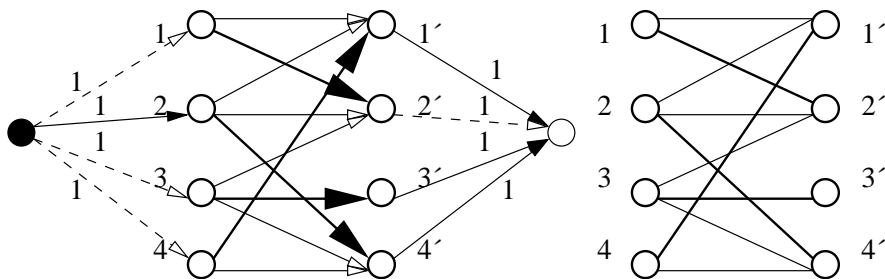


Figure 2.2: Perfect matching in a bipartite graph and corresponding network flow model.

The source is connected to every node in V by a directed arc of capacity 1, every node in W is connected to the sink by a directed arc of capacity 1, and every arc in E is directed from V to W and supplied with infinite capacity. The *maximum network flow problem* asks for a flow with maximum value $z(f)$. Obviously, a maximum integral flow in the special network constructed above corresponds to a matching with maximum cardinality. A *cut* in the network \mathcal{N} is a subset C of the node set N with $s \in C$ and $t \notin C$. The value of $u(C)$ is defined as

$$u(C) = \sum_{x \in C, y \notin C, (x,y) \in A} c(x,y) \quad (2.4)$$

where $c(x, y)$ is the capacity of the arc (x, y) .

Ford and Fulkerson's famous *Max Flow - Min Cut Theorem* [44] states that the value of a maximum flow equals minimum cut value. Such a theorem can be directly translated into the Konig Matching Theorem [76]. Given a bipartite graph G , a vertex cover (cut) in G is a subset of its vertices such that every edge is incident with at least one vertex in the set.

Theorem 2.1.2 (*Konig Matching Theorem [76]*) *In a bipartite graph, the minimum number of vertices in a vertex cover equals the maximum cardinality of a matching.*

Let us now formulate this theorem in terms of 0-1 matrices. Given a bipartite graph $G = (V, W; E)$ with $|V| = |W| = n$, we define 0-adjacency matrix Σ of G as a $(n \times n)$ matrix $\Sigma = (\varsigma_{ij})$ by

$$\varsigma_{ij} = \begin{cases} 0 & \text{if } (i, j) \in E \\ 1 & \text{if } (i, j) \notin E \end{cases} \quad (2.5)$$

A zero cover is a subset of the rows and columns of matrix Σ which contains all 0 elements. A row (column) which is an element of a zero-cover is called a covered row (covered column). Now we get

Theorem 2.1.3 *There exists an assignment ϕ with $\varsigma_{i\phi(i)} = 0$ for all $i = 1, \dots, n$, if and only if the minimum zero cover has n elements.*

Since a maximum matching corresponds uniquely to a maximum flow in the corresponding network \mathcal{N} , we can construct a zero-cover in the 0-adjacency matrix Σ by means of a minimum cut C in this network: if node $i \in V$ of the network does not belong to the cut C , then the row i is an element of the zero-cover. Analogously, if node $j \in W$ of the network belongs to the cut C , then column j is an element of the zero-cover.

2.2 The Linear Assignment Problem

A relatively simple problem in the allocation of indivisible resources, which is a direct application of the matching of two sets, is the task of creating a one to one association of the elements in each set. In the context we present, the allocations of a set of plants to a set of candidate sites. There are a variety of practical decision problems for which this is an adequate characterization. Due to our underlying interest in location theory, we will discuss the problem here in terms of assigning facilities to locations. Each facility, still on the drawing board, is supposed to be capable of achieving a given expected profit in each location, different locations having different suitabilities for a given economic

activity. The problem is then to find an assignment that makes the profit attainable from the location of facilities as large as possible.

It is clear that this problem is fully defined by its mathematical formulation, given below, independently of the locational interpretation. It is useful to remember that this approach gives an artificial picture of locational problems when compared with the complexities and degrees of freedom that can be found in reality. Any kind of rule to subdivide land or allowing the building of more than one plant is not explored, for instance. Other variables like production technology and resource availability are also ignored.

The profitability for the n^2 possible plant location pairs are represented as a square matrix; the element a_{ki} representing the fixed profit expected from the assignment of facility k to location i . If we choose to use a matrix $X = (x_{ki})$ to represent the assignment, a linear integer program for the problem can be obtained with the aid of equations (2.1) - (2.3), resulting

$$\max p = \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} \quad (2.6)$$

subject to (2.1) - (2.3)

This linear integer program has n^2 integer variables, and would be a very difficult one, if the linear relaxation of the integrality constraints would not create a set of doubly stochastic matrices. These matrices satisfy the assignment constraints, and from Theorem 2.1.1, these assignments correspond uniquely to the vertices of the associated polytope. This means that, the associated linear program has the same solution as the former integer program, which is a very comfortable property. In fact, Burkard [24] shows that linear assignment problems can be solved by only adding, subtracting and comparing the cost coefficients (see the *Hungarian Method*[79]).

However, if one needs a major level of detail, and requires a better description of reality, it is necessary to improve the model by adding more realistic effects and relationships.

2.3 The Quadratic Assignment Problem

The assumption that the profit obtainable from an economic activity at some location does not depend on the uses of other locations is quite inadequate and unrealistic in most practical situations. There are direct, physical, interactions between different production processes. The mere fact that scarce resources need to be utilized for the transportation of intermediate commodities between facilities appears to be sufficient to indicate the unsuitability of the linear model to describe the reality. In order to improve our capabilities of modeling the world of locational decisions, we introduce now the *Quadratic Assignment Problem*. Considering two sets of n facilities and n locations, and the installation profitability matrix $A = (a_{ki})$, as defined for the linear assignment problem, we

express the intertransportation cost, given a matrix $B = (b_{kl})$ of demands of intermediate commodities between facilities k and l , and a matrix $C = (c_{ij})$ of costs of transportation per unit of flow between the locations i and j , as

$$q = \sum_{(k,l)} \sum_{(i,j)} b_{kl} x_{ki} c_{ij} x_{lj} \quad (2.7)$$

We must observe that the transportation cost is independent of the facility assignment and the total demand of intermediate commodities is independent of location assignment. So, the quadratic assignment problem can be stated (in the profitability version) as $p - q$:

$$\max \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} - \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ij} x_{ki} b_{kl} x_{lj} \quad (2.8)$$

subject to (2.1) - (2.3)

The *Quadratic Assignment Problem (QAP)* remains among the most complex combinatorial optimization problems. The inherent difficulty for solving *QAP* is also reflected by its computational complexity. Sahni and Gonzalez [120] showed that *QAP* is $\mathcal{NP} - \text{Hard}$ and that even finding an approximate solution within some constant factor from the optimal value cannot be done in polynomial time. Recently it has been shown that even local search is hard in some instances, as can be seen in [29] and [107].

2.3.1 Alternative Problem Formulations, Linearizations and Bounds

There are different, but equivalent, mathematical formulations for *QAP* which stress different structural characteristics of the problem and lead to different solution approaches. In the form of an integer quadratic program, as stated above, it is very difficult to devise solution strategies. So it is useful to develop techniques to rewrite *QAP* as an integer linear program.

In the last forty-five years, many researchers working on *QAP* have proposed methods for linearizing the quadratic term in the objective function by introducing additional variables. The work of Lawler [80] is a fundamental linearization, deriving the well known *Gilmore-Lawler Bound (GLB)* and an entire family of correlated linearizations. The research of Kaufman and Broeckx [73], Frieze and Yadegar [46] and more recently Adams and Johnson [1], Hahn and Grant [59] and Ramakrishnan et al. [115] are extremely important on this matter. For a more complete survey on *QAP* affairs, see [24], [29], [11], [25], [27], [61], [13], [28] and [6]. The linearization of Adams and Johnson is reputed to dominate all the others [1] (excepting Hahn and Grant [59]), and is a mixed integer program with n^2 binary variables, $n^4 - 2n^3 + n^2$ continuous variables, and $n^4 - n^2 + 2n$ constraints, as presented here:

$$\max \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} - \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \sum_{j=1}^n d_{kilj} y_{kilj} \quad (2.9)$$

subject to (2.1) - (2.3) and:

$$\sum_{j=1}^n y_{kilj} = x_{ki}, \quad \forall i, k, l = 1, \dots, n, \quad i \neq j, \quad k \neq l \quad (2.10)$$

$$\sum_{l=1}^n y_{kilj} = x_{ki}, \quad \forall i, j, k = 1, \dots, n, \quad i \neq j, \quad k \neq l \quad (2.11)$$

$$y_{kilj} = y_{ljki}, \quad \forall i, j, k, l = 1, \dots, n, \quad i \neq j, \quad k \neq l, \quad (2.12)$$

$$y_{kilj} \geq 0, \quad \forall i, j, k, l = 1, \dots, n, \quad i \neq j, \quad k \neq l \quad (2.13)$$

Closely related to some linearizations are the polyhedral studies performed by Barvinok [12], Junger and Kaibel [70],[71] and Padberg and Rijal [106], designed to derive the *QAP polytope* for use with *Branch-and-Cut* methods. This family of linearizations usually produces strong linear programming relaxations, being on the other hand very difficult to solve. If the good linear programming lower bounds are desirable to obtain success in a *Branch-and-Cut* framework, the excessive computational cost to solve the programs is really a problem. Even the work of Junger and Kaibel [70],[71], over the linearization of Adams and Johnson[1] demands considerable computational efforts to attain substantial results.

Otherwise, many authors have chosen to work with *QAP* in its original quadratic form. Writing *QAP trace formulation*, we have:

$$\max \operatorname{tr}(A - CXB^T)X^T \quad (2.14)$$

subject to:

$$X \in \mathcal{X}_n. \quad (2.15)$$

The trace formulation was used by Finke, Burkard and Rendl [43] to introduce the eigenvalue bounds, a stronger class of lower bounds when compared to bounds obtained via mixed integer linear programming. The eigenvalue lower bounds are, however, very expensive in terms of computational time and deteriorate quickly when lower levels of the *Branch-and-Bound* tree are searched. Requirements for a good bound are not to be so hard to compute, be easily evaluated for subsets of the problem which occur after some branching, and, of course, to be tight. Recently, the work of Anstreicher et al. [5] about bounds for *QAP* is based on *Semi-Definite Programming* relaxations, and these new bounds are reported to be superior to all the other bounds available from *QAP* literature. Solutions for large (and hard) *QAPLIB* instances as *ste36a*, *ste36b* and *nug30* to optimality were obtained by Anstreicher et al. using a *Computational Grid* [6].

2.3.2 Flow Formulations for QAP

Starting with the aid of a flow formulation involving the intermediate flows between facilities and denoting by f_{ij}^{kl} the flow from location i to location j sent from facility k to facility l , one can write:

$$\max \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} - \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ij} f_{ij}^{kl} \quad (2.16)$$

subject to (2.1) - (2.3) and:

$$b_{kl} x_{ki} + \sum_{j=1}^n f_{ji}^{kl} = b_{kl} x_{li} + \sum_{j=1}^n f_{ij}^{kl}, \quad \forall i, k, l = 1, \dots, n \quad (2.17)$$

$$f_{ii}^{kl} = 0, \quad \forall i, k, l = 1, \dots, n \quad (2.18)$$

$$f_{ij}^{kl} \geq 0, \quad \forall i, j, k, l = 1, \dots, n \quad (2.19)$$

This was the first technique proposed to linearize QAP, due to Koopmans and Beckmann [77], and it works just like the other ones: introducing additional variables and constraints. However, this linearization is very weak, since constraints (2.17) can be satisfied with zero flows between facilities. The linear programming relaxation yields then a trivial solution, consequently producing low quality linear programming lower bounds.

Instead of working with the linearized Koopmans and Beckmann formulation, (2.16) - (2.19), we suggest to rewrite constraints (2.17) into two equivalent sets, representing the flow balance at the source and sink points for each commodity kl :

$$\max \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} - \sum_{(i,j), i \neq j} \sum_{(k,l), k \neq l} c_{ij} f_{ij}^{kl} \quad (2.20)$$

subject to (2.1) - (2.3) and:

$$-\sum_{j=1}^n f_{ij}^{kl} = -b_{kl} x_{ki}, \quad \forall i, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.21)$$

$$\sum_{i=1}^n f_{ij}^{kl} = b_{kl} x_{lj}, \quad \forall j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.22)$$

$$f_{ij}^{kl} \geq 0, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.23)$$

The above flow formulation imply nonzero flows and is able to produce better linear programming bounds when compared to the original Koopmans and Beckmann formulation. It is also well suited to the decomposition method presented in the next section, having n^2 binary variables, $n^4 - 2n^3 + n^2$ continuous

variables, and $n^4 - n^2 + 2n$ constraints. In fact, this formulation is not so strong as the formulation of Adams and Johnson [1], and it is possible to observe some weakness of the linear programming bounds as the problem size increases. On the other hand, this flow formulation is very easy to solve. The idea here is to obtain a well balanced mixed integer programming formulation, which sustain a reasonable linear programming lower bound, being not so difficult to solve as the other ones.

Pursuing this objective, one can add a new family of constraints to the above flow formulation that can make it easier. If the facility k is located at the site i and the facility l is placed at the site j , constraints (2.21) and (2.22) ensure that the flows from i to j and from j to i equal the demands for the intermediate commodities b_{kl} and b_{lk} . So, if $x_{ki} = 1$ and $x_{lj} = 1$ then $-f_{ij}^{kl} = -b_{kl}$ and $-f_{ji}^{lk} = -b_{lk}$, or simply, for $b_{kl} \neq 0$:

$$\begin{aligned} \frac{1}{b_{kl}} f_{ij}^{kl} &= 1, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \\ \frac{1}{b_{lk}} f_{ji}^{lk} &= 1, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \end{aligned}$$

yielding:

$$\frac{1}{b_{kl}} f_{ij}^{kl} = \frac{1}{b_{lk}} f_{ji}^{lk}, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l$$

and implying:

$$b_{lk} f_{ij}^{kl} = b_{kl} f_{ji}^{lk}, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.24)$$

These seemingly innocuous constraints are the key to balance the flow formulation stated above, equations (2.20) - (2.23), ensuring reasonable linear programming bounds and yet producing very easy linear programs. In fact, for the symmetric instances one set of flow balance equations, (2.21) or (2.22) can be dismissed. In order to try out all these formulations, comparing linear programming lower bounds and mixed integer programming solution time, a set of computational experiments was carried out.

2.3.3 A Brief Computational Experiment

At this point it is useful to discover how the flow formulation will behave, when up against some well documented instances from the literature, available in *QAPLIB*. In this initial experiment, an implementation using *ILOG CPLEX 7.0 Concert Technology* for Adams and Johnson linearization and our flow formulation was produced. These experiments were carried out in a SUN BLADE 100 workstation, equipped with one 500 MHz processor and 1 Gbyte of RAM memory. The *QAPLIB* instances selected to make part of the test, with sizes

n varying from 6 to 30, are shown in Tables 2.1, 2.2, 2.3 and 2.4. We are also solving purely pseudo-random instances, in the same range of sizes. These random instances are not Koopmans and Beckmann instances, since they do not sustain the triangular inequality, and are represented by names beginning with *rpqa* plus the size of the instance.

In Table 2.1, we show a comparison between linear programming bounds obtained by the flow formulation and Adams and Johnson linearization. In despite of the little degeneracy observed in our bounds, they are achieved at a low computing cost. Checking the instances with the higher bounds, it is possible to note that the flow formulation bounds improve quality when dealing with sparse demand matrices. This is not really a surprise, since Heffley [61] has concluded that the presence of sparse demand matrices could lead to integral assignments, when using Koopmans and Beckmann linearization. Because the flow formulation is in some sense derived from Koopmans and Beckmann model, we may expect that this property must be common to both. At this point, it is necessary to remark that in real life applications, one can expect sparse demand matrices instead of dense ones, as the problem size increases. In Figures (2.3) and (2.4) we have the obtained results in graphical form.

QAPLIB instance	Flow Formulation		Bound Quality	Adams and Johnson		Bound Quality	Integer Optimal
	lp bound	time[s]		lp bound	time[s]		
chr12a	8593.12	1	0.900	9552	725	1.000	9552
chr12b	7184	1	0.737	9742	508	1.000	9742
chr12c	10042.7	1	0.900	11156	1068	1.000	11156
chr15a	8621.94	4	0.871	9513	30146	0.961	9896
chr15c	9504	4	1.000	9504	3622	1.000	9504
had12	894	17	0.541	1621.54	2533	0.982	1652
had14	1300.5	62	0.477	2666.12	14778	0.979	2724
lipa10a	318.8	4	0.674	473	50	1.000	473
nug12	348	10	0.602	522.89	6597	0.905	578
nug15	621	86	0.540	1041	131923	0.905	1150
nug5	49	1	0.980	50	0	1.000	50
nug6	72	0	0.837	86	1	1.000	86
nug7	118	0	0.797	148	3	1.000	148
nug8	154	1	0.720	203.5	17	0.951	214
scr10	21958	2	0.816	26873.1	269	0.998	26922
scr12	25474	5	0.811	29827.3	4555	0.950	31410
tail0a	47953.3	3	0.355	131098	160	0.971	135028
tail0b	855788	1	0.723	1176140	248	0.994	1183760
tai5a	10747	0	0.833	12902	0	1.000	12902
tai6a	21427.8	1	0.728	29432	1	1.000	29432
tai7a	31730.1	0	0.588	53976	1	1.000	53976
tai8a	41952.2	1	0.541	77502	7	1.000	77502
tai9a	41816	2	0.442	93501	37	0.988	94622

Table 2.1: Linear programming bounds for both formulations under comparison and respective computing times.

It is time now to confront the flow formulation and Adams and Johnson linearization. To make our tests a little bit more realistic, we are setting profitabilities to install a facility in a given location. These terms are of capital importance, since they accomplish the heterogeneities of the environment, and translates the relation of the system under design to the external world (external world connections, location of external markets, policy of ground occupa-

tion, competition for locations and other interferences). They also serve as a point of connection with the well established location theory, linear by principle. The idea here is to verify the assumption that, as the p/q ratio increases, our model becomes more competitive. In this context, and since linear profitability matrices are not available on the instances of *QAPLIB*, we include linear profitabilities $p = (\sum_{(k,i)} a_{ki})$ varying from 0 to 10 times the magnitude of the quadratic transportation cost component $q = (\sum_{(i,j)} \sum_{(k,l)} c_{ij} f_{ij}^{kl})$, defining the linear/quadratic ratio p/q . These linear profitabilities were generated randomly with the aid of the standard pseudo-random number generator implemented on the GNU C compiler GCC version 3.0. It may also be observed that because the problems contained in *QAPLIB* are purely quadratic, there are no difference established between cost and demand matrices. This is not of real importance when dealing with formulations based on Lawler's linearization (since they pre-multiply b_{kl} and c_{ij}), but the flow formulation is not adapted to deal with zero transportation costs. This was responsible for some additional effort to rebuild the instances in a coherent way.

Tables 2.2, 2.3 and 2.4 presents the p/q ratio, and the computing times for the flow formulation and Adams and Johnson [1] linearization (2.9) - (2.13). The entries on Tables 2.2, 2.3 and 2.4 assigned with * are describing instances not solved in 24 hours of computation. In order to provide a better insight on analysis, we have plotted these results on Figures (2.5), (2.6), (2.7) and (2.8), in logarithmic scale.

Original instance	Problem size	Variables		p/q ratio	Flow form. time[s]	Adams and Johnson time[s]
		Integer	Continuous			
esc8a	8	64	3136	0	6	146
esc8b				0	15	710
esc8c				0	14	207
esc8d				0	6	196
esc8e				0	14	176
esc8f				0	5	202
nug6	6	36	900	0	2	1
nug7	7	49	1764	0	4	6
nug8	8	64	3136	0	15	227
rpqa7	7	49	1764	0	9	2
rpqa8	8	64	3136	0	55	83
rpqa9	9	81	5184	0	348	166
tai7a	7	49	1764	0	6	1
tai8a	8	64	3136	0	28	9
tai9a	9	81	5184	0	247	178
tai10a	10	100	8100	0	1197	1218

Table 2.2: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of integer mixed programming computing times.

From Tables 2.2, 2.3 and 2.4, we can realize that the linear term on the objective function is responsible for an expressive reduction of the computing times, sometimes of an order of magnitude. Beyond this, the flow formulation appears to balance two desirable qualities: a not so poor linear programming bound, easy to solve.

Original instance	Problem size	Variables		p/q ratio	Flow form. time[s]	Adams and Johnson time[s]
		Integer	Continuous			
chr12a	12	144	17424	0.426	4	584
chr12a				0.796	3	629
chr12b				0.383	5	516
chr12b				0.592	3	492
chr12c				0.314	3	803
chr12c				0.662	4	1217
chr15a	15	225	44100	0.631	16	14209
chr15a				0.856	6	10125
chr15b				0.726	36	77071
chr15b				1.817	14	60452
chr15c				0.944	19	3872
chr15c				0.807	9	3058
chr18a	18	324	93636	0.564	116	*
chr18a				1.304	41	*
chr18b				0.455	21	*
chr18b				1.864	7	*
chr20a	20	400	144400	1.047	76	*
chr20a				2.028	28	*
chr20b				0.824	47	*
chr20b				1.702	10	*
chr20c				1.056	108	*
chr22a	22	484	213444	0.304	138	*
chr22a				0.744	19	*
chr22b				0.293	96	*
chr22b				1.341	18	*

Table 2.3: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of mixed integer programming computing times.

In fact, for the problems with linear installation profitabilities, for some instances it is possible to find the integer optimal solution using our flow formulation before the completion of the linear programming solution of Adams and Johnson [1] formulation. The only class of test instances in which the flow formulation was defeated was for the instances *had***, just those that has some of the poorest linear programming bounds. In despite of that, the flow formulation lack of performance decreases as the ratio p/q grows.

These observations clearly suggest the existence of an equilibrium point between the bound quality and the cost to compute it. The natural conclusion here is that Adams and Johnson formulation is so hard to solve that his good linear programming bounds are overcome by the large computing times required to obtain them. Even in some purely quadratic instances it is possible to defeat Adams and Johnson computing times, as observed in Table 2.2.

We must also consider the solution of the modified versions of the instances *nug30* and *ste36a*, reported until now only by Anstreicher et al. [6], using large scale parallel computing. For certain conditions of the p/q ratio, these problems could be solved in computing times not superior to one hour, as reported in Table 2.4.

Based on the former discussion and on the good solution times acquired by our flow formulation, we introduce a Benders decomposition scheme for the problem.

Original instance	Problem size	Variables		p/q ratio	Flow form. time[s]	Adams and Johnson time[s]
		Integer	Continuous			
had12	12	144	17424	0.585	35	14
had12				1.534	2	3
had12				4.707	2	3
had14	14	196	33124	0.464	20	42
had14				1.312	8	7
had14				2.592	12	8
had16	16	256	57600	0.827	12	16
had16				1.036	12	11
had18	18	324	93636	0.689	209	74
had18				1.590	89	49
lipa10a	10	100	8100	2.375	2	1
lipa10a				5.770	1	1
lipa10b				0.693	6	4
lipa10b				1.440	1	1
lipa10b				2.524	1	1
nug12	12	144	17424	2.401	3	110
nug12				3.621	3	30
nug12				8.584	1	9
nug15	15	225	44100	0.990	61	1350
nug15				1.113	24	258
nug15				2.981	5	94
nug15				5.522	6	126
nug20	20	400	144400	0.434	1069	*
nug20				3.268	42	654
nug20				4.477	80	799
nug30	30	900	756900	0.769	2719	*
nug30				1.358	1779	*
scr10	10	100	8100	0.221	22	311
scr10				0.414	7	95
scr12	12	144	17424	0.188	146	13554
scr12				0.336	54	6854
ste36a	36	1296	1587600	0.843	2896	*
ste36a				1.199	2801	*
ste36a				1.871	2196	*
tai10b	10	100	8100	0.003	197	344
tai10b				0.013	96	714
tai10b				0.023	95	658

Table 2.4: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and a comparison of mixed integer programming computing times.

2.4 Benders Decomposition of the Problem

Benders partitioning method was published in 1962 [17] and it was initially developed to solve mixed integer programming problems. The computational success of the method to solve large scale multi-commodity distribution system design models has been confirmed since the pioneering paper of Geoffrion and Graves [53]. Magnanti and Wong [89] proposed a methodology to improve Benders decomposition algorithm performance when applied for solving mixed-integer programs. They introduced a technique for accelerating the algorithm convergence and developed a theory that distinguishes "good" formulations for those problems that have different, but equivalent, possible formulations. We can also remark the work of Geoffrion, on the generalized Benders method [52], and Balas and Bergthaller [10] revisiting the cut generation procedure. A Benders partitioning method essentially relies on a *projection* problem manipulation,

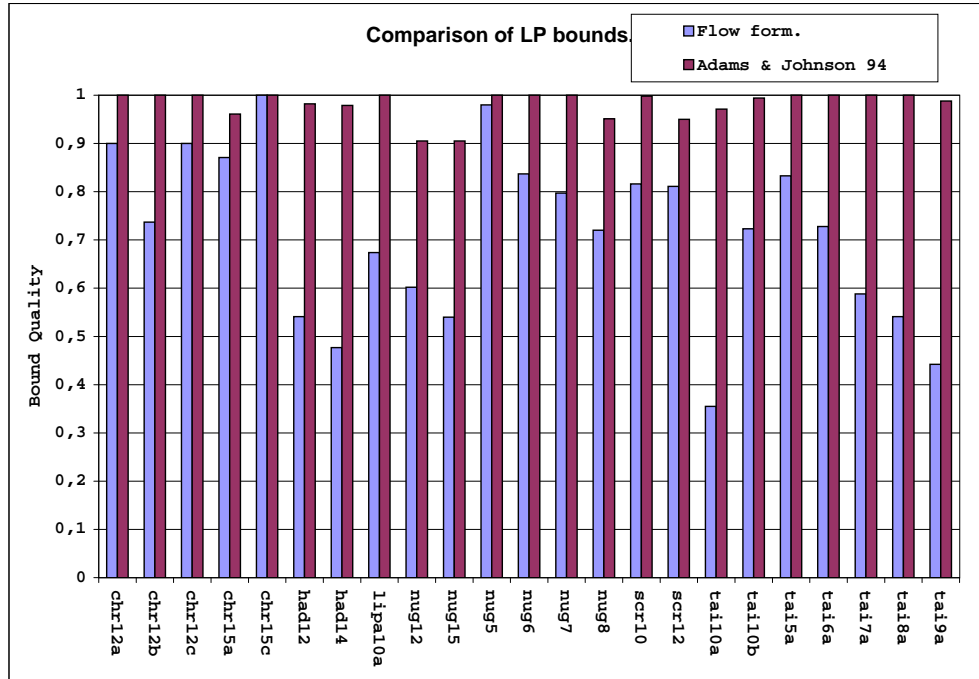


Figure 2.3: Comparison of linear programming bounds for both formulations.

that is then followed by the solution strategies of dualization, outer linearization and relaxation.

Starting with the flow formulation described by equations (2.20) - (2.23), from the viewpoint of mathematical programming we can conceive a projection of the problem onto the space of the *assignment* variables x , thus resulting the following implicit problem to be solved at a superior level:

$$\min - \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} + t(x) \quad (2.25)$$

subject to (2.1)-(2.3)

where $t(x)$ is calculated by the following problem to be solved at an inferior

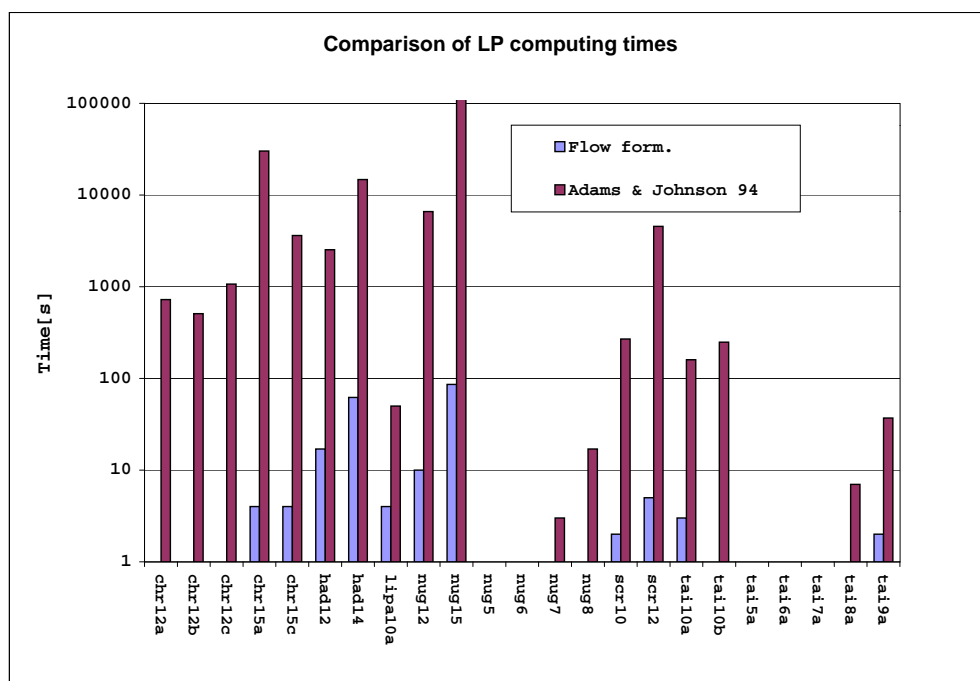


Figure 2.4: Comparison of lp computing times for both formulations.

level:

$$t(x) = \min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ij} f_{ij}^{kl} \quad (2.26)$$

subject to:

$$-\sum_{j=1}^n f_{ij}^{kl} = -b_{kl} x_{ki}, \quad \forall i, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.27)$$

$$\sum_{i=1}^n f_{ij}^{kl} = b_{kl} x_{lj}, \quad \forall j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.28)$$

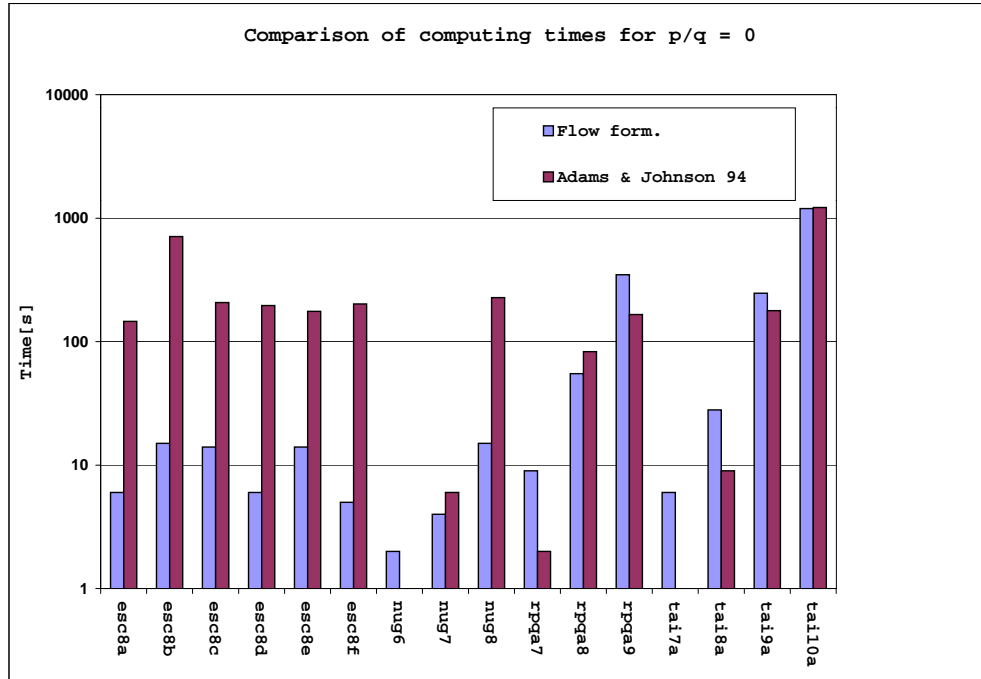


Figure 2.5: Comparison of mip computing times for both formulations.

$$f_{ij}^{kl} \geq 0, \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.29)$$

for x fixed.

The feasibility requirements related to the integer variables x implies that the facilities for which $x_{ki} = 1$ are such that the superior level solution is an *assignment* between the set of facilities and the set of locations. Thus, there is no need for further feasibility constraints on the domain of the projected problem (2.25), and the existence of the minimum in the subproblem (2.26)-(2.29) is ensured since we are minimizing a convex function in a nonempty set.

Since the subproblem has a linear objective function and linear constraints, the Karush-Kuhn-Tucker conditions are necessary and sufficient for optimality. With two associated vectors v_j^{kl} and u_i^{kl} of dual variables, and since there is

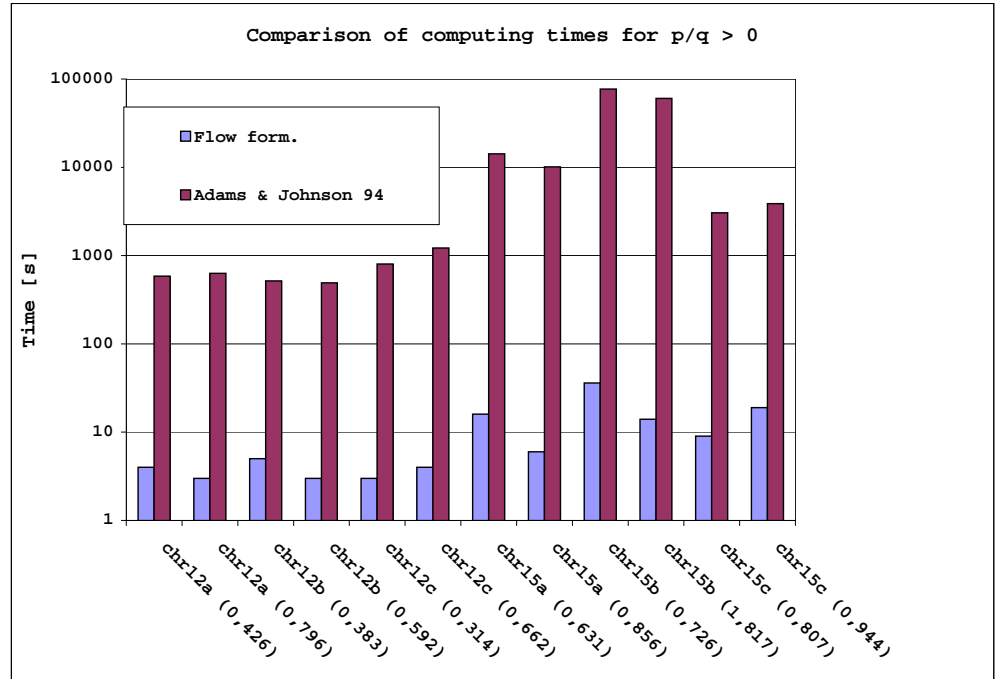


Figure 2.6: Comparison of mip computing times for both formulations.

no duality gap for any x which forms an assignment, the optimal value of the subproblem can be written as:

$$t(x) = \max \sum_{k=1}^n \sum_{l=1}^n \sum_{j=1}^n b_{kl} x_{lj} v_j^{kl} - \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n b_{kl} x_{ki} u_i^{kl} \quad (2.30)$$

subject to:

$$v_j^{kl} - u_i^{kl} \leq c_{ij}, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.31)$$

$$v_j^{kl} \in \mathcal{R}, \quad \forall j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.32)$$

$$u_i^{kl} \in \mathcal{R}, \quad \forall i, k, l = 1, \dots, n, i \neq j, k \neq l \quad (2.33)$$

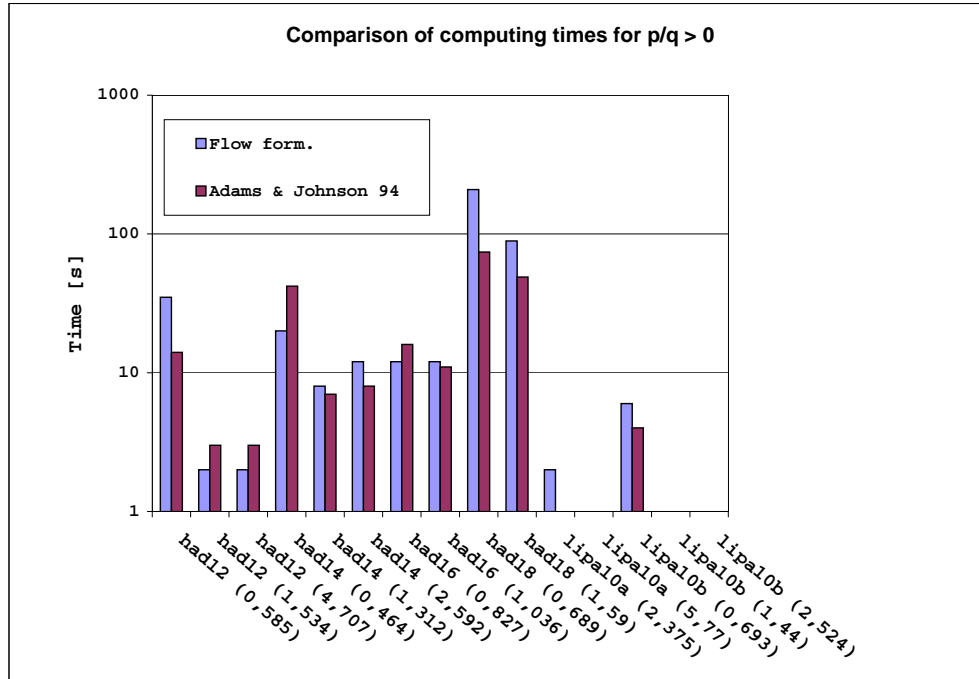


Figure 2.7: Comparison of mip computing times for both formulations.

for x fixed.

At this point, it is interesting to observe that the feasible solution set of the dual subproblem is always the same, independently of the assignment x . So, for every x , the value of the dual objective function underestimates the corresponding primal objective cost. If at a certain cycle h the subproblem has been solved for a given assignment x_{ki}^h the optimal value $t(x^h)$ occurs for $v_j^{kl} = v_j^{kl,h}$ and $u_i^{kl} = u_i^{kl,h}$ and is given by:

$$t(x^h) = \sum_{k=1}^n \sum_{l=1}^n \sum_{j=1}^n b_{kl} x_{lj}^h v_j^{kl,h} - \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n b_{kl} x_{ki}^h u_i^{kl,h} \quad (2.34)$$

Using the fact that a supremum is the least upper bound, the problem (2.25)

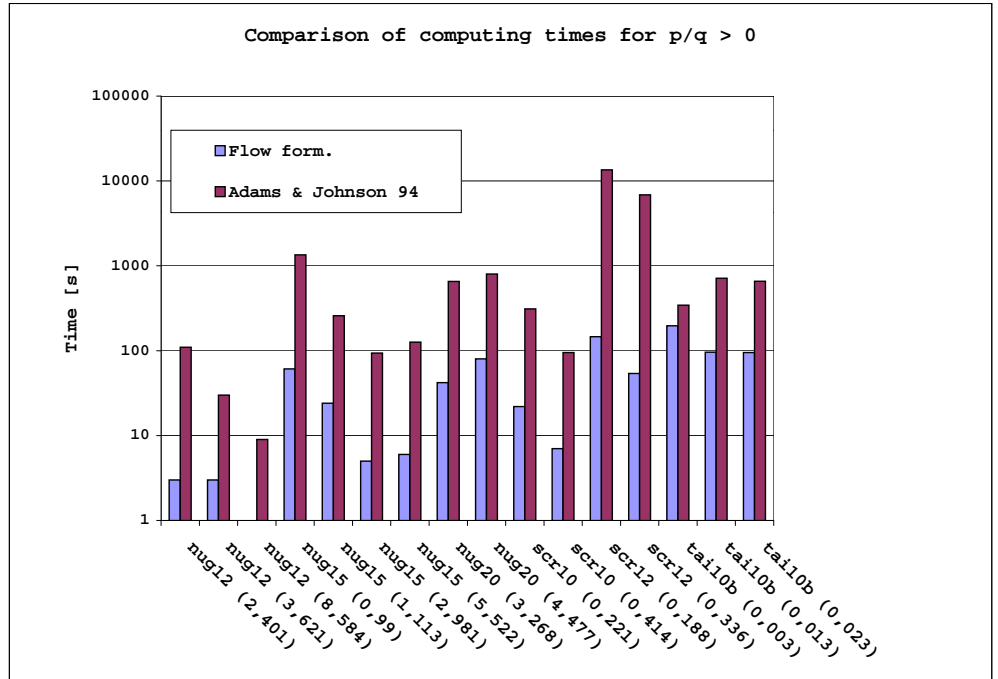


Figure 2.8: Comparison of mip computing times for both formulations.

is equivalent to the master problem:

$$\min - \sum_{k=1}^n \sum_{i=1}^n a_{ki} x_{ki} + \eta \quad (2.35)$$

subject to (2.1) - (2.3) and:

$$\eta \geq \sum_{k=1}^n \sum_{l=1}^n \sum_{j=1}^n b_{kl} x_{lj} v_j^{kl,h} - \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n b_{kl} x_{ki} u_i^{kl,h}, \forall h \quad (2.36)$$

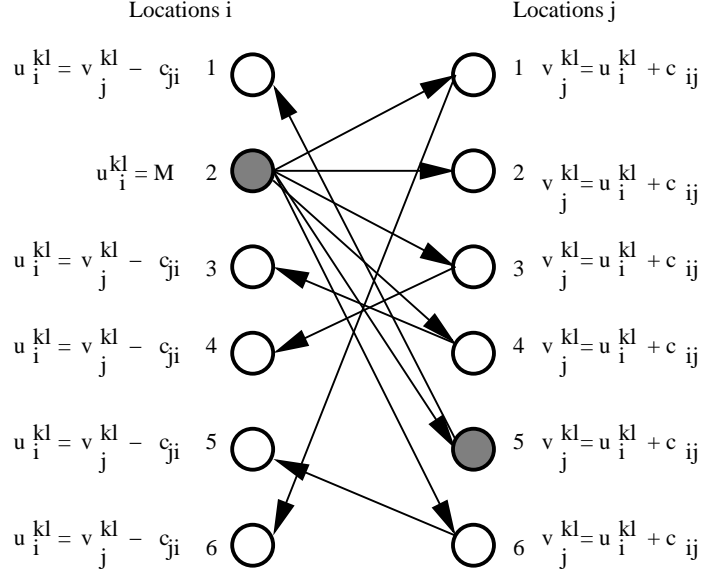


Figure 2.9: An example of automatic construction of a feasible solution for the dual subproblem.

2.4.1 Subproblems

For a fixed assignment associated with the matrix x_{ki}^h , the computation of a minimal cost flow $t(x^h)$ can be separated in a series of trivial network flow problems, one for each pair kl . We remark that, for an optimal solution $(x_{ki}^h, f_{ij}^{kl,h})$ for the primal problem, an associated optimal solution $(v_j^{kl,h}, u_i^{kl,h})$ for the dual subproblem (2.30) - (2.33) should minimize for each f_{ij}^{kl} the correspondent parcel of the associated Lagrangean function.

This dual problem has many feasible solutions, contrarily to the primal problem that has a unique trivial solution. Since $f_{ij}^{kl,h} = b_{kl}$ if $x_{ki}^h = 1$ and $x_{lj}^h = 1$ we have from the complementary slackness condition that:

$$\begin{aligned} v_j^{kl,h} - u_i^{kl,h} &\leq c_{ij}, \quad \forall i, j = 1, \dots, n, i \neq j, k \neq l \\ v_j^{kl,h} - u_i^{kl,h} &= c_{ij}, \quad \text{if } x_{ki}^h = 1 \text{ and } x_{lj}^h = 1 \end{aligned}$$

In such a way that we can obtain the following dual feasible solution, associated with the primal solution $f_{ij}^{kl,h}$ (see Figure (2.9)). Fixing a single variable $u_i^{kl,h}$, it is possible to construct:

$$v_j^{kl,h} = u_i^{kl,h} + c_{ij}, \quad \forall j = 1, \dots, n, i \neq j, k \neq l \quad (2.37)$$

And using the above defined $v_j^{kl,h}$, we can define the other variables $u_i^{kl,h}$:

$$u_i^{kl,h} = \max_{j, j \neq i} [v_j^{kl,h} - c_{ij}], \quad \forall i = 1, \dots, n, i \neq j, k \neq l \quad (2.38)$$

The systematic evaluation of the dual variables with meaningful values is a clue for an efficient implementation. Here, the two series of dual variables can be interpreted as price information. Each variable $v_j^{kl,h}$ represents the commodity price after flowing from facility k to facility l , if facility l is placed at location j . The variables $u_i^{kl,h}$ represents the commodity price before flowing from facility k to facility l , if facility k is placed at location i .

2.4.2 Enhancing the Benders Decomposition Algorithm with Flow Equilibrium Constraints

Our task now is to modify the above proposed scheme to accomplish constraints (2.24). It is possible to observe that (2.24) describes a coupling between the flows of commodities kl and lk . In order to perform the task, it is necessary to point that, for a fixed $x = x^h$, the primal subproblem for commodities kl and lk is, for $k \neq l$:

$$\min \sum_{i=1}^n \sum_{j=1}^n (c_{ij} f_{ij}^{kl} + c_{ji} f_{ji}^{lk})$$

subject to:

$$\begin{aligned} - \sum_{j=1}^n f_{ij}^{kl} &= -b_{kl} x_{ki}, \quad \forall i = 1, \dots, n, i \neq j \\ - \sum_{i=1}^n f_{ji}^{lk} &= -b_{lk} x_{lj}, \quad \forall j = 1, \dots, n, i \neq j \\ \sum_{i=1}^n f_{ij}^{kl} &= b_{kl} x_{lj}, \quad \forall j = 1, \dots, n, i \neq j \\ \sum_{j=1}^n f_{ji}^{lk} &= b_{lk} x_{ki}, \quad \forall i = 1, \dots, n, i \neq j \\ b_{lk} f_{ij}^{kl} &= b_{kl} f_{ji}^{lk}, \quad \forall i, j = 1, \dots, n, i \neq j \\ f_{ij}^{kl} &\geq 0, \quad \forall i, j = 1, \dots, n, i \neq j \\ f_{ji}^{lk} &\geq 0, \quad \forall i, j = 1, \dots, n, i \neq j \end{aligned}$$

The trivial and unique solution of this problem is $f_{ij}^{kl} = b_{kl}$ and $f_{ji}^{lk} = b_{lk}$, for $x_{ki} = 1$ and $x_{lj} = 1$, and $f_{ij}^{kl} = 0$ and $f_{ji}^{lk} = 0$ otherwise. This result leads to the dual subproblem for commodities kl and lk , for $x = x^h$, $k \neq l$:

$$\max b_{kl} \left(\sum_{j=1}^n x_{lj} v_j^{kl} - \sum_{i=1}^n x_{ki} u_i^{kl} \right) + b_{lk} \left(\sum_{i=1}^n x_{ki} v_i^{lk} - \sum_{j=1}^n x_{lj} u_j^{lk} \right)$$

subject to:

$$\begin{aligned} v_j^{kl} + b_{lk} \lambda_{ij}^{kl} - u_i^{kl} &\leq c_{ij}, \quad \forall i, j = 1, \dots, n, i \neq j \\ v_i^{lk} - b_{kl} \lambda_{ij}^{kl} - u_j^{lk} &\leq c_{ji}, \quad \forall i, j = 1, \dots, n, i \neq j \\ v_j^{kl} &\in \mathcal{R}, \quad \forall j = 1, \dots, n, i \neq j \\ u_i^{kl} &\in \mathcal{R}, \quad \forall i = 1, \dots, n, i \neq j \end{aligned}$$

Fixing at a reference value a single variable u_i^{kl} and also a single variable u_j^{lk} , making $\lambda_{ij}^{kl} = 0$ for i and j such that $x_{ki}^h = 1$ and $x_{lj}^h = 1$, it is possible to write:

$$\begin{aligned} v_j^{kl,h} &= u_i^{kl,h} + c_{ij}, \quad \forall j = 1, \dots, n \\ v_i^{lk,h} &= u_j^{lk,h} + c_{ji}, \quad \forall i = 1, \dots, n \end{aligned}$$

Using the above defined $v_j^{kl,h}$ and $v_i^{lk,h}$, we have for u_i^{kl} and u_j^{lk} :

$$\begin{aligned} u_i^{kl,h} - b_{lk} \lambda_{ij}^{kl,h} &\geq v_j^{kl,h} - c_{ij}, \quad \forall i, j = 1, \dots, n, i \neq j \\ u_j^{lk,h} + b_{kl} \lambda_{ij}^{kl,h} &\geq v_i^{lk,h} - c_{ji}, \quad \forall i, j = 1, \dots, n, i \neq j \end{aligned}$$

We must remark that, an enough high value of λ_{ij}^{kl} can eventually increase the value of $u_i^{kl,h}$ while eventually decreasing the value of $u_j^{lk,h}$. The idea here is to decrease the value of any component of vector u as far as possible, but never increasing the value of another component. Constructing a consistent dual solution we have to fix, for instance, all the components of vector $\lambda^h \geq 0$ as high as possible, while maintaining the idea of never increasing of any component of vector u^h :

$$u_i^{kl,h} = \max_{j, j \neq i} [v_j^{kl,h} - c_{ij}], \quad \forall i = 1, \dots, n, i \neq j$$

And determining $\lambda_{ij}^{kl,h}$ in such way that:

$$\begin{aligned} \lambda_{ij}^{kl,h} &= 0 \quad \text{for index } j \text{ that maximizes } v_j^{kl,h} - c_{ij} \\ \lambda_{ij}^{kl,h} &= \frac{1}{b_{lk}} (u_i^{kl,h} - v_j^{kl,h} + c_{ij}), \quad \forall i, j = 1, \dots, n, i \neq j \end{aligned}$$

After this step, we have a way to reduce some values of the origin prices u for commodity lk :

$$u_j^{lk,h} = \max_{i, i \neq j} [u_i^{lk,h} - b_{kl}\lambda_{ij}^{kl,h} - c_{ji}], \quad \forall i = 1, \dots, n, i \neq j$$

Now, we are ready to try out our decomposition algorithm over a set of instances, object of the next section.

2.5 Computational Experiments Using Enhanced Benders Decomposition

These experiments follows the same standards and adopted convention used to compare the flow formulation and Adams and Johnson formulation [1]. The tests were carried out in a Sun Blade 100 with one 500 MHz Ultra-SPARC processor and 1 Gbyte of RAM memory. The operational system is Solaris 5.8. The Benders decomposition algorithm was implemented in C++ with *CPLEX 7.0* application programming interface called *ILOG Concert Technology*. To perform the tests, instances from *QAPLIB*, by Burkard, Karish and Rendl [29], with sizes from $n = 8$ to $n = 36$ were selected. The *QAPLIB* instances selected to make part of the test, are shown in Tables 2.5 and 2.6. We have solved purely pseudo-random instances, in the same range of sizes. These random instances are not Koopmans and Beckmann instances, since they do not sustain the triangular inequality, and are represented by names beginning with *rpqa* plus the size of the instance. We are setting profitabilities to install a facility in a given location, from $p/q = 0$ to $p/q = 16$. Tables 2.5 and 2.6 presents the p/q ratio, and the computing times for the Benders decomposition algorithm.

Original instance	Problem size	Iterations h	Variables		p/q ratio	Benders algorithm time[s]
			Integer	Continuous		
esc8a	8	82	64	3136	0	473
nug5	5	33	25	400	0	11
nug6	6	188	36	900	0	1266
nug12	12	102	144	17424	1.810	663
		23			3.621	11
nug15	15	31			3.621	26
		11	225	44100	4.881	2
		15			6.303	5
nug20	20	63			1.258	661
		13	400	144400	3.992	11
		9			4.477	2
rou12	12	99	144	17424	2.630	199
ste36a	36	108	1296	1587600	3.853	1381
		40			5.780	108
tai5a	5	45	25	400	0	14
tai6a	6	188	36	900	0	1071
tai7a	7	657	49	1764	0	90437

Table 2.5: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition.

We are able now to compare the decomposition algorithm to the monolithic implementation of the flow formulation, evaluating the decomposition cost, given by loss of information at the superior level. This is done in Table 2.7.

Instance name	Problem size	Iterations h	Variables		p/q ratio	Benders algorithm time[s]
			Integer	Continuous		
rpqa16	16	40	256	57600	2.395	67
		37			2.412	56
		102			2.490	842
		47			2.667	94
		53			2.680	136
		33			2.701	53
		19			2.752	12
		20			4.158	12
		15			4.400	6
		20			4.508	12
rpqa25	25	74	625	360000	3.261	821
		40			4.483	124
		42			4.496	146
		36			4.598	101
		31			4.731	72
		30			4.771	63
		21			5.194	30
		21			5.381	29
		11			5.767	6
		26			5.801	40
		15			5.816	11
		20			6.071	22
		15			6.377	11
rpqa9	9	21	81	5184	3.037	5
		6			7.594	0
		5			11.071	0
		3			15.189	0

Table 2.6: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition.

Tables 2.8 and 2.9 presents an evolution of the computing times for Benders decomposition algorithm and for our monolithic implementation. These results are plotted in Figures (2.10), (2.11) and (2.12).

As one can see, the computing times for the flow formulation monolithic implementation are sometimes better than those obtained by Benders decomposition scheme. The exception occurs on the situations that we deal with the larger instances, with higher p/q ratios. This effect is due to the master problem strengthening, that accelerates the lower bound progression.

We can sustain that high p/q ratios better describe the cost structure of real large scale implementations, and that the linear parcel, that represents the profitability of a given location, can be considered in some cases more important than the transportation costs, from the economic point of view. This is specially true when we think in location theory, since the linear term gives the profitability of a location for an economic activity. The economic equilibrium condition is so found for $p/q = 1$, and all situations where $p/q > 1$ capture liquid profit for the considered optimal location for at least one activity. It is necessary to make clear that our objective, when we start to develop the decomposition scheme, was to go where no one has gone before: proceed with the solution of larger instances. This is impossible without decomposition since, for instances of size beyond 40, *CPLEX* crashes down due to lack of computer memory.

In fact, for large p/q ratios, it is possible to solve larger instances, without use

Original instance	Problem size	Iterations h	Variables		p/q ratio	Benders algorithm time[s]	Flow form. time[s]
			Integer	Continuous			
chr12a	12	55	144	17424	1.571	196	3
chr12b		31			2.555	59	4
chr12c		81			2.374	736	4
chr15a	15	142	225	44100	1.383	3349	8
chr15c		76			2.191	1283	3
chr15b		28			3.268	27	4
nug15	15	9	225	44100	1.894	2	7
nug20	20	30	400	144400	3.267	35	61
nug30	30	44	900	756900	1.889	188	1236
chr18a	18	68	324	93636	2.407	201	16
chr18b		3			21.40	0	11
chr20a	20	9	400	144400	10.82	2	19
chr20a		5			8.948	0	18
chr20a		4			3.871	1	18
chr20a		38			4.266	56	26
chr20a		50			4.728	99	20
chr20b		4			10.75	0	20
chr20b		4			6.854	1	20
chr20b		15			4.069	6	20
chr20b		45			2.268	65	22
chr22a		22			69	484	213444
had12	12	13	144	17424	2.531	4	3
had14	14	25	196	33124	2.591	19	13
had16	16	9	256	57600	1.035	2	12

Table 2.7: Problem dimensions for test instances, number of integer and continuous variables, p/q ratio and computing times for Benders decomposition and flow formulation.

of massive parallel computing, as can be seen in Table 2.10 where the computing where limited to 36 hours.

We are considering these results very expressive, since there is no solution report in the literature for any instance of size beyond 36, considering any available formulation or algorithm. Since any extension of a \mathcal{NP} -hard problem is also \mathcal{NP} -hard, the addition of information about the external environment, trough the linear profitabilities, do not makes QAP easier, in a theoretical point of view. This fact is confirmed by the computing times observed for the larger instances (Table 2.10).

2.6 Concluding Remarks

On the cases where we can define or compute heterogeneous profitabilities, it is possible to solve large instances of QAP , without an excessive computational cost or the use of massive parallel computing. This conclusion has its foundations on the pioneer work of Koopmans and Beckmann and also on the work of Hefley, many years later. The inclusion of heterogeneous profits for location is a natural step when considering location theory, and can introduce external environment influence on the location decision process, being more realistic.

Once established this, the new flow formulation has proved to unify desirable qualities for a good mathematical programming implementation: be easy to solve, giving good linear programming bounds. These two qualities are directly

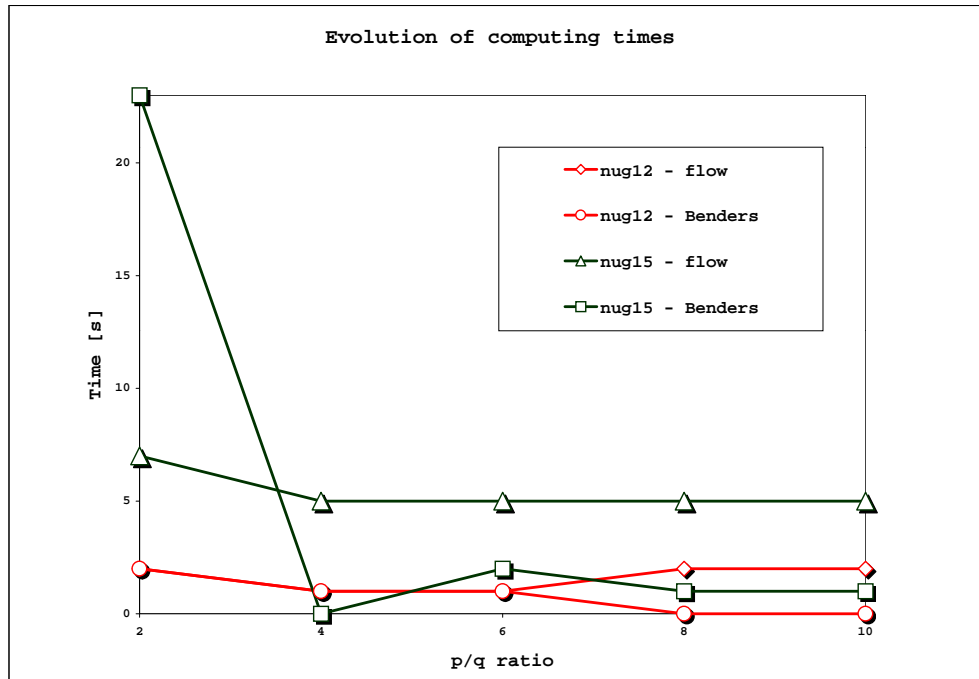
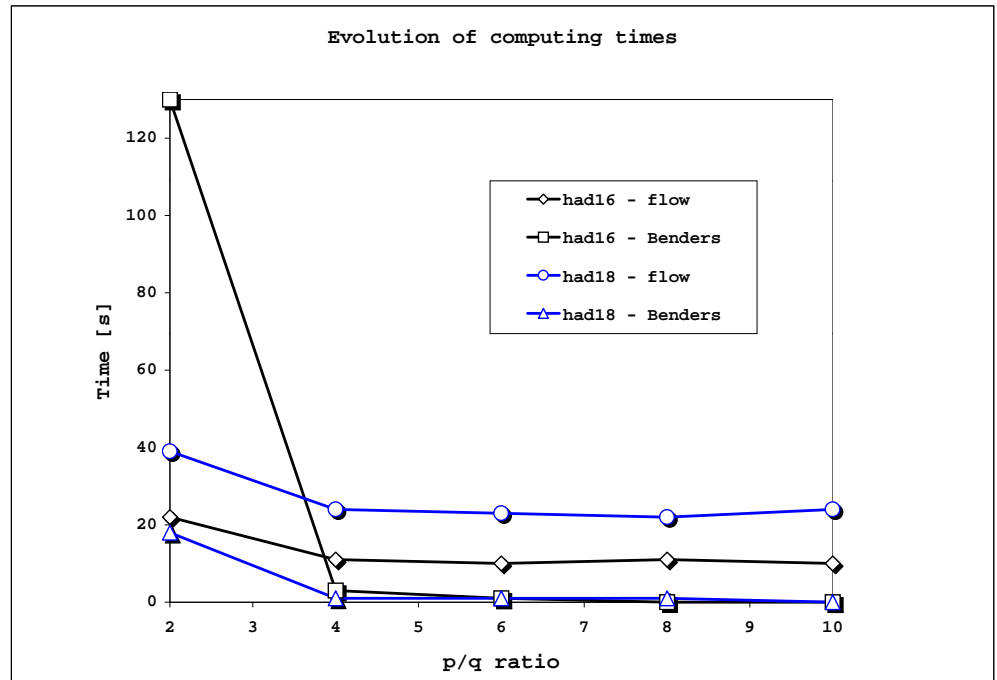


Figure 2.10: Evolution of computing times with p/q ratio.

responsible for the good computing times achieved, and also for the solution of larger instances at reasonable cost, until now obtainable only through the use of computational grids.

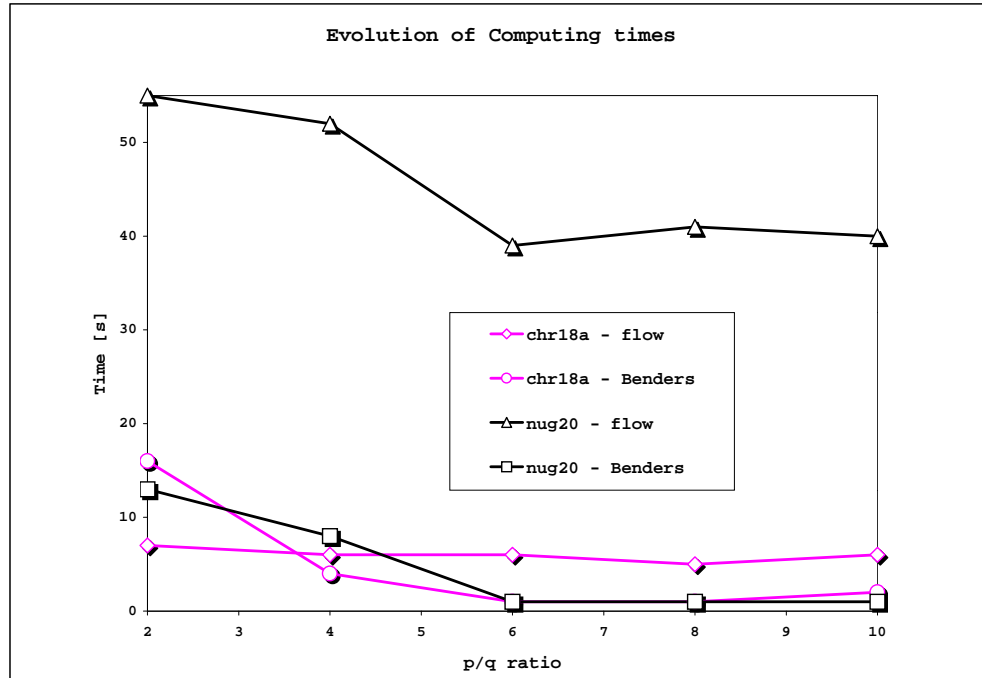
For the instances of size beyond 40, the Benders decomposition algorithm appears to be the best choice to find an exact solution, avoiding excessive space and time complexity, if we observe some conditions about the cost structure.

For future work, it is necessary to better explore the equilibrium between bound quality and cost of computation, detecting when and how to merge easy to compute and stronger and hard to compute formulations for a given problem.

Figure 2.11: Evolution of computing times with p/q ratio.

p/q ratio	0		1		2		4	
Instance	flow	Benders	flow	Benders	flow	Benders	flow	Benders
nug12	*	*	5	101	2	2	1	1
nug15	*	*	7	129	7	23	5	0
nug20	*	*	139	30751	55	13	52	8
had14	*	*	16	4	5	1	5	1
had16	*	*	12	27781	22	130	11	3
had18	*	*	31	3595	39	18	24	1
chr15a	*	*	6	511	7	32	5	4
chr15b	*	*	35	748	6	45	3	22
chr15c	*	*	19	469	5	37	2	89
chr18a	*	*	40	*	7	16	6	4
ste36a	*	*	2821	*	2196	*	1971	952

Table 2.8: Evolution of computing times for Benders algorithm and the flow formulation

Figure 2.12: Evolution of computing times with p/q ratio.

p/q ratio	6		8		10	
	flow	Benders	flow	Benders	flow	Benders
nug12	1	1	2	0	2	0
nug15	5	2	5	1	5	1
nug20	39	1	41	1	40	1
had14	5	1	5	2	7	2
had16	10	0	11	0	10	0
had18	23	1	22	1	24	0
chr15a	2	2	4	1	3	0
chr15b	3	3	3	1	2	0
chr15c	3	1	2	1	3	1
chr18a	6	1	5	1	6	2
ste36a	1964	1	1925	3	1947	3

Table 2.9: Evolution of computing times for Benders algorithm and the flow formulation

Original instance	Problem size	Iterations h	Variables		p/q ratio	Benders algorithm time[s]
			Integer	Continuous		
tho40	40	278	1600	2433600	2.161	43734
sko49	49	268	2401	5531904	8.386	57170
		100			10.240	4224
sko64	64	295	4096	16257024	8.303	134236
		118			9.707	6859

Table 2.10: Solution of larger instances using the Benders algorithm.

Chapter 3

The Placement of Electronics with Thermal Effects

3.1 Introduction

Nowadays, all the electronic and micro-electronic devices are migrating from controlled environment places (laboratories, offices) to the direct application ones (our houses, cars, and even clothes). This fact introduces a new element in the product reliability equation: the capability of maintaining design conditions during operation. The engineers and designers are now facing a new challenge: how to protect the most vulnerable parts of this kind of component against damage on the application environments? They are dealing with high temperatures, atmospheric residues, mechanical interference and vibration. Is it possible to create products which are just designed for maximum efficiency and ignore these operational conditions? The answer seems to be no. In fact, reliability is a well known component of the quality function deployment.

It is important to remark that the heat transfer efficiency is a strong constraint when designing more powerful computing machinery. This is a major reason for the recent efforts on dealing with thermal problems on the micro-electronic domain, as can be seen in Lorente, Wechsato and Bejan [81], Zuo, Hoover and Phillips [131], Visser and Kock [126] and Rocha, Lorente and Bejan [117]. The work of Wechsato et al. [127] is a good reference on how network flow models can be used to design an optimized distribution coolant network for electronics systems. Several efforts have been done to obtain solutions that compromise electronic components placement and temperature profiles, see Huang et al. [66], [65], [64] for MCM (Multi Chip Modules) design, and Queipo [113], [112]. For a more complete survey on the electronics cooling matter, we suggest to read Burmann et al. [30], Boyalakuntla and Murthy [22], Tucker [125], Ros-

ales et al. [118], EYK, Wen and Choo [100], Craig et al. [39] and Queipo et al. [111].

In order to overcome the problem, we must have a computer optimization algorithm which is capable of solving combinatorial placement problems in an efficient manner. It must be powerful enough to deal with the secondary metric — the thermal component. The exact optimization methods are not well succeeded in dealing with real size instances and the heat transfer associated problem is nonlinear and non-convex, although. In this work we design a Benders decomposition based algorithm that is capable to solve exactly the placement problem keeping good solutions for the maximum temperature rising on the surface board. The proposed algorithm is a heuristic. The models developed by Queipo [113] and Huang [65] and our approach are very similar, but instead of dealing with the thermal-placement combined problem with the aid of metaheuristics, we are proposing a performance guarantee heuristic.

In section 3.2, the thermal model is developed and the temperature penalty function is considered, being appreciated aspects involving the use of *Finite Volume Method* [108] to solve the *Energy Conduction Equation* and the concerning boundary conditions. In section 3.3, the computational experiments and the corresponding results are shown, where the test instances are viewed in a detailed way, resuming some concluding remarks and giving hints for future work.

On the placement design of electronic boards, one needs to place n electronic components to n established locations in a printed circuit card, building the complete electronic board. As proposed by Steinberg (see [29]), it is interesting to minimize the distance among components which has greater levels of interactivity and energy or data flow, in order to avoid excessive signal delays. This is a location problem which can be modeled as an instance of the *QAP*. On the other hand, if all the major heat sources are put together, one can create a so called “hot-spot” on the board: a specific region of high energy dissipation that causes usual heat sinks to present low efficiency. Then it becomes necessary to investigate the sensitivity of optimal placement solution, when a new quality criterion is introduced: the maximal surface temperature.

3.2 Thermal Modeling and Temperature Penalty Costs

It is necessary to develop the capability to simulate the thermal field behavior for a given assignment. The main equation for heat transfer phenomena is the well known *Energy Conduction Equation*, given here in two-dimensional form:

$$\kappa_z \frac{\partial^2 T}{\partial z^2} + \kappa_y \frac{\partial^2 T}{\partial y^2} + g(z, y, \tau) = \rho c_p \frac{\partial T}{\partial \tau} \quad (3.1)$$

where T is the temperature [$^{\circ}C$], z and y are the spatial coordinates [m], $g(z, y, \tau)$ is the heat source volumetric rate discrete distribution [W/m^3], τ is

the time $[s]$, κ is the thermal conductivity $[W/(^{\circ}C \cdot m)]$, ρ is the characteristic density $[kg/m^3]$ of the system and c_p is the constant pressure specific heat $[kJ/(^{\circ}C \cdot kg)]$. This second order partial differential equation is subject in each lateral side to the following boundary conditions:

$$\begin{aligned}
-\kappa_z A_z \frac{\partial T}{\partial z} \Big|_{z=z_1} &= h_1^{conv} A_z (T - T_{\infty}) \\
-\kappa_z A_z \frac{\partial T}{\partial z} \Big|_{z=z_2} &= h_2^{conv} A_z (T - T_{\infty}) \\
-\kappa_y A_y \frac{\partial T}{\partial y} \Big|_{y=y_1} &= h_3^{conv} A_y (T - T_{\infty}) \\
-\kappa_y A_y \frac{\partial T}{\partial y} \Big|_{y=y_2} &= h_4^{conv} A_y (T - T_{\infty})
\end{aligned} \tag{3.2}$$

and to an initial condition like:

$$T \Big|_{\tau=\tau_0} = T_0 \tag{3.3}$$

Here, h_i^{conv} , for each $i = 1, 2, 3, 4$, is the convective heat transfer coefficient $[W/(^{\circ}C \cdot m^2)]$ at each corresponding boundary. The natural and forced convective heat transfer over the horizontal surface is included as a general negative source term packed in $g(z, y, \tau)$, having the same form of (3.2), approximating the combined heat transfer coefficient by:

$$h_{surface}^{conv} = \frac{\kappa}{L} \cdot 0.664 Re^{1/2} Pr^{1/3} \tag{3.4}$$

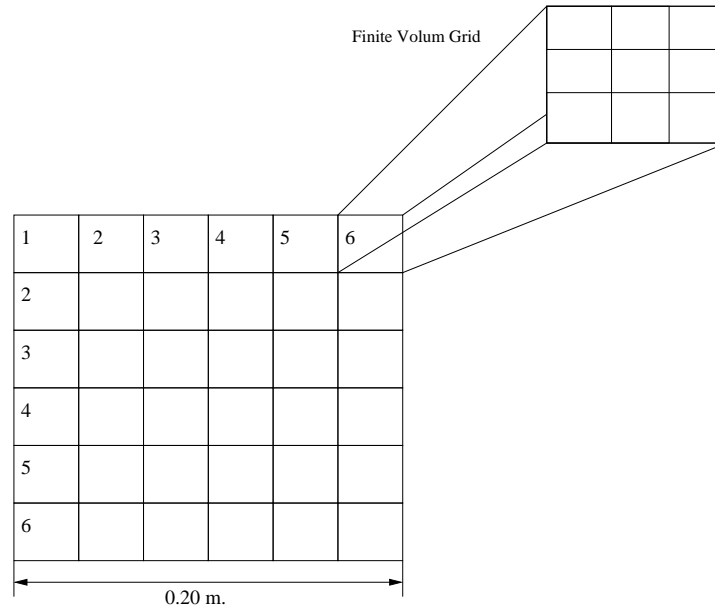
where Re is the associated *Reynolds Number*, Pr is the *Prandtl Number* and L is a fluid flow geometry dependent length $[m]$. To obtain good solutions for this model, we can use a simple version of the Finite Volume Technique [108]. In our discretization, we are using a mesh nine times the size of the test instance (Figure (3.1)). Only the two-dimensional isotropic steady state situation is under analysis. The Central Difference Interpolating Scheme was adopted, and an average heat source term in each volume is used to accomplish the discrete nature of heat source distribution. Since steady state heat conduction usually presents good solution properties for the associated Finite Volume equation set, we can not observe numerical diffusion, instabilities or other numerical degradation at this level of grid resolution. All the thermo-physical properties used to describe the thermal model are given in Table 3.1 (the boundary convective conditions was chosen as typical values in electronics equipments [16], [123]).

3.2.1 Maximum Temperature and Penalty Costs

When solved, the thermal model (equations (3.1)-(3.4)) can determine the highest temperature over the electronics board under investigation. It is interesting

Table 3.1: Thermo-physical properties for the thermal model.

Environment Temperature	25 °C
Total Dissipated Power	120 W
Lateral Board Dimension (L)	0.20 m
Thermal Conductivity (Glass Fiber - Epoxy)	$5.9 \cdot 10^{-1} W/(m \cdot K)$
Lateral Convective Heat Transfer Coefficients	$1.0 \cdot 10^{-4} W/(m^2 \cdot K)$

Figure 3.1: *QAP* instance and Finite Volume Grid representation.

to remark that, for each fixed assignment x given by the master problem (2.35)-(2.36), see chapter 2, a different temperature field and maximum temperature must be found. Since the source term $g(z, y, \tau)$ is given by the power dissipation of each electronic component being assigned to a given location, it is not trivial to obtain analytical solution for (3.1)-(3.4), considering the discrete nature of the power source distribution over the board (in fact, this singularity is the reason for the adoption of a Finite Volume technique). Beyond this, since the maximum temperature is not obtained explicitly, it is very difficult to establish a function correlating the maximum temperature and the assignment x , becoming virtually impossible to ensure mathematical properties like convexity, discarding approaches via generalized Benders decomposition [52].

It is necessary to point that the maximum temperature over the electronics board under study is an important design variable, since it determines partially

the reliability of the project on the application environment. If this temperature is very high, an entire new cooling system for the electronics board can be necessary, enhancing costs. Once defined a threshold, based on the design operational conditions, we can conceive a penalty function. This function would be responsible for taking into account the costs associated with maximum temperatures beyond the design established threshold, comprehending additional maintenance, cooling and environmental costs, for instance. In this case, we choose to use the following function to play this role:

$$c_{raise}^{Temp} = \begin{cases} 0 & , \text{ for } T_{max} < T_{threshold} \\ \mu(T_{max} - T_{threshold})^2 & , \text{ for } T_{max} \geq T_{threshold} \end{cases} \quad (3.5)$$

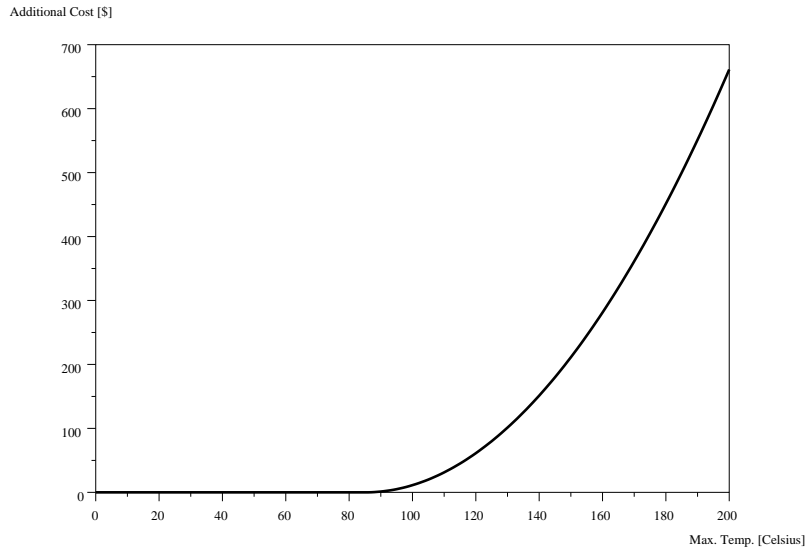


Figure 3.2: The penalty overheating cost function, for a threshold temperature of 85 Celsius.

Where c_{raise}^{Temp} is the cost [\$] associated with temperature raising T_{max} beyond the threshold $T_{threshold}$, and μ is an estimative of additional cost unit per Celsius degree [$\$/^{\circ}C^2$] (see Figure (3.2)). Unfortunately, even constructing (3.5) as a convex function of T_{max} , we remark that T_{max} is not smooth or continuous, considering the assignment variables x . But, this also means that the Pareto optimal solution for the combined problem is always between the following two bounds: the lower-bound, given by the placement problem optimal solution, and the upper-bound, given by the placement problem optimal solution added to the associated thermal cost component.

We can conceive now a performance guarantee heuristic to determine good solutions for the both point of views: thermal performance and placement cost. On the search for the optimal placement, we can examine the thermal cost component for each assignment solution, keeping the best upper bound. When the optimal placement solution is found, we just choose the lowest total cost (the best upper bound), combining placement and thermal cost components.

Defining the combined thermal-placement problem lower bound as the optimal solution for QAP ,

$$LB = QAP_{optimal} \quad (3.6)$$

and the combined thermal-placement problem upper bound as the optimal solution for QAP plus the associated overheating penalty,

$$UB = QAP_{optimal} + c_{raise}^{Temp} \quad (3.7)$$

we can then pick the best feasible solution found

$$BEST = (QAP + c_{raise}^{Temp})_{best}. \quad (3.8)$$

At this point we remark that the performance guarantee of a heuristic algorithm for a minimization problem is α ($\alpha \geq 1$) if the algorithm is guaranteed to deliver a solution φ whose value is at most α times the optimal value: $\varphi \leq \alpha\varphi^*$. Since we do not have the value of the global minimum for the combined thermal-placement problem (φ^*), we can only provide the following indirect relation:

$$LB \leq \varphi^*, BEST \leq \alpha LB, BEST \leq \alpha\varphi^* \text{ where } \alpha = UB/LB.$$

This is true because $c_{raise}^{Temp} \geq 0$ by construction, and it is now possible to define the optimality gap as:

$$GAP = \frac{BEST - LB}{LB}. \quad (3.9)$$

An illustration of this method is depicted in Figure (3.3).

3.3 Computational Experiments

3.3.1 Experiment Description

When designing our computational experiments, we are interested on making our test instances realistic. Following this objective, we can observe that the relationship between the system (the electronic board, in this case) and the application environment (external world connections, internal space restrictions, other interferences) has the same importance as the interrelations among the system components, being even more important sometimes.

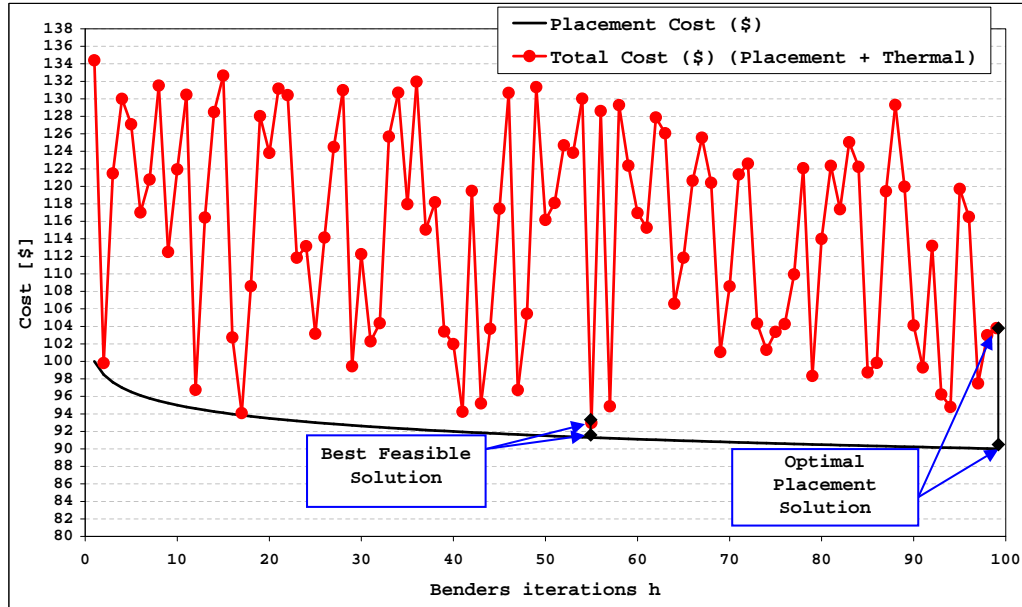


Figure 3.3: Evolution of bounds during the method execution.

In this context, we have chosen to establish linear installation costs $p = (\sum_{(k,i)} a_{ki})$ varying from near 1 to 4 times the magnitude of the quadratic cost component $q = (\sum_{(i,j)} \sum_{(k,l)} c_{ij} f_{ij}^{kl})$. These linear costs were generated randomly with the aid of the standard pseudo-random number generator implemented on the GNU C compiler GCC version 3.0. In fact, the addition of the linear installation costs was directly responsible for the efficiency of the solution procedure when dealing with larger instances, since this increases the master problem strength.

The set of test instances can be divided in two subsets. The first subset, composed by instances available in *QAPLIB* [29], with instance sizes from 6 to 36. The linear costs were generated randomly as described above. The name of the instances selected from *QAPLIB*, their size (number of variables), the mixed-integer linear program size (number of integer and continuous variables)

and the Finite Volume grid size are shown in Table 3.2. The second subset of test instances is also shown in Table 3.2, and it is composed by entirely randomly generated test instances, with sizes from 9 to 36. It is also possible to see in Table 3.2 the component disposition over the electronics board surface. The column for component distribution shows how the n components are being disposed over the surface board. It is necessary to point that the thermo-physical properties and the board dimension are not being changed from one instance to another. In this sense, when the instance size increases, we are just placing a larger number of smaller components.

Table 3.2: Test instances for computational experiments.

Name	Size	Integer Variables	Continuous Variables	Fin. Vol. grid	Component Distribution
nug06	6	36	1296	54	3×2
tail06	6	36	1296	54	3×2
nug08	8	64	4096	72	3×2
tail08	8	64	4096	72	4×2
esc08a	8	64	4096	72	4×2
esc08b	8	64	4096	72	4×2
esc08c	8	64	4096	72	4×2
esc08d	8	64	4096	72	4×2
esc08e	8	64	4096	72	4×2
esc08f	8	64	4096	72	4×2
tail09	9	81	6561	81	3×3
tail10	10	100	10000	90	5×2
lipa10a	10	100	10000	90	5×2
lipa10b	10	100	10000	90	5×2
rou10	10	100	10000	90	5×2
scr10	10	100	10000	90	5×2
nug12	12	144	20736	108	4×3
tail12a	12	144	20736	108	4×3
rou12	12	144	20736	108	4×3
scr12	12	144	20736	108	4×3
nug20	20	400	160000	180	5×4
ste36a	36	1296	1679616	324	6×6
ste36b	36	1296	1679616	324	6×6
rpqa9	9	81	6561	81	3×3
rpqa16	16	256	65536	144	4×4
rpqa25	25	625	390625	225	5×5
rpqa36	36	1296	1679616	324	6×6

The intensity of each component heat source is a relative inter-activity measurement, defined as having a direct reason with the energy/information flow matrix given on the instance under analysis, plus a small pseudo-random parcel. The heat sources were also adjusted to ensure physical consistency, resulting on temperature ranges typically observed in electronics equipments. As the size of the instance is increased, the heat sources are redistributed in such way that the same temperature ranges can be observed. We choose 85° Celsius as threshold, since we are designing an electronic board. The board under study is probably used as host for semi-conductive devices as microchips, and over this temperature, must experience some performance loss. Referential values for the heat source ranges and temperature threshold used here are available in [16] and [123]. The adopted value for the parameter μ in equation (3.5) was 5.

3.3.2 Numerical Results

Our computational experiments were carried out in a Sun Blade 100 Workstation equipped with a 500 MHz Ultra-SPARC processor and 1 Giga byte of RAM memory, running Solaris 5.8. The Benders decomposition algorithm was implemented in C++, using *ILOG CPLEX 7.0 Concert Technology* application programming interface, on the solution of the master problem. Table 3.3 shows the obtained results for the first set of tests, for the linear/quadratic cost ratio p/q kept between 1 and 2. This table lists instance name, instance size, number of Benders iterations h , linear/quadratic cost ratio p/q for the optimal solution, $Gap(\%)$ as defined by equation (3.9) and the execution time. Tables 3.4 and 3.5 points, respectively, to the second and the third sets of experiments. For all these tables, the entries marked by "*" could not be solved in 24 hours of computing.

Table 3.3: Results for computational experiments - first set.

Name	Size	h	Optimal p/q cost ratio	Gap (%)	Time [s]
nug06	6	9	1.25	0	10
tail06	6	8	0.95	0	11
nug08	8	12	1.13	8.3	120
tail08	8	10	1.49	2.5	117
esc08a	8	3	1.36	0	118
esc08b	8	5	1.08	1.5	103
esc08c	8	4	0.99	0.5	115
esc08d	8	4	1.13	4.3	106
esc08e	8	5	1.15	9.7	98
esc08f	8	4	1.33	6.4	123
tail09	9	20	1.18	12.1	305
tail10	10	16	1.45	10.5	419
lipa10a	10	30	1.34	5.4	383
lipa10b	10	31	1.24	0.7	372
rou10	10	32	1.19	8.2	385
scr10	10	33	1.15	11.8	369
nug12	12	37	1.45	33.1	521
tail12a	12	26	1.14	25.6	483
rou12	12	40	1.05	18.6	454
scr12	12	30	1.12	30.5	427
nug20	20	112	1.26	24.3	6019
ste36a	36	*	*	*	*
ste36b	36	*	*	*	*
rpqa9	9	42	0.98	12.9	358
rpqa16	16	105	1.61	15.4	464
rpqa25	25	148	1.18	16.1	3348
rpqa36	36	*	*	*	*

It is possible to note in these three tables that the gap between solutions can be responsible for a great amount of the total cost. Remembering that the heat source distribution is partially randomly generated, we can observe that there is no relation between the optimality gap and the size of the instance. This is also true for the ratio p/q . Otherwise, the number of Benders iterations and the solution time apparently reduces as the ratio p/q increases, as plotted in Figures (3.4) and (3.5) for the instances *nug12* and *nug20*. In order to compare the dif-

Table 3.4: Results for computational experiments - second set.

Name	Size	h	Optimal p/q cost ratio	Gap (%)	Time [s]
nug06	6	7	2.12	1.1	10
tail06	6	8	2.87	0.8	12
nug08	8	9	2.44	0.5	30
tail08	8	5	2.72	0.7	109
esc08a	8	3	2.6	2.1	32
esc08b	8	4	2.94	0.6	64
esc08c	8	2	2.55	0.9	86
esc08d	8	2	2.46	0.7	7
esc08e	8	2	2.24	1.7	42
esc08f	8	4	2.11	1.3	35
tail09	9	11	2.01	0.4	105
tail10	10	12	2.57	0.6	296
lipa10a	10	15	2.62	5.4	374
lipa10b	10	16	2.63	0.3	340
rou10	10	20	2.66	12.5	333
scr10	10	16	2.05	11.8	298
nug12	12	26	2.18	8.8	160
tail12a	12	26	2.82	14.9	282
rou12	12	22	2.64	11.3	351
scr12	12	13	2.45	4.7	187
nug20	20	60	2.85	8.3	981
ste36a	36	237	2.58	22.1	1876
ste36b	36	213	2.79	19.4	1381
rpqa9	9	19	2.47	2.7	358
rpqa16	16	54	2.23	12.3	464
rpqa25	25	74	2.83	21.1	821
rpqa36	36	348	2.76	28.7	2971

ference between only placement solutions and penalizing overheating solutions, Figures (3.6) and (3.7) show the temperature field over the board in both situations. They correspond to the best solution for the instance *ste36a* considering overheating cost, and the temperature field for optimal *QAP* solution of *ste36a*. As we can see, the method was capable to find a solution that has a total cost lower than optimal *QAP* solution. This solution designs a system that is colder than the first one.

3.4 Concluding Remarks

In this paper, an approach to solve the electronic board design problem, described as a quadratic assignment problem instance, has been presented. Another important quality solution criterion was under analysis: the maximal temperature over the board surface. Methods to study the thermal field behavior were discussed. The Benders decomposition algorithm was used as a solution technique to the proposed problem. The presented results report that overheating can respond for a great amount of the total cost, when considered.

It is interesting to remark that other quantities — depending on the selected application — could be used as secondary quality criteria for the solution. Also, depending on other design parameters, many of other kinds of information could

Table 3.5: Results for computational experiments - third set.

Name	Size	h	Optimal p/q cost ratio	Gap (%)	Time [s]
nug06	6	2	3.02	0.1	4
tail06	6	3	3.76	1.2	6
nug08	8	3	3.29	9.0	2
tail08	8	5	3.21	12.1	6
esc08a	8	4	3.92	10.1	5
esc08b	8	2	3.07	3.1	15
esc08c	8	3	3.69	3.1	10
esc08d	8	3	3.78	10.1	1
esc08e	8	3	3.43	5.2	23
esc08f	8	2	3.29	8.1	5
tail09	9	5	3.16	7.0	77
tail10	10	12	3.72	11.0	170
lipa10a	10	10	3.42	17.3	124
lipa10b	10	11	3.94	4.3	115
rou10	10	12	3.21	5.6	162
scr10	10	14	3.86	14.1	187
nug12	12	16	3.55	3.0	93
tail12a	12	12	3.48	2.2	105
rou12	12	14	3.42	21.2	261
scr12	12	8	3.84	12.1	119
nug20	20	13	3.26	27.3	195
ste36a	36	118	3.81	15.3	1379
ste36b	36	147	3.99	27.2	1130
rpqa9	9	8	3.27	5.4	14
rpqa16	16	34	3.11	17.2	35
rpqa25	25	53	3.75	21.2	379
rpqa36	36	242	3.22	20.1	1234

be considered for this particular problem: for instance, the average and minimal temperatures over the board.

For future work, it can be noted that a lot of real life problems can be viewed as instances of quadratic assignment problems and that it is sometimes possible to propose other quality criteria for the solution. Nowadays, this kind of solutions is valuable, since the environmental impact of any kind of large scale human implementation has becoming more important. It is desirable to have the lower transport cost for intermediate commodities in the design of a regional industrial complex, and to take into account the dispersion of heavy industrial atmospheric residues. There is no doubt that air quality is a great part of quality of life, and that environmental concerns must be in the top of the list of all modern industrial management. One could still deal with the noise level of machinery on the low plant layout design.

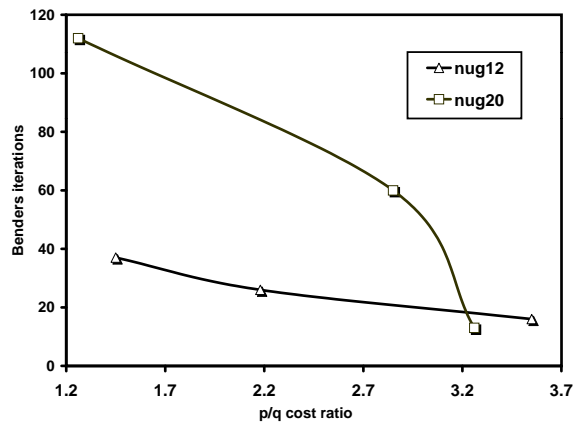


Figure 3.4: Number of Benders iterations versus p/q cost reason.

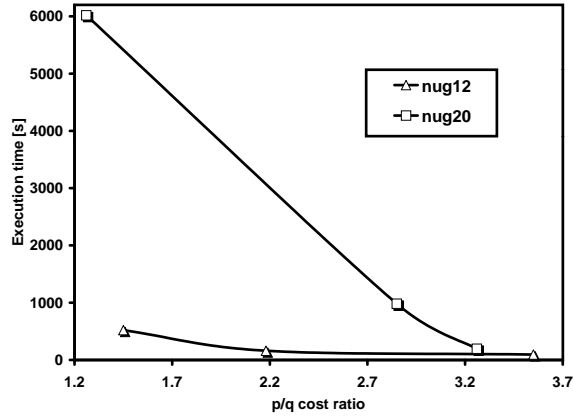


Figure 3.5: Execution time [s] versus p/q cost reason.

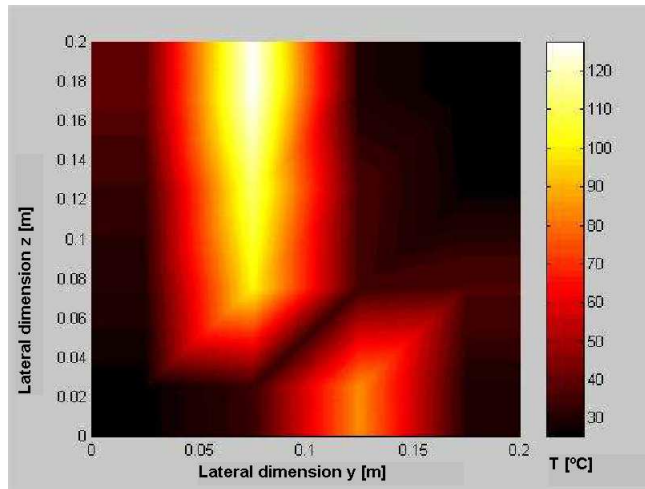


Figure 3.6: Temperature field for *ste36a* placement solution without overheating penalty.

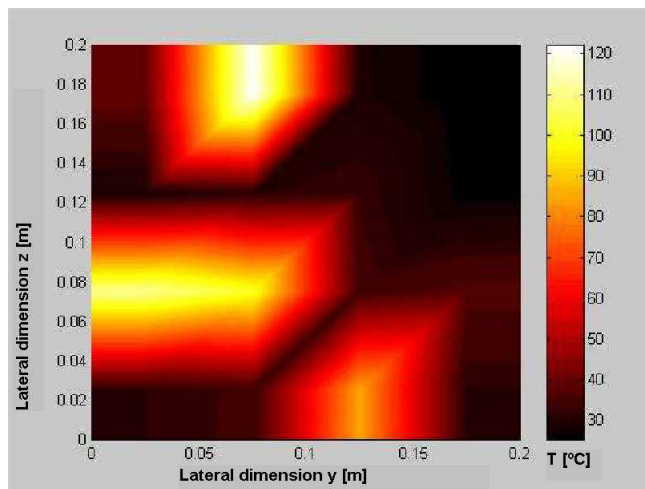


Figure 3.7: Temperature field for *ste36a* placement solution considering overheating penalty.

Chapter 4

The Local Access Network Design With Congestion Costs

4.1 Introduction

The tree network design problem consists of linking a supply node to its demand nodes to satisfy demands at minimal total cost. The problem may include *Steiner* or transshipment nodes. Each arc of the network has three associated costs: a variable operational cost depending on the flow through the arc, a nonlinear congestion cost that penalizes flows close to implicit capacities and a fixed cost for setting up the physical connection represented by the arc.

This problem can be viewed as a generalization of the Steiner tree problem on a directed graph [85]. In fact, if we neglect variable costs at the arcs we will have basically the Steiner problem, and in this sense we are treating a NP-hard problem, for which some computational strategies have been devised [87, 128, 67, 68, 75, 83]. On the other hand, if we neglect fixed costs on the arcs we have the single source transshipment problem, which can be solved easily [40].

As one can see, it is possible to use models for this problem to deal with any network flow problem which has a single source tree as optimal solution. The network design problem associated with centralized computer networks and the multi-party multicast tree construction problem are good examples. The last one has been treated with the aid of heuristics [69], but on the two versions treated in the literature, the Single Source Tree Networks - where we have a true root of the multi-party multicasting tree - and the Core Based Tree Networks - where a single node, the core, is chosen to play a role as the tree root - it is possible to accomplish the data to make the model faithful to the nature of the problem. To make a survey on correlated research work, we refer to [58], [122]

and [72]. Congested computer networks are studied in the work of Ferreira and Luna [42].

The provision of multi-point connections is one of most important services that will be required in future broadband communication networks which support distributed multimedia applications. Multimedia video-conferencing applications, for example, requires that audio and video be transmitted to multiple conference participants simultaneously. This requires that an efficient multicast capability be provided by the underlying network. An illustration of the problem is depicted in Figure 4.1.

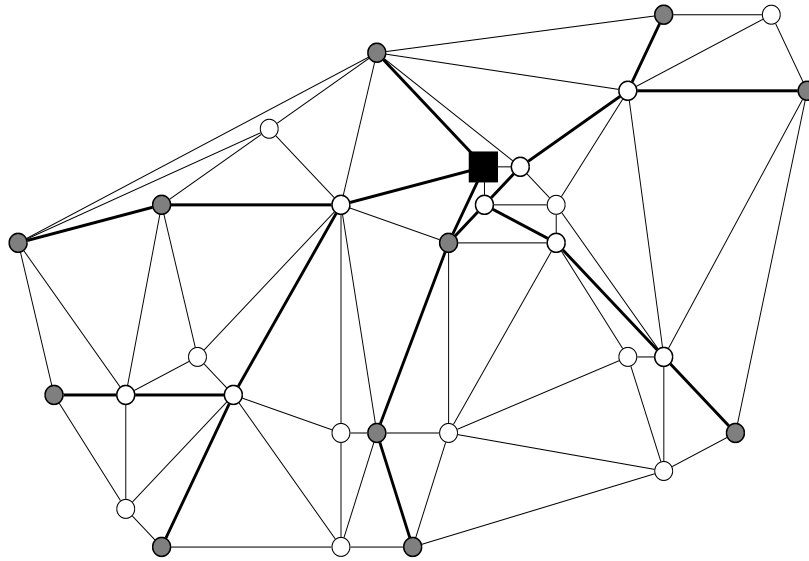


Figure 4.1: The tree network design problem.

An alternative class of tree network design problems is related to the minimal spanning tree, where a central node has to be linked to all the remainder nodes in the network at minimal cost. These problems have been studied by authors such as Gavish [47, 48, 49] and Gouveia [57]. This class of problems includes the capacitated and degree constrained minimal spanning tree problems, and some routing and scheduling problems. In most of these problems all the terminals have identical characteristics, and a single line type can be used for connecting the terminals. When the network has heterogeneous terminals, in the sense that they generate different amounts of traffic, and line costs that are dependent of the line capacities we have the Telepak problem. In [48] Gavish proposes one-flow formulations for these problems and applies Lagrangean

relaxation with the sub-gradient optimization procedure to solve the degree constrained minimal spanning tree problem, also solving the capacitated minimal spanning tree problem by Benders decomposition. In [49] a new formulation of the capacitated minimal spanning tree problem is presented as a zero-one integer programming problem and a combination of a sub-gradient optimization procedure and an augmented Lagrangean-based procedure is used to generate tight lower bounds. The original paper of Rothfarb and Goldstein [119] formulates the Telepak problem as a single commodity flow problem, and Gavish [48] shows that it can also be considered as a capacitated minimal spanning tree problem. Several heuristics have been developed for the problem by Goldstein [56] and Chandy and Russel [35]. Gavish [50] presented a mixed-integer mathematical programming formulation of the problem for a general cost structure. Hochbaum and Segev [63] propose two Lagrangean relaxations for the problem and apply the sub-gradient optimization algorithm to approximate the best Lagrangean multipliers.

In the context of telecommunication systems the local access network design problem corresponds to setting up a topology on an urban street network that minimizes the total cost of cables and underground piping infrastructure necessary to link a switching center and its subscribers. Each subscriber group has a known demand and it is supposed that the switching center is able to supply all subscribers demands. For each arc, the fixed cost is represented by the underground infrastructure and the flow dependent cost is represented by the cables to be installed in each arc. An optimal solution to this problem is a tree with root at the switching center. Local access networks with aerial cables can also be dealt with this formulation, and the fixed cost in such a case can include pole rents.

In a modern framework of applications, such as the multi-party multicast tree construction cited above, the fixed cost can be seen as a bandwidth leasing cost, due to the allocation of large capacity links used by multicasting participants. The variable costs, in this case, depends on data flow level and congestion level observed during the operation. The output is a low cost spanning tree representing the final computer conference configuration.

In Section 4.2 a mathematical programming formulation of the problem will be presented. In Section 4.3 Benders decomposition is applied to the problem. The proposed solution was implemented and experimental results are reported in Section 4.4. Section 4.5 closes this chapter with final remarks and conclusions.

4.2 Multi-commodity Flow Formulation

Consider a directed connected graph $G(V, E)$, where V denotes the set of nodes, and E is a collection of arcs representing pairs of nodes between which a direct transmission link can be installed. Suppose we have an origin node o (the local hub or switching center) that must be linked to a number of $|K|$ demand nodes (offices), each of them with a commodity flow requirement of d_k where $k \in K$ and $K \subseteq V$. With appropriate structural, operational, leasing and congestion

costs, the problem is to find a minimal cost arborescence that links the switching center to all the spatially distributed offices. It should be stressed that all flows originate at a *root* node, i.e. the switching center or the server in a computer network.

4.2.1 Variables and Parameters

Define the variables:

$$x_{ij} = \begin{cases} 1 & \text{if a directed transmission link is placed in arc } (i, j) \\ 0 & \text{if not;} \end{cases}$$

f_{ijk} : flow destined to demand node k , passing through arc (i, j) ;

g_{ij} : total flow of all commodities passing through arc (i, j) .

And also define the parameters:

b_{ij} : fixed (structural) cost to install a directed transmission link in arc (i, j) ; we suppose $b_{ij} = \beta d_{ij}$ where d_{ij} is the distance (in meters) between i and j , and β is the linkage structural cost per meter.

c_{ijk} : variable (operational) cost to transmit one unit of commodity k through arc (i, j) ; we suppose $c_{ijk} = \gamma^k d_{ij}$, $\forall k \in K$.

The model allows variable costs to be dependent on both the commodity and the arc. If the variable cost is independent from the commodity, we can make $\gamma^k = \gamma$, $\forall k$. We also assume that, for each arc (i, j) , is given an increasing function $\tau_{ij}(g_{ij})$ of the total flow passing through the arc. The leasing and congestion cost function of the model is assumed to be separable with respect to arcs, and each parcel $\tau_{ij}(g_{ij})$ is intended to integrate quality of service and expansion costs on arc (i, j) . Quality of service is typically an increasing function which measures congestion on the arc where commodities are considered as competitive users of a limited resource.

4.2.2 Mixed Integer Nonlinear Program

The mathematical model M is:

$$\min \sum_{(i,j) \in E} [b_{ij}x_{ij} + \tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk}f_{ijk}] \quad (4.1)$$

subject to:

$$\sum_{k \in K} f_{ijk} - g_{ij} \leq 0, \quad \forall (i, j) \in E \quad (4.2)$$

$$- \sum_{(o,j) \in E} f_{ojk} = -d_k, \quad \text{for node } o \text{ and } \forall k \in K \quad (4.3)$$

$$\sum_{(i,k) \in E} f_{ikk} = d_k, \quad \forall k \in K \quad (4.4)$$

$$\sum_{(i,j) \in E} f_{ijk} - \sum_{(j,l) \in E} f_{jlk} = 0, \quad \forall j \in V - \{o\} \text{ and } j \neq k \text{ and } \forall k \in K \quad (4.5)$$

$$f_{ijk} \leq d_k x_{ij}, \quad \forall (i,j) \in E \text{ and } \forall k \in K \quad (4.6)$$

$$f_{ijk} \geq 0, \quad \forall (i,j) \in E \text{ and } \forall k \in K \quad (4.7)$$

$$g_{ij} \geq 0, \quad \forall (i,j) \in E \quad (4.8)$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \quad (4.9)$$

The objective function (4.1) has three terms: the first one accounts for the total fixed cost of activating the arcs; the second term is a measure of leasing and congestion costs related with the use of the arcs; and the third term is associated with the operational cost of sending flows of all commodities from the source node to the demand nodes through the arcs. Constraints (4.2) account for the total flow of all the commodities passing through each arc (i, j) . Constraints (4.3) ensure that the total flow of commodity k that originates from the source node is equal to the demand of node k and constraints (4.4) impose that the total flow of commodity k arriving at demand node k is equal to its demand, d_k . Constraints (4.5) ensure, for each commodity, the flow conservation for each Steiner node of that commodity. The x and f coupling constraints (4.6) ensure that no flow is allowed on arc (i, j) unless the fixed cost b_{ij} is paid. The fact that the flow of any commodity through an arc is not negative is guaranteed by constraints (4.7). Finally, constraints (4.9) state that the variables x_{ij} are binary.

4.2.3 Theoretical Properties of the Linear and the Concave Versions

It is important to note that the continuous relaxation of model M , that we call here $MLin$, generates a *quasi-integral* polytope. A *quasi-integral* polytope is one in which the edges of the convex hull of its integer points are also edges of the polytope itself [62]. This means that there exists a path through extreme integer points that finds the optimal integer solution, if it exists.

Another important theoretical property is that, for $\tau_{ij}(g_{ij}) = 0$, $\beta = 0$ and $d_k = 1, \forall k \in K$ the model is reduced to the multi-commodity flow formulation of the Steiner problem in directed graphs, as presented by Maculan and others [38, 129, 88, 55]. As a result, the strong formulation given by the objective function (4.1) and constraints (4.3) to (4.9) also includes as particular case the linear programming formulation for the shortest directed spanning tree problem, also introduced by Maculan [86]. All these theoretical properties can help to understand why, in many instances of this particular case, a linear programming relaxation of model M can automatically lead to optimal integral solutions.

Another related result concerns to the single source concave cost flow problem, where $\tau_{ij}(g_{ij})$ is a concave cost function of the flow g_{ij} passing through the arc (i, j) , with $b_{ij} = 0$ and $c_{ijk} = 0$ for every commodity k . For this particular problem, it can be shown that a tree topology is associated to an optimal solution for the problem [114].

4.2.4 Convexification of Leasing and Congestion Costs

As we can see in [51], to use the generalized Benders decomposition procedure, the Benders subproblem must be a convex one. The function chosen here to represent congestion costs is the well known Kleinrock's delay function, presented by Gerla and Kleinrock in [54]. The possibility of capacity expansion in each network link is accomplished with the aid of leasing costs for each capacity, remembering that the installation cost of first level of capacity is treated as a fixed cost. In the framework of modern telecommunications systems or Internet multi-party multicasting applications, the capacity expansion of a link can be seen as an upgrade of the bandwidth which can be made available for a group of customers.

To manage this important feature together with the necessity of convexity in the Benders subproblem, we adopt here a convexification technique proposed in [84]. The cost function is assumed to be separable with respect to arcs and is intended to integrate quality of service (on the arc) and expansion cost. Quality of service is typically an increasing function which measures congestion on the arc where commodities are considered as competitive users of a limited resource.

Let π_{ij}^1 and π_{ij}^2 be the fixed costs of expanding the arc, to respectively, capacities q_{ij}^1 and q_{ij}^2 , $\tau_{ij}^0(g_{ij})$, $\tau_{ij}^1(g_{ij})$ and $\tau_{ij}^2(g_{ij})$ being the correspondent congestion cost functions before and after expansions. Then, the integrated arc cost function is defined on $[0, q_{ij}^2]$ by:

$$\sigma_{ij}(g_{ij}) = \min\{\tau_{ij}^0(g_{ij}), \tau_{ij}^1(g_{ij}) + \pi_{ij}^1, \tau_{ij}^2(g_{ij}) + \pi_{ij}^2\} \quad (4.10)$$

The univariate congestion functions $\tau_{ij}^0(g_{ij})$, $\tau_{ij}^1(g_{ij})$ and $\tau_{ij}^2(g_{ij})$ are increasing, proper convex, differentiable functions and depend, respectively, on the arc capacities q_{ij}^0 , q_{ij}^1 and q_{ij}^2 (such that $q_{ij}^0 < q_{ij}^1 < q_{ij}^2$). The derivatives $\tau_{ij}^{0'}$ (g_{ij}), $\tau_{ij}^{1'}$ (g_{ij}) and $\tau_{ij}^{2'}$ (g_{ij}) are increasing. We assume also the following hypothesis on these functions:

$$\tau_{ij}^0(0) = \tau_{ij}^1(0) = \tau_{ij}^2(0) = 0 \quad (4.11)$$

$$\tau_{ij}^2(g_{ij}) < \tau_{ij}^1(g_{ij}) < \tau_{ij}^0(g_{ij}), \quad \forall g_{ij} \in (0, q_{ij}^2) \quad (4.12)$$

$$\tau_{ij}^{2'}(g_{ij}) < \tau_{ij}^{1'}(g_{ij}) < \tau_{ij}^{0'}(g_{ij}), \quad \forall g_{ij} \in (0, q_{ij}^2) \quad (4.13)$$

$$\tau_{ij}^{0'}(q_{ij}^0) \geq \Omega \text{ and } \tau_{ij}^{1'}(q_{ij}^1) \geq \Omega \text{ and } \tau_{ij}^{2'}(q_{ij}^2) \geq \Omega \quad (4.14)$$

Observations:

1. Relation (4.11) trivially says that a null congestion cost results from a null flow.
2. Relation (4.12) means that the congestion cost decreases when capacity is added to an arc.
3. Relation (4.13) means that the marginal cost in an arc also decreases with the addition of capacity.

4. As the congestion increases with the flow, we force the derivative at the saturation level to be greater than a given high value Ω in relation (4.14).

Figure 4.2 illustrates the integrated arc cost function $\sigma_{ij}(g_{ij})$, as it is constructed on the basis of the congestion functions τ_{ij}^0 and τ_{ij}^1 and τ_{ij}^2 and of the fixed expansion cost π_{ij}^1 and π_{ij}^2 . Remark that, from the properties (4.11-4.12), when g_{ij} increases from 0, initially $\sigma_{ij}(g_{ij})$ assumes the value of $\tau_{ij}^0(g_{ij})$. Because of (4.13) the difference between the congestion costs $\tau_{ij}^0(g_{ij})$ and $\tau_{ij}^1(g_{ij})$ increases until reaching the value π_{ij}^1 , when a break-even point $g_{ij} = \xi_1$ appears with neither loss nor gain for expanding capacity. For $g_{ij} > \xi_1$, it becomes more economical to expand the arc capacity and the integrated function $\sigma_{ij}(g_{ij})$ assumes the value of $\tau_{ij}^1(g_{ij}) + \pi_{ij}^1$, and for $g_{ij} > \xi_2$, it is more economical to expand arc capacity for q_{ij}^2 and the function assumes the value of $\tau_{ij}^2(g_{ij}) + \pi_{ij}^2$. The values of the break-even points ξ_1 and ξ_2 are implicit in the solution of the equations $\tau_{ij}^0(\xi_1) = \tau_{ij}^1(\xi_1) + \pi_{ij}^1$ and $\tau_{ij}^1(\xi_2) = \tau_{ij}^2(\xi_2) + \pi_{ij}^2$. Figure 4.2 also shows the convex hull function $\tau_{ij}(g_{ij})$ and the tangent points g_{ij0} and g_{ij2} of the unique supporting line of σ_{ij} on $[0, q_{ij}^2]$, in such a way that $\tau_{ij}^{0'}(g_{ij0}) = \tau_{ij}^{2'}(g_{ij2})$. The idea is that, for each specific class of convex functions satisfying properties (4.11-4.14), we can determine g_{ij0} , ξ_1 , ξ_2 and g_{ij2} , thus characterizing explicitly the convex hull, $\tau_{ij}(g_{ij})$, of the integrated function of congestion and expansion costs, $\sigma_{ij}(g_{ij})$. It results that problem *MBend* is a convex program for a fixed network topology, given by x , in such a way that we can conceive a Benders decomposition strategy to solve it.

4.3 Benders Decomposition of the Problem

Benders partitioning method was published in 1962 [17] and was initially developed to solve mixed integer programming problems. The computational success of the method to solve large scale multi-commodity distribution system design models has been confirmed since the pioneering paper of Geoffrion and Graves [53]. Magnanti and Wong [89] proposed a methodology to improve Benders decomposition algorithm performance when applied to solve mixed-integer programs. They introduced a technique for accelerating the algorithm convergence and developed a theory that distinguishes "good" formulations amongst different, but equivalent, possible formulations. In [90] Benders decomposition is applied to solve the uncapacitated network design problem and adapted to be as efficient as possible, when solving problems with undirected arcs. Now we specialize the method to cope with nonlinear congestion costs and discrete capacity assignment. This is done as an application of generalized Benders decomposition [51] for an adequate variant of model M .

4.3.1 Problem Manipulations

Benders partitioning method essentially relies on a *projection* problem manipulation, that is then followed by the solution strategies of dualization, outer lin-

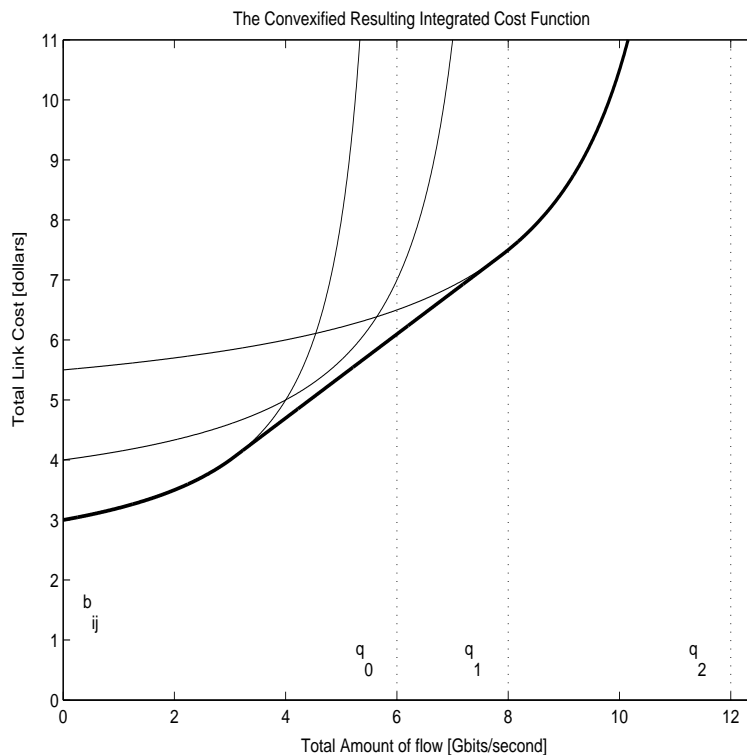


Figure 4.2: An example of convexified integrated leasing and congestion cost function.

earization and relaxation. From the viewpoint of mathematical programming we can conceive a projection of problem M onto the space of the *topological* variables x , thus resulting in the following implicit problem to be solved at a superior level:

$$\min_{x \in X} \sum_{(i,j) \in E} b_{ij} x_{ij} + v(x) \quad (4.15)$$

where $X = \{x \mid \text{for } x \text{ fixed there exist feasible flows satisfying (4.2) - (4.8)}\}$ and where $v(x)$ is calculated by the following problem to be solved at an inferior level:

$$v(x) = \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk}] \text{ subject to (4.6) for } x \text{ fixed} \quad (4.16)$$

where $G = \{(f, g) \mid f \geq 0 \text{ and } g \geq 0 \text{ satisfying (4.2) - (4.5)}\}$.

The flow feasibility requirement related to a topological variable $x \in X$ implies that the components for which $x_{ij} = 1$ include an arborescence rooted

at the origin o and destined to every demand node $k \in K$. Thus, there is no need for further feasibility constraints on the domain of the projected problem (4.15), and the existence of the minimum in the subproblem (4.16) is assured since we are minimizing a convex function in a nonempty compact set.

Since the subproblem has a convex differentiable objective function and linear constraints, the Karush-Kuhn-Tucker conditions are necessary and sufficient for optimality and the problem is amenable to dualization techniques [51]. With an associated vector $\lambda \geq 0$ of dual variables, the idea is to dualize the subproblem with respect to the coupling constraints (4.6). Since there is no duality gap, for any $x \in X$ the optimal value of the subproblem 4.16 can be given by

$$v(x) = \max_{\lambda \geq 0} \left[\min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk} + \sum_{k \in K} \lambda_{ijk} (f_{ijk} - d_k x_{ij})] \right] \quad (4.17)$$

or

$$\begin{aligned} v(x) = \max_{\lambda \geq 0} \left[\sum_{(i,j) \in E} \sum_{k \in K} -\lambda_{ijk} d_k x_{ij} + \right. & (4.18) \\ \left. \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} (c_{ijk} + \lambda_{ijk}) f_{ijk}] \right] \end{aligned}$$

The whole problem (4.15) is then equivalent to

$$\begin{aligned} \min_{x \in X} \left\{ \sum_{(i,j) \in E} b_{ij} x_{ij} + \max_{\lambda \geq 0} \left[\sum_{(i,j) \in E} \sum_{k \in K} -\lambda_{ijk} d_k x_{ij} + \right. \right. & (4.19) \\ \left. \left. \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} (c_{ijk} + \lambda_{ijk}) f_{ijk}] \right] \right\} \end{aligned}$$

or, using the fact that a supremum is the least upper bound, problem (4.1) to (4.9) is equivalent to the master problem

$$\min_{t, x \in X} \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.20)$$

subject to:

$$\begin{aligned} t \geq \sum_{(i,j) \in E} \sum_{k \in K} -\lambda_{ijk} d_k x_{ij} + & (4.21) \\ \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} (c_{ijk} + \lambda_{ijk}) f_{ijk}] \text{ for all } \lambda \geq 0 \end{aligned}$$

Generalized Benders decomposition solves problem (4.20)-(4.22) by the strategy of relaxation, i. e., ignoring all but a few constraints (4.22). If at a certain

cycle h the subproblem has been solved for a given network design x^h and the optimal multiplier vector λ^h has been recovered, then from (4.19) the optimal value $v(x^h)$ occurs for $\lambda = \lambda^h$ and is given by

$$v(x^h) = - \sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k x_{ij}^h + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} (c_{ijk} + \lambda_{ijk}^h) f_{ijk}] \quad (4.22)$$

From (4.22) it follows that, associated with λ^h , one has the constraint

$$t \geq - \sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k x_{ij}^h + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} (c_{ijk} + \lambda_{ijk}^h) f_{ijk}] \quad (4.23)$$

and using the value of the minimum given by (4.23) it results the following cut based on x^h and λ^h

$$t \geq v(x^h) + \sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k (x_{ij}^h - x_{ij}) \quad (4.24)$$

We analyze now the subproblem to provide further detail on the choices made.

4.3.2 Subproblems

For a fixed arborescence A^h , associated with the vector x^h , the computation of a minimal cost flow $v(x^h)$ can be separated in a series of trivial network flow problems. Let C_{ok}^h be the path, from the source node to the demand node k , that has been defined by the master problem of iteration h . The subproblem to be solved is

$$\min_{g \geq 0, f \geq 0} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk}] \quad (4.25)$$

subject to the coupling arc flow constraints (4.2), that is

$$\sum_{k \in K} f_{ijk} - g_{ij} \leq 0, \quad \forall (i,j) \in E$$

and the constraints (4.3) to (4.6) for a fixed binary vector $x = x^h$.

Since $\tau_{ij}(g_{ij})$ is an increasing function, constraints (4.2) are satisfied with strict equality in an optimal solution. A unique flow $f_{ijk} = d_k$ can be assigned to every arc that belongs to the path C_{ok}^h from the source node o to the demand node k , thus resulting in a unique optimal solution (f^h, g^h) associated with the

arborescence x^h . The construction of an associated optimal multiplier vector can be started with the dualization of the subproblem with respect to the coupling constraints (4.2). With the correspondent dual variables $\theta \geq 0$, the optimal value $v(x^h)$ can be computed as

$$v(x^h) = \max_{\theta \geq 0} d(\theta) \quad (4.26)$$

where the dual function $d(\theta)$ is evaluated inducing the inherent separability of each commodity flows, since

$$d(\theta) = \min_{g \geq 0, f \geq 0} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk}] + \sum_{(i,j) \in E} \theta (\sum_{k \in K} f_{ijk} - g_{ij})$$

and so

$$d(\theta) = \sum_{k \in K} \min_{f_k \geq 0} \sum_{(i,j) \in E} (c_{ijk} + \theta_{ij}) f_{ijk} + \sum_{(i,j) \in E} \min_{g_{ij} \geq 0} (\tau_{ij}(g_{ij}) - \theta_{ij} g_{ij}) \quad (4.27)$$

where each f_k refers to the vector of commodity k flows that is feasible in the correspondent constraints (4.3) to (4.6) for $x = x^h$.

We remark that, for an optimal solution (f^h, g^h) for the primal problem (4.25)-(4.2), an associated optimal solution θ^h for the dual problem (4.26) should minimize for each $g_{ij} \in E$ the corresponding parcel of the Lagrangean function in (4.27), what implies

$$\theta_{ij}^h = \tau'(g_{ij}^h) \quad \forall (i, j) \in E \quad (4.28)$$

As a consequence of fixing this unique optimal vector θ^h , we can now state in detail the primal-dual linear programming pair to be solved separately for each commodity $k \in K$ for any given x^h .

Primal subproblem for commodity k when $x = x^h$

$$\min \sum_{(i,j) \in E} (c_{ijk} + \theta_{ij}^h) f_{ijk}^h \quad (4.29)$$

subject to:

$$- \sum_{(o,j) \in E} f_{ojk}^h = -d_k \quad \text{for the root } o \quad (4.30)$$

$$\sum_{(i,k) \in E} f_{ik}^h = d_k \quad (4.31)$$

$$\sum_{(i,j) \in E} f_{ijk}^h - \sum_{(j,l) \in E} f_{jlk}^h = 0 \quad \forall j \in V - o \text{ and } j \neq k \quad (4.32)$$

$$-f_{ijk}^h \geq -d_k x_{ij}^h \quad \forall (i, j) \in E \quad (4.33)$$

$$f_{ijk}^h \geq 0 \quad \forall (i, j) \in E \quad (4.34)$$

The trivial and unique solution of the problem is:

$$f_{ijk}^h = \begin{cases} d_k & \text{if } (i, j) \in C_{ok}^h \subseteq A^h \\ 0 & \text{otherwise} \end{cases} \quad (4.35)$$

Dual subproblem for commodity k when $x = x^h$

The dual problem associated with the subproblem given by the objective function (4.29) and constraints (4.30), (4.31), (4.32), (4.33) and (4.34) is:

$$\max_{p^h, \lambda^h \geq 0} d_k(p_{kk}^h - p_{ok}^h - \sum_{(i,j) \in E} x_{ij}^h \lambda_{ijk}^h) \quad (4.36)$$

subject to:

$$p_{jk}^h - p_{ik}^h - \lambda_{ijk}^h \leq c_{ijk} + \theta_{ij}^h \quad \forall (i, j) \in E \quad (4.37)$$

This dual problem has many feasible solutions, contrarily to the primal problem that has a unique trivial solution. Since $f_{ijk}^h = d_k > 0 \forall (i, j) \in C_{ok}^h \subseteq A^h$ we have from the complementary slackness condition that:

$$p_{jk}^h - p_{ik}^h - \lambda_{ijk}^h = c_{ijk} + \theta_{ij}^h \quad \forall (i, j) \in C_{ok}^h \subset A^h \quad (4.38)$$

in such a way that we can construct, associated with the primal solution x^h , the following dual feasible solution:

$$p_{ok}^h = 0 \quad \forall k \in K, \text{ for the origin node } o \quad (4.39)$$

$$p_{jk}^h = p_{ik}^h + c_{ijk} + \theta_{ij}^h, \quad \forall (i, j) \in C_{ok}^h \subset A^h \quad (4.40)$$

$$p_{ik}^h = p_{ik}^0, \quad \forall i \in V - V^h \quad (4.41)$$

$$\lambda_{ijk}^h = 0, \quad \forall (i, j) \in C_{ok}^h \subset A^h \quad (4.42)$$

$$\lambda_{ijk}^h = p_{jk}^h - p_{ik}^h - c_{ijk} - \theta_{ij}^h, \quad \forall (i, j) \in E - A^h \quad (4.43)$$

$$\text{such that } p_{jk}^h - p_{ik}^h > c_{ijk} + \theta_{ij}^h$$

$$\lambda_{ijk}^h = 0, \quad \forall (i, j) \in E - A^h \text{ such that } p_{jk}^h - p_{ik}^h \leq c_{ijk} \quad (4.44)$$

The systematic evaluation of the dual variables with meaningful commodity values is a clue for an efficient implementation. Here the two series of dual variables can be interpreted as price information. Each variable p_{ik}^h represents the price of establishing communication k ($k \in K$) from the origin node o until node i ($i \in V$) in iteration h ($h = 1 \dots H$). On the other hand, each variable λ_{ijk}^h gives for commodity k the value of an additional unit of capacity at arc $(i, j) \in E$. The dual variable λ_{ijk}^h evaluates for commodity k the maximal reduction in the operational and leasing and congestion costs that could be gained with the introduction of arc (i, j) in the solution. In the case of transportation systems,

it can also be understood as a tax to be paid with the use of arc (i, j) in order to maintain the distribution agents with no positive profit. Remark that the dual solution set (4.37) represents spatial prices for which there is no positive profit for any distribution agent that pays the cost $c_{ijk} + \theta_{ij}^h$ to ship commodity k through arc (i, j) .

4.3.3 Master Problem

With the objective of producing a feasible solution from the master problem of the Benders decomposition method we have added some redundant constraints to the model M . We call this extended model $MBend$. The new constraints try to impose that the arcs implied by a solution to the master problem constitute a tree from the origin node o to all the demand nodes $k \in K$. Nevertheless, these constraints alone are not sufficient to guarantee this. For some problem instances, a solution to the master problem can imply cycles in the generated topology and our algorithm treats this special situation accordingly. The redundant constraints are the following:

$$\sum_{(o,j) \in E} x_{oj} \geq 1, \quad \text{for node } o \quad (4.45)$$

$$\sum_{(i,k) \in E} x_{ik} = 1, \quad \forall k \in K \quad (4.46)$$

$$\sum_{(l,j) \in E} x_{lj} - \sum_{(i,l) \in E} x_{il} \geq 0, \quad \forall l \in V - K - o \quad (4.47)$$

$$\sum_{(i,l) \in E} x_{il} \geq \frac{\sum_{(l,j) \in E} x_{lj}}{\sum_{(l,j) \in E} 1}, \quad \forall l \in V - K - o \quad (4.48)$$

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i, j) \in E \quad (4.49)$$

Constraint (4.45) imposes that at least one arc leaves the origin node. Constraints (4.46) establish that only one arc arrives at each demand node. Constraints (4.47) ensure that the number of arcs leaving a Steiner node is not smaller than the number of arcs arriving at it. Constraints (4.48) express the fact that if at least one arc leaves node l , then at least one arc enters node l . Constraints (4.49) avoid the occurrence of cycles involving two arcs.

The master problem consists of searching for x variables with the following objective function:

$$\min \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.50)$$

subject to the Benders cut constraints

$$t \geq v(x^h) - \sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k x_{ij} \quad h = 0, 1, 2, \dots, H \quad (4.51)$$

and the constraints (4.45), (4.46), (4.47), (4.48), (4.49) and (4.9).

We remark that the Benders cut constraint (4.51) results from (4.24) and from the fact that $\sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k x_{ij}^h = 0$, since, by construction, at any Benders cycle h , for any arc $i, j \in E$, either $x_{ij}^h = 0$ or, from (4.42), $\lambda_{ijk}^h = 0$ for $x_{ij}^h = 1$. The parameter $H + 1$ evaluates the number of cuts that can be taken into account at each Benders iteration. For given h and k the number of constraints (4.51) provide a lower bound on the cost of the flow that leaves the origin node to the demand node k . The variable t that appears in the objective function (4.50) is the best known lower bound on the sum of operational cost with leasing and congestion costs.

4.3.4 Algorithm

The implemented Benders decomposition algorithm is presented below. First the shortest path from the origin node to all the demand nodes and the minimum spanning tree of a reduced graph are computed. These two feasible solutions are used to initialize the dual variables. Next, in each cycle, one master problem and a series of subproblems are solved until the difference between the lower and the upper bound becomes small enough, for a given tolerance parameter. The steps of the algorithm are the following:

1. Use Dijkstra's algorithm to find the shortest path from the origin o to every node of the network. Let E^0 be the arcs of the arborescence that contains the shortest paths to all the nodes and let $T(V^0, A^0)$ be the corresponding arborescence that links the origin o to all demand nodes $k \in K$ ($x_{ij}^0 = 1 \forall (i, j) \in A^0$ and $x_{ij}^0 = 0 \forall (i, j) \in E - A^0$). Make

$$\begin{aligned} p_{ok}^0 &= 0 \quad \forall k \in K, \text{ for the origin node } \{o\} \\ p_{jk}^0 &= p_{ik}^0 + c_{ijk} + \theta_{ij}^0 \quad \forall (i, j) \in A^0, \forall k \in K \\ p_{jk}^0 &= p_{ik}^0 + c_{ijk} \quad \forall (i, j) \in E^0 - A^0, \forall k \in K \\ \lambda_{ijk}^0 &= 0 \quad \forall (i, j) \in E^0, \forall k \in K \end{aligned}$$

Compute the cost associated with $T(V^0, A^0)$ (the sum of the fixed cost of the arcs in A^0 plus the sum of the variable costs for sending the flow requirement of each commodity k from the origin o to the demand node k). This value gives an initial upper bound, $UB = \sum_{(i,j) \in A^0} b_{ij} x_{ij}^0 + \sum_{k \in K} d_k p_{kk}^0$, and (x^0, f^0) is an incumbent solution. Also, the shortest paths solution provides the minimal total variable cost among all possible arborescences, and thus we can use it to initialize a lower bound, $LB = \sum_{k \in K} d_k p_{kk}^0$.

2. Use Prim's algorithm to find a minimal spanning tree in a reduced graph that contains only the origin and the demand nodes $k \in K$. This reduced graph is constructed as proposed by Mehlhorn in [95] using a single shortest-path computation. Let $T(V^1, A^1)$ be the associated Steiner arborescence, that is contained in the original graph $G(V, E)$, and that links the origin o to all demand nodes $k \in K$ ($x_{ij}^1 = 1 \forall (i, j) \in A^1$ and $x_{ij}^1 = 0 \forall (i, j) \in E - A^1$). C_{ok}^1 is the set of the arcs in the path from the origin to the demand node k through A^1 . Set the iteration counter $h = 1$.
3. Compute the values of the dual variables as indicated by the equations (4.39)-(4.44). A new value for the upper bound is calculated and if this value is less than the current upper bound then the current upper bound is updated. Add a new cut to the master problem.
4. Solve the master problem. It provides a lower bound for the problem. If the upper bound is close enough to the lower bound, for the given tolerance, then **stop**.
5. Solve the subproblem. To solve it, initially verify if the arcs selected in the master problem imply an arborescence from the origin to all demand nodes. If yes, set $h = h + 1$, let $T(V^h, A^h)$ be the arborescence that links the origin o to all demand nodes $k \in K$, contained in the original graph $G(V, E)$, let C_{ok}^h be the set of the arcs in the path from the origin to the demand node k through A^h , and go to step 3. Else, the topological solution of the master problem is infeasible in the subproblem in the sense that it generates a cycle. In this case, the cycle is identified and a constraints to avoid it is added to the master problem model and no new lower bound is generated. To identify the cycle in the path from the origin to a demand node, a backward search from the demand node is executed until a node is repeated. Let n be this node. A cycle avoiding constraint is generated considering the arcs from the demand node to n in both directions. The sum of the variables x_{ij} corresponding to these arcs must not be greater than the number of arcs from the demand node to n (considering only one direction of the arcs and the second time that n is found). After the cycle avoiding constraints are added, the master problem must again be solved, then go to step 4.

4.3.5 Avoiding Cycles

As it was explained earlier, the model *MBend* uses the constraints (4.45 - 4.49) to accomplish the necessity of computing a feasible solution at the higher (master problem) level which, in this case, is a tree. We use this redundant constraints to guarantee the feasibility of flow balance and avoid to use type 2 cuts in Benders decomposition (extremal rays). However, depending on the demand vector structure and on the network connectivity level, constraints (4.45 - 4.49) are not powerful enough to ensure the calculation of a structural solution which prevents cycles in the source/demand path.

This explains the use, just in case, of the strategy described on step 5 of the algorithm given in the last subsection. This strategy is no more but the inclusion of a type 2 cut, and the consequent loss of one master problem iteration, just to ensure feasibility. Depending on the instance to be solved, this strategy may be responsible for unnecessary computational effort.

To deal with this, we devise an alternative strategy, which preserves the flow balance at the higher level without the necessity of constraints (4.45 - 4.49). This approach just uses the link between variables f_{ijk} and g_{ij} , as implied by constraint (4.2), to construct a new master problem. Instead of working only in the space of topological variables x , in this new formulation we work in the space of both the x and g variables, in such a way that the model is able to avoid cycles because it sustains the flow balance. We call this model *MMBend*, and the new master problem is plotted below.

$$\min \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.52)$$

subject to:

$$- \sum_{(o,j) \in E} g_{oj} = - \sum_{k \in K} d_k \quad (4.53)$$

$$\sum_{(i,k) \in E} g_{ik} - \sum_{(k,j) \in E} g_{kj} = d_k, \quad \forall k \in K \quad (4.54)$$

$$\sum_{(i,j) \in E} g_{ij} - \sum_{(j,l) \in E} g_{jl} = 0, \quad \forall j \in V - K - \{o\} \quad (4.55)$$

$$g_{ij} \leq \sum_{k \in K} d_k x_{ij}, \quad \forall (i,j) \in E \quad (4.56)$$

$$g_{ij} \geq 0, \quad \forall (i,j) \in E \quad (4.57)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in E \quad (4.58)$$

and the Benders Cut:

$$t \geq v(x^h) - \sum_{(i,j) \in E} \sum_{k \in K} \lambda_{ijk}^h d_k x_{ij} \quad h = 0, 1, 2, \dots, H \quad (4.59)$$

With this model, we force the flow balance at the master level, with constraints (4.53)-(4.57), and guarantee the feasibility on the structural variables x . On the other hand, the number of variables being treated by the master problem increases by $|E|$. This is also onerous, but, for larger amounts of products being shipped, may justify the alternative decomposition strategy.

4.4 Computational Results

Computational tests were carried out in a Sun Blade 100 with a 500 MHz Ultra-SPARC processor and 1 Gbyte of RAM memory. The operational system

is Solaris 5.8. The Benders decomposition algorithm was implemented in C with *GLPK 3.1* [91] application programming interface.

Table 4.1 shows the dimensions of the networks and of the corresponding mixed integer programming formulations for the tested problems. The test problems can be divided into three classes: the first class contains problems that are Euclidean graphs randomly generated using a procedure similar to that presented in [3]. This procedure has been used extensively for creating testing instances of the Steiner problem. The second class was based in data obtained from a geographical information system and uses realistic costs and distances based on different locations of Brazil. The third class of problems refers to some of Beasley's graphs [15] for Steiner Problems. We have chosen as the origin node the first demand node of each selected Beasley's graph, and we have assigned one unit of flow for each of the remainder demand nodes. These problem instances were selected to perform a direct extension of the work of Randazzo and Luna [116].

Problem Number	V	E	K	β/γ	Number of Variables in Model M		
					Integer	Continuous	
CLASS 1	1	12	36	3	1	36	108
	2	16	30	4	1	30	120
	3	16	60	4	1	60	240
	4	20	60	3	1	60	180
	5	25	80	4	1	80	320
	6	30	90	5	1	90	450
	7	35	100	6	1	100	600
	8	40	110	7	1	110	770
	9	46	120	8	1	120	960
	10	50	130	8	1	130	1040
	11	55	140	8	1	140	1120
	12	60	150	9	1	150	1350
	13	65	170	10	1	170	1700
	14	70	200	12	1	200	2400
	15	80	220	12	1	220	2640
CLASS 2	16	10	21	8	10	21	168
	17	12	26	10	10	26	260
	18	14	31	12	10	31	372
	19	16	36	14	10	36	504
	20	18	40	16	10	40	640
	21	20	48	18	10	48	864
	22	21	52	19	10	52	988
	23	22	55	20	10	55	1 100
	24	24	47	22	10	47	1 034
	25	25	50	20	10	50	1 000
CLASS 3	B1	50	126	8	10	126	1 008
	B2	50	126	12	10	126	1 512
	B5	50	200	12	10	200	2 400
	B6	50	200	24	10	200	4 800
	B16	100	200	16	10	200	3 200

Table 4.1: Network Dimensions for Test Problems.

Six different experiments were conducted to verify the algorithm performance and to determine the behavior of the solutions founded as the network load increases. In order to create the arc cost function for the problem instances presented in Table 4.1, six levels of implicit capacities were proposed $(q_{ij}^0, \dots, q_{ij}^6)$, and five expansion costs also $(\pi_{ij}^1, \dots, \pi_{ij}^5)$. The implicit capacity level q_{ij}^0 was defined *ad hoc* and capacity levels 1 to 5 were produced just doubling the arc

capacity at each expansion level. Indexing the implicit capacity level by l , we have

$$q_{ij}^{l+1} = 2 q_{ij}^l, \quad \forall i, j \in E, \quad l = 0, \dots, 5 \quad (4.60)$$

The capacity expansion costs $(\pi_{ij}^1, \dots, \pi_{ij}^5)$ were proposed in a non-increasing realistic way, enabling scale economies for higher capacity levels. Another restriction imposed over the expansion costs determines that their sum cannot be greater than 50% of the fixed arc installation cost b_{ij} .

The main difficulty in this phase of our work was the unavailability of other popular methods for solving mixed integer nonlinear programs. This makes a comparison of optimal methods hard to obtain. We so choose to compare between linear and nonlinear versions of the test instances. The linear versions were obtained setting the congestion costs equal to zero. This little exercise is able to show us how different are the linear and nonlinear solutions.

The gap between linear and nonlinear solutions was defined by

$$GAP = 100 \left(\frac{Optimum_{nonlinear} - Optimum_{linear}}{Optimum_{linear}} \right) \quad (4.61)$$

In our first experiment, the demand vector of each instance was chosen to guarantee a low level of network load, leading to arc flows below the first implicit capacity level. The computational effort was due to the execution of the 30 problem instances proposed in Table 4.1 using the formulations *MBend*, *MMBend* and a linear run obtained setting congestion costs equal to zero.

To produce the second experiment, all demand vectors were multiplied by a factor of two, and this process was repeated six times, in order to generate data for all the six experiments under focus. With this policy, we have incremented the total network load beyond a factor of twelve. With each of the 30 problem instances being tested for six different levels of network load, using formulations *MBend*, *MMBend* and the linear version, we have 540 different runs.

However, the first five experiments has presented an analogous behavior, always deriving the same optimal structure for linear and nonlinear instances. Beyond this, the gap observed between linear and nonlinear results was almost the same for experiments 1 to 5, as well as the numbers of Benders iterations and the computing times.

For the gap between nonlinear and linear solutions, this can be explained by the following effect: allowing six levels of expansion for the implicit capacities, the tangent point g_{ij0} used to define the unique supporting line (convex hull) of the integrated arc cost function is located very near to zero. In this case, even small arc flows are computed with the aid of the linear portion of the convex approximation. Combining the effect described above with the economies of scale suggested by the existent relationship between parameters β and γ , it is possible to conclude that we are always obtaining as optimal solution an arborescence that is very much like the shortest path tree or the minimum spanning tree. This tendency is repeated until the network load level becomes closer to the maximum installed capacity q_{ij}^5 , where the congestion costs become

explosive. Then, we choose to synthesize the results obtained for experiments 1 to 5 in the Tables 4.2 and 4.3.

Problem Number	Original Formulation			Cycle Avoiding Formulation	
	Iterations	For Cycle Avoiding	Time [s]	Iterations	Time [s]
1	5	0	3.6	4	3.26
2	5	0	4.28	5	3.54
3	6	0	4.3	4	3.1
4	5	0	12.2	3	6.22
5	5	0	15.5	2	7.68
6	6	0	20.86	3	11.9
7	7	0	25.06	5	19.2
8	5	0	23.06	7	30.72
9	6	0	76.82	4	70.02
10	7	0	324	3	231.8
11	5	0	274.66	4	256.42
12	4	0	479	5	411.88
13	7	0	687.64	3	684.68
14	5	0	690.5	3	713
15	6	0	741	3	773.84
16	39	2	36.68	16	42.28
17	79	4	72.6	17	79.44
18	78	31	136.9	15	134.3
19	62	9	138.06	22	127.56
20	234	14	693.9	115	599.28
21	41	10	456.42	23	435.1
22	31	8	609.18	15	656.5
23	120	6	734.76	76	753.56
24	86	7	753.8	53	717.72
25	112	11	836.18	85	833.28
B1	5	0	18.86	3	19.78
B2	6	0	18.6	4	13.98
B5	3	0	130.7	3	141.24
B6	4	0	370.58	5	355.06
B16	6	0	418.74	5	364.56

Table 4.2: Average computing time, number of Benders iterations and number of cycle avoiding constraints for experiments 1 to 5.

In these tables (4.2 and 4.3) we have shown the number of Benders iterations and CPU time for all the test instances. For the original formulation *MBend* were also plotted the number of iterations for cycle avoiding. The experiments are described in this way: Table 4.2 compares the original formulation *MBend* and the cycle avoiding formulation *MMBend* and Table 4.3 compares the results for the cycle avoiding formulation and the linear formulation results. The obtained results for experiment 6, are presented in tables 4.4 and 4.5 in the same manner.

Structural differences were observed in many of the instances between linear and nonlinear optimal solutions, for all the six experiments conducted. For convenience, we are just trying to report these differences through the number of different active arcs founded in the optimal solution for linear and nonlinear cases, as can be seen in the last column of Tables 4.3 and 4.5. This is a direct evidence that congestion costs can affect network design.

For all the experiments the cycle avoiding formulation has presented very competitive results, being smoothly superior to the original one (*MBend*).

As is possible to guess, for experiments belonging to classes 2 and 3, were we have a relation β/γ equal to 10, there must be some tendency to obtain

Problem Number	Cycle Avoiding Formulation		Linear Results		Nonlinear/linear GAP (%)	Number of Different Arcs
	Iterations	Time [s]	Iterations	Time [s]		
1	4	3.26	5	3.3	24.742	2
2	5	3.54	6	2.92	25.94	4
3	4	3.1	4	3.28	15.386	3
4	3	6.22	5	7.08	26.096	2
5	2	7.68	5	7.86	25.91	3
6	3	11.9	5	14.34	25.25	3
7	5	19.2	7	18.58	24.818	4
8	7	30.72	7	25.42	24.532	3
9	4	70.02	4	72.68	16.912	3
10	3	231.8	5	217.4	24.764	5
11	4	256.42	7	244.12	25.138	3
12	5	411.88	4	432.8	26.65	3
13	3	684.68	7	618.76	25.824	6
14	3	713	5	700.8	24.642	4
15	3	773.84	5	789.4	25.012	2
16	16	42.28	19	45.32	26.23	5
17	17	79.44	18	82.44	24.96	4
18	15	134.3	19	118	25.784	4
19	22	127.56	27	118.48	24.456	2
20	115	599.28	116	573.52	34.728	3
21	23	435.1	28	416.58	25.63	6
22	15	656.5	21	621.88	25.896	5
23	76	753.56	77	799.96	16.188	3
24	53	717.72	56	763	25.558	2
25	85	833.28	93	854.94	24.434	6
B1	3	19.78	5	18.96	25.788	6
B2	4	13.98	6	15.98	14.788	5
B5	3	141.24	3	160.98	24.948	5
B6	5	355.06	3	358.18	34.822	3
B16	5	364.56	3	367.58	25.954	6

Table 4.3: Average computing time, number of Benders iterations, Nonlinear/linear gap and number of different arcs for experiments 1 to 5.

solutions that are essentially the minimum spanning trees (higher fixed costs). This tendency is reversed only for the last experiment, where the congestion cost component becomes more expressive, justifying the adoption of solutions that are more similar to the shortest path tree.

Among the 30 problems, those who belong to class 2 were the more difficult to solve. This can be explained by the use of denser networks and demand vectors, and by the presence of alternative optimal solutions. In this case, the Benders decomposition algorithm spends a long time testing equal cost solutions, before advancing for a better upper bound. The solution times for all the tested instances is acceptable, if we remember that we are solving a mixed integer nonlinear program. In practice, our approach renders a nonlinear problem with congestion costs as much difficult as the standard linear programming version [116].

4.5 Conclusions

We have presented a multi-commodity flow formulation for the local access network design problem with congestion costs. We have studied an exact method to solve the problem, based on generalized Benders decomposition. We have also

Problem Number	Original Formulation			Cycle Avoiding Formulation	
	Iterations	For Cycle Avoiding	Time [s]	Iterations	Time [s]
1	6	0	4.6	7	3.4
2	5	0	4.8	7	4.4
3	11	0	5.5	7	3.4
4	6	0	12.9	4	5.8
5	5	0	16.5	2	7.5
6	10	0	22.4	3	11.8
7	11	0	27	6	19.6
8	4	0	24.2	9	30.9
9	10	0	78.6	4	69.7
10	9	0	325.5	3	231.5
11	7	0	275.7	6	255.5
12	8	0	480.3	5	412.2
13	12	0	687.3	2	685.4
14	6	0	689.3	5	712.7
15	9	0	742.3	6	774.7
16	41	4	37.2	18	41.3
17	82	4	73.7	18	79.4
18	80	40	137.9	16	135
19	65	12	139.7	25	127.8
20	234	16	694.7	117	599.9
21	46	11	457.7	24	438.5
22	34	10	607.3	16	656.1
23	120	8	735.1	77	746
24	85	7	754.4	57	717.6
25	115	12	837.4	89	833.2
B1	6	0	20.1	3	19.5
B2	6	0	20.6	7	14.4
B5	4	0	131.8	5	140.6
B6	9	0	383.2	5	354.2
B16	8	0	420.2	6	365.9

Table 4.4: Computing time, number of Benders iterations, number of cycle avoiding constraints for experiment 6.

confirmed the possibility of exactly solving such problems with free software mathematical programming packages, and for this purpose we have a surprisingly good alternative to commercial softwares.

Yet the computational experiments have been limited, our experience suggests that a certain number of these modeling and solution strategies can be applied to the frequently occurring problems where the congestion cost component is important. Our conclusion is that generalized Benders decomposition emerges as a good method, being enough robust to deal with mixed integer nonlinear programs of this type. The relevance of this type of model is now increasing because emerging Internet services requires larger bandwidth in the local access network and also because applications concerning multi-party multicasting tree construction should take into account congestion costs.

Another question for investigation is the generation of approximate solutions by heuristics with the objective of adding cuts to the Benders master problem in a preprocessing phase (in the current implementation our preprocessing phase only includes the solution of a shortest path problem and a minimal spanning tree problem). These new cuts might reduce the total number of Benders cycles, but the necessary time to solve each master problem might be increased since the master problem would have more constraints. The trade off between these two aspects should be better evaluated.

Problem Number	Cycle Avoiding Formulation		Linear Results		Nonlinear/linear GAP (%)	Number of Different Arcs
	Iterations	Time [s]	Iterations	Time [s]		
1	7	3.4	4	3.3	71.32	3
2	7	4.4	8	3	37.57	5
3	7	3.4	3	4.5	55.54	3
4	4	5.8	6	7.7	33.94	3
5	2	7.5	6	7.5	57.13	3
6	3	11.8	6	15.3	57.05	3
7	6	19.6	7	19.3	81.2	5
8	9	30.9	9	25.2	60.19	4
9	4	69.7	4	72.9	42.26	3
10	3	231.5	6	218.8	58.68	6
11	6	255.5	8	245.5	32.55	3
12	5	412.2	4	434.1	58.4	3
13	2	685.4	7	619.4	42.55	7
14	5	712.7	4	701.6	35.01	5
15	6	774.7	8	790.1	59.89	2
16	18	41.3	21	46.1	57.95	5
17	18	79.4	17	84.1	73.9	5
18	16	135	19	117.3	77.74	4
19	25	127.8	26	119.8	56.3	3
20	117	599.9	117	574.1	77.29	4
21	24	438.5	29	416.9	61.68	7
22	16	656.1	21	622	63.77	5
23	77	746	79	801	37	4
24	57	717.6	58	764	34.91	3
25	89	833.2	95	854.7	76.39	6
B1	3	19.5	7	20.3	55.25	6
B2	7	14.4	6	15.1	83.27	5
B5	5	140.6	4	162.9	42.75	5
B6	5	354.2	5	357.9	39.59	4
B16	6	365.9	3	369.3	39.3	7

Table 4.5: Computing time, number of Benders iterations, and Nonlinear/linear gap for experiment 6.

Chapter 5

Integrating Facility Location and Network Design

5.1 Problem Description

As already pointed in chapter one, the main goal of this work is to deal with a combination of local access network design and facility location under a *QAP* characterization. Our motivation is the design of a system involving two levels: the higher level, typically the server-to server network, in which we are dealing with the task of choosing the optimal location of all servers, and the lower level, typically the client-server access network, in which we are dealing with the task of choosing the optimal network design (naturally, including congestion costs). These two problems cannot be solved in a separated way: the choice of a server location has impact on the maximum quality of the service provided to the final customers. On the other hand, the local access network cost can influence the choice of a location for a server. See Figures (5.3) and (5.4).

Problems that involve network design and facility location arise in very different application areas. The design of "Hub-and-Spoke Systems" is a fundamental stone for optimized air traffic, and can also be applied in the field of third part logistic provision. A survey about hub-and-spoke systems must include the work of O'Kelly [101] [102] [105] [104], O'Kelly and Skorin-Kapov [103], Aykin [7] [8] [9], Zapfel and Wasner [130], Drezner and Wesolowsky [41], Campbel [32] [33] [31] and Pirkul et al. [109] [110]. For network design/location models we advice the work of Daskin and Melkote [97] [96].

We are searching for the best location of servers which minimize the total system cost. In this first approach we will make the assumption that the server-to-server network is technologically different of the local access network. This assumption is consistent with reality if we are talking about telecommunication

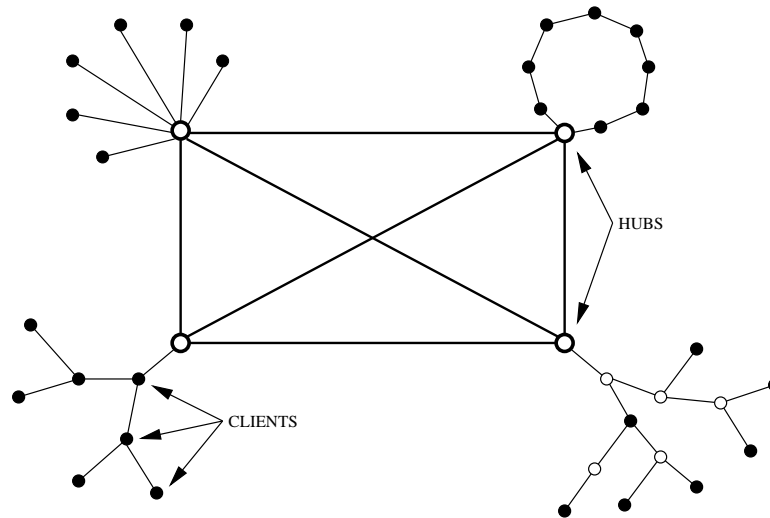


Figure 5.1: Hub-and-spoke system with different kinds of local access networks.

networks, where the the local access network is usually made of copper cable based electronics, and the server-to-server network is optical. This assumption is also well applied to transportation networks, where the local access network is composed by urban streets and avenues and the server-to-server network is made up of highways. This separation is reasonable since the capacities and flows associated with each level are different in some orders of magnitude and are treated in different ways.

In the literature survey discussed above, it is clear that one can have all kinds of local access networks (star trees, spanning trees, Steiner trees and rings) as we can see in Figure (5.1) [74]. However, the most commonly used model for the local access network infra-structure is the star tree. This can be explained since the first field of application of hub-and-spoke models was for air traffic optimization. The fundamental assumptions of these models arise from the special features of this kind of application. For example, a given flow never passes across more than two hubs, and there is not activation cost for a given link. These assumptions are well suited for air traffic and even for some transportation systems, but not for telecommunications systems and for third-party logistics systems. The first integer-mixed quadratic programming models for hub-and-spoke systems were proposed by Aykin [8] and O’Kelly [102]. More recent models (and linearizations) were covered by Campbell in [33]. The best suited to our former integer programming toolkit are the *p-Hub Median Problem*, where the number of hubs to locate is fixed, and the *Uncapacitated Hub Location Problem*, where we do not know the number of hubs to locate *a priori*. We are

able now to discuss the formulation and the decomposition scheme for several of these models.

5.2 Mathematical Programming Formulations

To associate the local access network design problem with the quadratic assignment problem, our location model, a first approach is to imagine a city/state/country to be target of our experiment. Points that are origins or destinations of some flow of commodities are dispersed over this region, that also contains some candidate sites to locate the hubs (or servers) that will concentrate the traffic and re-route it from its origin to the correct destination. This is depicted in Figure (5.3).

On the traditional hub-and-spoke models, it is necessary to deal with the client-server assignment. This task can be done in reasonable time because the local access network model is as simpler as possible (the star tree structure). This means that there is no additional costs to route flows from the clients to the corresponding server as in a multi-connected congested network. For this kind of models, a pair of clients $i, j \in I$ is interconnected by a hub-and-spoke structure that can contain one or two hubs (servers) $k, l \in K$. The integer-mixed quadratic programming formulations usually multiply two client-hub assignment $0-1$ variables to design the interconnection between i and j , and the linearized versions just replace the product $x_{ik}x_{lj}$ by a variable f_{ijkl} that describes the flow between origin i and destination j being routed through the pair of hubs kl . A cost matrix c_{ijkl} is constructed adding the costs to interconnect client i and hub k , plus the cost to flow from hub k to hub l and finally to client j : $c_{ijkl} = c_{ik} + c_{kl} + c_{lj}$. If the number of hubs to locate is given, we have the *p-Hub Median Problem*, that we are writing as close to *QAP* as possible, as a flow formulation:

$$\min \sum_{k \in K} a_k x_k + \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} \sum_{l \in K} c_{ijkl} f_{ijkl} \quad (5.1)$$

subject to:

$$\sum_{k \in K} x_k = p \quad (5.2)$$

$$\sum_{k \in K} f_{ijkl} \leq w_{ij} x_l, \quad \forall i, j \in I, i \neq j, l \in K \quad (5.3)$$

$$\sum_{l \in K} f_{ijkl} \leq w_{ij} x_k, \quad \forall i, j \in I, i \neq j, k \in K \quad (5.4)$$

$$\sum_{k \in K} \sum_{l \in K} f_{ijkl} = w_{ij} \quad (5.5)$$

$$f_{ijkl} \geq 0, \quad \forall i, j \in I, k, l \in K \quad (5.6)$$

$$x_k \in \{0, 1\} \quad \forall k \in K \quad (5.7)$$

If we just know the maximum number of possible hubs $|K|$, we have the *Uncapacitated Hub Location Problem*:

$$\min \sum_{k \in K} a_k x_k + \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} \sum_{l \in K} c_{ijkl} f_{ijkl} \quad (5.8)$$

subject to:

$$\sum_{k \in K} x_k \leq |K| \quad (5.9)$$

$$\sum_{k \in K} f_{ijkl} \leq w_{ij} x_l, \quad \forall i, j \in I, i \neq j, l \in K \quad (5.10)$$

$$\sum_{l \in K} f_{ijkl} \leq w_{ij} x_k, \quad \forall i, j \in I, i \neq j, k \in K \quad (5.11)$$

$$\sum_{k \in K} \sum_{l \in K} f_{ijkl} = w_{ij} \quad (5.12)$$

$$f_{ijkl} \geq 0, \quad \forall i, j \in I, i \neq j, k, l \in K \quad (5.13)$$

$$x_k \in \{0, 1\} \quad \forall k \in K \quad (5.14)$$

As these problems are written close to the *QAP*, we can use all the structure of *QAP* decomposition to find a Benders decomposition scheme for them. For *P-Hub Median Problem* (Equations (5.1)-(5.7)), fixing the structural variables x at a given iteration h , the primal subproblem in f can be written as:

$$\min \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} \sum_{l \in K} c_{ijkl} f_{ijkl} \quad (5.15)$$

subject to:

$$\sum_{k \in K} f_{ijkl} \leq w_{ij} x_l^h, \quad \forall i, j \in I, i \neq j, l \in K \quad (5.16)$$

$$\sum_{l \in K} f_{ijkl} \leq w_{ij} x_k^h, \quad \forall i, j \in I, i \neq j, k \in K \quad (5.17)$$

$$\sum_{k \in K} \sum_{l \in K} f_{ijkl} = w_{ij} \quad (5.18)$$

$$f_{ijkl} \geq 0, \quad \forall i, j \in I, i \neq j, k, l \in K \quad (5.19)$$

If a dual variable λ_{ij} is associated with constraint (5.18) for each commodity ij , a set of dual variables ν_{ijk} is associated with constraints (5.17) and a set of dual variables ν_{ijl} is associated with constraints (5.16), the dual subproblem for the commodity ij is:

$$\max \left[w_{ij} \lambda_{ij} - \sum_{l \in K} w_{ij} x_l^h \nu_{ijl} - \sum_{k \in K} w_{ij} x_k^h \nu_{ijk} \right] \quad (5.20)$$

subject to:

$$\lambda_{ij} - \nu_{ijl} - v_{ijk} \leq c_{ijkl}, \forall k, l \in K \quad (5.21)$$

$$\lambda_{ij} \in \mathcal{R} \quad (5.22)$$

$$v_{ijk} \geq 0, \forall k \in K \quad (5.23)$$

$$\nu_{ijl} \geq 0, \forall l \in K \quad (5.24)$$

Implying the master problem:

$$\min \sum_{k \in K} a_k x_k + \eta \quad (5.25)$$

subject to:

$$\eta \geq \sum_{i,j \in I} \left[w_{ij} \lambda_{ij}^h - \sum_{l \in K} w_{ij} \nu_{ijl}^h x_l - \sum_{k \in K} w_{ij} v_{ijk}^h x_k \right], \forall h \in H \quad (5.26)$$

$$\sum_{k \in K} x_k = p \quad (5.27)$$

$$x_k \in \{0, 1\} \quad \forall k \in K \quad (5.28)$$

To determine the optimal solution for the dual subproblem at iteration h , remark that the routing of the commodity ij can pass through one or two hubs, as depicted in Figure (5.2).

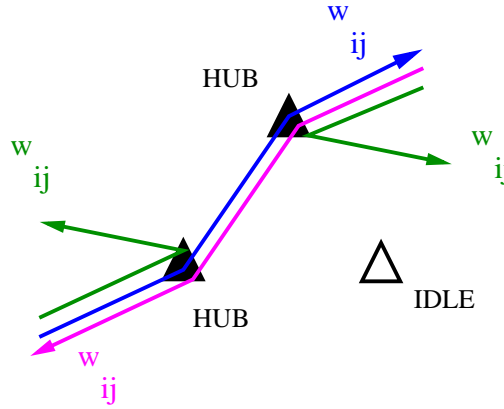


Figure 5.2: Possible routes for the commodity ij for a given $x = x^h$.

The optimal solution of the primal subproblem for the commodity ij , equations (5.15)-(5.19), is the costless route, discarding locations that do not host any hub:

$$w_{ij} \cdot \min_{(k,l)} \{c_{ijkl} | y_k^h = y_l^h = 1\} \quad (5.29)$$

The dual variables computation with economical interpretation (meaningful values) is a clue for an efficient implementation. Here, λ_{ij} is the highest possible price difference between i and j and ν_{ijk} and ν_{ijl} are additional taxes paid to route the flow ij through hubs k and l . Using the complementary slackness, the optimal dual solution for commodity ij is given by:

$$\lambda_{ij}^h = \min_{(k,l)} \{c_{ijkl} | y_k^h = y_l^h = 1\} \quad (5.30)$$

$$\nu_{ijk}^h = 0, \text{ if } y_k^h = 1 \quad (5.31)$$

$$\nu_{ijk}^h = \max \{0, \max_l \{\lambda_{ij}^h - c_{ijkl}\}\}, \text{ if } y_k^h = 0 \quad (5.32)$$

$$\nu_{ijl}^h = 0, \text{ if } y_l^h = 1 \quad (5.33)$$

$$\nu_{ijl}^h = \max \{0, \max_k \{\lambda_{ij}^h - c_{ijkl}\}\}, \text{ if } y_l^h = 0 \quad (5.34)$$

To adapt our decomposition scheme to the *Uncapacitated Hub Location Problem*, we must observe that the master problem does not have information about the flow constraints. To determine only feasible solutions when decomposing the *Uncapacitated Hub Location Problem* it is necessary to add a single constraint in x to the master problem ensuring the installation of at least one hub, replacing the fixed number of hubs to locate by a maximum number of possible hubs:

$$\min \sum_{k \in K} a_k x_k + \eta \quad (5.35)$$

subject to:

$$\eta \geq \sum_{i,j \in I} \left[w_{ij} \lambda_{ij}^h - \sum_{l \in K} w_{ij} \nu_{ijl}^h x_l - \sum_{k \in K} w_{ij} \nu_{ijk}^h x_k \right], \forall h \in H \quad (5.36)$$

$$\sum_{k \in K} x_k \leq |K| \quad (5.37)$$

$$\sum_{k \in K} x_k \geq 1 \quad (5.38)$$

$$x_k \in \{0, 1\} \quad \forall k \in K \quad (5.39)$$

5.2.1 Improving the Design of the Local Access Network

The former hub-and-spoke formulations deal with the client-server interconnection cost in a very simple way. This is not interesting in some applications where the major costs are just in the local access network. In fact, traditional hub-and-spoke models optimize the client-server assignment while solving a quadratic

location model. However, to improve the treatment of the local access network also dealing with the client-hub assignment it is not easy. Our idea here is to detach the two network levels, optimizing the local access network and solving the associated quadratic location problem in a separated way.

It is natural to suppose that the city/state/country under study is divided in a set of regions. This kind of partitioning, as suggested by Figure (5.3), can be induced by a series of different factors like ground occupation policy, population density, geographical interference and other social, politic and economic criteria. The installation of at least a server per region is ensured and natural. Inside each region, there exists a set of candidate locations to host the server, that are interconnected each other and to the final customers by a potential local access network. The role of the server is to deal with the local access demands and concentrate all the traffic, correctly addressing it to the other regions in the city/state/country.

We define $I_k \subset I$ as the set of all candidate sites for the location of server k . To implement this partitioning scheme, we define a binary matrix P that has $p_{ki} = 1$ if the site i can host the server k and $p_{ki} = 0$, otherwise.

To overcome the local access network design problem, let us consider first the graph $G = (V, E)$ that represents the potential local access network for a given region, as suggested in Figure (5.3). We are making the assumption that each region is self-contained, what means that we are not considering the arcs of the potential local access network that interconnect one region to another. If inside this particular region we have $|I_k|$ candidate sites to host a server, the local access network design problem with congestion costs will be solved $|I_k|$ times, each one considering a different candidate site $i \in I_k$ as the origin of flow. This task will be performed with the aid of the methods and algorithms treated in chapter 4. Since this is not expensive, and also can be carried out using parallel computing, we will assume that can be done in reasonable time.

Now, given all the optimal local access network design costs, for each server candidate location, we will use them to define a linear server installation cost matrix $A = (a_{ki})$. This matrix has as many lines as servers and as many columns as the sum of all the candidate sites of all servers, having the same dimensions of P . The integration of the two levels of design is ensured by the incorporation of the local access network costs into a_{ki} .

We are now capable to adapt our models and algorithms for QAP to work on this problem. Modifying equations (2.20)-(2.23) for our network design/facility location model, we remember that there are some candidate sites for a facility, but only one possible facility for each location. We can now re-write the QAP model as:

$$\min \sum_{k \in K} \sum_{i \in I_k} a_{ki} x_{ki} + \sum_{(i,j) \in E} \sum_{k \in K, l \in K, k \neq l} c_{ij} f_{ij}^{kl} \quad (5.40)$$

subject to:

$$\sum_{i \in I_k} x_{ki} = 1, \quad \forall k \in K \quad (5.41)$$

$$- \sum_{j \in I, i \neq j} f_{ij}^{kl} = -b_{kl}x_{ki}, \quad \forall i \in I, k, l \in K, k \neq l \quad (5.42)$$

$$\sum_{i \in I, i \neq j} f_{ij}^{kl} = b_{kl}x_{lj}, \quad \forall j \in I, k, l \in K, k \neq l \quad (5.43)$$

$$f_{ij}^{kl} \geq 0, \quad \forall i, j, k, l = 1, \dots, n, i \neq j, k \neq l \quad (5.44)$$

$$x_{ki} \in \{0, 1\} \quad \forall k \in K, i \in I_k \quad (5.45)$$

For the model given by equations (5.40)-(5.45), there is an underlying assumption: the hub-to-hub interconnecting backbone network is already given. This assumption is consistent with the usual applications, specially if we devise the air traffic, transportation logistics and telecommunication networks. This assumption also provides an interesting approach to use Benders method, since for x fixed, our subproblems are not different from those obtained for QAP decomposition. We can avoid the development of a new mechanism to evaluate the dual variables, using the decomposition algorithm developed in chapter 2. We just need to adapt our master problem, equations(2.35)-(2.36), taking into account the new set of constraints in x . Rewriting the master problem:

$$\min \sum_{k \in K} \sum_{i \in I_k} a_{ki}x_{ki} + \eta \quad (5.46)$$

subject to:

$$\eta \geq \sum_{k, l \in K, k \neq l} \sum_{j \in I} b_{kl}x_{lj}v_j^{kl, h} - \sum_{k, l \in K, k \neq l} \sum_{i \in I} b_{kl}x_{ki}u_i^{kl, h}, \quad \forall h \in H \quad (5.47)$$

$$\sum_{i \in I_k} x_{ki} = 1, \quad \forall k \in K \quad (5.48)$$

$$x_{ki} \in \{0, 1\} \quad \forall k \in K, i \in I_k \quad (5.49)$$

The dual prices $u_i^{kl, h}$ and $v_j^{kl, h}$ are evaluated for the pair kl , fixing a single variable $u_i^{kl, h}$ at the iteration h , using the same scheme developed in chapter 2:

$$v_j^{kl, h} = u_i^{kl, h} + c_{ij}, \quad \forall j \in I, i \neq j, k \neq l \quad (5.50)$$

and using the above defined $v_j^{kl, h}$, we can define the other variables $u_i^{kl, h}$:

$$u_i^{kl, h} = \max_{j \in J, j \neq i} [v_j^{kl, h} - c_{ij}], \quad \forall i \in I, i \neq j, k \neq l \quad (5.51)$$

This model is well suited to improve the local access network design, but we still have just two hubs between each pair of clients. If we want to generalize our

model, we must obtain scale-economies when designing this system, sometimes using alternative routes to address flow for a given origin-destination pair. This feature can be accomplished if we rewrite the above formulation, in such a way that becomes possible to route flows through more than two hubs.

5.2.2 Generalized Model Including Hub Transshipment and Network Design

The first logical step to generalize our network design location model is to enable transshipment between servers. Transshipment can be particularly interesting if we set up congestion costs between hubs. Rewriting constraints (5.42) and (5.43) as a single transshipment constraint, we can overcome the task. It is necessary to remark that in this case we will not observe trivial solutions when writing *QAP* with a transshipment constraint [77], since each facility is constrained to its own region:

$$\min \sum_{k \in K} \sum_{i \in I_k} a_{ki} x_{ki} + \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} \sum_{l \in K} \gamma d_{ij} f_{ij}^{kl} \quad (5.52)$$

subject to:

$$\sum_{j \in I} f_{ji}^{kl} - \sum_{j \in I} f_{ij}^{kl} = b_{kl} x_{li} p_{li} - b_{kl} x_{ki} p_{ki}, \quad \forall i \in I, k, l \in K \quad (5.53)$$

$$\sum_{i \in I_k} x_{ki} = 1, \quad \forall k \in K \quad (5.54)$$

$$f_{ij}^{kl} \geq 0, \quad \forall i, j \in I, k, l \in K \quad (5.55)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k \in K, i \in I_k \quad (5.56)$$

The key for the above formulation is the set of constraints (5.53), that uses the matrix P to create conditions for nodes and arcs outside regions k and l to be used for transshipment of the commodity kl . Here, the matrix D is a distance matrix established between all the candidate sites of all regions, and γ is a cost per unit flow per distance for transportation. From our experience on *QAP*, we can see that one set of assignment constraints is not necessary, since a server of a given region can not be located outside this region. We must observe also that this formulation (that includes transshipment) is derived from the original Koopmans and Beckmann formulation for *QAP*, and has $o(n^2 m^2) + o(n^2) + o(n)$ variables and $o(nm^2) + o(n)$ constraints. We remark that local access network contributes with the majority of the cost. Since the existence of linear costs makes *QAP* more suitable to solve, once defined the server locations all that remains is a multicommodity minimum cost flow problem.

If it is necessary to design the backbone, we can add network design at the upper level. Introducing the variables y_{ij} that decide the installation of the arc ij , it is possible to rewrite Equations (5.52)-(5.56) as:

$$\min \sum_{k \in K} \sum_{i \in I_k} a_{ki} x_{ki} + \sum_{i \in I} \sum_{j \in I} \beta d_{ij} y_{ij} + \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} \sum_{l \in K} \gamma d_{ij} f_{ij}^{kl} \quad (5.57)$$

subject to:

$$\sum_{j \in I} f_{ji}^{kl} - \sum_{j \in I} f_{ij}^{kl} = b_{kl} x_{li} p_{li} - b_{kl} x_{ki} p_{ki}, \quad \forall i \in I, k, l \in K \quad (5.58)$$

$$\sum_{i \in I_k} x_{ki} = 1, \quad \forall k \in K \quad (5.59)$$

$$f_{ij}^{kl} \leq b_{kl} y_{ij}, \quad \forall i, j \in I, k, l \in K \quad (5.60)$$

$$f_{ij}^{kl} \geq 0, \quad \forall i, j \in I, k, l \in K \quad (5.61)$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \in I \quad (5.62)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k \in K, i \in I_k \quad (5.63)$$

The above formulation appears to be much more complex than its transshipment-only counterpart. Beyond this, the two above formulations also prepare the ground to deal with congestion costs when considering flows between hubs. It is time now to develop computational experience on the use of all the formulations discussed.

5.3 Computational Experiences

In all the following experiments we are not interested in to solve a lot of different instances of a given formulation, making a detailed study of the structure of each one, but only in to draw the borders, verifying how far each formulation or decomposition scheme can go with conventional computational resources. All the implementations were produced using *ILOG CPLEX 7.0 Concert Technology* and were carried out in a SUN BLADE 100 workstation, equipped with one 500 MHz processor and 1 Gbyte of RAM memory.

5.3.1 Benders Decomposition for the p-Hub Median Problem and the Uncapacitated Hub Location Problem

In this small set of experiments the demand and cost matrices of the original *QAPLIB* instances were taken, adding hub installation pseudo-randomic costs. Table 5.1 shows the experience with the *p-Hub Median Problem*, plotting the number of clients and hubs for each instance, the number of Benders iterations, the *p/q* ratio and the computing time. As we can see, Benders decomposition algorithm is capable to solve large instances in reasonable time. In despite of that, we experience some problems with the larger instances *sko49* and *sko64*, since the subproblem computation sometimes slows down due to memory lack of our workstation, what opens opportunity to the successful use of parallel

Original instance	Number of clients	Number of hubs	Number of flow variables	Benders iterations	p/q cost ratio	Time[s]
nug7	7	2	2058	8	0.000	0
nug7		2		4	0.252	0
nug7		2		3	0.504	0
nug7		2		3	0.756	0
nug7		5		3	6.784	0
nug12	12	1	19008	6	0.009	0
nug12		2		6	0.102	0
nug12		2		4	0.306	0
nug12		3		26	0.000	3
nug12		3		5	0.172	0
nug12		3		4	0.343	0
nug12		3		3	0.515	0
nug12		3		3	0.859	0
nug12		4		22	0.000	2
nug12		4		3	0.317	0
nug12		4		3	0.951	0
nug12		4		3	1.585	0
nug12		6		3	5.862	1
nug15	15	3	47250	72	0.000	24
nug15		3		12	0.108	1
nug15		3		8	0.137	0
nug15		3		7	0.205	0
nug15		4		7	0.285	1
nug15		4		5	0.358	0
nug15		5		6	0.396	0
nug15		6		3	0.511	0
nug20		20		1	152000	9
nug20	3		8	0.123		1
nug20	4		9	0.220		0
nug20	5		13	0.088		1
nug20	5		9	0.175		1
nug20	5		8	0.263		0
nug20	5		7	0.312		1
nug20	5		7	0.312		0
nug20	6		6	0.406		1
nug20	7		3	0.544		0
nug30	30		3	783000		105
nug30		3	53		0.027	25
nug30		4	45		0.047	19
nug30		4	30		0.071	10
nug30		5	24		0.093	8
ste36a	36	2	1632960	54	0.034	43
sko49	49	6	5647152	133	0.053	482
sko49		6		38	0.107	75
sko64	64	6	16515072	72	3.340	408

Table 5.1: Benders decomposition for the p-Hub Median Problem.

computing. In Table 5.2 we have the experiments with the *Uncapacitated Hub Location Problem*, showing the same parameters emphasized in Table 5.1.

We believe that rewrite these classical hub-and-spoke models as flow formulations was decisive when considering the performance of the Benders algorithm. We also believe that these flow formulations can be enhanced, granting the solution of very large instances with the aid of parallel computing.

Original instance	Number of clients	Number of hubs	Number of flow variables	Benders iterations	p/q cost ratio	Time [s]
had12	12	8	19008	14	0.219	1
had14	14	8	35672	5	0.089	0
had14		8		3	0.148	1
had16	16	12	61440	9	0.051	0
had16		12		7	0.062	0
had16		12		6	0.071	0
had16		12		4	0.109	0
had18	18	12	99144	7	0.056	1
had18		12		7	0.058	0
nug7	7	1	2058	5	0.165	2
nug7		2		6	0.252	0
nug7		3		6	0.252	1
nug7		4		6	0.252	1
nug7		7		7	0.277	0
nug12	12	2	19008	7	0.102	0
nug12		6		8	0.172	1
nug12		7		12	0.190	1
nug12		8		8	0.172	1
nug12		9		8	0.172	0
nug12		10		7	0.053	0
nug12		10		6	0.071	1
nug12		10		5	0.080	0
nug12		10		5	0.106	0
nug12		10		11	0.136	1
nug12		10		3	0.150	0
nug12		10		9	0.205	0
nug12		10		15	0.399	0
nug15		15		6	47250	8
nug15	7		6	0.066		0
nug15	7		6	0.070		0
nug15	10		8	0.062		1
nug15	10		6	0.079		0
nug15	10		6	0.083		0
nug15	10		23	0.188		2
nug15	10		33	0.296		2
nug15	10		3	0.413		0
nug30	30	6	783000	6	0.089	2
nug30		6		5	0.118	2
nug30		6		11	0.227	3
nug30		6		7	0.325	2
nug30		10		8	0.101	2
nug30		10		8	0.118	2
nug30		10		6	0.130	1
nug30		10		12	0.358	4
nug30		10		14	0.399	4
nug30		10		12	0.432	3
sko49	49	6	5647152	14	0.250	26
sko49		10		14	0.250	26
sko49		30		112	0.000	341
sko49		30		9	0.169	16
sko49		30		8	0.187	14
sko49		30		13	0.267	24
sko49		30		5	0.312	8
sko49		49		14	0.165	25
sko49		49		13	0.202	24
sko64	64	49	16515072	4	0.396	18
sko64		49		3	0.792	11
ste36a	36	10	1632960	15	0.437	8
ste36a		18		9	0.483	5
ste36a		18		9	0.640	4
ste36a		18		9	0.644	5

Table 5.2: Benders decomposition for the Uncapacitated Hub Location Problem.

5.3.2 The Integrated Model: QAP + Local Access Network Design With Congestion Costs

We report now our experiences with the integrated model, described in section 5.2.1. The solution of the integrated model must be made in two phases: first evaluating the local access costs as suggested in chapter 4, and then, embedding the local access costs on the a_{ki} coefficients, solving the model given by equations (5.40)-(5.45) in his decomposed form. The solved instances are purely pseudo-randomic, do not sustaining the triangular inequality, and are represented by names beginning with *net* plus the size of K and the size of I . They were produced in a realistic way, ensuring that costs to install infrastructure are always equal or superior to operational costs, with the aid of a generator written in C++ and compiled with GNU GCC 3.0.

Table 5.3 shows the number of servers (hubs), the total number of possible locations and consequently the number of flow variables, the number of iterations of Benders algorithm, the p/q cost ratio and the computing times.

Problem name	Number of servers	Possible locations	Number of flow variables	Benders iterations	p/q cost ratio	Time[s]
net624	6	24	17424	7	1.711	0
net824	8	24	32448	30	2.537	3
net1030	10	30	81300	161	1.361	137
net1030	10	30	81300	4	7.843	0
net1236	12	36	171504	7	8.061	1
net1248	12	48	304704	3	1.835	1
net1260	12	60	475920	26	1.582	6
net1260	12	60	475920	3	15.820	1
net1560	15	60	756900	70	1.668	33
net2060	20	60	1369200	14	2.490	5
net20100	20	100	3802000	114	2.423	181
net20100	20	100	3802000	15	4.847	23
net20100	20	100	3802000	3	121.166	2
net30150	30	150	19579500	54	4.685	391
net40200	40	200	62408000	85	5.316	2177

Table 5.3: Computational results for the integrated model.

As we can see, the computational experiments have been quite limited, but the Benders decomposition scheme is able to obtain solutions for very large instances. Since the growing of the flow variables is really explosive in this formulation, limiting the size of problems that are solvable inside the computer main memory, parallel computing is a reasonable alternative to the solution of larger instances.

5.3.3 Testing the Hub Transshipment Network Design Model

The instances solved here are also purely pseudo-randomic. These randomic instances do not sustain the triangular inequality, and are represented by names beginning with *rnet* plus the size of K and the size of I . They were produced in a realistic way, ensuring that costs to install infrastructure (servers and arcs interconnecting them) are always superior to operational costs.

In Table 5.4, we show the obtained results for the experience with model (5.57)-(5.63). This table shows the numbers of variables (integer and continuous), the quality of the linear programming bound and the solution time. Table 5.4 shows that the linear programming relaxation is very strong. This can be explained by two reasons: the location problem uses linear installation costs, what induces indivisible locations (see chapter 2), and for a fixed server location, what remains is a multicommodity network design problem, having an extremely strong formulation (see [116]). On the other hand, the linear programs generated by this location/network design model are very hard to solve, what explains the large computing times. It was also not possible to solve problems beyond 15 servers and 45 candidate sites, since *CPLEX* always crashes down due to lack of memory beyond these values. This fact suggests that the use of Benders decomposition for this problem is straightforward and must add solution power, if well developed.

problem name	Number of Variables		Bound Quality	Install. Cost	Network Costs (ND + OC)	Optimal Cost	Computing Time[s]
	Integer	Continuous					
net624.dat	24	20736	0.954	369	17821	18190	6
net824.dat	24	36864	0.977	3744	33207	37113	12
net824.dat			0.955	468	33369	33837	9
net1030.dat	30	90000	0.959	457	55648	56105	46
net1030.dat			0.997	27609	61438	89047	36
net1030.dat			1	218970	80183	326153	19
net1030.dat			0.941	175176	106171	379747	200
net1030.dat			0.926	175176	98798	345974	108
net1030.dat			0.992	109485	90017	252002	644
net1030.dat			0.987	109485	81485	229970	42
net1030.dat			0.96	109485	80183	216668	28
net1030.dat			0.963	31968	65460	117828	882
net1030.dat			0.914	29438	62433	105771	97
net1030.dat			0.98	27609	61503	91352	45
net1030.dat			0.912	27609	61114	89047	37
net1236.dat			36	186624	0.965	252768	165929
net1236.dat	0.929	242112			146485	451597	2967
net1236.dat	0.961	235088			137776	403464	109
net1236.dat	0.925	227632			130608	359054	36
net1236.dat	0.945	35994			110298	146292	265
net1236.dat	0.969	753			94189	94942	537
net1248.dat	48	331776	0.917	71392	81069	152461	70
net1545.dat	45	455625	0.993	44949	100938	145887	292

Table 5.4: Report for a brief experiment using the Hub Transshipment Network Design model.

5.4 Concluding Remarks

The obtained results concerning location/network design problems with interdependence on the location of servers are very expressive. Future improvements of the Benders decomposition algorithms of this chapter will make possible to solve very larger instances at reasonable time, if we use parallel computing, specially on the field of interacting hub facilities systems design.

The task of integration of the models of local access network design and facility location with interdependency, using a *QAP* framework, was well ac-

completed. The produced flow formulation can handle with large instances at reasonable cost.

Our Benders decomposition algorithms are robust enough to deal with large scale hard to solve mixed integer programming problems, designing very complex systems.

It is always important to remember that our local access network is capacitated in this case, computing congestion costs, what ensures the quality of service maintenance for the final customer.

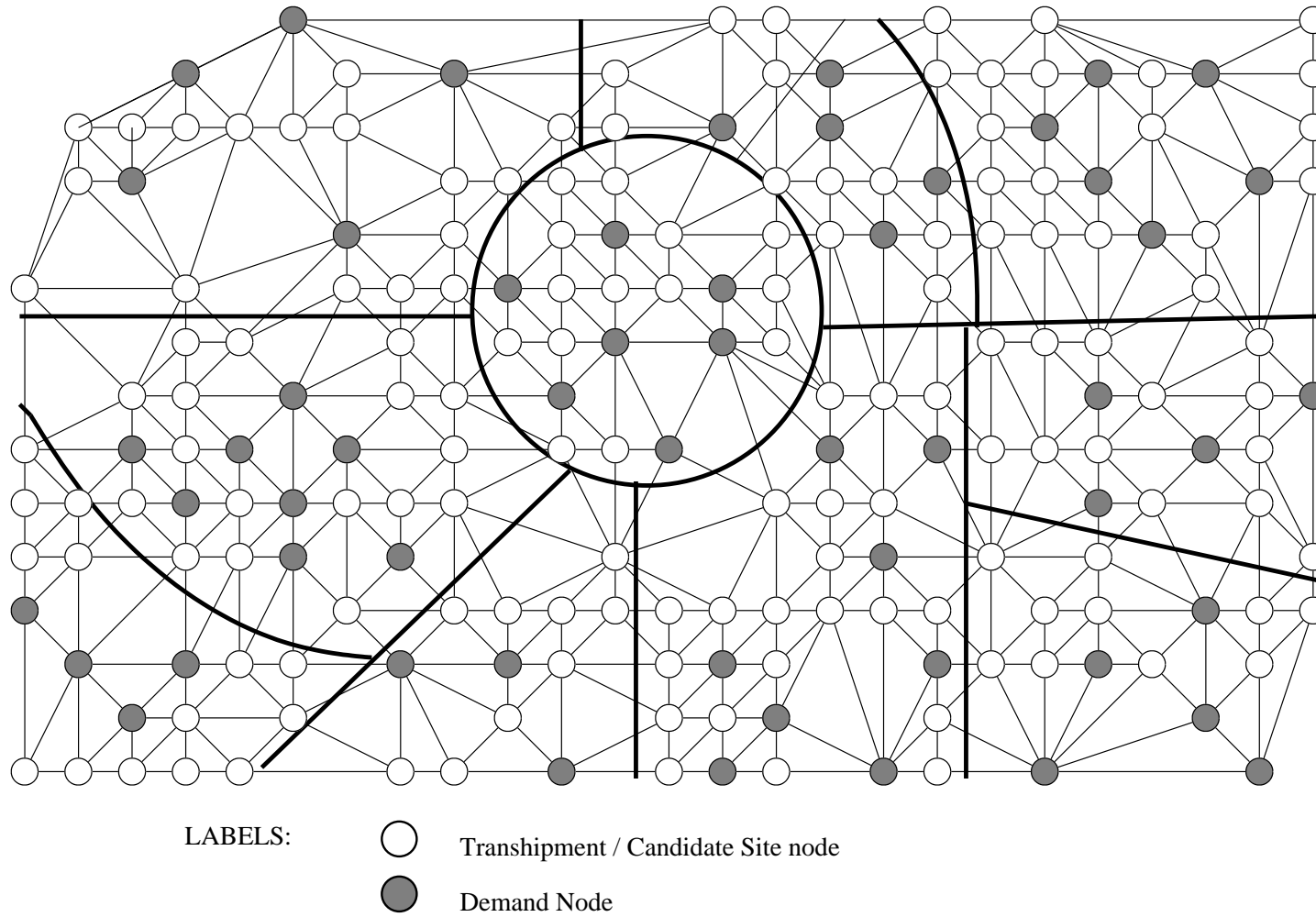
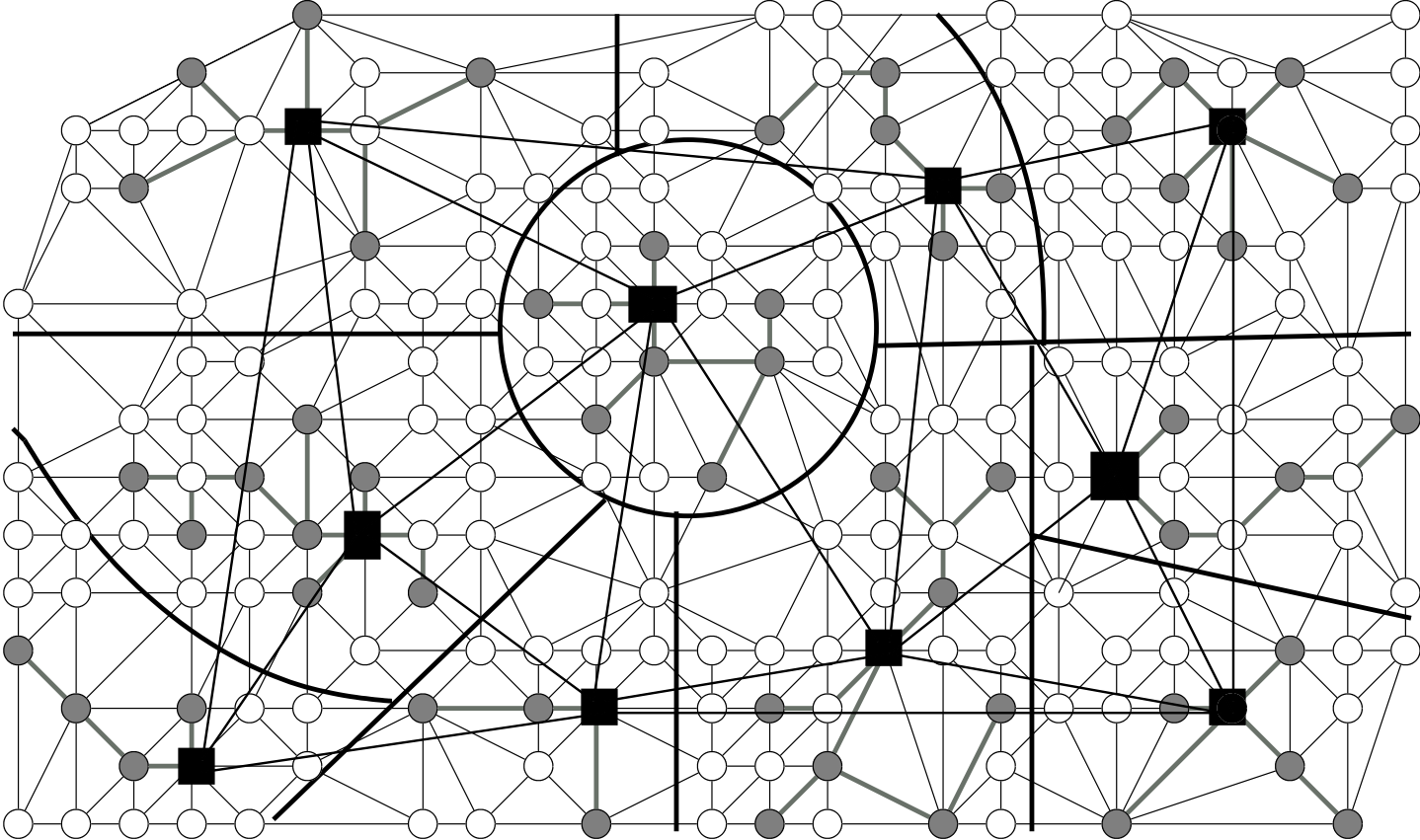


Figure 5.3: A city partitioned in regions








- LABELS:
-  Transshipment / Candidate Site node
 -  Demand Node
 -  Server-to-Server Network
 -  Client-Server Network
 -  Server

Figure 5.4: A feasible solution

Chapter 6

Conclusions and Future Work

6.1 Contributions

6.1.1 The Quadratic Assignment Problem

On the cases where we can define or compute heterogeneous profitabilities, it is possible to solve large instances of *QAP*, without an excessive computational cost or the use of massive parallel computing. This conclusion has its foundations on the pioneer work of Koopmans and Beckmann and also on the work of Heffley, many years later. The inclusion of heterogeneous profits for location is a natural step when considering location theory, and can introduce external environment influence on the location decision process.

Once established this, the new flow formulation has proved to unify desirable qualities for a good mathematical programming implementation: be easy to solve and to grant good linear programming bounds. These two qualities are directly responsible for the good computing times achieved, and also for the solution of large instances at reasonable cost, until now obtainable only through the use of computational grids.

For the instances of size beyond 40, the Benders decomposition algorithm appears to be a good choice to find an exact solution, avoiding excessive space and time complexity, if we observe some conditions about the cost structure, as mentioned in chapter 2.

The new flow formulation and the associated Benders decomposition scheme are original contributions of this work. There is no solution report for instances of *QAP* from $n = 30$ to 64, without the aid of computational grids, even for $p/q > 0$. The current literature of this problem do not address $p/q \neq 0$, losing the *QAP* application framework.

6.1.2 The Placement of Electronics With Thermal Effects

As a subproduct of the solution of large QAP instances, we could derive a performance guarantee heuristic for the electronics placement with thermal effects.

The approach of the specific literature for this problem is with the aid of meta-heuristics, as Genetic Algorithms and Simulated Annealing. This fact makes the performance guarantee heuristic presented in chapter 3 another original contribution of this work.

It is interesting to point that other quantities — depending on the selected application — could be used as secondary quality criteria for the solution. Also, depending on other design parameters, a lot of other kinds of information could be considered for this particular problem. For instance, the maximal temperature over an electronics board or the minimal dispersion of heavy industrial atmospheric residues.

6.1.3 The Local Access Network Design With Congestion Costs

We have also presented a multi-commodity flow formulation for the local access network design problem with congestion costs. We have studied an exact method to solve the problem, based on generalized Benders decomposition. We have also confirmed the possibility of exactly solving such problems with free software mathematical programming packages, and for this purpose we have a surprisingly good alternative to commercial softwares.

Yet the computational experiments have been limited in chapter 4, our experience suggests that a certain number of these modeling and solution strategies can be applied to the frequently occurring problems where the congestion cost component is important. Our conclusion is that generalized Benders decomposition emerges as a good method, since it is enough robust to deal with mixed integer nonlinear programs. The relevance of this type of model is now increasing because emerging Internet services requires larger bandwidth in the local access network and also because applications concerning multi-party multicasting tree construction should take into account congestion costs.

Once established the importance of the congestion costs, the solution of the generated nonlinear mixed-integer programs is already an original contribution of this work.

6.1.4 Integrating Facility Location and Network Design

The obtained results concerning location/network design problems with interdependence on the location of serves are very expressive, as pointed in chapter 5. Future improvements of the Benders decomposition algorithms of chapter 5 will make possible to solve very large instances at reasonable time, if we use parallel computing, specially on the field of interacting hub facilities systems design.

The task of integration of the models of local access network design and facility location with interdependency, using a *QAP* framework, was well accomplished. The produced flow formulation can handle with large instances at reasonable cost.

Our Benders decomposition algorithms are robust enough to deal with large scale hard to solve mixed integer programming problems, designing very complex systems.

It is always important to remember that our local access network is capacitated in this case, computing congestion costs, what ensures the quality of service maintenance for the final customer.

6.2 Hints for Future Work

We are about to obtain expressive results concerning location/network design problems with interdependency on the location of facilities. The future development of a Benders decomposition algorithm of this problem will make possible to solve large instances at reasonable computational cost.

It is also important to remember that our local access network is capacitated, and on the solution of network design at this level, we are computing congestion costs, ensuring the quality of service maintenance for the final customer.

For future work, when considering *QAP*, it is necessary to better explore the equilibrium between bound quality and cost of computation, detecting when and how to merge easy to compute and stronger and hard to compute formulations for a given problem.

In the field of *QAP* applications, it can be noted that a lot of real life problems can be viewed as instances of quadratic assignment problems and that it is sometimes possible to propose other quality criteria for the solution. Nowadays, this kind of solution is valuable, since the environmental impact of any kind of large scale human implementation has becoming more important. It is desirable to have the lower transport cost for intermediate commodities in the design of a regional industrial complex, and to take into account the dispersion of heavy industrial atmospheric residues. There is no doubt that air quality is a great part of quality of life, and that environmental concerns must be in the top of the list of all modern industrial management. One could still deal with the noise level of machinery on the low plant lay-out design.

About local access network design, another question for investigation is the generation of approximate solutions by heuristics with the objective of adding cuts to the Benders master problem in a preprocessing phase (in the current implementation our preprocessing phase only includes the solution of a shortest path problem and a minimal spanning tree problem). These new cuts might reduce the total number of Benders cycles, but the necessary time to solve each master problem might be increased since the master problem would have more constraints. The trade off between these two aspects should be better evaluated.

Concerning the formulations and decomposition schemes presented in chapter 5, additional computational experience is necessary. The investigation of

the cost trade offs for hub-and spoke systems is very important and must add a comprehensive picture of this complex kind of system. It is also interesting to compare the monolithic implementation of traditional hub-and-spoke models and the decomposed versions. The development of a well suited decomposition scheme for the Hub Transshipment Network Design model can add solution power, making possible to solve large instances.

Bibliography

- [1] H.P. Adams and T. Johnson. Improved linear programming bounds for the quadratic assignment problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16:43–77, 1994.
- [2] C.H. Aikens. Facility location models for distribution planning. *European Journal of Operations Research*, 22:263–279, 1985.
- [3] Y.P. Aneja. An integer linear programming approach to Steiner problem in graphs. *Networks*, 10:167–178, 1980.
- [4] K.M. Anstreicher. Eigenvalue bounds versus semidefinite relaxations for the quadratic assignment problem. *SIAM Journal of Optimization*, 11:254–265, 2001.
- [5] K.M. Anstreicher and N.W. Brixius. A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming Series A*, 89:341–357, 2001.
- [6] K.M. Anstreicher, N.W. Brixius, J.P. Goux, and J. Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming Series B*, 91:563–588, 2001.
- [7] T. Aykin. On the location of hub facilities. *Transportation Science*, 22:155–157, 1988.
- [8] T. Aykin. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operations Research*, 46:409–411, 1990.
- [9] T. Aykin. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79:501–523, 1994.
- [10] E. Balas and C. Bergthaller. Benders method revisited. *Journal of Computational and Applied Mathematics*, 9(1):3–12, 1983.
- [11] E. Ballas and M.J. Saltzman. An algorithm for the tree-index assignment problem. *Operations Research*, 39:150–161, 1991.

- [12] A. I. Barvinok. Computational complexity of orbits in representations of symmetric groups. *Adv. Soviet. Math.*, 9:161–182, 1992.
- [13] M.S. Bazaraa and H.D. Sherali. Benders partitioning scheme applied to a new formulation of the quadratic assignment problem. *Naval Research Logistics Quarterly*, 12:29–41, 1980.
- [14] J.E. Beasley. An algorithm for solving large capacitated warehouse location problems. *European Journal of Operations Research*, 33:314–325, 1988.
- [15] J.E. Beasley. An sst-based algorithm for the Steiner problem in graphs. *Networks*, 19:1–16, 1989.
- [16] A. Bejan, G. Tsatsaronis, and M. Moran. *Thermal Design and Optimization*. Interscience, 1st edition, 1995.
- [17] J. F. Benders. Partitioning procedures for solving mixed integer variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.
- [18] R.T. Berger, C.R. Coullard, and M.S. Daskin. Modeling and solving location routing problems with route-length constraints. *Transportation Science*, 1996.
- [19] O. Berman, D.I. Ingco, and A.R. Odoni. Improving the location of minimum facilities through network modification. *The Annals of Operations Research*, 40:1–16, 1992.
- [20] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Rev Univ. Nac. Tucuman*, A(5):147–151, 1946.
- [21] R.R. Boorstyn and H. Frank. Large-scale network topological optimization. *IEEE Transactions on Communications*, COM-25:29–47, 1977.
- [22] D.S. Boyalakuntla and J.Y. Murthy. Hierarchical compact models for simulation of electronic chip packages. *IEEE Transactions on Components and Packaging Technologies*, 25(4):629–634, 2002.
- [23] R. E. Burkard. *Discrete Location Theory: Locations with spatial interactions: the quadratic assignment problem*, volume 1. Wiley and Sons, 1 edition, 1991.
- [24] R.E. Burkard. Selected topics on assignment problems. Technical Report 175, Technische Universität Graz, Austria, November 1999.
- [25] R.E. Burkard and E. Cela. Heuristics for biquadratic assignment problem and their computational comparison. *European Journal of Operations Research*, 83:283–300, 1995.
- [26] R.E. Burkard and E. Cela. The quadratic assignment problem: theory and algorithms. *Kluwer Academic Publishers*, 1(1), 1998.

- [27] R.E. Burkard and E. Cela. Linear assignment problems and extensions. *Handbook of Combinatorial optimization*, 4(1):221–300, 1999.
- [28] R.E. Burkard, E. Cela, P. Pardalos, and L.S. Pitsoulis. The quadratic assignment problem. *Handbook of Combinatorial optimization*, 3(1):241–339, 1998.
- [29] R.E. Burkard, S. Karisch, and F. Rendl. QAPLIB - A quadratic assignment problem library. *European Journal of Operations Research*, 17:115–119, 1991.
- [30] P. Burmann, A. Raman, and S.V. Garimella. Dynamics and topology optimization for piezoelectric fans. *IEEE Transactions on Components and Packaging Technologies*, 25(4):592–600, 2002.
- [31] J. F. Campbell, G. Stiehr, A. T. Ernst, and M. Krishnamoorthy. Solving hub arc location problems on a cluster of workstations. *Parallel Computing*, 29:555–574, 2003.
- [32] J.F. Campbell. Continuous and discrete demand hub location problems. *Transportation Research B*, 27B(6):473–482, 1993.
- [33] J.F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operations Research*, 72:387–405, 1994.
- [34] J.F. Campbell. A survey of network hub location. *Studies in Locational Analysis*, 6:31–49, 1994.
- [35] M. Chandy and K. M. Rusell. The design of multipoint linkages in a teleprocessing tree network. *IEEE Transactions on Communications*, COM-21:1062–1066, 1972.
- [36] N. Christofides and J.E. Beasley. Extensions to a lagrangean relaxation approach for the capacitated warehouse location problem. *European Journal of Operations Research*, 12:19–28, 1983.
- [37] R. Church and C. ReVelle. The maximal covering location problem. *Regional Science Association Papers*, 32:101–118, 1974.
- [38] A. Claus and N. Maculan. Une nouvelle formulation du problème de Steiner sur un graphe. Technical Report 280, Centre de Recherche sur les Transports, Université de Montréal, Canada, 1983.
- [39] K.J. Craig, D.J. Kock, and P. Gauche. Minimization of heat sink mass using CFD and mathematical optimization. *Journal of Electronics Manufacturing*, 121(3):143–147, 1999.
- [40] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1962.

- [41] Z. Drezner and G.O. Wesolowsky. Network design: Selection and design of links and facility location. *Transportation Research Part A*, 37:241–256, 2003.
- [42] R.P.M. Ferreira and H. P. L. Luna. Discrete capacity and flow assignment algorithms with performance guarantee. *Computer Communications*, 26:1056–1069, May 2003.
- [43] G. Finke, R.E. Burkard, and F. Rendl. Quadratic assignment problems. *Annals of Discrete Mathematics*, 31:61–82, 1987.
- [44] L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
- [45] P.M. Franca and H.P.L. Luna. Solving stochastic transportation-location problems by generalized Benders decomposition. *Transportation Science*, 16:113–126, 1982.
- [46] A.M. Frieze and L. Yadegar. On the quadratic assignment problem. *Discrete Applied Mathematics*, 5:89–98, 1983.
- [47] B. Gavish. Topological design of centralized computer networks - formulations and algorithms. *Networks*, 12:355–377, 1982.
- [48] B. Gavish. Formulations and algorithms for the capacitated minimal directed tree. *Journal of the ACM*, 30:118–132, 1983.
- [49] B. Gavish. Augmented Lagrangean based algorithms for centralized network design. *IEEE Transactions on Communications*, COM-33:1247–1257, 1985.
- [50] B. Gavish. Topological design of telecommunication networks - Local access design methods. *Annals of Operations Research*, 33:17–71, 1991.
- [51] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory And Applications*, 10:237–260, 1972.
- [52] A.M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
- [53] A.M. Geoffrion and G.W. Graves. Multicommodity distribution system design by Benders decomposition. *Management Science*, 20:822–844, 1974.
- [54] M. Gerla and L. Kleinrock. On the topological design of distributed computer networks. *IEEE Transactions on Communications*, 25:48–60, 1977.
- [55] M. X. Goemans and Y.-S. Mying. A catalogue of Steiner tree formulations. *Networks*, 23:19–28, 1993.
- [56] M. C. Goldstein. Design of long-distance telecommunication networks - the Telepak problem. *IEEE Transactions on Circuit Theory*, CT-20:186–192, 1973.

- [57] L. Gouveia. A comparison of directed formulations for the capacitated minimal spanning tree problem. *Telecommunication Systems*, 1:51–76, 1993.
- [58] K.S.S. Gupta and P.K. Srimani. Core-based tree with forwarding regions (cbt-fr); a protocol for reliable multicasting in ad hoc networks. *Journal of Parallel and Distributed Computing*, 61:1249–1277, 2001.
- [59] P. Hahn and T. Grant. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research*, 46(6):912–922, 1998.
- [60] S.L. Hakimi. Optimum location of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.
- [61] D.R. Heffley. The quadratic assignment problem: A note. *Econometrica*, 40(6):1155–1163, 1972.
- [62] J. Hellstrand, T. Larsson, and A. Migdalas. A characterization of the uncapacitated network design polytope. *Operations Research Letters*, 12:159–163, 1992.
- [63] D.S. Hochbaum and A. Segev. Analysis of a flow problem with fixed charges. *Networks*, 19:291–312, 1989.
- [64] Y.J. Huang and S.L. Fu. Thermal placement design for mcm applications. *Journal of Electronic Packaging*, 122(2):115–120, 2000.
- [65] Y.J. Huang, S.L. Fu, S.L. Jen, and M.H. Guo. Fuzzy thermal modeling for mcm placement. *Microelectronics Journal*, 32(10):836–868, 2001.
- [66] Y.J. Huang, M.H. Guo, and S.L. Fu. Reliability and routability consideration for mcm placement. *Microelectronics Reliability*, 42(1):83–91, 2002.
- [67] F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 22:55–89, 1992.
- [68] F. K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem*. North-Holland, 1992.
- [69] P. J. Juell, D. Brekke, and R. Vetter. A multicast tree constructions algorithm for large multiparty conferences. *Telecommunication Systems*, 17:299–321, 2001.
- [70] M. Junger and V. Kaibel. On the SAQP polytope. *SIAM Journal of Optimization*, 11:444–463, 2000.
- [71] M. Junger and V. Kaibel. Box-inequalities for quadratic assignment polytopes. *Math. program. Ser. A*, 91:175–197, 2001.

- [72] D. Katanyutaveetip. Real-time optimal multicast routing. *Computer Communications*, 2002. Article in Press.
- [73] L. Kaufman and F. Broeckx. An algorithm for the quadratic assignment problem using Benders decomposition. *European Journal of Operations Research*, 2:204–211, 1978.
- [74] J.G. Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6:307–335, 1998.
- [75] T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998.
- [76] D. Konig. Graphok es matrixok. *Mat. Fiz. Lapok*, 38:116–119, 1931.
- [77] T.C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.
- [78] A.A. Kuehn and M.J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9:643–666, 1963.
- [79] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–98, 1955.
- [80] E.L. Lawler. The quadratic assignment problem. *Management Science*, 9:586–599, 1963.
- [81] S. Lorente, W. Wechsato, and A. Bejan. Optimization of tree shaped flow distribution structures over a disc disc-shaped area. *Int. Journal of Energy Research*, 27(8):715–723, 2003.
- [82] A. Losch. *The Economics of Location*. New Haven, 1 edition, 1954. Translated from German edition 1944.
- [83] A. Lucena and J. E. Beasley. A branch and cut algorithm for the Steiner problem in graphs. *Networks*, 31:39–59, 1998.
- [84] H. P. L. Luna and P. Mahey. Bounds for global optimization of capacity expansion and flow assignment problems. *Operations Research Letters*, 26:211–216, 2000.
- [85] H.P.L. Luna, N. Ziviani, and R.M.B. Cabral. The telephonic switching centre network problem: Formalization and computational experience. *Discrete Applied Mathematics*, 18:199–210, 1987.
- [86] N. Maculan. A new linear programming formulation for the shortest s-directed spanning tree problem. *Journal of Combinatorics, Information & Systems Sciences*, 11:53–56, 1986.
- [87] N. Maculan. The Steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.

- [88] N. Maculan, D. Arpin, and S. Nguyen. Le problème de Steiner sur un graphe orienté: Formulations et relaxations. *Matemática Aplicada e Computacional*, 7:109–118, 1988.
- [89] T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–483, 1981.
- [90] T.L. Magnanti, P. Mirchandani, and R.T. Wong. Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:112–154, 1986.
- [91] A. Makhorin. GLPK - The GNU linear programming kit. Technical Report 1, Moskow Aviation Institute, Moscow, Russia, 2001. available in <http://www.gnu.org/glpk>.
- [92] G.R. Mateus and H.P.L Luna. Decentralized decision-making and capacitated facility location. *The Annals of Regional Science*, 26:361–377, 1992.
- [93] G.R. Mateus, H.P.L. Luna, and A.B. Sirihal. Heuristics for distribution network design in telecommunication. *Journal of Heuristics*, 6(Special Number on Telecommunications):131–148, 2000.
- [94] G.R. Mateus and J.M. Thizy. Exact sequential choices of locations in a network. *Annals of Operations Research*, 86:199–219, 1999.
- [95] K. Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.
- [96] S. Melkote and M.S. Daskin. Capacitated facility location/network design problems. *European Journal of Operations Research*, 129:481–495, 2001.
- [97] S. Melkote and M.S. Daskin. An integrated model of facility location and transportation network design. *Transportation Research Part A*, 35:515–538, 2001.
- [98] E.S. Mills. The efficiency of spatial competition. *The Regional Science Association Papers*, 25:71–82, 1970.
- [99] T.S. Motzkin. The assignment problem. In *Proceedings of Symposia in Applied Mathematics*, Numerical Analysis. McGraw-Hill, 1956.
- [100] E.Y.K. Ng, C.P. Tso, Z.M. Wen, and K.F. Choo. Numerical simulation of flow and conjugate heat transfer in a microchannel for electronics cooling. *Journal of Electronics Manufacturing*, 9(2):141–153, 1999.
- [101] M. O’Kelly. The location of interacting hub facilities. *Transportation Science*, 20:92–106, 1986.

- [102] M. O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 1986.
- [103] M. O’Kelly, D. Bryan, , D. Skorin-Kapov, and J. Skorin-Kapov. Hub network design and multiple allocation: A computational study. *Location Science*, 4:125–138, 1986.
- [104] M. O’Kelly and M.W. Horner. Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, 9:255–265, 2001.
- [105] M. O’Kelly and H.L. Miller. The hub network design problem. *Journal of Transport Geography*, 2(1):31–40, 1994.
- [106] M.W. Padberg and M.P. Rijal. *Location, Scheduling, Design and Integer Programming*. Kluwer Academic Publishers, Boston, 1 edition, 1996.
- [107] P. Pardalos, F. Rendl, and H. Wolkowickz. The quadratic assignment problem: a survey of recent developments. *DIMACS Series on Discrete Mathematics and Computer Science*, 16:1–42, 1994.
- [108] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1st. edition, 1980.
- [109] H. Pirkul and J. Current. The hierarquical network design problem with transshipment facilities. *European Journal of Operational Research*, 52:338–347, 1991.
- [110] H. Pirkul and D.A. Schiling. An efficient procedure for designing single allocation hub and spoke systems. *Management Science*, 44:235–242, 1998.
- [111] N.V. Queipo, R. Devarakonda, and J.A.C. Humphrey. Genetic algorithms for thermosciences research - application to the optimized cooling of electronic components. *Int. Journal of Heat and Mass Transfer*, 37(6):893–908, 1994.
- [112] N.V. Queipo and G.F. Gil. Multiobjective optimal placement of convectively cooled electronic components on printed wiring boards. *IEEE Transactions on Components Packaging and Manufacturing Technology*, 21(1):142–153, 1998.
- [113] N.V. Queipo and G.F. Gil. Multiobjective optimal placement of convectively and condutively cooled electronic components on printed wiring boards. *Journal of Electronic Packaging*, 122(2):152–159, 2000.
- [114] M. Queiroz and C. Humes. The projected pairwise multicommodity flow polyhedron. *Applied Mathematics Letters*, 14(4):443–448, May 2001.
- [115] K.G. Ramakrishnan, M.G.C. Resende, B. Ramachandran, and J.F. Penky. *Tight QAP Bounds Via Linear Programming*. World Scientific Publishing Co., Singapore, 1 edition, 2002.

- [116] C.D. Randazzo and H.P.L. Luna. A comparison of optimal methods for local access uncapacitated network design. *The Annals of Operations Research*, 106:263–286, 2001.
- [117] L.A.O. Rocha, S. Lorente, and A. Bejan. Constructal design for cooling a disc-shaped area by conduction. *Int. Journal of Heat and Mass Transfer*, 45(8):1643–1652, 2002.
- [118] J.L. Rosales, A. Ortega, and J.A.C. Humphrey. A numerical simulation of the convective heat transfer in confined channel flow past square cylinders: comparison of inline and offset tandem pairs. *Int. Journal of Heat and Mass Transfer*, 44(3):587–603, 2001.
- [119] B. Rothfarb and M. C. Goldstein. The one-terminal Telepak problem. *Operations Research*, 19:156–169, 1971.
- [120] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of Association of Computing Machinery*, 23:555–565, 1976.
- [121] P.A. Samuelson. Spatial price equilibrium and linear programming. *American Economic Review*, 42:284–303, 1952.
- [122] L. Schwiebert and R. Chintalapati. Improved fault recovery for core based trees. *Computer Communications*, 23:816–824, 2000.
- [123] D.S. Steinberg. *Cooling Techniques for Electronic Equipment*. Interscience, 2nd edition, 1991.
- [124] C. Toregas, R. Swain, C. ReVelle, and L. Bergmann. The location of emergency service facilities. *Operations Research*, 19:1363–1373, 1971.
- [125] P.G. Tucker. Models aid the analysis of electronics cooling. *Microwaves and RF*, 40(6):95–96, 2001.
- [126] J.A. Visser and Kock D.J. Optimization of heat sink mass using the DYNAMIC-Q numerical optimization. *Communications in Numerical Methods in Engineering*, 18(10):721–727, 2002.
- [127] W. Wechsato, S. Lorente, and A. Bejan. Optimal tree-shaped networks for fluid flow in a disc disc-shaped body. *Int. Journal of Heat and Mass Transfer*, 45(25):4911–4924, 2002.
- [128] P. Winter. Steiner problem in networks: a survey. *Networks*, 17:129–167, 1987.
- [129] R.T. Wong. A dual ascent algorithm for the Steiner problem in directed graphs. *Mathematical Programming*, 28:271–287, 1984.
- [130] G. Zapfel and M. wasner. Planning and optimization of hub-and-spoke transportations networks of cooperative third-party logistics providers. *International Journal of Production Economics*, 78:207–220, 2002.

- [131] Z.J. Zuo, L.R. Hoover, and A.L. Phillips. Advanced thermal architecture for cooling of high power electronics. *IEEE Transactions on Components and Packaging Technologies*, 25(4):629–634, 2002.