Hub Location Models in Public Transport Planning

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte

Dissertation

von

Shahin Gelareh

Erstgutachter: Prof. Dr. Stefan Nickel

Zweitgutachterin: Prof. Dr. Elena Fernández

Datum der Disputation: 21. April 2008

D 386

Dedicated to

Agha, Baba, Rahimeh, Azin
and the memory of Azizeh.

Preface

In the name of God

First of all, I must be thankful of who owns the creation and supports me throughout my life. There is no doubt that all what I gained so far owes to his auspice and attention.

The outcome of this research, whatever it is, has strongly been influenced by many people who accompanied me during the whole or part of the work in my private and academic life. I take this opportunity to express my sincere thanks to them and appreciate their support.

The most, I am grateful and owing my family, parents and especially my father, Shaban Ali Gelareh and my grandfather, Abd Al-Rahim Chaloshtar Mokari who always supported me throughout my education. They initiated the first steps and continuously supported me during this work to reach this milestone. They have been and will always be paragons in my mind. Forever, I will be indebted to Ms. Sedigheh Makaremi and her family for their endless support.

I am very much indebted to my wife, Rahimeh Nematian Monemi for her endless patience in recent years which shared with me. Her support strongly influenced my work.

I would like to thank Prof. Dr. Stefan Nickel as my supervisor for offering me the opportunity, necessary scientific freedom and support to carry out this dissertation. Without his support and patience, this dissertation would not reach to this point. Special thanks to Prof. Dr. Elena Fernández from Universitat Politécnica de Catalunya as the second referee of my thesis. Many thanks to Prof. Dr. Horst. W. Hamacher for offering me the opportunity in Hub Location workshop. I would like to thank Dr. Alfredo Marín-Pérez from Universidad de Murcia (Spain) and Prof. Dr. Bernard Fortz from Université Libre de Bruxelles for our inspiring and fruitful discussion and their useful hints.

Armin, Azin, Bahram, Maziar, Bobak, Shahriar Makaremi, Sassan Khajehali and Nazira Heinrich deserve my gratitude for their support.

Finally, I acknowledge the Department of Optimization of Fraunhofer Institute for Industrial Mathematics (ITWM) for offering me a motivating scientific environment to carry out this work.

Kaiserslautern, May 2008 Shahin Gelareh

Contents

Chapter 1

Introduction

Among thousands of reasons for motivating so much of accomplished and on-going research and studies in logistics, transportation and telecommunications, we can point to: 1) the globalization, 2) increase in the number of highly competitive service providers, 3) emerge of new economic powers and new markets, 4) crowded transportation networks, 5) daily increase in the origin-destination services in terms of passengers, commodities and so on, 6) continuous increase in the mass of multi-media data transfer in telecommunication and IT industries, 7) population growth, 8) energy concerns and global warming, 9) environmental pollution and, 10) safety and security policies.

Operations Research (OR) plays a key role and offers very efficient tools for such studies. Mathematical models capable of describing such systems, simulations tools to analyze and predict behavior by what-if scenarios etc. are among those offered by OR. The research in such areas often center on the study of network structures and architectures. Several configurations have already been proposed and many promising issues are explored. Among all problems arising in this growing body, Hub Location Problems (HLP) have received a lot of attention in the last two decades. This new topology showed itself to be superior in terms of system performance, reliability and applicability, which results.

1.1 Hub Location Problems

The idea of hub-and-spoke networks initiated from Goldman [35] in 1969, which was followed by O'Kelly [62, 63] in 1986 incepted the primitive study of hub-and-spoke networks on a plane. O'Kelly was also the one who proposed the first mathematical formulation of hub-and-spoke networks as a quadratic integer programming model in 1987 [64]. Since that time, many researchers have been working on different classes of problems in HLPs; both in theoretical aspects and different applications. Also, many contributions are devoted to them. This amount of attention is mainly due to the necessity and implication of *modern* infrastructures for both transportation and telecommunication systems [12]. Among this amount of work on HLPs, public transport applications have received least amount of attention while we believe deserve much more.

The term *modern* reflects the functionality of these configurations that does not follow the traditional network service modes. In traditional service networks, every demand point expecting to meet its demand from an origin, is served by that origin through a direct interaction. Contrarily, in such modern systems, some of the nodes in the network are selected to build up an intermediate structure. Subsequently, demands of many destinations are served by origins via this intermediate structure. This structure (the so called hub-level network) is in charge of handling the routing of flow. It exploits the economy of scale arising from its special functionality that motivates making use of this new topology (the impact of this scale economy is drawn by a discount factor applied to any transportation cost inside the hub-level network). These new models and structures are studied in the context of Hub Location Problems. The location of hub facilities is fundamental to the design of a hub-and-spoke network because, it affects the total transportation cost of the system, the throughput level at central facilities and hence, the service time and congestion [64].

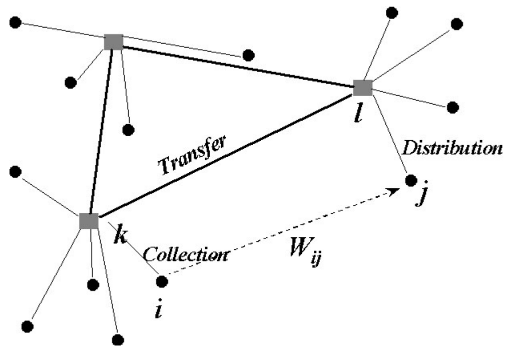


Figure 1.1: A typical Hub Location Problem network

The hub location problems are usually NP-hard problems [45]. A typical HLP network is depicted in Figure 1.1. The rectangles stand for hub nodes and the bold face circles for spoke nodes. The hub nodes together with the edges connecting them is called the hub-level and the rest, spoke-level network. In any origin-destination path there exists at least one hub element (hub node or edge). Rather than directly connecting any pair of locations, all the paths are handled by the hub-level network.

As an example of application of such networks in telecommunication we have the following: A call request is sent from i (a spoke node) to its related local call center k to make a connection to j; the call center (hub node) that receives plenty of such requests per second, checks whether it is authorized to fulfill the request (i.e. the destination is assigned to it) or it has to be handled by another call center; if yes, it establishes this connection together with all other calls which address j, else delivers this request together with all others which should be handled by l to it; l establishes a connection to j.

Although, there exists a lot of work on HLPs related to telecommunication and transportation, however, the especial attributes of other applications and also classical assumptions of HLPs are not always suitable for the public transport and are not adequately depicting the facts and features of this application. Subsequently, regarding the behavior and characteristic of public transport systems, HLPs deserve an exclusive part in such studies.

Public transportation is a very solicitous/challenging problem in many developed and developing countries. In developed countries, sample issues can be for example the effect of incompetence of transportation systems on the energy policies and concerns, quality of life or performance of other systems and services. This shows itself in form of degraded service-level. In the developing countries, it is important to improve and bring up the systems that somehow are affected by the inefficiency in the performance of public transport systems. This includes the customer service systems, energy policies and systems which contribute as indicators of development in such countries.

We propose a mathematical model for the application of such a structure in public transport planning [34].

1.2 Multi-Period Planning

In reality, decisions are often not made just for a single period. Usually, when a project concerns some construction activities and significant investments, there exists a planning horizon with several periods for which decisions are made. That is, the initial configuration of the network might change due to many factors (for instance the expected changes in the mass of transportation, re-shaped spatial distribution of the flows and many other economic issues). In each period of the planing horizon, the configuration of the previous period is updated and evolved until the end of the planning horizon is met. The planning horizon length may differs, however in terms of public transport, we expect it to be a long term (e.g. 10 years, 30 years etc.).

Briefly speaking, in a multi period planning approach, the transport network evolves throughout the planning horizon. Decisions about how the network should evolve are not made just in an improvident and myopic way. This is usually a function of changes in the patterns of distribution of flow density, emerging technologies, financial issues etc.

While modeling for a planning horizon, many other aspects of real-life can also be taken into account. For example, benefiting from the full performance of a hub facility in a period t, is only possible in the subsequent periods $(t+1,\, t+2,\, \dots)$ if we invest on maintenance operations thus overcoming the depreciation of facilities. Of course, when a hub facility no longer acts as a hub-level element, it incurs additional costs to transform it to a spoke element. This concerns both hub nodes and hub edges. Concerning the hub nodes, it may amount to the cost of re-training or firing of employees, ceasing costs etc. In the hub edge level, it concerns the removal of elements of fast-lines which need to be paid for maintenance and holding costs as well as a non-ignorable removal cost to shut down and uninstall (e.g. special navigation systems, particular type of vehicles, special service centers etc.).

1.3 Solution Procedures

Our models here are proposed for real-life applications. Such mathematical models of the problems usually lead to very large and hard-to-solve problems, in many cases to NP-hard problems. This is the indispensable feature of many real life applications. The combinatorial aspect of problems in network design, transportation and logistics are important issue that usually makes real-life size problems in these areas very hard to solve.

Usually, depending on how difficult a class of problems is, for a given machine specification, existing standard solvers are only capable of solving instances with very limited size. The more precisely the model tries to approximate the reality, the smaller the size of instances which can be solved to optimality using existing standard solvers. By exploiting the special structure of models and problems, it may be possible to solve larger instances of the problems and/or even reduce the computational time.

Some decomposition procedures exploiting the special structure of the problems already exist. They try to decompose and separately solve smaller and easier-to-solve parts of the problems to achieve the original optimal solutions. Among these approaches, Lagrangian relaxation, column generation and Benders decomposition are shown to be useful and applicable for many problems, especially combinatorial optimization problems. We use some of them as far as they can help to efficiently solve instances of our problems.

In many real-life applications, it may suffice to retrieve very good solutions in a very short amount of time. In many cases, the mathematical model itself might not be precisely depicting the reality and thus be a rough approximation of real-life problem. Another issue is the existence of approximated data which additionally imposes some imprecision. In these cases, it may not be really necessary to insist on solving the problem to optimality since this may take days or months of computations (which is quite normal in most of real-life applications with combinatorial nature). Often, an expected-to-be-good-enough solution can be found in significantly smaller amount of time (like some seconds or minutes). The dramatic increase in the size of the search space, non-stationary nature of the environment and the necessity for real-time responses motivated many researchers into solving combinatorial problems using heuristic techniques.

1.4 Overview

This dissertation aims to propose new mathematical models for hub location problems in public transport in which the economy of scale is exploited from consolidation of flow. Both, single and multi-period scenarios are taken into account. In addition to that, efficient solution methods are proposed to solve instances of such problems.

The second chapter introduces the Hub Location Problems and their characteristics along with a literature review with respect to the special attributes presented by each variant in this class of problems.

The third chapter presents a new (base) model for hub location problems applied on public transport. This model is a mixed integer problem and is applied on the Civil Aeronautics Board (CAB) [64] and Australian Post (AP) [27] datasets from OR-Library [1]. The new model generalizes the classical hub location model by relaxing certain assumptions. The quality of our model is compared against the existing HLP models for this application. In addition to the base model, some other variants of the base model in which more realistic aspects of the application are drawn are also proposed.

The fourth chapter proves the integrality property of a chosen sub-problem for a Lagrangian relaxation proposed to solve instances of the model.

In the fifth chapter, a strengthening of the formulation by means of some valid inequalities and preprocessing is carried out. The computational results report the strength of the final formulation.

[1] www.people.brunel.ac.uk/~mastjjb/jeb/info.html. The CAB instances are based on the airline passenger interaction between 25 cities in U.S. in 1970. This dataset is a part of a 100-city data set related to the 100 larger places in U.S. urban system and amounted for 51% of the total flow observed in U.S. . The AP is based on a postal delivery information between 200 postal districts in Sydney. Also, the flow matrix of AP is not symmetrical.

In the sixth chapter, an exact algorithm is proposed to solve instances of the problem. Also, different modifications of the standard Benders decomposition are proposed.

As we will see in the seventh chapter, a heuristic method is developed to solve large instances of the problem which can not solved by exact methods. This heuristic method is a kind of greedy neighborhood search equipped with some improvement procedures (diversification and intensification local searches). The quality of the heuristic method against the standard solver is also examined.

In the eighth chapter, the first multi-period HLP model and its variants for public transport planning are proposed. These models generalize the single period model of chapter 2.

The ninth chapter deals with the extension of the neighborhood search for the multi-period model. The quality of the heuristic solution is studied here.

Finally, in the tenth chapter, this work is concluded and some avenues for further development in future work are suggested.

Chapter 2

Hub Location Problems

In this chapter, after introducing the hub location problems and their properties, we review the literature in a classified manner.

2.1 Problem Description

The classical problem is stated as follows: *Let G be a complete graph $G = (V, E)$ where $V = \{v_1, v_2 \ldots, v_n\}$ is the set of all vertices. The elements of V are assumed to represent origins and destinations and at the same time are potential points for establishing hubs. The flow between node i and node j is w_{ij} and the distance from node i to node j is d_{ij}, where the distance satisfies the triangle inequality. The aim is to designate some of these vertices as hubs and minimize the total flow cost in network. Each origin-destination path consists of three components: collection from origin to the first hub, transfer between the first hub and last hub and distribution from the last hub to the destinations. Paths containing only one hub node are also allowed. The parameters α, χ and δ are discount factors related to each of the three components, respectively.*

In these networks, some of the nodes are selected to act as so called *hub nodes*. Any non-hub node is known as a *spoke node*. Consequently, a *hub-level network* is formed by connecting pairs of hub nodes by a *hub edge*. Eventually, each spoke node will be allocated to these hub nodes by *spoke edge* links. The network composed of the spoke nodes and spoke edges in different areas called *spoke-level*, *tributary* or *access* networks, by different authors.

A hub node can simultaneously have three different functionalities (*see* Figure 2.1), namely:

i). *Consolidation* (concentration) of flows that receives, in order to have a larger flow and letting economy of scale to be exploited;

ii). *Switching* (transfer) which allow the flows to be re-directed at the node;

iii). *Distribution* (decomposition) of large flows into smaller ones.

A hub node receives flows from many origins and *consolidates* (accumulates) them. This consolidated flow splits up into several groups of accumulated flows (concentrated pieces) according to their final destinations. Each of these groups contains flows of many destinations and will be sent through hub edge(s). This happens at all hub

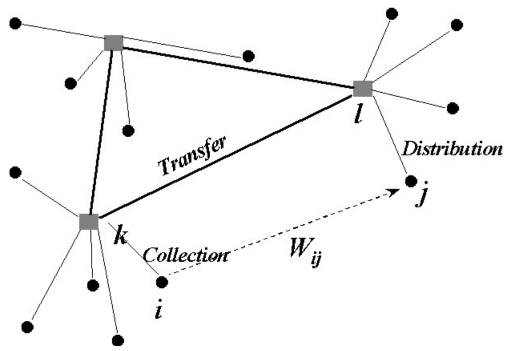


Figure 2.1: A typical Hub Location Problem network

nodes in the network. Each piece of flow at its last visited hub node along the path in the hub-level network joins to other pieces coming from different hub nodes in a similar way. This concentrated flow should be again decomposed to meet the demand of the current hub node and also spoke nodes assigned to.

Thus, hubs are intermediate points along the paths followed by origin-destinations [12]. Of course, the situation in which a hub node is itself an origin or destination is also allowed.

Figure 2.1 sheds more light on the topology of such structures. The rectangles stand for hub nodes and the bold face circles for spoke origin/destination points. When compared with a classical transportation network with the same number of nodes (like in Figure 2.2), the number of connections is much smaller. In the latter case, very small and sometimes ignorable amount of flow may exist on some links, where this is less likely to happen in the network in Figure 2.1.

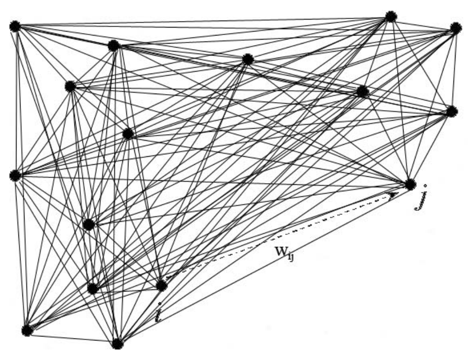


Figure 2.2: A Classical Transportation Network.

Although, a fully interconnected structure like in Figure 2.2 rarely exists in the reality, however, it can be used to clarify the idea of HLP networks.

2.1.1 Examples of Real-Life Applications

Two other well-know applications of HLPs are telecommunications and cargo systems.

Telecommunications

A simple and clear example can be the call centers which connect us from our house to another houses. In Figure 2.1, consider a person at location i wanting to make a call to the house in location j. In order to make this call, in practice, such procedure takes place: we dial a number from i and our request is delivered to the call-center at k. This call center determines whether this call should be delivered to a destination which is allocated to itself or to a destination which is allocated to another hub node, say l. In the first case, it switches directly to the destination and in the latter, it delivers this request to the operator at the call-center l. Now, l delivers the call to the destination, namely j.

The flow is the packets of multimedia data and the flow cost is the cost of keeping this connection active for a certain time interval.

Cargo and Transportation

As an example, a real hub network of a transportation company in North America is depicted in Figure 2.3. In this figure, the bold faced rectangles stand for hub nodes and the edges connecting them are hub edges. This hub network is used to transport the shipments of an express shipment company. Rather than existing direct connections between pairs of spoke cities like A and B, shipments is carried to the first hub, traverse the hub edges and will finally be delivered to the hub node where the customer(s) are assigned to.

In Figure 2.4, flows of origin A is delivered to the hub node of Omaha city and transported by the company vehicles through hub edges and customer B eventually receives his shipment from the hub node in Memphis city. Obviously, fewer links are used in this network in comparison to traditional networks and many direct links are omitted.

Here, in Figure 2.4, the hub-level network is *not* a complete graph and the number of hub edges in a path can be more than one.

For urban transportation systems, we have an analogy as depicted in Table 2.1 between elements of a public transport network and HLP networks.

Figure 2.5 depicts such an urban transport network for the metropolis of Munich in Germany. Usually, in a metropolitan transport network, the distance between origins

Figure 2.3: Hub-level network of an express shipment company in North America.

Figure 2.4: Shipment from A to B in the hub-and-spoke network of a transportation company in North America.

and destinations are quite considerable. The inefficiency of the transportation network can have significant effects on the service level of many other service sectors.

As one can see in Figure 2.6, the city center and its nearby areas, main street boule-

Table 2.1: Hub-and-Spokes vs. urban traffic.

Urban Traffic	Hub-and-Spoke
Buses and taxis	Spoke edge vehicles
Subways, Metro and fast-lines	Hub edge facilities
Subways and Metro stations	Hub nodes
Bus and taxi stations	Spoke node

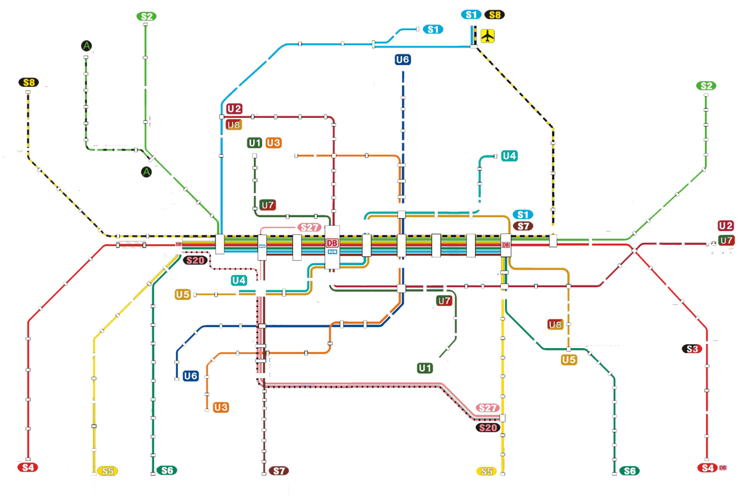


Figure 2.5: Public Transport Network of Munich, Germany.

vards and highways which are the most demanding regions (regarding the density of flow) in each city are the locations where deserve to receive fast-lines.

In the other words, fast-lines are the links connecting centers of districts to each other and to the city center. The spoke-levels are buses which are internal to the districts.

2.2 Traits of Hub-and-Spoke Networks

Like any other structure, hub-and-spoke networks have some advantages and also disadvantages. In this section we briefly note some of them.

2.2.1 Advantages

Some major advantages of these structures are listed here:

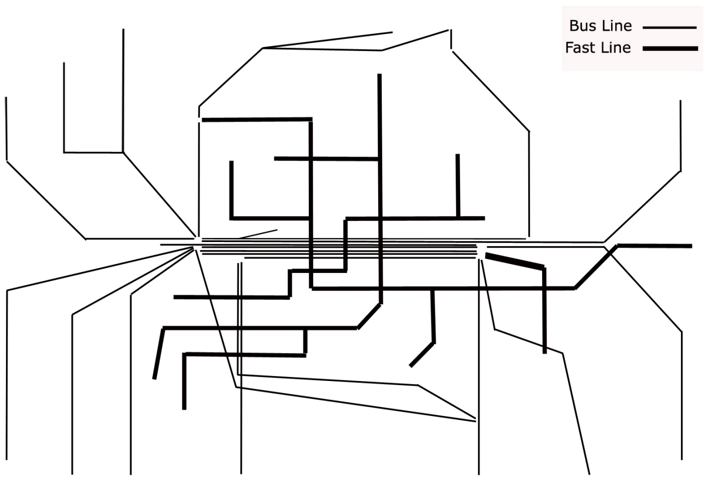


Figure 2.6: Public Transport Network of Munich, Germany (Hub-and-spoke).

1. *Economies of Scale*: The reduction of cost per unit of flow of commodity or passenger caused by the consolidation of flows on larger connections, trucks or aircrafts (increase in service). The more increase in service, the more reduction there is in service cost of additional demands.

2. *Economies of Scope:* The situation in which the cost of performing multiple jobs simultaneously (shared transshipment facilities), is more efficient than performing each job separately. Here, hubs are susceptible to perform three roles.

3. *Locational Issues*:

   (a) Central geographic position

   (b) High demand in area

   (c) Distance to a hub of a competitive company

   (d) Cultural/Economic importance

   (e) Weather conditions

   (f) Infrastructure

4. *Multiplier Effect:* The effect arising when a small change in investment on establishing hub facilities makes a non-proportionate change in the aggregated demand. In another point of view, the construction project employs worker and indirectly, it will stimulate employment in laundries, restaurants and service industries in the vicinity of facility.

5. *Economies of Density:* This condition arise when the cost of service is reduced due to the increase in the demand density rather than the distributed demand and lower density.

2.2.2 Disadvantages

Some disadvantages of such structures can be as follows:

1. Longer travel times and higher costs of some routes,
2. Capacity overload,
3. Higher risk of accident (congestion phenomena) and,
4. Missing connecting facilities due to the unforeseen delay (interrupt) at some parts of the network.

2.3 Variants

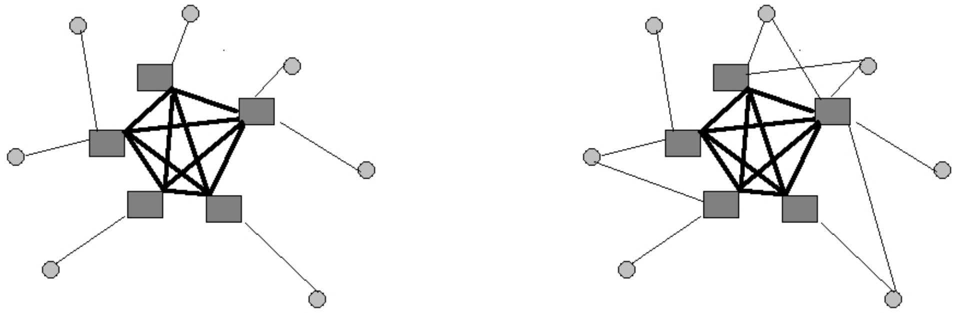
Some of the most important variants of HLPs are as follows:

- **p-Hub Median Problem (pHMP):** Given the number of hubs ,p , the objective is to locate them and minimize the transportation cost;
- **Hub Location Problem (HLP):** The objective is to minimize the total costs. The total cost is the cost of establishing hub facilities (nodes and/or edges) plus transportation costs;
- **p-Hub Center Problem (pHCP):** Given the number of hubs, p, the objective is to minimize the maximum cost for each origin-destination pair, on each single link or for movements between a hub and origin/destination;
- **Hub Covering Problem (HCP):** The objective is to minimize the total transportation cost. Transportation costs should not exceed a certain threshold (on origin-destination pairs, each link or between hub and origin/destination);
- **Hub Arc Location Problem (HALP):** Given the number of hub arcs, q, the objective is to locate hub arcs and minimize the total transportation cost. The hub network is not necessarily a complete graph. Moreover, there is no necessity to have discount on all the hub edges. The location of hub nodes are identified from the location of hub arcs.

This dissertation will be based on the Hub Location Problem (HLP), i.e. the second variant.

2.3.1 Single vs. Multiple Allocation

A Hub Location Problem, in general, includes two simultaneous problems: locating hub nodes and allocating spoke nodes to hubs.
With respect to the allocation schemes, HLPs can be categorized into two classes. The first class, in which a single hub node should receive (and deliver) the whole flow of an origin (or destination) node (hub or spoke) assigned to, is known as *Single Allocation (SA)*. In the second class, the activities of a spoke node can be processed by more than one hub node. The latter is called *Multiple Allocation (MA)*.



(a) Single allcoation **(b) Multiple allcoation**

Figure 2.7: Assignment schemes in HLPs

In Figure 2.7, **(a)** depicts a single allocation scheme and **(b)** a multiple allocation configuration.

2.3.2 Capacities

Different types of capacities are also considered for different variants. For instance,

- Capacity on the amount of flow arriving to the hub nodes from non-hub nodes,
- Capacity on the all through traffic and,
- Capacity on the hub arcs.

However, in general capacity is considered on hub nodes or hub edges.

Some of the variants, like *Uncapacitated Single Allocation p-Hub Median Problem* (USApHMP) has received most attention from the works on HLPs.

2.4 Applications

As mentioned previously, two of the most important and well-known areas of application of hub location problems are telecommunication and transportation. So far, many contributions have been devoted to them and many researchers have been focusing on them.

In telecommunication, flows to be sent are digital data which are supposed to be delivered from origins to destinations. Such applications include video conferences, telephone networks, distributed processing and so on. Hub nodes are multiplexors, gates, switches etc. The hub edges are considered as different types of physical media, like optic fibre, coaxial cable etc. [12]. With progressive advances, new technologies result in new types of hub nodes and edges.

In transportation applications, like public transport, air passenger, air freight, express shipment (over night delivery), large trucking system, postal delivery and rapid transit, the demands are physical flows in form of passengers or goods to be transferred between origins and destinations. Also, different transportation vehicles can be considered for hub level facilities like, buses, trucks, trains, taxis, planes, fast-lanes etc. The hub facilities in postal delivery applications are post offices where the items for different destinations are collected, sorted into several groups and distributed. The hub edges can be seen as the planes or trains transferring shipments between districts corresponding to hub nodes. In public transport, the hubs are regional airports, central railway stations or bus terminals. People from corresponding regions (geographical zones containing the nodes assigned to a given hub node) are transferred there to use the inter-hub facilities working on the hub edges. The hub edges are widebody aircrafts flying longer distances or take more people or special type of trains and/or buses with special functionalities. In freight transportation, hubs are the break-bulk terminals where the smaller trucks unload (load) their shipments onto (from) larger vehicles or any other transportation facilities operating on the hub edges.

In this dissertation, we concentrate on the application in public transport planning and propose HLP models customized for this application.

2.5 State of the Art

Hakimi in 1964 [37], showed that in order to find the optimal location of a single switching center that minimizes the total wire length in a communication network, one can limit oneself to finding the vertex median of the corresponding graph. In other words, he generalized the Goldstein's vertex optimality of a tree to vertex optimality of a weighted graph (the absolute median of a graph is always at a vertex of graph). In 1965 [38], he showed that optimal locations of switching center(s) in a graph of communication network are at p-median of the corresponding weighted graph. In general he emphasized on the node optimality of one-median and p-median problems in a weighted graph.

HLPs originally initiate from the idea of Goldman [35] in 1969. In his work he proved that Hakimi's argument holds even for more general cases. He proposed two formulations. In the first model he assumed that materials are sent from origins to destinations and pass through a processing as well as a collect-and-dispatch center. The unit transportation cost of a source-center may be different from unit transportation cost of the processed material or in other words center-destination. In the second formulation, he assumed that flows can be processed in more than one center and transportation costs of origin-center, center-center and center-destination may be totaly different. Therefore, the problem is to find p centers and assign the flows to the center(s), aiming to result in a minimum of transportation cost. This is essentially the p-hub median problem [12] though he used the term *center* instead.

For the first time O'Kelly [62, 63] paved the way for the future study of hub location problems. These works dealt with one and two hub systems in the plane. However, in this work, we concentrate on the discrete hub location problem. The first work in this area is again due to O'Kelly in 1987 [64], where he proposed the first mathematical formulation (a quadratic model) for hub location problems. He presented the formulation for *Single Allocation p-Hub Median Problem* (SApHMP) which is also known as *Uncapacitated Single Allocation p-Hub Median Problem* (USApHMP).

There are some reviews devoted to HLPs on a discrete network. Among these reviews, we refer readers to the latest two reviews [12] and [4] where one can also find more details about other works and reviews.

As mentioned earlier, most of the works on HLPs are devoted to the p-hub median problems. Therefore, we briefly go along by citing the works accomplished on hub location problem and p-hub median problem. Although, there are other variants of HLPs, but we just restrict ourself to these variants and the works close to the subject of our work.

2.5.1 Formulations

In this section we review the work on the formulation approaches while distinguishing between specific attribute(s) each one possesses (single and multiple allocation schemes etc.).

Nomenclature

Throughout this dissertation, unless mentioned otherwise, we refer to the variants of the HLPs by the following: Each problem is referred as a XYZ, where X indicates the capacity policies (Capacitated (C) or Uncapacitated (U)), Y for allocation mode (Single Allocation (SA) or Multiple Allocation (MA)) and Z for problem type like:

- p-Hub Median Problem (pHMP).
- Hub Covering Problem (HCP).
- p-Hub Center Problem (pHCP).

For example, an *Uncapacitated Single Allocation p-Hub Median Problem* is referred as USApHMP, and whenever there is no capacity or both assignment schemes are considered, we may omit the indicators and use SApHMP or UpHMP (pHMP).

Single Allocation

In the formulation context as mentioned earlier, the first formulation is proposed by O'Kelly [64] for SApHMP applied to airline passenger transport. The model is a quadratic model and includes n^2 binary variables and $2n+1$ linear constraints. In this model no intrahub cost is assumed. He also explained why the objective function is not guaranteed to be convex.

However, the first linear integer programming for pHMP was proposed by Campbell [9] in 1994. That is a formulation with a total number of $n^4 + n^2 + n$ variables where $n^2 + n$ variables are binary. The number of linear constraints is $4n^4 + 2n^2 + 1$. Again, in 1996 [10], he presented another integer formulations for SApHMP.

Skorin-Kapov *et al.* in 1996 [78], presented an MIP formulation for the USApHMP. The model was very tight and uses $2n^3 + n^2 + n$ linear constraints and $n^4 + n^2$ variables where n^2 are binary. In most cases (96% of instances) the LP solution was optimal for CAB instances and for the rest it generated lower bounds less than 0.1% below the optimal. They proved the optimality of best known solutions of their earlier tabu search. Where the optimal solution was not integer, they set a few variables to binary values based on the best-known solutions. They extended the range of optimally solved problems.

In 1996 [27], Ernst and Krishnamorthy presented a new LP formulation for SApHMP, which requires fewer variables and constraints than those in literature. They affirmed that even if their model is not tighter than that of Skorin-Kapov in 1994 (which was a working paper and published in 1996 [78]), it yields optimal solutions in a significantly less computational time and is less memory intensive. This model has $2n^2 + n + 1$ linear constraints and $n^3 + n^2$ variables where n^2 are binary.

O'Kelly *et al.* in 1996 [66], tried to use other existing formulations and improved the linearization scheme for both single and multiple allocations. They derived new formulations with $(n^4 - 3n^3 + 5n^2 - n)/2$ variables and $(2n^3 - n^2 - n + 2)/2$ linear constraints for the multiple allocation and $(n^4 - 3n^3 + 7n^2 - 3n)/2$ variables and $n^3 + 1$ linear constraints for the single allocation problem. These new formulations allowed them to do more extensive computation and solve even larger instances than those have been solved in literature until that time.

In 2001, Ebery [24] presented a new MIP for USApHMP. This formulation uses fewer variables than those previously presented in the literature. This is also the first mixed integer linear program for single allocation hub location problems requiring only $O(n^2)$ variable and $O(n^2)$ constraints. He showed that the model is more effective both in terms of computational time and memory usage. Moreover, lager instances could be solve.

Other work can be found in Sohn and Park in 1998 [80] on USApHMP. They studied the case when the unit flow cost is symmetric and proportional to the distance and improved the formulation of [66].

Although, the formulation proposed by Ebery in 2001 is the best known model among single allocation models. However, with respect to the computational time the model of Ernst and Krishnamorthy [27] keeps to be the best ([4]).

Multiple Allocation

The first model for multiple allocation problem is due to the work of Campbell in 1992 [8]. He formulated MApHMP as a linear integer program. It employs $n^4 + n$ binary variables. The total number of (linear) constraints is $2n^4 + n^2 + 1$. In 1994 [9], he showed that in the absence of capacity constraints, the total flow from each origin to each destination is routed via the least cost hub pair. Therefore, it is not necessary for any of the n^4 flow variables to be binary in MApHMP. He proposed formulations for UpHMP and UHLP. Again, in 1996 [10], he presented other integer formulations for MApHMP.

In 1996, Skorin-Kapov *et al.* [78] presented a new MIP for UMApHMP. This model uses totally n^4+n variables where only n variables are binary and includes $2n^3+n^2+1$ linear constraints. The model reported a tight LP bound. The LP solution of most of the CAB instances were integer and generated lower bounds less than 1% below the optimal value of MApHMP.

In 1998 [28], a new model for UMApHMP was proposed by Ernst and Krishnmoorthy based on the idea of their earlier work on the single allocation problem. It uses $2n^3 + n^2 + n$ variables n of which are binary and $4n^2 + n + 1$ linear constraints. They also showed that their model is superior to the LP-based model of [78] and allows solving larger size instances.

Sohn and Park in 1998 [80] proposed a model for UMApHMP.

In 2004, Boland *et al.* [7] exploited the characteristic of optimal solutions in the model of [28]. They proposed some preprocessing and improved the lower bound.

In 1999, Sasaki *et al.* [76] introduced a special case of MApHMP by considering only one hub in each route. This problem is known as the *1-stop* problem.

Fixed Cost, Threshold and Capacities

The answer to question of optimal number of hubs for a given set of interactions between a number of fixed nodes redounded to incorporating new aspects in the problem. In order to make the number of hubs an endogenous part of the problem, one can either make the operating cost of hubs explicit or consider an amount of available budget for.

For the first time, O'Kelly in 1992 [65] proposed the incorporation of fixed costs as hub setup cost in the objective function. In this model the number of hubs rather than being fixed beforehand is a decision variable. The model is the quadratic integer model of O'Kelly [64] where fixed cost terms are incorporated.

In 1994, Campbell [9] also suggested making use of a threshold as the minimum flow needed to allow service on a spoke link. He incorporated fixed costs for spoke edges in pHMP.

In 1998, Abdinnour-Helm and Venkataramanan [3], proposed a new quadratic integer formulation for the UHLP based on the idea of multi-commodity flows in networks.

In 1998, Sohn and Park [80] proposed improved MIP formulations for UMApHMP and USApHMP where fixed cost for hub edges is considered.

Nickel *et al.* [61] in 2001 proposed a model for HLPs. They assumed fixed costs not only for hubs, but also fixed costs for hub edges and spoke edges.

When the number of hubs is not in a priori fixed, in addition to multiple and single allocation variants, some capacity policies can also be considered. The first models dealing with the single/multiple uncapacitated/capacitated hub location problems belong to Campbell in 1994 [9].

In 1994, Aykin [6] proposed a hub location model with fixed cost and capacity on the hubs. This model uses $n^4 + n$ binary variables and $2n^2 + 2n + 1$ linear constraints.

In 1999, Ernst and Krishnamoorthy [30] presented two formulations for CSAHLP. The second and better one uses fewer variables and constraints than those existing in literature until that time. This was the first time that the capacitated variants of the problem is solved in literature.

In 2000, Ebery *et al.* [25], presented a new MIP formulation and its modification for CMAHLP. This model has $2n^3 + n^2 + n$ variable where n are binary and includes $2n^2 + 2n$ linear constraints. Although, this new formulation contains fewer variables and constraints in its time, it was weaker. However, they could solve larger problems and in less amount of computational time.

Boland *et al.* [7] suggested making use of preprocessing and cutting for MAHLP. They used flow cover constraints for the capacitated case in order to improve the computational time. They identified some properties of the problem and exploited for applying preprocessing.

Labbé *et al.* in 2005 [53] proposed a *Quadratic Capacitated Hub Location Problem with Single Allocation* (QHL) (QHL). They presented two relaxation of this model, *Linear Capacitated Hub Location Problem with Single Allocation* (LHL) with capacity on nodes (on the amount of traffic passing through) and *Uncapacitated Hub Location Problem with Single Allocation* (UHL) (UHL).

In the same year, Yaman [84] proposed a model for telecommunications with modular capacity. This model is known as *Uncapacitated Hub Location Problem with Modular Arc Capacities* (HLM) and the capacity restricts the amount of traffic transiting through the hub. Furthermore, fixed cost, both for hubs and hub edges, were considered.

Again, Yaman and Carello [85] in 2005 proposed a different idea in their new model. In this new model the cost of using an edge is not linear, rather, is stepwise. The capacity of a hub restricts the amount of traffic transiting through the hub rather than the incoming traffic. The problem considered is again the *Single Assignment Hub Location Problem with Modular Link Capacities* (HLMC).

Marín in 2005 [56], proposed a new capacitated model for MAHLP in which the capacity is considered as an upper bound on the total flow coming directly from the origins. Furthermore, it is assumed that the flow between a given origin–destination pair can be spilt into several routes. He proposed a tight integer linear programming formulation for the problem along with some useful properties of the optimal solutions to speed up the resolution. He could solve instances of medium size very efficiently and outperformed others given in the literature.

In 2007, Wagner [83] improved the formulation of [81] (the so called *Cluster Hub location Problem* (CHLP)). This new model has less variable while LP relaxation is tight. Moreover, larger problem instances can be solved.

A capacitated version of the $1 - stop$ problem is also studied in [75].

2.5.2 Polyhedral Studies

Hamacher *et al.* [40] identified the dimension and derived some classes of facets for the HLP polyhedron. They proposed a general lifting rule to lift facets from *Uncapacitated Facility Location Problem* (UFLP) to the UMAHLP. In their strong formulation, all the constraints were facet defining.

In [52], Labbé and Yaman studied the USAHLP to find facet-defining inequalities that can be separated in polynomial time. They proposed two formulations based on the multi-commodity flow variables. Two projection schemes for the flow variables of these two models were presented and extreme rays of the projection cone for the single commodity case were identified.

In addition to that, in 2005, Labbé *et al.* [53] studied the polyhedral properties of SAHLP. They introduced variants of QHL and some facet-defining and valid inequalities. This includes facet-defining inequalities involving only assignment variables and valid inequalities involving both assignment and traffic variables. Other valid inequalities and separation algorithms were also presented.

Yaman [84] in 2005, presented some valid inequalities, some results that give the optimal lifting coefficients of some variables as well as families of facet defining inequalities.

Marín [57] exploited the knowledge of the polyhedron of set-packing problem for UHLP and presented some facet-defining valid inequalities when the cost structure satisfies the triangle inequality. He called the problem as *Uncapacitated Euclidean Hub Location Problem* (UEHLP) (UEHLP).

In 2006, Marín *et al.* [58], presented a new tight integer formulation for UMAHLP by studying the polyhedron of the problem. This model allows one or two visits to hubs. In this generalized model, the cost does not need to satisfy the triangle inequality. He used the intersection graphs to create clique inequalities that proved to be better. So far, this model is the best-known model.

2.5.3 Others

In 2005, Campbell *et al.* [13] proposed a new model which they called *Hub Arc Location Problem*. Rather than locating discrete hub facilities, this model locates hub arcs, which have reduced unit flow costs.

There are other works which deal with the different cost structures, study and analysis of impacts of the scale economy and so on (see [67, 41] .

In 2007, Costa *et al.* [21] proposed an interactive bi-criteria approach for the CSAHLP. Instead of using capacity constraints to limit the amount of flow that can be received by the hubs, they introduce a second objective function to the model (besides the traditional cost minimizing function) that tries to minimize the time to process the flow entering the hubs. Two bi-criteria single allocation hub location problems were presented. The first one minimizes the total service time and the second one minimizes the maximum service time on the hubs. They also discarded the capacity constraints and analyzed the impact of these limits on different non-dominated solutions.

2.5.4 Applications

Applications of HLPs in transportation and telecommunications were reported in many work. Hall in 1989 [39], examined the application of hub-and-spoke networks for configuration of an overnight package air network. The specific impact of time-frame restriction on the configuration of a transportation network with emphasis on the air transportation over a large region with multiple time zones was studied.

O'Kelly and Lao [68] in 1991, presented a linear programming approach to solve the mode choice problem in a hub-and-spoke network. Their network structure includes two hub nodes: A master hub (Dayton, Ohio) that is connected to all the nodes and a minihub (Los Angeles, California) performing as a regional sorting center. The LP formulation decides which hub to be served by truck rather than air and afterwards the cities that should be connected to the minihub will be determined. All the activities are constrained by the delivery schedule determined by the time zone.

In 2005, Raccunica and Wagner [72] applied HLPs on the freight rail network. This model allows for non-linear and concave cost functions on different segments. They exploited the available knowledge on the polyhedral of HLPs and proposed a linearization procedure along with two efficient variable-reduction heuristics for its resolution. They provided an analysis of the model on a full-scale dataset of the Alpine region in Europe.

In 2007, Jeong *et al.* [44], studied the European freight rail way system. They proposed a mathematical model for their case study.

Iyer and Latriff [42] in 1990, considered centralized and decentralized strategies and used hub nodes in a network of service within a guaranteed time.

In 1983, Powell and Sheffi [71] studied less-than-truckload (LTL) motor carrier. Freights are consolidated in end-of-line terminals and sent to breakbulk terminals. The freights are consolidated by unloading, sorting and reloading. Although their aim was to propose a heuristic algorithm to solve this problem, they also presented an IP formulation in order to represent the complexity and implication of designing heuristics instead.

LTLs were again studied in 2007 on a case study in Brazil [23]. For further details one may refer to [11].

In 1996, Jaillet *et al.* [43] applied HLPs for capacitated airline networks. They presented three linear integer programming models corresponding to three different service policies. They applied their work on data of up to 39 cities in U.S. .

Kuby and Gray [51] applied it on package delivery in U.S and in 2006 Cetiner *et al.* [18] on a case study of Turkish postal delivery system.

For telecommunication applications we refer to the work of Klinswicz [49] in 1998 and Carello *et al.* [17]. Another work in 2005 [85], was due to Yaman and Carello.

For a review of models and methods in freight transportation one may refer to [22].

Aversa in 2005 [5] proposed an MIP model for locating a hub port in the East coast of South America.

Cordeau *et al.* [20] in 1998 surveyed the studies on rail transportation problems. The review mainly concentrates on routing and scheduling problems.

In 2001, Nickel *et al.* [61] proposed new mathematical models for application of HLPs in urban public transport network. As mentioned earlier, in one of their models (*Generalized Public Transport* (GPT)), they considered fixed cost for hub nodes, hub edges and also spoke edges. They relaxed some classical assumptions of HLPs and their models are customized for public transport planning. Their first model (PT) has $2n^4 + n^2 + n$ variables, where $n^2 + n$ are binary and the second one, GPT, has $2n^4 + 2n^2 + n$ variables where $2n^2 + n$ are binary. Both models use linear constraints of $O(n^4)$. They used CAB dataset for their work.

2.5.5 Solution Methods

It is already proven in literature that pHMP is NP-hard, even if the location of hubs are known (*see* [45]). That is, the allocation part of the problem still remains NP-hard (in fact it is equal to quadratic assignment problem (QAP)). Therefore, as the problem size increases, some difficulties to solve instances of the problem raises. One expect a lot of studies especially on the heuristic strategies aiming to solve instances of HLPs to high quality solutions. Available solution methods in literature are divided into two category: Exact and heuristics.

Exact Methods

Aykin in 1994 [6], proposed a branch-and-bound algorithm for CSApHMP.

In 1996 Ernst and Krishnamoorthy [27], presented a branch-and-bound algorithm for SApHMP where they used the solution of their simulated annealing as upper bound. Another variant of branch-and-bound can be found in [48].

In 1998, Abdinnour-Helm and Venkataramanan [3] applied a sophisticated approach. The bounds are obtained by employing the underlying network structure of the problem.

In 1999, Ernst and Krishnamoorthy [30], proposed an LP based branch-and-bound algorithm for CSAHLP. The upper bound is achieved from a heuristic.

In 2000, Ebery *et al.* [25] developed a branch-and-bound for CMAHLP. They used upper bounds from a shortest-path based heuristic and a lower bound from a dual ascent based on a disaggregated model formulation.

In 1998, Ernst and Krishnamoorthy [29] presented a shortest-path based branch-and-bound for both USApHMP and UMApHMP. However, the algorithm performs better for the case of multiple allocation.

Mayer and Wagner in 2002 [59], proposed a branch-and-bound algorithm for UMAHLP. The so called HubLocator algorithm could solve the instances up to size 40.

In 1999, Sasaki *et al.* [76] presented a branch-and-bound method for the *1-stop* problem. In 2003, Sasaki and Fukushima [75] proposed a branch-and-bound based on the lagrangian relaxation bounding strategy.

In 2005, Marín [56] used commercial branch-and-bound code for solving CMAHLP. The lower bound was achieved from the LP relaxation and the upper bound from a heuristic which acts on the solution of the LP.

Branch-and-cut is also applied to variants of problems. One can refer to [53] for QHL and also [85] for a similar model. They applied the knowledge about the polyhedral studies in both work.

Aykin in 1994 [6], used a Lagrangian relaxation for the routing sub-problem of a CSApHMP after the hub locations were fixed by branch-and-bound or a greedy-interchange heuristic.

In 1998, Pirkul and Schilling [70] presented a Lagrangian relaxation equipped with a subgradient and a cut constraint. They also proposed a measure of quality to evaluate their solutions quality. The model is based on the tight LP of Skorin-Kapov *et al.* [78] for SAHLP.

Elhedi and Hu [26] in 2005 proposed a Lagrangian heuristic for the problem of hub-and-spoke network design with congestion at hubs. They assumed a convex cost function that increases exponentially as more flows are assigned to hubs. First they linearized this non-linear term and then Lagrangian heuristic is applied.

Marín [57] in 2005 took the advantage of his tight model which benefits from the applied knowledge of the polyhedron of the problem and used the Lagrangian relaxation technique to achieved a very efficient relax-and-cut algorithm for UEMAHLP.

In the dual approaches we refer to [48] for a dual ascent and dual adjustment algorithm. Together with an upper bound construction algorithm, it offers a stand-alone algorithm. However, they also incorporated them into a branch-and-bound algorithm.

Sung and Jin in 2001 [81], considered a hub network design problem and presented a model under non-restrictive network policy allowing spoke-to-spoke edges. The model is then solved by a coupling of dual ascent and dual adjustment.

Mayer and Wagner [59] in 2002, employed a dual ascent inside the HubLocator.

In 2007, Cénovas *et al.* [16], presented a heuristic method based on dual-ascent for the UMAHLP which appeared in [58]. They could solve the instances up to 120 nodes and at reasonably fast speed. This heuristic is embedded in a branch-and-bound framework and prepares a good lower bound for hub nodes of branching tree.

Wagner [83] in 2007 proposed a dual-based solution method for *Cluster Hub Location Problem* that outperforms the heuristic proposed by Sung and Jin (in 2001 [81]) . This not only provided optimal solutions, but also needed less computational time.

In 1998, Ernst *et al* [28] proposed an enumeration algorithm by explicitly enumerating all possible p hubs from among n nodes for UMApHMP.

In 2002, Klincewicz [50] proposed an optimal enumeration approach for FLOWLOC model of O'Kelly and Bryan [67] that treats the economies of scale by means of piecewise-linear concave cost functions on the interhub arcs.

Campbell *et al.* in 2003 [15], proposed an enumeration-based parallel solution method for solving hub arc location problem. Although this paper was published in 2003, it essentially appeared after the two other papers of Campbell *et al.* [13, 14] which were published in 2005. In these works, they introduced the hub-arc location problems.

Campbell *et al.* in 2005 [14], proposed two novel enumeration-based approaches to finding solutions to hub arc location problems.

In 1998 [80], Sohn and Park proposed a shortest-path based algorithm for MApHMP.

Study of lower bound is also reported by O'Kelly *et al.* [69]. They improved lower bounds of related LP of quadratic model of O'Kelly [65] and presented new lower bounds for HLP in presence of triangle inequality.

Pirkul and Schilling in 1998 [70], prepared tightest bounds of any heuristic to that date by using cut constraints in the subgradient optimization.

Benders decomposition approaches for UMAHLP for the formulation in [40] is considered by R.S. de Camargo *et al.* in [74]. They decomposed the problem following the Benders scheme and solve the sub-problem for each origin and destination by inspection.

In [73], Rodriguez-Martin and Salazar-Gonzalez presented an MIP model and proposed solution method is a Double Benders Decomposition.

Heuristic Methods

There are much work has also been done on the development and design of heuristics and meta-heuristics. We only cite some of the most well-known and effective ones in literature.

Perhaps, Ernst and Krishnamoorthy in 1996 [27], have been one of the firsts to develop a Simulated Annealing (SA) algorithm for the p-hub median problem.

In 1999, Ernst and Krishnamoorthy [30] developed an SA for CSAHLP. They used the solutions as the upper bound in a branch-and-bound.

In 2001, Abdinnour-Helm [2] presented an SA for pHMP of O'Kelly [64]. Results have been compared against the tabu search of Skorin-Kapov [77], MAXFLOW and ALLFLOW. They showed that in general, tabu search and simulated annealing outperform the two others.

In 1998 [3], Abdinnour-Helm and Venkataramanan proposed the first Genetic Algorithm (GA) for UHLP.

In 2005, Topcuoglu *et al.* [82] proposed a robust GA that could solve all the CAB instances to optimality. For AP instances, solutions are obtained in a fraction of best run times reported in literature ([1]). For larger instances it considerably surpasses its predecessors with respect to both quality and computational time.

In 2007, seemingly not aware of [82], Cunha *et al.* proposed another GA [23]. They embedded a fast and simplified mechanism to allow non-improving solutions to be accepted with a given probability.

Several Tabu Search (TS) algorithms were also proposed in literature. In [47] the first TS was proposed for SApHMP. In 1994, Skorin-Kapov and Skorin-Kapov [77] developed another TS based on *1-exchange* for the same problem. They showed that their so called TABUHUB outperforms both HEUR1 and HEUR2 of O'Kelly [64].

Klincewicz in 2002 [50], presented a heuristic based on the TS for FLOWLOC model.

In 2004, Carello *et al.* [17] developed a TS for application of HLPs in telecommunications.

Yaman and Carello [85], proposed a TS for the HLMC with new cost structure and capacity policies. They used TS for the location part and another local search for the assignment problem.

Greedy Randomized Adaptive Search Procedure (GRASP) was also proposed for USApHMP in [47], and in [50] for the model given in [67].

A hybrid of genetic algorithm and tabu search is proposed by Abdinnour-Helm [1] for USAHLP. They showed that this hybrid had better performance than pure GA. The results stated that GA component is useful to diversify the search and TS to localize it.

Other hybrid algorithm in literature is due to Chen [19] in 2007 for USAHLP. He presented two approaches in order to find upper bound on the number of hubs and proposed a Simulated Annealing (SA) equipped with a tabu list and an improvement method. Proposed heuristic is capable of obtaining optimal solutions for all small-scaled problems very efficiently. This algorithm outperformed that of [82].

Smith *et al.* [79] took the advantage of the quadratic model of O'Kelly [64] to be mapped on a Hopfield neural network. A Hopfield network is used to find the local optimum of a energy function, which can be translated to the local minimum of the optimization problem.

There are also some other heuristics proposed which may not be classified into the well-known categories.

Powell and Sheffi in 1983 [71], proposed a local improvement procedure which starts with a reasonable initial solution and then test a large number of small changes in solution looking for improvements.

O'Kelly in 1987 [64], presented two greedy and interchange heuristics (Heur1 and Heur2). In both of them a complete enumeration of different patterns is considered. In Heur1, each node is allocated to the nearest hub but in the Heur2, which results in a tighter bound, allocation to first *or* second hub node is taken into account.

In 1991, Klincewicz [46] developed some heuristics for SApHMP. He developed exchange and clustering heuristics. The exchange heuristics, for a given incumbent set of nodes, examines the existence of other promising candidate(s) in single and double exchange manner (*1-exchange* and *2-exchange*). The clustering heuristic, first clusters the nodes into p groups and then based on a special measurement, finds a hub node for each one to serve the cluster. In the assignment part of the problem, in addition to the distance-based measurement of O'Kelly [64] that tries to minimize the costs on the spoke edges, he proposed other approaces. A *common traffic* criterion aiming to minimize the cost on the hub edges and a multi-criteria assignment procedure to compromise between these two sorts of assignments based on a weighted sum of two measurements. A drawback of such approach is due to the parameters of weight used in the weighted sum. Computational results showed than the double exchange heuristic has been considerably better that others.

In 1992, O'Kelly [65] proposed a heuristic that addressed the optimal number and location problem. This gives an upper bound on the solution. A lower bound is then developed by underestimating the quadratic contribution to the objective and finally attempts to improve the selection by systematic adjustment of the assignments.

Campbell [10] in 1996, proposed two heuristics ALLFLO and MAXFLO for single and multiple problems, respectively. The multiple assignment problem is easier due to the degrees of freedom. From a solution of multiple assignment problem which is a lower bound on the optimal solution of single assignment, a solution to the single assignment is developed. The ground idea is a greedy-interchange heuristic.

In 1996, Jalliet *et al.* [43] proposed a mathematical programming based heuristic using valid inequalities and local improvements.

In 1998, Ernst *et al.* [28] proposed a heuristic based on the shortest-path. For a given set of hub nodes, the allocation part of the problem can be solved very efficiently by all-pairs shortest-path algorithm. They used a modified Floyd-Warshall algorithm to find the shortest paths.

In 1999, Sasaki *et al.* [76] proposed a greedy heuristic for the *1-stop* problem.

In 1999, Ernst and Krishnamoorthy [30] proposed a random descend algorithm for CSAHLP.

In 2000, Ebery *et al.* [25] developed a shortest-path based heuristic for CMAHLP.

In 2004, Carello *et al.* [17] developed two more heuristics namely, random multi-start and iterated local search for CSAHLP.

Chapter 3

Mathematical Formulations

In this chapter, we propose a new mathematical model of HLPs applied to public transport. The computational results substantiate the superiority of this model to the existing models for the application and use it as a basis to derive other applicable variant models.

3.1 Classical Assumptions

In classical HLP models, four main assumptions were always considered:

Ass. a) The hub-level network is a complete graph.

Ass. b) Using inter-hub connections has a lower price per unit than using spoke connections. That is, it benefits from a discount factor $\alpha, (0 < \alpha < 1)$.

Ass. c) Direct connections between the spoke nodes are not allowed.

Ass. d) Costs are proportional to the distance or in other words, the triangle inequality holds.

These assumptions have always been considered in the literature in the initial work on HLPs. However, thereafter some authors suggested to relax some of them with pertinence of the applications.

Theorem 3.1.1 (Path Length)**.** *If all the four assumptions hold, any origin-destination path contains no more than one hub edge (or two hub nodes).*

Proof. From Ass. (a) and Ass. (d) it follows that transportation in the hub-level is always directly accomplished. Together with Ass. (c), every flow is routed via either 1 or 2 hub nodes. Hence the proof is completed. □

Some of the prototypical works dealing with relaxation of classical assumptions are noted in the following.

In 2001 [61], Nickel *et al.* proposed two models for urban public transport where they relaxed some of these assumptions. In general, they relaxed Ass. (a), Ass. (c) and Ass. (d).

After that in 2005, Campbell *et al.* [13, 14] introduced the Hub Arc Location Problem that in general relaxed Ass. (a) and Ass. (b). Their cost structure satisfies the triangle inequality and there is no direct connection between the spoke nodes. However, it is not necessary to have discount on all of the hub arcs. In addition to these work, Marín in [56, 58] relaxed Ass. (d).

There already exist other work that modify (relax) some of these assumptions, however, the mentioned ones are from among the most well-known ones in literature.

3.2 Mathematical Models

In this section we review some of the initial formulations proposed for HLPs and consequently, focus on the models designed for public transport planning.

3.2.1 Primary Models

In initial models for HLPs, it is usually assumed that there are three different cost factors corresponding to each of the spoke-hub, hub-hub and hub-spoke edge components. The parameters χ, α and δ are discount factors corresponding to each of these three components, respectively. Therefore, in a general form and according to Theorem 3.1.1, for a given pair of origin-destination (i and j) the cost of traveling from node i to node j via hub nodes k, l amounts to,

$$\chi d_{ik} + \alpha d_{kl} + \delta d_{lj},$$

where d_{pq} is the traveling cost between nodes p and q, $\alpha < \chi$, $\alpha < \delta$ and $0 < \alpha < 1$ is considered as a discount factor.

As mentioned earlier, the first formulation in this area is an IP formulation proposed by O'Kelly [64] for USApHPM which uses a quadratic term in the objective function follows.

(SApHMP)

$$Min \quad \sum_i \sum_j W_{ij}[\sum_k X_{ik}C_{ik} + \sum_{k,m} X_{ik}X_{jm}\alpha C_{km} + \sum_m X_{jm}C_{jm}] \tag{3.1}$$

$$s.t. \quad (n-p+1)X_{kk} - \sum_i X_{ik} \geq 0, \qquad \forall k, \tag{3.2}$$

$$\sum_{k \in N} X_{ik} = 1, \qquad \forall i, \qquad (3.3)$$

$$X_{ik} \leq X_{kk}, \qquad \forall i, k, \qquad (3.4)$$

$$X_{ik} \in \{0, 1\}, \qquad \forall i, k, \qquad (3.5)$$

where $X_{ik} = 1$, if node i is assigned to hub node k, 0 otherwise. Moreover, $X_{kk} = 1$, if node k is designated to act as a hub node, 0 otherwise. The objective function (3.1) measures the total flow cost of origins to the first hubs, first hubs to the second ones and second ones to the destinations, respectively.
The constraints (3.2) ensure that exactly p hub nodes should be selected. The constraints (3.3) indicate that each node, is either a hub node or assigned to exactly one hub node. In constraints (3.4) it is stated that a node can only be assigned to a hub node or itself, if it is a hub node. Here, intrahub flow, namely flow from a hub to itself does exist.

The first linear formulation is due to Campbell [8] in 1994. This model follows:

(UMAHLP)

$$Min \quad \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \sum_{j \in N} W_{ij} Cijkl X_{ijkl} + \sum_{k} F_k Y_k \qquad (3.6)$$

$$s.t. \quad \sum_{k \in N} \sum_{l \in N} X_{ijkl} = 1, \qquad \forall i, j, \qquad (3.7)$$

$$X_{ijkl} \leq Y_k, \qquad \forall i, j, k, l, \qquad (3.8)$$

$$X_{ijkl} \leq Y_l, \qquad \forall i, j, k, l, \qquad (3.9)$$

$$Y_k \in \{0, 1\}, \qquad \forall k, \qquad (3.10)$$

$$X_{ijkl} \geq 0, \qquad \forall i, j, k, l, \qquad (3.11)$$

where $Y_k = 1$ if node k acts as a hub, 0 otherwise and X_{ijkl} is the fraction of flow from i to j traversing the hub edge $k - l$. The objective function (3.6) is to minimize total cost composed of the flow transition costs and the hub establishment costs. Moreover, no constraints on the number of hubs to be established exists. Constraint (3.7) ensures that the whole flow originated at i and destined to j, will be eventually received by j and no part of the flow can be lost (flow conservation). In (3.8) and (3.9) it is ensured that the flow of i to j is allowed to be transferred through the nodes k and/or l, only if they are hub nodes.

The model includes n^4 real variables, n integer and has $2n^4 + n^2$ constraints. In this model Campbell assumed that $C_{ijkl} = d_{ik} + \alpha d_{kl} + d_{lj}$, $(\chi = \delta = 1)$ and the triangle inequality holds.

Many other models are proposed in literature. However, these two models are between the most effective, initiative and fundamental models which paved the way for further studies in the area of HLPs.

All the four classical assumptions are valid for these formulations.

3.2.2 Available HLP Models for Public Transport

The first models for public transport applications, to the best of our knowledge, are due to Nickel *et al.* in 2001 [61]. They proposed two models that will be explained prior to presenting our new models.

In general, in application like public transport networks, the hub-level graph is not necessarily a complete graph (see Figure (2.4)). The triangle inequality might not hold in the cost structure or the fixed cost for establishing a direct hub edge can be higher than the cost for non-direct transportation. Therefore Ass. (a) and Ass. (d) can be relaxed. In order to achieve a more realistic model, Ass (c) can be relaxed, too. Because, often there exist direct connections between some spoke nodes which can be used in a cheaper way than routing via hub nodes.

In the absence of the triangle inequality (Ass. (d)), at least in the hub-level network, if an undirected path (sequence of hub edges) is preferred to a direct hub arc connection, then no flow uses this direct hub edge as a part of the path it traverses to reach the destination. Even, the flow between two end-points of this direct connection does not traverse the direct edge. Therefore, necessity for relaxation of Ass. (a) in situations that setup costs are incorporated into the model is again confirmed.

In the following, the first model (PT) relaxes Ass. (a) and Ass. (d). Later (in GPT) they showed that Ass. (c) also can be relaxed.

Pubic Transport (PT)(Nickel *et al.* 2001)

A set $\mathcal{E} = \{\{i, j\} \in V^2 : i \leq j\}$ is defined to be the set of all edges that can be established in the overall hub-level and spoke-level networks. Although definition of such a set can be omitted by a subtle modification of fixed setup costs, it avoids from the existence of a large number of constraints (even before the preprocessing phase of MIP solver) if they are not really necessary. It should be noted that, in situations that the network is directed, the term $(i \leq j)$ should be removed from the definition of $\mathcal{E}$. The variable X_{ijkl} is defined to be the fraction of flow from i to j which traverse hub edge $\{k, l\}$ and variable S_{ijkl} is defined to be the fraction of flow from i to j which traverse the spoke edge $\{k,l\}$. The binary variable Y_{kl} is defined to be 1 if the edge $\{k,l\}$ is established as hub edge, 0 otherwise and $H_k = 1$ if node k receives a hub and 0, otherwise.

(PT)

$$Min \quad \sum_{i\in N}\sum_{j\in N} W_{ij}[\sum_{\{k,l\}\in\mathcal{E}} \alpha d_{kl}(X_{ijkl}+X_{ijkl}) + \sum_{k\in N} d_{ik}S_{ijkl} + \sum_{l\in N:l\neq i} d_{lj}S_{ijlj}] + \sum_{\{k,l\}\in\mathcal{E}} I_{kl}Y_{k,l} + \sum_{k\in N} F_k H_k \tag{3.12}$$

$$s.t. \quad \sum_{l\in N}(X_{ijkl}+S_{ijkl}-X_{ijlk}-S_{ijlk}) = \tag{3.13}$$

$$\begin{cases} +1, & \forall i,j,k\in V: k=i, i\neq j,\\ -1, & \forall i,j,k\in V: k=j, i\neq j,\\ 0, & \forall i,j,k\in V: k\neq i, k\neq j,\end{cases}$$

$$\sum_{l\in N}(X_{iiil}+S_{iiil}) = 1, \quad \forall i, \tag{3.14}$$

$$\sum_{l\in N}(X_{iili}+S_{iili}) = 1, \quad \forall i, \tag{3.15}$$

$$X_{ijkl} \leq Y_{kl}, \quad \forall i,j,\{k,l\}\in\mathcal{E}, \tag{3.16}$$

$$X_{ijlk} \leq Y_{kl}, \quad \forall i,j,\{k,l\}\in\mathcal{E}, \tag{3.17}$$

$$S_{ijik} \leq H_k, \quad \forall i,j,k,l: k\neq j, \tag{3.18}$$

$$S_{ijkj} \leq H_k, \quad \forall i,j,k,l: k\neq i, \tag{3.19}$$

$$S_{ijij} \leq H_i + H_j, \quad \forall i,j, \tag{3.20}$$

$$S_{ijkl} = 0, \quad \forall i,j,k,l: k\neq i, l\neq j, \tag{3.21}$$

$$Y_{kl} \leq H_k, \quad \forall\{k,l\}\in\mathcal{E}, \tag{3.22}$$

$$Y_{kl} \leq H_l, \quad \forall\{k,l\}\in\mathcal{E}, \tag{3.23}$$

$$S_{ijkl}, X_{ijkl} \geq 0, \quad \forall i,j,k,l, \tag{3.24}$$

$$Y_{kl}, H_k \in \{0,1\}, \quad \forall k,l. \tag{3.25}$$

The objective function (3.12) reflects the total cost. This cost amounts to the transportation costs plus the hub edges and hub nodes setup costs. The constraints (3.13) are the flow conservation at the origins, destinations and intermediate nodes, respectively. Constraints (3.14) and (3.15) are needed only if the intrahub flows or the flows from i to i (intra-flow) exist at the hubs. For example, in applications of postal delivery, the packages received by the post office of districts are sorted at hubs and some of them might be sent back to the another destination in the same district (remain in the same region) and no other districts. However, this is usually not the case in public transport unless switching between different platforms or stations in the same terminal requires considerable amount of resources (cost or time). Constraints (3.16) and (3.17) ensure that the flow of X_{ijkl} should be passed through the hub edge $\{k,l\}$ and analogously for X_{ijlk}. The spoke edges should start or end with a hub node, which is stated in (3.18) and (3.19). Constraints (3.20) ensure that, direct connections should also be passed through at least one hub. Spoke edges are only allowed for the first and last parts of any path, which is stated in (3.21). Finally, constraints (3.22) and (3.23) ensure that end-points of a hub edge are hub nodes.

A special feature of PT is that, it allows spoke edges to be established between pairs of hub nodes.

Generalized Pubic Transport (GPT)(Nickel *et al.* 2001)

This model is an extension to the previous model because it also permits making use of direct edges between two non-hub (spoke) nodes, in contrast with the necessity of using paths containing at least one hub node in between. This model additionally considers the cost incurred by establishing the spoke edges. It decides on establishing two types of edges with the goal of minimizing the total cost. In addition to the previously existing variables, Z_{kl} for all $\{k,l\} \in \mathcal{E}$ are introduced. $Z_{kl} = 1$ if the edge $\{k,l\}$ is chosen to be a spoke edge and 0, otherwise. A new parameter J_{kl}, stands for the cost incurred by establishing an undirected spoke edge between the nodes k and l. Usually, in real life problem this cost is considerably less than a hub edge setup cost. Another new variable C_{ijk} is defined to be 1, if the type of edge changes along the path from i to j at node k and 0, otherwise. Eventually, parameter q is introduced to be the maximum allowed number of such switches along any path.

(GPT)

$$\begin{aligned}
Min \quad & \sum_{i\in N}\sum_{j\in N} W_{ij}[\sum_{\{k,l\in\mathcal{E}\}} d_{kl}(\alpha(X_{ijkl}+X_{ijkl})+S_{ijkl}+S_{ijlk})] \\
& + \sum_{\{k,l\}\in\mathcal{E}} I_{kl}Y_{k,l} + \sum_{\{k,l\}\in\mathcal{E}} J_{kl}Z_{k,l} + \sum_{k\in N} F_k H_k && (3.26)\\
s.t. \quad & (3.13), (3.14), (3.15), (3.16), (3.17), (3.22), (3.23) \\
& C_{ijl} \geq \sum_{k\in N}(X_{ijkl} - X_{ijlk}), && \forall i,j,l : l \neq i,j, \quad (3.27)\\
& C_{ijl} \geq \sum_{k\in N}(X_{ijlk} - X_{ijkl}), && \forall i,j,l : l \neq i,j, \quad (3.28)\\
& \sum_{l\in N} C_{ijl} \leq q, && \forall i,j, \quad (3.29)\\
& S_{ijkl} \leq Z_{kl}, && \forall i,j\ \{k,l\}\in\mathcal{E}, \quad (3.30)\\
& S_{ijlk} \leq Z_{kl}, && \forall i,j\ \{k,l\}\in\mathcal{E}, \quad (3.31)\\
& S_{ijkl}, X_{ijkl} \geq 0, && \forall i,j,k,l, \quad (3.32)
\end{aligned}$$

$$Y_{kl}, Z_{kl}, H_k \in \{0,1\}, \qquad \forall \{k,l\} \in \mathcal{E}, \qquad (3.33)$$

$$C_{ijl} \in \{0,1\}, \qquad \forall i,j,l. \qquad (3.34)$$

By removing the constraints (3.27), (3.28) and (3.29), we might have origin-destination paths with more than three links along in an optimal solution. Using these constraints and the parameter $q \in \mathbb{N} \cup \{0\}$, we can control the number of switchings in every path. The constraints (3.27) and (3.28) detects whether a switching takes place or not. By employing the variables C_{ijk}, the constraint (3.29) limits the number of switchings by a maximum of q. However, if the number of switches does not matter, these constraints can be simply dropped from the model without any harm. The constraints (3.30) and (3.31) ensure that the flow of S_{ijkl} is only sent through a spoke edge $\{k,l\}$.

$2n^4$ real variables and at most $n^3 + n^2$ integer variables are used in this model when the edges are undirected. In the case of undirected network, the set $\mathcal{E}$ contains at most $\frac{n(n-1)}{2}$ edges (if there is no flow from i to itself). In a directed network, the cardinality of $\mathcal{E}$ (with the same assumption) is at most $n^2 - n$.

The model also has special properties. For example, in an undirected network it can happen that in the optimal solution, we have one spoke edge and also one hub edge between two hub nodes but in different directions, simultaneously.

Moreover, if $q = 0$, then between two hubs at the same time and in the same direction both types of edges may exist. Since, a flow originated at a spoke node wants to stay on the spoke edge and flow started from a hub node does not change its edge type along the path to destination, if selects the first edge among the hub edges.

3.2.3 New Model for Public Transport Application

A new mathematical model is proposed to act as a basis for our further models for this application. This model which we refer to by HUB LOCATION PROBLEM FOR PUBLIC TRANSPORT (HLPPT) as is depicted in the following is an MIP model [32, 34].

The following are the attributes we considered for our model:

Attr. a) Connected hub-level network rather than a complete one.

Attr. b) The cost structure neither necessarily satisfies the triangle inequality nor any other special structure.

Attr. c) To ensure some levels of reliability, there exist the possibility of allowing multiple connections between the spoke nodes and the hub-level network.

While a connected hub-level network is assumed in HLPPT, in [61], they did not introduce any alternative assumption for the relaxation of Ass. (a).

Moreover, in [61], the existence of a spoke edge between hub nodes are allowed, which can be a threat for connectivity of hub-level network.

Attr. c can guarantee that if an unpredicted failure happens in a (or more) transportation link(s), an alternative one always exists.

In this model, there is no constraint on the number of links in any origin-destination path.

In HLPs, with the assumptions of Ass. a - Ass. d, once the hub nodes are known, the remaining problem in the multiple allocation is to find the cheapest routes (although in single assignment scheme it is again NP-hard problem, we are not dealing with the single assignment in our new model).

In HLPPT, the problem is first to locate the hub-nodes, second to choose the connecting hub edges so that it results in a connected hub-level graph and then in the third step routing the flows. In the second step, neither the number of hub edges is known a priori nor the way they should be connected to make an optimal connected graph. In fact, the second step can be considered as the problem of assigning an unknown and finite number of edges to pairs of hub nodes so that it results in a connected graph. Table 3.1 sheds some light on this fact.

While in both SAHLP and MAHLP hub edges are identified as soon as the hub nodes are located, in HLPPT, locating the hub edges is another problem.

Table 3.1: HLPPT vs MAHLP and SAHLP

	HLPPT	MAHLP	SAHLP
Locating hubs	$\surd$	$\surd$	$\surd$
Selecting hub edges	$\surd$ [a]	$\times$	$\times$
Allocation	Polynomial	Polynomial	NP-hard[b]

[a] in special case reduces to QAP.
[b] QAP.

Therefore, in terms of difficulty of the problem, one can say that HLPPT is more difficult than MAHLP. When compared to SAHLP, this problem is not easier, if not more complicated. Because, if the number of hub nodes are known (say q) by some way, the allocation of $n-q$ spoke nodes to q hub nodes is not more difficult than allocation of unknown and finite number of hub edges to pairs of hub nodes so that it makes a connected sub-graph. In a special case this problem will be reduced to QAP and therefore is also NP-hard.

The variables in this model are defined as follows: $x_{ijkl}=1$ if the optimal path from i to j traverses the hub edge $k-l$ and 0, otherwise. Also, $a_{ijk}=1$ if the optimal path from i to j traverses the spoke edge $i-k$ while i is not hub and 0, otherwise and $b_{ijk}=1$ if the optimal path from i to j traverses the spoke edge $k-j$ while j is not hub and 0, otherwise. In addition, $e_{ij}=1$ if the optimal path from i to j traverses $i-j$ and either i or j is a hub and 0, otherwise. For the hub-level variables, $y_{kl}=1, k<l$, if the hub edge $k-l$ is established, 0 otherwise and $h_k=1$ if k is used as a hub node, 0 otherwise.

The transportation cost for a given flow with origin i and destination j is the sum of, (i) the cost of sending the flow from i to the first hub node, (ii) the cost of traversing one or more hub edges discounted by the factor α $(0<\alpha<1)$ and (iii) the cost of transition on the last spoke edge. The proposed mathematical formulation follows:

(HLPPT)

$$
\begin{aligned}
Min \quad & \sum_{i}\sum_{j\neq i}\sum_{k}\sum_{l\neq k}\alpha W_{ij}C_{kl}x_{ijkl}+\sum_{i}\sum_{j\neq i}\sum_{k\neq i,j}W_{ij}C_{ik}a_{ijk}+ \\
& \sum_{i}\sum_{j\neq i}\sum_{k\neq i,j}W_{ij}C_{kj}b_{ijk}+\sum_{i}\sum_{j\neq i}W_{ij}C_{ij}e_{ij}+ \\
& \sum_{k}F_kh_k+\sum_{k}\sum_{l>k}I_{kl}y_{kl} && (3.35)\\
s.t. \quad & \sum_{l\neq i}x_{ijil}+\sum_{l\neq i,j}a_{ijl}+e_{ij}=1, && \forall i,j\neq i, \quad (3.36)\\
& \sum_{l\neq j}x_{ijlj}+\sum_{l\neq i,j}b_{ijl}+e_{ij}=1, && \forall i,j\neq i, \quad (3.37)\\
& \sum_{l\neq k,i}x_{ijkl}+b_{ijk}=\sum_{l\neq k,j}x_{ijlk}+a_{ijk}, && \forall i,j\neq i,k\neq i,j, \quad (3.38)\\
& y_{kl}\leq h_k,\quad y_{kl}\leq h_l, && \forall k,l>k, \quad (3.39)\\
& x_{ijkl}+x_{ijlk}\leq y_{kl}, && \forall i,j\neq i,k,l>k, \quad (3.40)\\
& \sum_{l\neq k}x_{kjkl}\leq h_k, && \forall j,k\neq j, \quad (3.41)\\
& \sum_{k\neq l}x_{ilkl}\leq h_l, && \forall i,l\neq i, \quad (3.42)\\
& e_{ij}\leq 2-(h_i+h_j), && \forall i,j\neq i, \quad (3.43)
\end{aligned}
$$

$$a_{ijk} \leq 1 - h_i, \qquad \forall i, j \neq i, k \neq i, j, \tag{3.44}$$

$$b_{ijl} \leq 1 - h_j, \qquad \forall i, j \neq i, l \neq i, j, \tag{3.45}$$

$$a_{ijk} + \sum_{l \neq j,k} x_{ijlk} \leq h_k, \qquad \forall i, j \neq i, k \neq i, j, \tag{3.46}$$

$$b_{ijk} + \sum_{l \neq k,i} x_{ijkl} \leq h_k, \qquad \forall i, j \neq i, k \neq i, j, \tag{3.47}$$

$$e_{ij} + 2x_{ijij} + \sum_{l \neq j,i} x_{ijil} + \sum_{l \neq i,j} x_{ijlj} \leq h_i + h_j, \quad \forall i, j \neq i, \tag{3.48}$$

$$x_{ijkl}, y_{kl}, h_k, a_{ijk}, b_{ijk}, e_{ij} \in \{0, 1\}. \tag{3.49}$$

The objective (3.35) is the total cost of transportation plus hub nodes and edges setup costs. The constraints (3.36)-(3.38) are the flow conservation constraints. In (3.39), it is ensured that both end-points of a hub edge are hub nodes. The constraints (3.40) ensure that a flow in its path to destination if passes through more than one hub in the hub-level network, it traverses hub edge(s) connecting these nodes. In (3.41) ((3.42)) it is ensured that only a flow with origin (destination) of hub type is allowed to select a hub edge to depart from origin (arrive to the destination). Constraints (3.43)-(3.45) check the end-points of spoke edges. Any flow from i to j, if enters to (depart from) a node other than i and j, that node should be a hub node. This is ensured by (3.46) ((3.47)). Selection of edges on the path between origin and destination i and j depends on the status of i and j whether both, none or just one of them is a node. This is checked by (3.48). In an uncapacitated environment, as also mentioned in ([9]), only hub node and hub edge variables need to be considered as binary variables (even in the case of our model, only hub edge variables need to be binary values. However, the computational experiences suggest to keep the integrality of hub node variables as well). Therefore, the constraints 3.49 can be replaced by,

$$x_{ijkl}, a_{ijk}, b_{ijk}, e_{ij} \in (0, 1), h_k, y_{kl} \in \{0, 1\}. \tag{3.50}$$

From now on, whenever we talk about HLPPT we are referring to the model of (3.35)-(3.48) together with the constraint (3.50).

3.2.4 HLPPT vs. PT

In our new model, we emphasize on the real willingness of passengers who use public transport services. Some personal communication with passengers revealed that, for example, passengers who arrived to the hub-level network do not like to change their vehicle (train) type inside this network. That is, if they enter a hub node they prefer to use the fast-line as long as they did not reach to the last hub node where their destination is assigned to (assume an urban public transport network where people have to frequently change the vehicle type from ground to the underground and vice versa, just in a city-wide distance. Besides the inconveniences, it increases the likehood of missing a connection).

Although the existing models could be modified to include this new feature, HLPPT also possesses other properties.

Theorem 3.2.1. *In presence of the triangle inequality, the hub level network is a tree.*

Proof. The model always reports a connected hub-level. This can be deduced from the constraints of HLPPT.

In a multiple allocation which we considered here for HLPPT, each origin sends its flow through the shortest-path to the destination. Moreover, shortest path between a pair of origin-destination traverses the shortest path between any pair of hub nodes along this path. In addition to that, there exists only one unique shortest path between pairs of origin-destination. The latter, is due to the minimizing objective function. Therefore, there is no more than one path between any pairs of origin-destination in the hub-level network. The hub-level network is a tree. □

It is well-known that the problem is an NP-Hard problem which even small size instances of that cannot be solved to optimality in a reasonable amount of time. Our new HLPPT model, as we will show later, paves the way for preparing a good basis for decomposition algorithms as well as (meta-)heuristics which may enable us to solve larger size instances to both optimality or high quality solutions.

To have a comparison between our new model and existing PT model, some modification to the PT model should be taken into account. By adding new constraints and avoiding spoke edge connections between the hub nodes, the following comparable model is achieved.

(Comparable PT (CPT))

$$
\begin{aligned}
Min \quad & \sum_{i}\sum_{j\neq i}\sum_{k}\sum_{l\neq k} W_{ij}d_{kl}(\alpha X_{ijkl} + S_{ijkl}) + \\
& \sum_{k,l>k} I_{kl}Y_{k,l} + \sum_{k\in N} F_k H_k && (3.51)\\
s.t. \quad & (3.13), (3.16), (3.17), (3.18), (3.19), (3.20), (3.21), && (3.52)\\
& (3.22), (3.23), (3.24), (3.25), && (3.53)\\
& S_{ijkl} + S_{ijlk} \leq 2 - H_k - H_l, \qquad \forall i, j, k, l > k. && (3.54)
\end{aligned}
$$

Table 3.2: Comparison between HLPPT and CPT

	number of constraints	Number of variables	
		binary	continuous
CPT	$6n^4 + 3n^3 + 2n^2$	$\frac{n(n-1)}{2} + n$	$2n^4$
HLPPT	$n^4 + 5n^3 + 7n^2$	$\frac{n(n-1)}{2} + n$	$n^4 + 2n^3 + n^2$

As one can see in Table 3.2, in our new model HLPPT, the number of constraints is much smaller than that of CPT. Roughly speaking, it contains almost less than $\frac{1}{6}$ of constraints in CPT. With respect to the number of variables, although they both use the same number of binary variables, the number of continuous variables in HLPPT is considerably less than in CPT.

Due to the complexity and memory usage of MIP solution methods like branch-and-bound algorithm and also hardware restrictions, it is very helpful to have a more compact and of course tighter model.

3.2.5 Computational Comparison

In this section, we are going to solve instances of AP and CAB data set using both CPT and HLPPT models. We compare *root node gaps(r.n.g)*, *cpu time usage(c.t.u)* and problem size that can be solved by each one. Here, we set $I_{kl} = 500C_{kl}$ and $F_i = 5000$ for AP instances and $I_{kl} = 200C_{kl}$ and $F_i = 2000$ for CABs.

Table 3.3: Comparison between HLPPT and CPT on CAB instances

	CPT		HLPPT	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
CAB 5	27.08	0.50	*opt*	0.03
CAB 10	36.30	19.81	*opt*	0.42
CAB 15	64.35	461.63	*opt*	2.27
CAB 20	59.95	4596.49	*opt*	9.09
CAB 25	77.38	$\gg$	*opt*	28.23

Table 3.4: Comparison between HLPPT and CPT on AP instances

	CPT		HLPPT	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.1	*opt*	0.03
AP 10.4	39.74	26.58	38.99	8.81
AP 15.6	39.87	1055.46	67.75	318.24
AP 20.8	39.89	12564.14	42.75	3683.07
AP 25.10	51.15	> 1 d[a]	44.55	56839.31
AP 30.12	N.A.[b]	N.A.	43.27[c]	N.A.

[a] N.A., Not able to solve the instance.
[b] The root note relaxation was solved.
[c] d: day.

As it is depicted in the Table 3.3, there is a considerable difference between the root node gaps of HLPPT and CPT on CAB instances. In fact, HLPPT can be used to solve all the instances of CAB dataset just in the root node to integral optimal solution. That is the LP relaxation bounds coincide with the MIP optimal value. It is also clear that HLPPT overtakes the other one in terms of computation time which is a fraction of CPT. In Table 3.4, again, the superiority of HLPPT to CPT with respect to the computational time is substantiated. For $n = 30$, CPT cannot even load the model in the memory where it takes more than 1.4 GB of memory and `CPLEX 9.1` can not handle it. In Contrary, HLPPT can load, perform the primary computation and not only emphasize the feasibility of problem but also the root relaxation is solved successfully and gap is reported. However, `CPLEX 9.1` failed when proceeded.

In general, HLPPT proves to be superior to the CPT. With respect to the computational effort and the cpu time usage, obviously HLPPT outperforms CPT when `CPLEX 9.1` is used to solve both instances of AP and CAB. Especially, in the case of CAB instances, HLPPT can be used to solve even more than 500 times faster for some instances in comparison with CPT.

All the reported instances, are solved on a AMD Opteron (tm) Processor 250, 2.4 GHz and 1 GB of RAM.

3.3 Extensions to the Base Model of HLPPT

In Subsection (3.2.3), a basis model namely HLPPT is proposed. Now, we are going to introduce other extensions. For example, other assumptions like Ass. (c) is going to be relaxed to let direct connections between spoke nodes. In our application, occurrences of such situations are very likely. Some other aspects like incorporation of additional costs for unforeseen delays are also important issues. Special configuration of hub-level network is also important in some other applications. Moreover, decisions made by the planners are not usually made for a single period, rather a time horizon is behind most of the decisions. These are among the motivations to propose new models.

3.3.1 HLPPT Under Non-Restrictive Network Policy

In HLPPT, a flow emanated from one node and destined to another one must be passed through at least one hub node. It is also possible that flow between two spoke nodes transits via spoke edges or in other words, there may be direct connections between spoke nodes (if it provides a better total cost). A new model, we call it Non-Restrictive HLPPT (NRHLPPT), which treats this variant can be expressed follows.

Mathematical Model

The new model would be as it follows:

(NRHLPPT)

$$
\begin{aligned}
Min \quad & \sum_{i}\sum_{j\neq i}\sum_{k}\sum_{l\neq k}\alpha W_{ij}C_{kl}x_{ijkl} + \sum_{i}\sum_{j\neq i}\sum_{k\neq i,j} W_{ij}C_{ik}a_{ijk} + \\
& \sum_{i}\sum_{j\neq i}\sum_{k\neq i,j} W_{ij}C_{kj}b_{ijk} + \sum_{i}\sum_{j\neq i} W_{ij}C_{ij}(e_{ij}+s_{ij}) + \\
& \sum_{k} F_k h_k + \sum_{k}\sum_{l>k} I_{kl}y_{kl} + \sum_{k}\sum_{l>k} J_{kl}z_{kl} \\
s.t. \quad & \sum_{l\neq i} x_{ijil} + \sum_{l\neq i,j} a_{ijl} + e_{ij} + s_{ij} = 1, && \forall i, j\neq i, \\
& \sum_{l\neq j} x_{ijlj} + \sum_{l\neq i,j} b_{ijl} + e_{ij} + s_{ij} = 1, && \forall i, j\neq i, \\
& \sum_{l\neq k,i} x_{ijkl} + b_{ijk} = \sum_{l\neq k,j} x_{ijlk} + a_{ijk}, && \forall i, j\neq i, k\neq i,j, \\
& (3.39), (3.40), (3.41), (3.42), (3.43),
\end{aligned}
$$

$$(3.44), (3.45), (3.46), (3.47), (3.48),$$
$$s_{kl} \leq z_{kl}, \quad s_{kl} \leq z_{lk}, \qquad \forall k, l > k,$$
$$x_{ijkl}, a_{ijk}, b_{ijk}, e_{ij}, s_{ij}, z_{ij} \in (0, 1), h_k, y_{kl} \in \{0, 1\}.$$

In addition to the existing variables and parameters of HLPPT, new parameters J_{kl} and new variables z_{kl} and s_{ij} are incorporated. Usually in real-life, the direct spoke edges are available and are not subjected to a setup cost, therefore a maintenance cost is charged for keeping the existing edge operative and recovering the depreciations. The parameter J_{kl} for all $k, l > k$ is the amount that is charged to keep this type of edges in a ready-to-service status. Variables z_{kl} for all $k, l > k$ are introduced to take 1, if the spoke-to-spoke edge $k - l$ is selected to be used as a direct connection between two spoke nodes, 0 otherwise. The variable s_{ij} is introduced to take 1 if the optimal path from i to j traverses spoke-to-spoke edge $i - j$ while none of i and j are hub nodes and 0, otherwise. Although the variables s_{ij} are defined to take binary values but with the similar argument mentioned earlier, they can also be relaxed.

3.3.2 HLPPT with Delay Considerations (DHLPPT)

Delay time or connecting time are usually non-ignorable aspects of every transportation system. The delay time addresses the amount of time between getting down from a transportation vehicle and waiting for the next possibility to get on. This connecting time implicitly imposes some costs.

These costs can also have other interpretations. For example, if the passengers from origin i to destination j lose their connecting train at station k due to an unanticipated delay and the statistics of such occurrences already exist, this cost can be interpreted as the average cost of missing for the passengers of origin i to destination j at hub node k.

Mathematical Model

To extend our existing base model to incorporate delay costs, we consider the delay to appear where the type of edge changes. That is, when passengers change the spoke edge and wait to get on the fast-line vehicles or they get down from a fast-line and take the spoke edge type facilities to reach destinations. Clearly, that is the most important place where any unforseen delay in the hub-level network imposes the amount of time the passenger has to wait.

A new variable $\delta_{ijk} = 1$, if the path from i to j which traverses the spoke edge $i - k$ or $k - j$ does not traverse via a single hub node in the path from i to j and 0, otherwise. The parameter C_{ijk}^{del} is the cost per person of the corresponding delay.

(DHLPPT)

$$
\begin{aligned}
Min \quad & \sum_i \sum_{j \neq i} \sum_k \sum_{l \neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j \neq i} \sum_{k \neq i,j} W_{ij} C_{ik} a_{ijk} + \\
& \sum_i \sum_{j \neq i} \sum_{k \neq i,j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j \neq i} W_{ij} C_{ij} e_{ij} + \\
& \sum_k F_k h_k + \sum_k \sum_{l > k} I_{kl} y_{kl} + \sum_i \sum_{j \neq i} \sum_{k \neq i,j} C_{ijk}^{del} \delta_{ijk} \\
s.t. \quad & (3.36), (3.37), (3.38), (3.39), \\
& (3.40), (3.41), (3.42), (3.43), \\
& (3.44), (3.45), (3.46), (3.47), (3.48), \\
& \delta_{ijk} \geq a_{ijk} - b_{ijk}, \qquad \forall i, j \neq i, k \neq i, j, \\
& \delta_{ijk} \geq b_{ijk} - a_{ijk}, \qquad \forall i, j \neq i, k \neq i, j, \\
& x_{ijkl}, a_{ijk}, b_{ijk}, e_{ij} \in (0, 1), h_k, y_{kl}, \delta_{ijk} \in \{0, 1\}.
\end{aligned}
$$

Now, the transportation cost between i and j is measured by the transportation from i to the first hub node; plus the delay cost if the origin is a spoke node and destination is not a spoke node allocated to the first hub node; plus the transportation cost on the hub-level network; plus the delay time for catching the first possibility to travel to the destination if the destination is a spoke node and origin is not a spoke node allocated to the last hub node.

3.4 Additional HLP Models Derived from HLPPT

We use this opportunity to introduce other variants. Although they might not be directly used in the public transport applications, the flexibility of our formulation lead to the emergent of such variants. These variants, as for some of them reported in literature are useful in IT and telecommunication applications.

3.4.1 Tree-Shaped HLPPT (TSHLPPT)

Often, due to some reasons, where deficiency of resources, political and geographical issues are some of them it is required to have just a single path between each pair of origin-destination.

In presence of the triangle inequality as a result of Theorem 3.2.4, the hub-level network is a tree. However, if the triangle inequality does not hold, due to the connectivity of hub-level network, in order to have a tree-shaped one, we only need to refer to the property of tree graphs.

Proposition 3.4.1. *For a given graph G(V,E), the following statements are equivalent:*

i). G is a tree,

ii). Any two vertices of G are connected by a unique path,

iii). G is connected and $|E| = |V| - 1$*,*

iv). G is connected and has no cycle.

Therefore we have to add the following constraint.

$$\sum_{k,l>k} y_{kl} = \sum_{k} h_k - 1. \tag{3.55}$$

Mathematical Model

TSHLPPT model would be as it follows:

(TSHLPPT)

$$\begin{aligned}
Min \quad & \sum_{i}\sum_{j\neq i}\sum_{k}\sum_{l\neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_{i}\sum_{j\neq i}\sum_{k\neq i,j} W_{ij} C_{ik} a_{ijk} + \\
& \sum_{i}\sum_{j\neq i}\sum_{k\neq i,j} W_{ij} C_{kj} b_{ijk} + \sum_{i}\sum_{j\neq i} W_{ij} C_{ij} e_{ij} + \\
& \sum_{k} F_k h_k + \sum_{k}\sum_{l>k} I_{kl} y_{kl} \\
s.t. \quad & (3.36), (3.37), (3.38), (3.39) \\
& (3.40), (3.41), (3.42), (3.43) \\
& (3.44), (3.45), (3.46), (3.47), \\
& (3.48), (3.50), \\
& \sum_{k,l>k} y_{kl} = \sum_{k} h_k - 1.
\end{aligned}$$

3.4.2 Polygonal HLPPT (PHLPPT)

A polygonal HLPPT or PHLPPT, is an HLPPT where its hub-level network is a polygon (triangle, rectangle, pentagon, etc.).

Proposition 3.4.2. *In each polygonal graph $G(V,E)$, $deg(V) = 2$.*

Mathematical Model

THLPPT is depicted in the sequel:

(PHLPPT)

$$\begin{aligned}
Min \quad & \sum_i \sum_{j\neq i} \sum_k \sum_{l\neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j\neq i} \sum_{k\neq i,j} W_{ij} C_{ik} a_{ijk} + \\
& \sum_i \sum_{j\neq i} \sum_{k\neq i,j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j\neq i} W_{ij} C_{ij} e_{ij} + \\
& \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} \\
s.t. \quad & (3.36), (3.37), (3.38), (3.39) \\
& (3.40), (3.41), (3.42), (3.43) \\
& (3.44), (3.45), (3.46), (3.47), \\
& (3.48), (3.50) \\
& \sum_{l\neq k} (y_{kl} + y_{lk}) = 2h_k, \qquad \forall k, \\
& \sum_k h_k = q.
\end{aligned}$$

where q is the number of corners in the polygon.

3.5 Multi-Period Hub Location Model for Public Transport

So far, we dealt with single period planning. That means, there was no transition from one period t to another consecutive one, $t+1$ and all the activities have been supposed to be carried out at a given time. Very often, decisions in such construction projects which are resource-demanding and long-lasting are not made for a single period. In other words, more detailed study of such systems can be accomplished by defining the project on a planning horizon with several periods. Here, we aim to model the problem to consist of planning for T periods ($T \geq 2$). In this section we pave the way of proposing, to the best of our knowledge, the first multi-period HLP model for public transport planning.

Mathematical Model

Some assumptions should be considered for this model. These assumption are:

Ass. i) If one node is selected to be a hub in a period $t \in \{1, .., T\}$, it will be performing as hub node until the end of planning horizon.

Ass. ii) If there has been a hub edge established in the network in any period, it will always be available as a hub edge.

We call the preliminary model proposed for this scenario by PRE-MPHLPPT:

(PRE-MPHLPPT)

$$
\begin{aligned}
Min \quad & \sum_{t \in T}[\sum_{i}\sum_{j \neq i}\sum_{k}\sum_{l \neq k} \alpha^t W_{ij}^t C_{kl}^t x_{ijkl}^t + \sum_{i}\sum_{j \neq i}\sum_{k \neq i,j} W_{ij}^t C_{ik}^t a_{ijk}^t + \\
& \sum_{i}\sum_{j \neq i}\sum_{k \neq i,j} W_{ij}^t C_{kj}^t b_{ijk}^t + \sum_{i}\sum_{j \neq i} W_{ij}^t C_{ij}^t e_{ij}^t + \\
& \sum_{k} F_k^t h_k^t + \sum_{k}\sum_{l > k} I_{kl}^t y_{kl}^t] \\
s.t. \quad & \sum_{l \neq i} x_{ijil}^t + \sum_{l \neq i,j} a_{ijl}^t + e_{ij}^t = 1, && \forall i, j \neq i, \\
& \sum_{l \neq j} x_{ijlj}^t + \sum_{l \neq i,j} b_{ijl}^t + e_{ij}^t = 1, && \forall i, j \neq i, \\
& \sum_{l \neq k,i} x_{ijkl}^t + b_{ijk}^t - \sum_{l \neq k,j} x_{ijlk}^t - a_{ijk}^t = 0, && \forall i, j \neq i, k \neq i, j, \\
& y_{kl}^t \leq \sum_{t'=1}^{t} h_k^{t'}, && \forall k, l > k, t \in T, \\
& y_{kl}^t \leq \sum_{t'=1}^{t} h_l^{t'}, && \forall k, l > k, t \in T, \\
& x_{ijkl}^t + x_{ijlk}^t \leq \sum_{t'=1}^{t} y_{kl}^{t'}, && \forall i, j \neq i, k, l > k, t \in T, \\
& \sum_{l \neq k} x_{kjkl}^t \leq \sum_{t'=1}^{t} h_k^{t'}, && \forall j, k \neq j, t \in T, \\
& \sum_{k \neq l} x_{ilkl}^t \leq \sum_{t'=1}^{t} h_l^{t'}, && \forall i, l \neq i, t \in T,
\end{aligned}
$$

$$e_{ij}^t \leq 2 - \sum_{t'=1}^{t} (h_i^{t'} + h_j^{t'}) \qquad \forall i, j \neq i, t \in T,$$

$$a_{ijk}^t \leq 1 - \sum_{t'=1}^{t} h_i^{t'}, \qquad \forall i, j \neq i, k \neq i, j, t \in T,$$

$$b_{ijl}^t \leq 1 - \sum_{t'=1}^{t} h_j^{t'}, \qquad \forall i, j \neq i, l \neq i, j, t \in T,$$

$$a_{ijk}^t + \sum_{l \neq j,k} x_{ijlk}^t - \sum_{t'=1}^{t} h_k^{t'} \leq 0, \qquad \forall i, j \neq i, k \neq i, j, t \in T,$$

$$b_{ijk}^t + \sum_{l \neq k,i} x_{ijkl}^t - \sum_{t'=1}^{t} h_k^{t'} \leq 0, \qquad \forall i, j \neq i, k \neq i, j, t \in T,$$

$$e_{ij}^t + 2x_{ijij}^t + \sum_{l \neq j,i} (x_{ijil}^t + x_{ijlj}^t) \leq \sum_{t'=1}^{t} (h_i^{t'} + h_j^{t'}), \qquad \forall i, j \neq i, t \in T,$$

$$\sum_{t'=1}^{t} h_k^{t'} \leq 1, \qquad \forall k, t \in T, \tag{3.56}$$

$$\sum_{t'=1}^{t} y_{kl}^{t'} \leq 1, \qquad \forall k, l > k, t \in T, \tag{3.57}$$

$$x_{ijkl}^t, a_{ijk}^t, b_{ijk}^t, e_{ij}^t \in (0, 1), y_{kl}^t, h_k^t \in \{0, 1\}.$$

With respect to these assumptions, we have two more constraints to guaranty these conditions. These constraints are (3.56) and (3.57). Interpretation of variables is the same as before and of course, with an additional index $t \in T$ indicating the time period this variable belongs to.

Although, this model works fine, it has a big drawback. That is, in the optimal solution, it establishes all the facilities in the first period. In an abstract sense, this is quite logical to establish all the facilities in the first period and take the advantage of economy of scale arisen from their functionalities throughout the planning horizon. In fact, there is no optimal solution for this model which opens a hub facility in a period $1 < t \leq T$, if the parameters are chosen realistically. This is because, it can be established in the $t = 1$ and its advantages be used in the period 1 as well as other periods. However, this is not something we expect in reality.

The occurrence of such situations is mainly due to the fact that, there are also other issues which are missed in the model and should be taken into account in a multi-period planning. Other types of costs, should also be incorporated into the model. In presence of this type of cost, opening a hub facility does not only lead to benefit but also incurs some additional costs.

Later on in chapter 8 we will refer to the multi-period approach in detail.

Chapter 4

Lagrangian Relaxation and Integrality Property

In this chapter we are going to present a Lagrangian relaxation approach for HLPPT. At first glance, this Lagrangian approach is expected to be a useful tool and the only concern is whether it will converge at a reasonable amount of time.

Actually, after implementing and running the algorithm, interesting results were observed. A trivial decomposition of the problem into a sub-problem and a set of dualized constraints turned out to leave the integrality property in the sub-problem.

In this chapter, we prove that the convex hull of feasible solutions of the sub-problem is equal to the convex hull of its LP relaxation. In general, such a proof, which is based on the concept of Totally Unimodularity (TU), is in the complement of $\mathcal{NP}$, i.e. $\mathcal{CNP}$.

4.1 Lagrangian Approach for HLPPT

A trivial decomposition of HLPPT results in a sub-problem and a bunch of complicating constraints which should be dualized.

(HLPPT)

$$\begin{aligned}
Min \quad & \sum_i \sum_{j\neq i} \sum_k \sum_{l\neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j\neq i} \sum_{k\neq i,j} W_{ij} C_{ik} a_{ijk} + \\
& \sum_i \sum_{j\neq i} \sum_{k\neq i,j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j\neq i} W_{ij} C_{ij} e_{ij} + \\
& \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} && (4.1) \\
s.t. \quad & (3.36), (3.37), (3.38), (3.41), (3.42) && (4.2) \\
& (3.46), (3.47), (3.48), (3.50), && (4.3) \\
& y_{kl} \leq h_k, \quad y_{kl} \leq h_l, & \forall k, l > k, & \quad (4.4) \\
& e_{ij} \leq 2 - (h_i + h_j), & \forall i, j \neq i, & \quad (4.5) \\
& a_{ijk} \leq 1 - h_i, & \forall i, j \neq i, k \neq i, j, & \quad (4.6) \\
& b_{ijl} \leq 1 - h_j, & \forall i, j \neq i, l \neq i, j, & \quad (4.7) \\
& x_{ijkl} + x_{ijlk} \leq y_{kl}, & \forall i, j \neq i, k, l > k. & \quad (4.8)
\end{aligned}$$

Now, a Lagrangian relaxation of the problem is developed as follows. We shall relax all the constraints, except those which only include the hub-level variables and those of (4.4)-(4.8) in a Lagrangian fashion. Since the right-hand sides are 0, 1 or 2 (in the similar scale), it is not necessary to re-scale the constraints. Also, appropriate multipliers following the lagrangian algorithm, are defined. Eventually, the sub-gradient algorithm iterations can be used to solve the problem.

Actually, after implementing the approach, we observed that even if the convergence appears fast enough (which has not been usually the case!) the lower bound will never be better than that of the LP relaxation. The reason is the subject of the next section.

4.2 Integrality Property

Our aim to apply Lagrangian relaxation is so that, at least for some instances, we can find lower bounds better than those that can be found by LP relaxation (although we should expect at least as good as the LP bounds) and of course fast enough. That is, we solve the Lagrangian objective function on the convex hull of a subset of constraints and expect such a bound. This bound may later be used in a branch-and-bound system instead of an LP bound, in measuring the quality of solution of a heuristic etc. .

Assume the problem (P) as follows:

$$(P) : \underset{x}{min}\{fx|Ax \leq b, Cx \leq d, x \in X\} \tag{4.9}$$

where X contains sign restrictions on x and integrality restrictions, i.e. $X = \mathbb{R}^{n-p} \times \mathbb{R}^p$, $X = \mathbb{R}^{n-p}_+ \times \mathbb{R}^p_+$ or $X = \mathbb{R}^{n-p}_+ \times \{0, 1\}^p$. $Ax \leq b$ are assumed complicating in the sense that P without them would be simpler to solve (see [36]).

Definition 4.2.1. The Lagrangian relaxation of (P) relative to the complicating constraints $Ax \leq b$, with non negative Lagrangian multiplier λ, is the problem,

$$(LR_\lambda) : \underset{x}{min}\{fx + \lambda(Ax \leq b)|Cx \leq d, x \in X\}. \tag{4.10}$$

Theorem 4.2.1. *The Lagrange dual (LR) is equivalent to the primal relaxation*

$$(PR) : \underset{x}{min}\{fx|Ax \leq b, x \in Co\{x \in X|Cx \leq d\}\}, \tag{4.11}$$

in the sense that $v(LR) = v(PR)$. Here, $Co(Y)$ stands for the convex hull of Y.

Corollary 4.2.1. *If $Co\{x \in X|Cx \leq d\} = \{x|Cx \leq d\}$, then $v(LP) = v(PR) = v(LR) \leq v(P)$.*

Definition 4.2.2. One says that (LR) has Integrality Property if $Co\{x \in X|Cx \leq d\} = \{x|Cx \leq d\}$.

If Corollary 4.2.1 holds, then the Lagrangian bound is equal to (cannot be better than) the LP bound.

We claim that the polytop of the following constraints which belong to the constraint set of our LR_λ for HLPPT possesses the integrality property.

$$y_{kl} - h_k \leq 0, \qquad \forall k, l > k, \quad (4.12)$$

$$y_{kl} - h_l \leq 0, \qquad \forall k, l > k. \quad (4.13)$$

Definition 4.2.3. An $m \times n$ integral matrix A is totally unimodular (TU) if the determinant of each square sub-matrix of A is equal to 0, 1, or -1.

Theorem 4.2.2 (Polyhedron Integrality)**.** *If A is TU, then $P(b) = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$ is integral for all $b \in \mathbb{Z}^m$ for which it is not empty.*

Proposition 4.2.3. *If the $(0, 1, -1)$ matrix A has no more than two nonzero entries in each column, and if $\sum_i a_{ij} = 0$ if column j contains two non-zero coefficients, then A is TU.*

Proposition 4.2.4. *The following statements are equivalent.*

1. *A is TU.*
2. *The transpose of A is TU.*
3. *(A,I) is TU.*
4. *A matrix obtained by a pivot operation on A is TU.*
5. *A matrix obtained by interchanging two rows (columns) of A is TU.*
6. *A matrix obtained by multiplying a row (column) of A by -1 is TU.*
7. *A matrix obtained by deleting a unit row (column) of A is TU.*
8. *A matrix obtained by duplicating columns (rows) of A is TU.*

Theorem 4.2.5. *The coefficient matrix of the constraint set (4.12) and (4.13) is TU.*

Proof. The transpose of the coefficient matrix of constraint set (4.12) and (4.13) together with the equivalency conditions of Proposition 4.2.4 satisfies the conditions of Proposition 4.2.3. Therefore, it is TU. □

4.3 Sub-Problem Integrality

According to our experiences, constraints (3.36), (3.37), (3.38), (3.41), (3.42), (3.46), (3.47) and (3.48) are the most complicating constraints. Their appearance in the sub-problem together with the computational efforts which one experiences while dealing with the Lagrangian relaxation and the sub-gradient algorithm, does not lead to an efficient solution method. That is, the appearance of any of these constraints in the sub-problem makes it dramatically harder to solve even by inspection, if possible.

As mentioned earlier, the following constraints are selected to form the sub-problem.

$$x_{ijkl} + x_{ijlk} \leq y_{kl}, \qquad \forall i, j \neq i, k, l > k, \quad (4.14)$$
$$e_{ij} \leq 2 - (h_i + h_j), \qquad \forall i, j \neq i, \quad (4.15)$$
$$a_{ijk} \leq 1 - h_i, \qquad \forall i, j \neq i, k \neq i, j, \quad (4.16)$$
$$b_{ijl} \leq 1 - h_j, \qquad \forall i, j \neq i, l \neq i, j, \quad (4.17)$$
$$y_{kl} \leq h_k, \qquad \forall k, l > k, \quad (4.18)$$
$$y_{kl} \leq h_l, \qquad \forall k, l > k. \quad (4.19)$$

Now, we shall show that the sub-problem possesses the integrality property.

Theorem 4.3.1. *The matrix of constraint set of (4.14)-(4.19) is TU. Moreover, its polyhedron possesses the integrality property.*

Proof. We are going to show that according to Proposition 4.2.4, without loss of generality, for each $i, j \neq i$, the matrix made by columns corresponding to a proposed *lexicographical ordering* of variables is an *identity* matrix. Moreover, these matrices can build up the matrix of coefficients of (4.14)-(4.19) which is going to be proven as a TU matrix.
For a given $i, j \neq i$, a lexicographical ordering of variables of constraints (4.14)-(4.19) is defined by:

$$\underbrace{x_{ijkl}}_{\forall k,l:l\neq i,k\neq j,l}, \underbrace{a_{ijm}}_{\forall m\neq i,j}, \underbrace{b_{ijm}}_{\forall m\neq i,j}, e_{ij} \qquad \forall i, j \neq i. \quad (4.20)$$

The complete matrix can be represented by,

$$\underbrace{x_{12kl}}_{k\neq 2,l\neq 1,l\neq k}, \underbrace{a_{12m}}_{\forall m\neq 1,2}, \underbrace{b_{12m}}_{\forall m\neq 1,2}, e_{12}, \ldots, \underbrace{x_{pqkl}}_{k\neq q,l\neq p,l\neq k}, \underbrace{a_{pqm}}_{\forall m\neq p,q}, \underbrace{b_{pqm}}_{\forall m\neq p,q}, e_{pq}, \ldots,$$

$$\underbrace{x_{N\ N-1\ kl}}_{\forall k\neq N-1, l\neq N, l\neq k}, \underbrace{a_{N\ N-1\ m}}_{\forall m\neq N-1,N}, \underbrace{b_{N\ N-1\ m}}_{\forall m\neq N-1,N}, e_{N\ N-1}, \underbrace{y_{kl}}_{\forall l>k}, \underbrace{h_k}_{\forall k}. \tag{4.21}$$

Every column of matrix built by (4.20), is a unit column where exactly a single '1' therein exists. Since, in none of the constraints two types of variables appears in each row from the first column until the variable e_{ij}, only one '1' exists, except for $x_{ijkl}, l \neq i, k \neq j$ where two x variables are associated with each of the constraints. Moreover, in each row there exist(s) one or two +1 or -1 correspond to the variables h and y.

According to Proposition 4.2.4 by pivoting on each row of matrix we achieve a matrix which contains 1's in the diagonal for each of the matrices in the form of (4.20). Since, the variables are ordered in such a form that build (4.21), we can have a matrix of the form,

$$\begin{gathered}\underbrace{x_{12kl}}_{k\neq 2, l\neq 1, l\neq k}, \underbrace{a_{12m}}_{\forall m\neq 1,2}, \underbrace{b_{12m}}_{\forall m\neq 1,2}, e_{12}, \ldots, \underbrace{x_{pqkl}}_{k\neq q, l\neq p, l\neq k}, \underbrace{a_{pqm}}_{\forall m\neq p,q}, \underbrace{b_{pqm}}_{\forall m\neq p,q}, e_{pq}, \ldots, \\ \underbrace{x_{N\ N-1\ kl}}_{\forall k\neq N-1, l\neq N, l\neq k}, \underbrace{a_{N\ N-1\ m}}_{\forall m\neq i,j}, \underbrace{b_{N\ N-1\ m}}_{\forall m\neq i,j}, e_{N\ N-1}.\end{gathered} \tag{4.22}$$

Therefore, (4.22) is a matrix with the values '1' in its diagonal.

By adding the columns of variables h and y, one can pivot again to keep the row unit vectors. This act does not affect the parts related to the constraints (4.18)-(4.19). Now, we have a matrix which contains an identity matrix block in the form,

$$\begin{pmatrix} I & 0 \\ 0 & HY \end{pmatrix}. \tag{4.23}$$

Here, HY stands for the coefficients of the constraints (4.18)-(4.19).
In order to prove the totally unimodularity of (4.23), it suffices to show that any square sub-matrix of it has the determinant of 0,-1 or +1. Choose a square sub-matrix of (4.23), say B:

Case 1. B has all its entries only from I and therefore determinant is either 1 or 0.

Case 2. B has its entries only adopted from HY which is TU, according to the Theorem 4.2.5.

Case 3. B has entries from both I and HY. Therefore, either it has some zero rows (columns) which determinant is 0, or there is exactly one '1' in rows (columns) from I that determinant is the determinant of a square sub-matrix from HY. In any case the

(4.23) is TU.
Moreover, since RHS of (4.14)-(4.19) is integral, therefore according to the Theorem 4.2.2 the polyhedron is integral and proof is completed.

□

Theorem 4.3.2. *Any matrix which contains only an identity block and a TU sub-matrix block, is a TU matrix.*

Proof. Similar to the proof of the Theorem (4.3.1). □

We close this chapter by having already proven that, for the suggested sub-problem, the integrality condition holds. In addition to that, based on our computational experiences; i) the dualized constraints are very complicating and, ii) adding even one set of constraints from the dualized ones to the sub-problem is not a good idea. Even in this case some behaviors in the computational results is observed that even if not indicating the integrality property, displays a very poor performance.

Chapter 5

Strengthened Formulation

In this chapter we are going to examine the strength of our formulation. That is, how would the integrality gap be, or in other words, how well the LP relaxation polytop surrounds the MIP polytop of HLPPT. Subsequently, we will introduce several classes of valid inequalities to yield an equally strong possible formulation. The idea of introducing these kinds of inequalities is inspired by [57]. Our inequalities are capable of cutting off parts of the polytop of LP relaxation that do not contain the optimal solution of the MIP of HLPPT. By means of some preprocessing, we reduce the number of variables to half and considerably decrease the number of constraints. This accelerates the resolution and allows larger instance to be solved.

In [57], the formulation of the problem is strengthened by means of powerful valid inequalities obtained through the study of the intersection graph of an associated set packing problem. A generic transformation of the problem to a set packing problem is accomplished and the intersection graphs of the new model are found.

We would like to remind that, in Table 3.3, it has been observed that HLPPT reports the solutions for all the CAB instances to the optimality just at the root node while it indicates existing gaps for the AP instances. Therefore, we only deal with AP instances in this chapter.

Eventually, we will close this chapter by having already reported a very strong formulation where its LP relaxation can solve some of the instances of our problem to integer optimality. The rest can be solved at the root node after adding some cuts by the solver itself. Just for one instance the branching proceeds and proves optimality of the solution at the first node of the branch-and-bound tree. This time, larger instances of the problem can also be solved.

5.1 Improving Valid Inequalities

In this section some of the improving valid inequalities will be suggested. We go step by step and report the progress of improvement and the implication of introducing new classes of valid inequalities.

(A): The integrality gap and the computational time of those problem instances which were solved by `CPLEX 9.1` on our machine are reported in the Table 5.1.

Table 5.1: Computational Results.

	HLPPT	
Instance	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03
AP 10.4	38.99	8.81
AP 15.6	67.75	318.24
AP 20.8	42.75	3683.07
AP 25.10	44.55	56839.31
AP 30.12	43.27	N.A.

(B): Two constraints (5.1) and (5.2) are added. New results are depicted in Table 5.2. The underlying idea is that, there exists at least one hub edge (and as a result at least two hub nodes) in the optimal solution of any instance.

$$\sum_{k,l>k} y_{kl} \geq 1, \tag{5.1}$$

$$\sum_{k} h_k \geq 2. \tag{5.2}$$

From now on, in each table, after the instance name column, the first two columns are the best former results and the second two columns stand for the results after adding the inequalities. The third two columns are the improvement achieved after adding new constraint(s).

Table 5.2: Computational Results.

	HLPPT$^{(A)}$		HLPPT$^{(B)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	38.99	8.81	26.30	12.58	12.69	-3.77
AP 15.6	67.75	318.24	29.04	514.35	38.71	-196.11
AP 20.8	42.75	3683.07	24.19	3282.89	18.56	400.18
AP 25.10	44.55	56839.31	53.81	29755.70	-9.26	27083.61

As one can observe in Table 5.2, in general the formulation becomes stronger. Regarding the computational time, as the instance size grows, this new model can solve the instances much faster. Even in the last case it succeeds to reduce the computational time up to 52% of the former one.

(C): Another valid inequality which is found by exploiting the optimal solution of LP problems is (5.3). It is derived from a basic property of connected graphs: the number of hub edges is greater or equal to one less than the number of hub nodes. If the values of h, and/or y are fractional it is observed that this constraint can be violated.

$$\sum_{k,l>k} y_{kl} \geq \sum_{k} h_k - 1. \tag{5.3}$$

Table 5.3: Computational Results.

	HLPPT$^{(B)}$		HLPPT$^{(C)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	26.30	12.58	*opt*	0.75	26.30	11.83
AP 15.6	29.04	514.35	*opt*	14.56	29.04	499.79
AP 20.8	24.19	3282.89	14.79	570.70	9.40	2712.19
AP 25.10	53.81	29755.70	14.02	5050.35	39.79	24705.35

As it can be observed in Table 5.3, this cut drastically affects the strength of our formulation. The root node gap as well as the computational time are significantly reduced. Two of the other problems (in total 3 instance) are solved to optimality by just solving them as linear programming model.

Remark. An important result is that, all the formulations derived from HLPPT$^{(C)}$ (as long as some thing else is not said explicitly) help the solver to avoid branching. That is, the solver is able to add enough cuts so that the problem can be solved just at the root node rather than following the branching process.

(D): Assume that a hub edge is partially opened, say $0 < y_{ij} = \beta < 1$. For a given pair of origin-destination, the spoke part of this hub edge should not be used more than $0 < 1 - \beta < 1$ as a spoke edge. Constraints (5.4) and (5.5) are introduced.

$$a_{ijk} + e_{ij} \leq 1 - (y_{ij} + y_{ji}), \qquad \forall i, j \neq i, k \neq i, j, \tag{5.4}$$

$$b_{ijk} + e_{ij} \leq 1 - (y_{ij} + y_{ji}), \qquad \forall i, j \neq i, k \neq i, j. \qquad (5.5)$$

Table 5.4: Computational Results.

	HLPPT$^{(C)}$		HLPPT$^{(D)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.75	*opt*	0.89	0	-0.14
AP 15.6	*opt*	14.56	*opt*	15.03	0	-0.47
AP 20.8	14.79	570.70	14.69	421.52	0.10	149.18
AP 25.10	14.02	5050.35	13.75	4684.67	0.27	365.68

Table 5.4 reports that the computational times of those time consuming instances (size 20 and 25) are considerably decreased and the integrality gap decreased as well.

(E): The following two sets of inequalities (5.6) and (5.7) showed some improvement on the formulation strength. The idea is that, if the hub edge y_{kl} is established, even if the triangle inequality does not hold it must be used for the direct transportation of flow between its end-points.

$$y_{kl} - x_{klkl} = 0, \qquad \forall k, l > k, \qquad (5.6)$$
$$y_{kl} - x_{lklk} = 0, \qquad \forall k, l > k. \qquad (5.7)$$

Table 5.5: Computational Results.

	HLPPT$^{(D)}$		HLPPT$^{(E)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.01	0	0.02
AP 10.4	*opt*	0.89	*opt*	0.66	0	0.23
AP 15.6	*opt*	15.03	*opt*	13.13	0	1.90
AP 20.8	14.69	421.52	14.62	393.82	0.07	37.21
AP 25.10	13.75	4684.67	13.65	5509.14	0.10	-824.47

The integrality gap decreased and the computational times except for the last one decreased (see Table 5.5). Although, in the last case the run-time has increased, experience shows to be worthwhile having these cuts added before admitting the coming ones.

(F): Constraints (5.8) and (5.9) are another two sets of valid inequalities that made some improvements. If the nodes i and j are both hub nodes, then the whole flow emanated from i and/or destined to j should be sent or received through a hub edge.

$$\sum_{l \neq i} x_{ijil} \geq h_i + h_j - 1, \qquad \forall i, j \neq i, \tag{5.8}$$

$$\sum_{l \neq j} x_{ijlj} \geq h_i + h_j - 1, \qquad \forall i, j \neq i. \tag{5.9}$$

Table 5.6: Computational Results.

	HLPPT$^{(E)}$		HLPPT$^{(F)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.01	*opt*	0.03	0	-0.02
AP 10.4	*opt*	0.66	*opt*	0.91	0	-0.25
AP 15.6	*opt*	13.13	*opt*	11.52	0	1.61
AP 20.8	14.62	393.82	14.55	404.95	0.07	-11.13
AP 25.10	13.65	5509.14	13.49	4828.40	0.16	680.74

Adding these constraints can slightly improve the strength of the formulation and also improves the computational time of a major challenging instance of 25 (see Table 5.6).

(G): If $\sum_{k,l>k} y_{kl} > 1$ in the LP relaxation optimal solution, it means that there should be more than one hub edge in the optimal solution of the problem. Therefore, upon this assumption, if an edge is a hub edge, or in terms of linear programming, if it is partially hub edge then the sum over all partial hub edges connected to it should be at least equal to it.

$$y_{kl} \leq \sum_{m \neq k} (y_{km} + y_{mk}) + \sum_{m \neq k} (y_{ml} + y_{lm}), \qquad \forall k, l > k. \tag{5.10}$$

Table 5.7: Computational Results.

	HLPPT$^{(F)}$		HLPPT$^{(G)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.91	*opt*	0.86	0	0.05
AP 15.6	*opt*	11.52	*opt*	11.25	0	0.27
AP 20.8	14.55	404.95	14.55	401.23	0	3.72
AP 25.10	13.49	4828.40	13.49	4795.38	0	33.02

As Table 5.7 reports, the computational times are decreased.

(H): If a node is a hub node, or in terms of linear programming, if it is partially hub node, then the sum over all partial hub edges connected to it should be at least equal to it.

$$h_k \leq \sum_{m \neq k} (y_{km} + y_{mk}), \qquad \forall k. \tag{5.11}$$

Table 5.8: Computational Results.

	HLPPT$^{(G)}$		HLPPT$^{(H)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.86	*opt*	0.67	0	0.19
AP 15.6	*opt*	11.25	*opt*	9.17	0	2.08
AP 20.8	14.55	401.23	14.55	320.23	0	81
AP 25.10	13.49	4795.38	13.36	2252.90	0.13	2542.48

The computational times are considerably reduced and the formulation is slightly tightened (see Table 5.8).

(I): If a node i is not a hub node, therefore as the first link in the path to its destination, j, it must use a spoke edge to send its flow.

$$\sum_{l \neq i,j} a_{ijl} + e_{ij} \geq 1 - h_i, \tag{5.12}$$

$$\sum_{l \neq j,j} b_{ijl} + e_{ij} \geq 1 - h_j. \tag{5.13}$$

Table 5.9: Computational Results.

	HLPPT$^{(H)}$		HLPPT$^{(I)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.67	*opt*	0.53	0	0.25
AP 15.6	*opt*	9.17	*opt*	8.52	0	2.26
AP 20.8	14.55	320.23	14.55	584.28	0	-264.05
AP 25.10	13.36	2252.90	13.36	6033.74	0	-3780.84

Although adding these constraints increases the computational time and does not improve the formulation, our experience shows that the existence of these constraints are beneficial before admitting the preprocessing and size reduction which will be explained in the next section (see Table 5.9).

5.2 Preprocessing

Preprocessing is a very useful tool in order to improve the performance of many solution methods. Most of the general purpose solvers use some kind of general preprocessing. By exploiting the special structure of HLPPT, some trivial but useful and successful preprocessing can improve the computational performance.

Theorem 5.2.1 (Preprocessing)**.** *In an optimal solution of* HLPPT *and for all* i, j, k, l *we have,*

$$e_{ij} = e_{ji}, \qquad \forall j \neq i, \tag{5.14}$$
$$a_{ijk} = b_{jik}, \qquad \forall j \neq i, k \neq i, j, \tag{5.15}$$
$$x_{ijkl} = x_{jilk}, \qquad \forall j \neq i, l \neq k, i \neq l, j \neq k. \tag{5.16}$$

Proof. Since the shortest path between two points is symmetric, therefore the flow of i to j in the optimal solution traverses the same path as j to i. Hence, all the elements of the path are common and coincide. □

5.2.1 Effect of Preprocessing on Computational Time

It has been observed that after this preprocessing, the run-time considerably decreased. In the Table 5.10, the computational time prior to and after the preprocessing are reported.

Table 5.10: Preprocessing.

	$\mathsf{HLPPT}^{(before)}$		$\mathrm{HLPPT}^{(after)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.53	*opt*	0.31	0	0.22
AP 15.6	*opt*	8.52	*opt*	3.69	0	4.83
AP 20.8	14.55	584.28	14.55	84.38	0	499.9
AP 25.10	13.36	6033.74	13.36	1090.12	0	4943.62

So far, the resulted formulation is considerably tight and computational time is drastically reduced after being subjected to a strengthening and preprocessing phase as represented in Table 5.10. Here, in the case of AP 20.8, the root node cuts ends up to a solution with the gap of 0.46%. Afterward, the optimal solution is found in the first node of the branch and bound method in `CPLEX 9.1`.

5.2.2 Size Reduction by Exploiting Symmetry

Since the symmetry holds in the shortest path between pairs of origin-destination, it is possible to only consider those constraints and variables addressing the flow of i to j where $j > i$. This can halve the number of such constraints and variables. That is, we define $W'_{ij} = W_{ij} + W_{ji}, \forall j > i$ and remove all the constraints as well as all the flow variables that are defined for $j \leq i$. Therefore, instead of doing preprocessing as we have done in Subsection 5.2.1, the formulation size is reduced.

The problem size of instances are quite huge. Loading and resolution of a more compact model is considerably faster than loading a larger model and dropping the constraints and variables by the preprocessing like in Subsection 5.2.1. Table 5.11 depicts the results of this type of preprocessing and compares it against that of the previous Subsection.

Comparing the initial and final computational times as is depicted, Table 5.12 shows that the improvement is absolutely considerable. For example, the instance of size 25

Table 5.11: Preprocessing by Symmetry.

	$\text{HLPPT}^{(Full)}$		$\text{HLPPT}^{(Halved)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	*opt*	0.03	0	0
AP 10.4	*opt*	0.31	*opt*	0.09	0	0.22
AP 15.6	*opt*	3.69	*opt*	2.30	0	1.39
AP 20.8	14.55	84.38	14.55	62.02	0	22.36
AP 25.10	13.36	1090.12	13.36	900.50	0	293.62
AP 30.12	N.A.[a]	$\gg$	17.40	5530.00	-	∞

which was solved in about 16 hours is now solved 15 minutes (63.11 times faster). Furthermore, the problem of size 30 that could not be solved is solved in approximately 1.5 hours. The model is so suitable for the solver that it can be used to solve almost all the possible instances at the root node by adding appropriate cuts. It should also be mentioned that, again for AP 20.8, the solver proved the optimality of its solution at the first node of the branch and bound.

Table 5.12: Final Comparison.

	$\text{HLPPT}^{(Initial)}$		$\text{HLPPT}^{(Final)}$		Improvement	
	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u	r.n.g(%)	c.t.u
AP 5.2	*opt*	0.03	0.00	0.03	0	0
AP 10.4	38.99	8.81	0.00	0.09	38.99	8.72
AP 15.6	67.75	318.24	0.00	2.30	67.75	315.94
AP 20.8	42.75	3683.07	14.55	62.02	20.20	3640.78
AP 25.10	44.55	56839.31	13.36	900.50	31.19	56042.81
AP 30.12	43.27	N.A.	17.40	5530.00	42.09	∞

From now on, whenever we refer to HLPPT, this last version which uses almost half of the variables is meant.

Chapter 6

Benders Decomposition Approach for HLPPT

Benders decomposition is a classical solution approach initially proposed for MIPs and is widely used for combinatorial optimization problems (COPs). The underlying idea of this approach is to decompose the problem into two smaller parts, namely a Master Problem (MP) and a Sub-Problem (SP). The master problem contains only the integer variables of the original problem, while the sub-problem is a linear programming problem with the continuous variables.

Since some of the constraints and variables are removed from the original problem to obtain the MP, it is considered as a relaxation. A solution to the sub-problem can be achieved by fixing the variables of the MP as parameters in the SP. The SP will be in charge of solving the remaining part of the original problem and generating cut(s). Corresponding to an optimal solution of the MP in each iteration, a single cut is generated from the SP dual solution. These cuts are added, one per iteration, to the master problem in order to cut off parts of the feasible space which contain no optimal values for the integer variables of the original problem and thus achieve a better approximation of the optimal solution.

The master and sub-problem are iteratively solved to optimality until no improving cut can be generated (although some works discard the necessity of resolution of the MP to optimality). The optimal solution of the master and sub-problem, jointly, represents the optimal solution of original problem.

6.1 Motivation of Applying on HLPs

HLPs are NP-Hard problems for which even moderate size instances cannot be solved to optimality in a reasonable amount of time. Furthermore, in some instances, hardware restrictions are also a constraining element. One expects that the decomposition of the problem into two smaller ones can be helpful to solve larger instances to optimality or even provide a good approximation of the optimal value, while it lies within a lower and an upper bound.

Benders decomposition algorithm, as cited earlier, was already proposed for HLPs in [73] and [74]. However, in both papers, the models with the classical assumptions were considered.

The structure of HLPPT in the form of multiple allocation benefits from the fact that once the hub-level network structure is determined and fixed, the remaining part is just a network flow problem which looks for the all-pairs shortest paths (perhaps with respect to some additional conditions).

A suitable model which can be tackled by Benders decomposition looks like the following MIP,

$$
\begin{aligned}
\text{P} : Min \quad & cx + fy && (6.1)\\
s.t. \quad & Ax + By \leq b, && (6.2)\\
& Dy \leq d, && (6.3)\\
& x \geq 0, y \in \{0, 1\}, && (6.4)
\end{aligned}
$$

or equivalently,

$$
\begin{aligned}
\text{MP} : Min \quad & fy + \eta(y)\\
s.t. \quad & Dy \leq d,\\
& y \in \{0, 1\},
\end{aligned}
$$

where,

$$
\begin{aligned}
\text{SP} : \eta(y) = \quad Min \quad & cx\\
s.t. \quad & Ax \leq b - By,\\
& x \geq 0.
\end{aligned}
$$

Indeed, our HLPPT fits in this structure. It is clear that HLPPT instances (and all its variants which are proposed or will be proposed later on) have finite optimal values for the solutions x^* and y^* (all the variables are bounded and positive, coefficients in the objective function are positive and the problem is a minimization problem).

Consider the SP and its dual, namely SPD:

Primal Sub-Problem (SP)

$$
\begin{aligned}
Min \quad & cx\\
s.t. \quad & \pi : Ax \leq b - By,\\
& x \geq 0.
\end{aligned}
$$

Sub-Problem Dual (SPD)

$$
\begin{aligned}
Max \quad & \pi(b - By) && (6.5)\\
s.t. \quad & \pi A \geq c, && (6.6)\\
& \pi \geq 0. && (6.7)
\end{aligned}
$$

Evidently, the primal is parameterized on the right hand side by y. Assuming that SP is finite implies that the primal is finite for at least one value of $y|Dy \leq d, y \in \{0, 1\}$. By the duality theorem of linear programming, the dual has to be feasible.

By the feasibility of the dual in LPs, we have $\pi A \geq c$. This dual feasible region is independent of y.

Corollary 6.1.1. *SP is finite for all* $y : Dy \leq d, y \in \{0, 1\}$.

Assuming that SP is feasible implies that the primal is feasible for at least one value of $y|Dy \leq d, y \in \{0, 1\}$. By the duality theorem of linear programming, the dual has to be finite.

The dual is finite if and only if,

$$\pi^j(b - By) \leq 0, \quad j = 1, \ldots, q,$$

where q is the number of extreme rays of the dual feasible region $\pi A \geq c$.

These constraints should be added to the master problem in order to guarantee the boundedness of the dual or, equivalently the feasibility of the primal.

Each time a feasible solution is found, the value $fy + \eta(y)$ is an upper bound on the optimal value of the original problem. Therefore, the optimal solution of the master problem should be better or at least as good as the previous iteration. This is accomplished by adding the cuts corresponding to each extreme point of the sub-problem dual to the master problem. These cuts are all in the form of:

$$\pi^i(b - By) \leq fy + \eta(y), \quad i = 1, \ldots, p,$$

where p is the number of extreme points of the dual feasible space.

Finally the full master problem follows:

(MP)

$$\begin{aligned} \eta(\theta) : Min \quad & fy + \theta \\ s.t. \quad & Dy \leq d, \\ & y \in \{0,1\}, \\ & \theta \geq \pi^j(b - By), \quad j = 1, \ldots, p, \\ & 0 \geq \pi^j(b - By), \quad j = 1, \ldots, q. \end{aligned}$$

By iteratively solving the master and the sub-problem, in each iteration, k, we achieve a lower and an upper bound as follows:

$$\begin{aligned} & Z^k_{LB} = fy^k + \theta^k \\ & Z^k_{UB} = fy^k + \eta(y^k). \end{aligned}$$

The structure of HLPPT is suitable for such a Benders algorithm (as it is explained above). One can expect this algorithm to perform as an efficient tool in order to solve instances of HLPPT.

6.2 Benders Approach for Single Period HLP in Public Transport

Following the Benders algorithm, the master problem would be:

(MP1)

$$\begin{aligned} Min \quad & \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} & & \\ s.t. \quad & y_{kl} \leq h_k, & \forall k, l > k, \quad & (6.8) \\ & y_{kl} \leq h_l, & \forall k, l > k, \quad & (6.9) \\ & \sum_{k,l>k} y_{kl} \geq 1, & & (6.10) \\ & y_{kl}, h_k \in \{0,1\}, & \forall k, l \neq k. \quad & (6.11) \end{aligned}$$

Having no hub node, just one hub node or, in the worst case, having two hub nodes without any hub edge does not make sense and destroys the structure of the HLPPT. Because the economy of scale is exploited by transportation of consolidated flows via hub edges, at least one hub edge and as a result at least two hub nodes must exist in the whole hub-and-spoke network.

The constraint (6.10) is added to the master problem to ensure that there exists at least one hub edge in any optimal solution of MP1. The sub-problem for the fixed variables of y_{kl} and h_k as RHS parameters is also considered from the compacted model after the preprocessing step of Subsection 5.2.2 as follows:

(SP)

$$
\begin{aligned}
Min \quad & \sum_{i}\sum_{j>i}\sum_{k}\sum_{l\neq k}\alpha(W_{ij}+W_{ji})C_{kl}x_{ijkl}+\sum_{i}\sum_{j>i}\sum_{k\neq i,j}(W_{ij}+W_{ji})C_{ik}a_{ijk}+ \\
& \sum_{i}\sum_{j>i}\sum_{k\neq i,j}(W_{ij}+W_{ji})C_{kj}b_{ijk}+\sum_{i}\sum_{j>i}(W_{ij}+W_{ji})C_{ij}e_{ij}+ \\
& \sum_{k}F_kh_k+\sum_{k}\sum_{l>k}I_{kl}y_{kl} && (6.12)\\
s.t. \quad & \sum_{l\neq i}x_{ijil}+\sum_{l\neq i,j}a_{ijl}+e_{ij}=1, && \forall i,j>i, \ (6.13)\\
& \sum_{l\neq j}x_{ijlj}+\sum_{l\neq i,j}b_{ijl}+e_{ij}=1, && \forall i,j>i, \ (6.14)\\
& \sum_{l\neq k,i}x_{ijkl}+b_{ijk}=\sum_{l\neq k,j}x_{ijlk}+a_{ijk}, && \forall i,j>i,k\neq i,j, \ (6.15)\\
& x_{ijkl}+x_{ijlk}\leq y_{kl}, && \forall i,j>i,k,l>k, \ (6.16)\\
& \sum_{l\neq k}x_{kjkl}\leq h_k, && \forall j,k<j, \ (6.17)\\
& \sum_{k\neq l}x_{ilkl}\leq h_l, && \forall i,l>i, \ (6.18)\\
& e_{ij}\leq 2-(h_i+h_j), && \forall i,j>i, \ (6.19)\\
& a_{ijk}\leq 1-h_i, && \forall i,j>i,k\neq i,j, \ (6.20)\\
& b_{ijl}\leq 1-h_j, && \forall i,j>i,l\neq i,j, \ (6.21)\\
& a_{ijk}+\sum_{l\neq j,k}x_{ijlk}\leq h_k, && \forall i,j>i,k\neq i,j, \ (6.22)\\
& b_{ijk}+\sum_{l\neq k,i}x_{ijkl}\leq h_k, && \forall i,j>i,k\neq i,j, \ (6.23)\\
& e_{ij}+2x_{ijij}+\sum_{l\neq j,i}x_{ijil}+\sum_{l\neq i,j}x_{ijlj}\leq h_i+h_j, && \forall i,j>i, \ (6.24)\\
& x_{ijkl},a_{ijk},b_{ijk},e_{ij}\in(0,1). && (6.25)
\end{aligned}
$$

Proposition 6.2.1. *The binary sub-problem, BSP, and SP are equivalent, in the* *sense that their optimal values coincide. That is,* $Z^*_{BSP} = Z^*_{SP}$.

6.2.1 Master Problem

A critical drawback of the master problem is that it can easily result in a disconnected sub-graph at the hub-level network. This hub-level network does not work for our model since we assumed that a flow emanated from a hub node and destined to another hub node must be routed only through hub edges. Therfore, a disconnected hub-level structure as a solution to this master problem is infeasible for the HLPPT.

The number of such a disconnected hub-level networks is much higher compared to the connected ones. Therefore, the likehood of frequently observation of disconnected hub-level networks in the MP1 is higher than those of connected ones. Moreover, adding a cut corresponding to each of such hub-level networks to MP1 results in a very hard-to-solve MP1 (one may even encounter tens of them between two consecutive feasible configurations). This means that even in the initial iterations of Benders algorithm before reaching good lower and upper bounds, we are faced with a very complicated MP1. This master problem is populated by dozens of cuts corresponding to each visited unbounded SPD.

We have to add more constraints to avoid the possibility of disconnectivity. By doing that, we can only work with the feasible hub-level configurations and only the cuts corresponding to the extreme points are added to the model rather than those of extreme rays.

We have been trying to use either some modifications of the sub-tour elimination constraints and cut-set constraints which have been worked for the traveling salesman problem (TSP), or some ideas from the Steiner tree problem. However, none of these ideas have been efficient or even applicable.

By using a dummy node "0", the master problem is replaced by a non-simultaneous flow problem [54]. Here, the cost interpretation coincides with that of MP1. The flow source is the dummy node. By doing this, we can always have a connected sub-graph as the hub-level network.

Let $G(V, E)$ be a connected graph, where $V = \{1, 2, 3, \ldots, n\}$ is the set of nodes or vertices and E the set of edges. Let $G_d = (V, A)$ be a directed graph deriven from G, where $A = \{(i, j), (j, i) | \{i, j\} \in E\}$. That is, each edge e is associated with two arcs (i, j) and $(j, i) \in A$. Two new graphs $G^0 = (V_0, E_0)$ and $G^0_d = (V_0, A_0)$ where $V_0 = V \cup \{0\}$, $E_0 = E \cup \{\{0, j\} | j \in V\}$, $A_0 = A \cup \{(0, j), (j, 0) | j \in V\}$ are defined.

Let $h = (h_i)_{i \in V} \in \{0,1\}^{|V|}$, $y = (y_u)_{u \in E_0} \in \{0,1\}^{|E_0|}$ two $0-1$ vectors, and $z_{ij}^k \geq 0, (i,j) \in A_0$, $k \in V'$ where V' is a subset of V, and z_{ij}^k is a real flow in the arc $(i,j) \in A_0$ having 0 as source and k as destination. $E(i)$ is considered as the set of edges $e \in E$ such that an endpoint is i, $\Gamma^+(i) = \{j|(i,j) \in A_0\}$ and $\Gamma^-(i) = \{j|(j,i) \in A_0\}$, $m = |E|$ and $n = |V|$ [54].

(MP)

$$Min \quad \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl},$$

$$s.t. \quad \sum_{j \in \Gamma^+(0)} z_{0j}^k - h_k = 0, \qquad \forall k \in V, \quad (6.26)$$

$$\sum_{j \in \Gamma^+(i)} z_{ij}^k - \sum_{j \in \Gamma^-(i)} z_{ji}^k = 0, \qquad \forall i \in V - \{k\}, k \in V, \quad (6.27)$$

$$\sum_{j \in \Gamma^+(k)} z_{kj}^k - \sum_{j \in \Gamma^-(k)} z_{jk}^k + h_k = 0, \qquad \forall k \in V, \quad (6.28)$$

$$z_{ij}^k \leq y_{ij}, \qquad \forall \{i,j\} \in E_0, k \in V, \quad (6.29)$$

$$z_{ji}^k \leq y_{ij}, \qquad \forall \{i,j\} \in E_0, k \in V, \quad (6.30)$$

$$y_{ij} \leq h_i, \qquad \forall \{i,j\} \in E, \quad (6.31)$$

$$y_{ij} \leq h_j, \qquad \forall \{i,j\} \in E, \quad (6.32)$$

$$\sum_{j \in V} y_{0j} = 1, \qquad \forall i,j = 1, \ldots, n, \quad (6.33)$$

$$z_{ij}^k \geq 0, \qquad \forall (i,j) \in A_0, k \in V, \quad (6.34)$$

$$y_{ij} \in \{0,1\}, \{i,j\} \in E_0, h_k \in \{0,1\}, k \in V. \quad (6.35)$$

For a given $k \in V$, if the vertex h_k is selected to be in the sub-graph, one unit flow that is destined to it should be sent from the dummy node. Constraints (6.26) ensure this fact. Constraints (6.27) guarantee that the flow conservation holds for each vertex h_i as an interior node of the flow path from the dummy node to another node k. Constraints (6.28) state that a 1 unit of flow is the amount which is received by the node h_k and no flow sediments under any node along the flow path to this node. A flow can exist if the corresponding edge exists and an edge can exist if its corresponding end-points are chosen to be in the sub-graph. These are guaranteed by constraints (6.29)-(6.30) and (6.31)-(6.32), respectively. In addition, constraints (6.33) guarantiee that the whole 1 unit of flow emanated from the dummy node enters the sub-graph only via one node. The interpretation of the objective function is the same as for the MP1.

It should also be considered that the sub-graph associated with $h_i = 0$ for all $i \in V$ is also a connected graph. Therefore, in order to have a sub-graph with at least two nodes, it suffices to add one of the following two constraints (one may add both of (6.36) and (6.37) to achieve a stronger formulation):

$$\sum_{\{i,j\}\in E} y_{ij} \geq 1, \tag{6.36}$$

or

$$\sum_{i\in k} h_i \geq 2. \tag{6.37}$$

Theorem 6.2.2 ([54]). *All vectors h and y satisfying (6.26)-(6.35) and (6.36)(or (6.37)) are associated with connected sub-graphs of G.*

6.2.2 Sub-Problem

The sub-problem dual would be:

SPD:

$$\begin{aligned}
Max \quad & -\sum_{i,j>i}(u_{ij}+v_{ij}) - \sum_{i,j>i}\sum_{k\neq i,j}(s_{ijk}+w_{ijk})h_k - \sum_{j,k>j} p_{jk}h_k \\
& -\sum_{i,l>i} q_{il}h_l - \sum_{i,j>i} e_{ij}(2-h_i-h_j) - \sum_{i,j>i} d_{ij}(h_i+h_j) \\
& -\sum_{i,j>i}\sum_{k\neq i,j}(a_{ijk}(1-h_i)+b_{ijk}(1-h_j)) - \sum_{i,j>i}\sum_{k,l>k} o_{ijkl}y_{kl} \\
s.t. \quad & u_{ij}+v_{ij}+p_{ji}+q_{ij}+o_{ijij}+2d_{ij} \geq -\alpha*(W_{ij}+W_{ji})*C_{ij}, \quad \forall i,j>i, \\
& v_{ij}+r_{ijk}+w_{ijk}+q_{ij}+o_{ijkj}+d_{ij} \geq -\alpha*(W_{ij}+W_{ji})*C_{kj}, \quad \forall i,j>i,k\neq i,j, \\
& u_{ij}+p_{ji}+d_{ij}+s_{ijl}-r_{ijl}+o_{ijil} \geq -\alpha*(W_{ij}+W_{ji})*C_{il}, \quad \forall i,j>i,l\neq i,j, \\
& r_{ijk}-r_{ijl}+s_{ijl}+w_{ijk}+o_{ijkl} \geq -\alpha*(W_{ij}+W_{ji})*C_{kl}, \quad \forall i,j>i,k,l\neq i,j, \\
& u_{ij}-r_{ijk}+s_{ijk}+a_{ijk} \geq -(W_{ij}+W_{ji})*C_{ik}, \quad \forall i,j>i,k\neq i,j, \\
& v_{ij}+r_{ijk}+w_{ijk}+b_{ijk} \geq -(W_{ij}+W_{ji})*C_{kj}, \quad \forall i,j>i,k\neq i,j, \\
& u_{ij}+v_{ij}+d_{ij}+e_{ij} \geq -(W_{ij}+W_{ji})*C_{ij}, \quad \forall i,j>i, \\
& d_{ij},e_{ij},p_{ij},q_{ij},a_{ijk},b_{ijk},s_{ijk},w_{ijk},o_{ijkl} \in \mathbb{R}^+, \\
& u_{ij},v_{ij},r_{ijk} \text{ free in sign.}
\end{aligned}$$

Consequently, the generated cut for MP looks like the following inequality:

$$-\sum_{i,j>i}\Bigg((u_{ij}+v_{ij}) + \sum_{k\neq i,j}(s_{ijk}+w_{ijk})h_k + p_{ji}h_i + q_{ij}h_j + \sum_{k,l>k} o_{ijkl}y_{kl}$$

$$+ \sum_{k \neq i,j} (a_{ijk}(1-h_i) + b_{ijk}(1-h_j)) + d_{ij}(h_i + h_j) + e_{ij}(2 - h_i - h_j))\Bigg)$$
$$\leq \eta. \tag{6.38}$$

6.3 Speeding Up the Convergence

The classical Benders algorithm which uses the proposed MP and SPD (cf. to the second column of Table 6.1) is very time demanding even for solving a small instance of size 10. In order to speed up the convergence we add constraints (5.3), (5.10) and (5.11) which are shown to be improving inequalities for the MP. The computational results are reported in Table 6.1 and they show the computational effort for instances of size 5 and 10 before and after adding these constraints.

cf.

Table 6.1: Speeding up the convergence.

Instance	**Classical Benders Algorithm**	
	Before (sec.)	After (sec.)
AP5.2	0.50	0.28
AP10.4	738.23	41.31

Figure 6.1 visualizes the results of Table 6.1 for the instance of size 10 from the AP dataset. It is clear that the bounds of Benders algorithm after adding the improving constraints converge much earlier than the algorithm without them. Furthermore, the computational time for the small size instances of 10 is reduced to approximately $\frac{1}{18}$.

From now on, we keep these constraints always in our MP and whenever we talk about MP we refer to this new one with the better results.

6.4 Splitting the Sub-Problem: Strength of the Cuts

Sometimes even solving this LP model of SPD is not cheap at all or is not possible as the problem size grows, mainly due to the hardware limitation. It is our experience that even for moderate size problems (e.g. $N \simeq 50$) the instance grows out of proportions and it cannot be run on our computer.

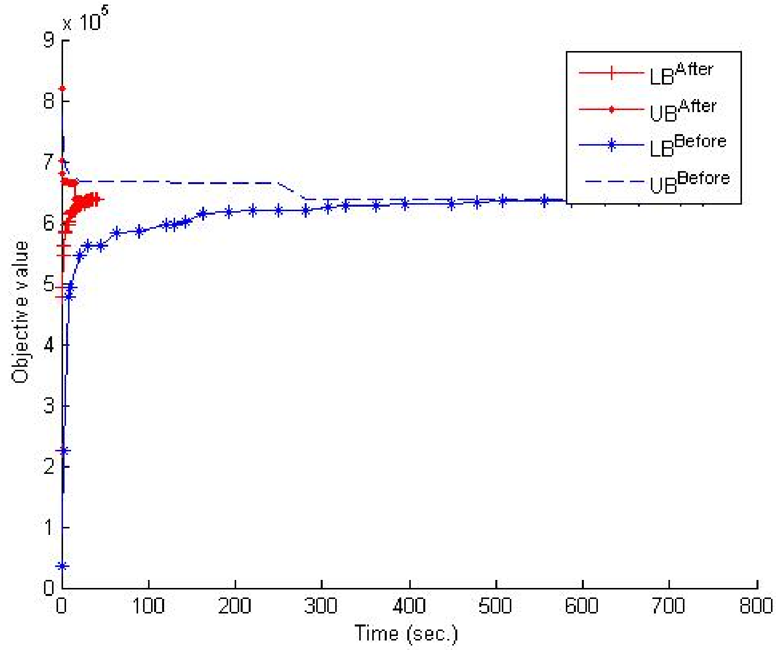


Figure 6.1: Speeding up the convergence.

It should be clear that SPD is composed of $\frac{n(n-1)}{2}$ independent LPs. Therefore, we separate parts of this SPD for each $i, j > i$ as follows:

SPD$_{i,j>i}$

$$
\begin{aligned}
Max \quad & -(u_{ij} + v_{ij}) - \sum_{k \neq i,j} (s_{ijk} + w_{ijk}) h_k - p_{ji} h_i - q_{ij} h_j \\
& -d_{ij}(h_i + h_j) - e_{ij}(2 - h_i - h_j) \\
& - \sum_{k \neq i,j} (a_{ijk}(1 - h_i) + b_{ijk}(1 - h_j)) - \sum_{k,l>k} o_{ijkl} y_{kl} \\
s.t. \quad & u_{ij} + v_{ij} + p_{ji} + q_{ij} + o_{ijij} + 2d_{ij} \geq -\alpha * (W_{ij} + W_{ji}) * C_{ij}, \\
& v_{ij} + r_{ijk} + w_{ijk} + q_{ij} + o_{ijkj} + d_{ij} \geq -\alpha * (W_{ij} + W_{ji}) * C_{kj}, \\
& u_{ij} + p_{ji} + d_{ij} + s_{ijl} - r_{ijl} + o_{ijil} \geq -\alpha * (W_{ij} + W_{ji}) * C_{il}, \\
& r_{ijk} - r_{ijl} + s_{ijl} + w_{ijk} + o_{ijkl} \geq -\alpha * (W_{ij} + W_{ji}) * C_{kl}, \\
& u_{ij} - r_{ijk} + s_{ijk} + a_{ijk} \geq -(W_{ij} + W_{ji}) * C_{ik}, \\
& v_{ij} + r_{ijk} + w_{ijk} + b_{ijk} \geq -(W_{ij} + W_{ji}) * C_{kj}, \\
& u_{ij} + v_{ij} + d_{ij} + e_{ij} \geq -(W_{ij} + W_{ji}) * C_{ij}, \\
& d_{ij}, e_{ij}, p_{ij}, q_{ij}, a_{ijk}, b_{ijk}, s_{ijk}, w_{ijk}, o_{ijkl} \in \mathbb{R}^+, \\
& u_{ij}, v_{ij}, r_{ijk} \text{ free in sign.}
\end{aligned}
$$

For each $i, j > i$, we have a simpler LP which can be solved much more efficiently using standard LP solvers.

Each of the generated cuts of the form (6.38) is also composed of $\frac{n(n-1)}{2}$ different independent terms. If we look at it more carefully, each of $ij|_{j>i}$-th term is generated from the corresponding $\text{SPD}_{i,j>i}$. That is, this constraint is an aggregation of $\frac{n(n-1)}{2}$ parts. For a given $i, j > i$, we can generate the following parts of the cut from SPD_{ij}:

$$
\begin{aligned}
&-\Bigg((u_{ij}+v_{ij}) + \sum_{k\neq i,j}(s_{ijk}+w_{ijk})h_k + p_{ji}h_i + q_{ij}h_j + \sum_{k,l>k} o_{ijkl}y_{kl} \\
&+ \sum_{k\neq i,j}(a_{ijk}(1-h_i) + b_{ijk}(1-h_j)) + d_{ij}(h_i+h_j) + e_{ij}(2-h_i-h_j))\Bigg) \leq \eta_{ij},
\end{aligned}
\tag{6.39}
$$

where $\eta = \sum_{ij>i}\eta_{ij}$ and $\eta_{ij} \geq 0, \forall i, j > i$.

Briefly speaking, instead of solving one large LP problem as a sub-problem and generating the corresponding cut, one can generate such a cut by solving $\frac{n(n-1)}{2}$ easier parts of SPD (see Algorithm 2). Such a (sub-)cut for the ij-th part of SPD, namely $\text{SPD}_{ij|_{j>i}}$, is in the form of (6.39).

The first splitting strategy we consider is used to generate the single cut by aggregation of $\frac{n(n-1)}{2}$ sub-cuts. If we use *Single Cut1 (SC1)* to call the traditional Benders algorithm, this approach will be called *Single Cut2 (SC2)*.

The sub-problem of HLPPT, namely SP, is a network problem and well-known due to its high degeneracy. This means that SPD has multiple optimal solutions. In fact, it turns out that, though solving the separated parts of SPD is much easier because of the separation but the aggregated cuts are much weaker than those of SC1.

According to our observation, even for small size problems, SC1 outperforms SC2. In the Figure 6.2, we show that for a problem instance of size $n = 10$, SC2 converges in a higher number of iterations compared to SC1 (367 iterations against 30). In Figure 6.2, the lower and upper bounds of SC1 converge earlier than those of SC2. Computational results are depicted in Table 6.2 at the end of next section.

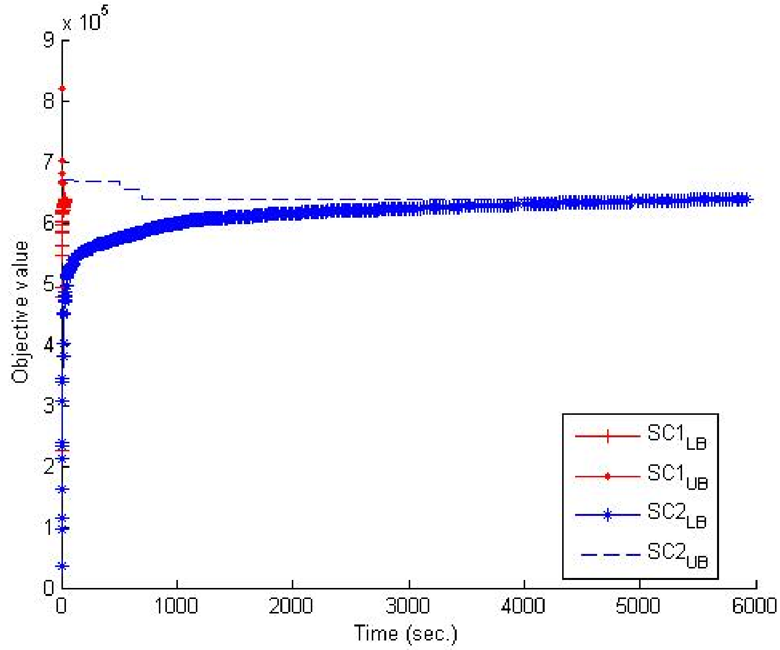


Figure 6.2: Convergence of SC1 vs. SC2.

6.5 Strengthening the Cuts

We now aim to improve our cuts to accelerate the convergence of our approaches. The idea originates from the technique initially proposed by Magnanti and Wong [55] for classical Benders algorithm.

Definition 6.5.1 (Dominating Cut)**.** Let π^1 and π^2 be two optimal solutions of the dual of sub-problem which are used to generate the cuts of the form $\pi^1(b - By) \leq \theta$ and $\pi^2(b - By) \leq \theta$. A cut generated from the extreme point π^1 dominates (is stronger than) a cut generated from the extreme point π^2, if:

$$\pi^1(b - By) \geq \pi^2(b - By), \tag{6.40}$$

for all $y \in Y$ with strict inequality for at least one point.

Definition 6.5.2 (Pareto-Optimal Cut)**.** A cut that is dominated by no other cut is a pareto-optimal cut.

Let Y^{LP} be the polyhedron defined by the linear relaxation of Y, and $ri(Y^{LP})$ be the relative interior of Y^{LP}. The following problem yields a Pareto-optimal cut for the general problem of (6.1)-(6.4):

$$\begin{aligned}
Max \quad & \pi(b - By^o) && (6.41)\\
s.t. \quad & uA \leq c, && (6.42)\\
& v(\overline{y}) = u(b - B\overline{y}), && (6.43)\\
& u \geq 0. && (6.44)
\end{aligned}$$

where $y^o \in ri(Y^{LP})$ and $v(\overline{y})$ is the optimal value of the sub-problem, when variables y are fixed to $\overline{y}$. The objective function maximizes the strength of the cut for y^o while the constraints define the feasible space as the optimal solutions of (6.5)-(6.7).

A relative interior point of the relaxed convex hull of relaxed variables h and y, $Co((h, y)^{LP})$ is chosen in such a way that just the constraints (6.31) and (6.32) hold and the variables take only factional values. As a rule of thumb, we would suggest to select the following point:

$$\begin{aligned}
h_k^o &= 0.5, && \forall k,\\
y_{kl}^o &= 0.5, && \forall k, l > k.
\end{aligned}$$

We applied this strategy to our cuts and the results changed significantly. Let these approaches be denoted by $SC1^d$ and $SC2^d$ for the case of using non-dominated cuts.

Firstly, we compare SC1 and $SC1^d$ in Figure 6.3.

As depicted in Figure 6.3 for a given instance of size 10, $SC1^d$ obviously outperforms SC1. Convergence is met much faster than before.

Figure 6.4 depicts the considerable difference between the performances of SC2 and $SC2^d$. The convergence of $SC2^d$ is 837 times faster than SC2.

Finally, the convergence behavior of $SC1^d$ and $SC2^d$ relative to each other is depicted in Figure 6.5.

By comparing Figure 6.2 and Figure 6.5, it is clear that the convergence times are significantly reduced after applying the pareto-optimal cut strategy. The difference between the convergence times are considerable. However, for larger instances $SC2^d$ outperforms $SC1^d$ as depicted in Table 6.2, although for $n = 10$, $SC1^d$ seems to work faster than $SC2^d$(see Figure 6.5). In the next subsection, we will report the computational results extensively.

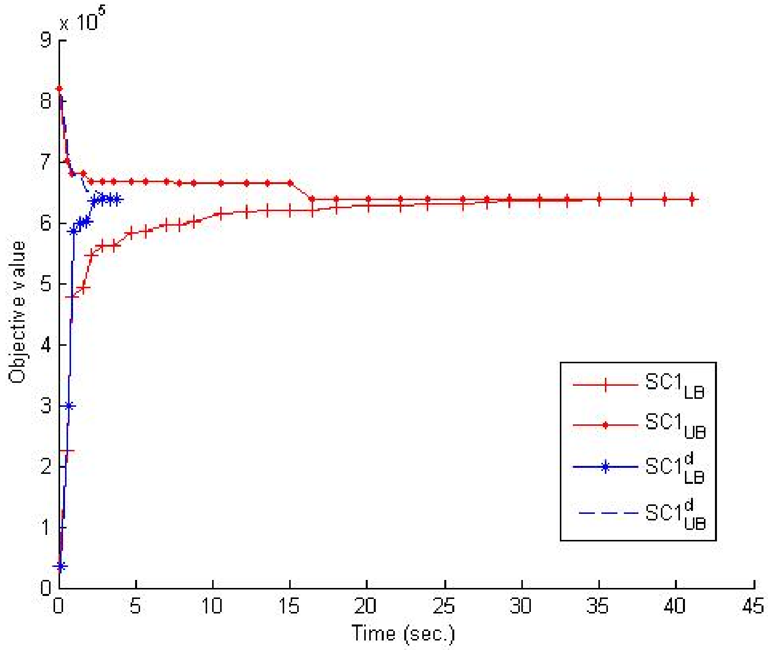


Figure 6.3: Convergence of SC1 vs. SC1^d.

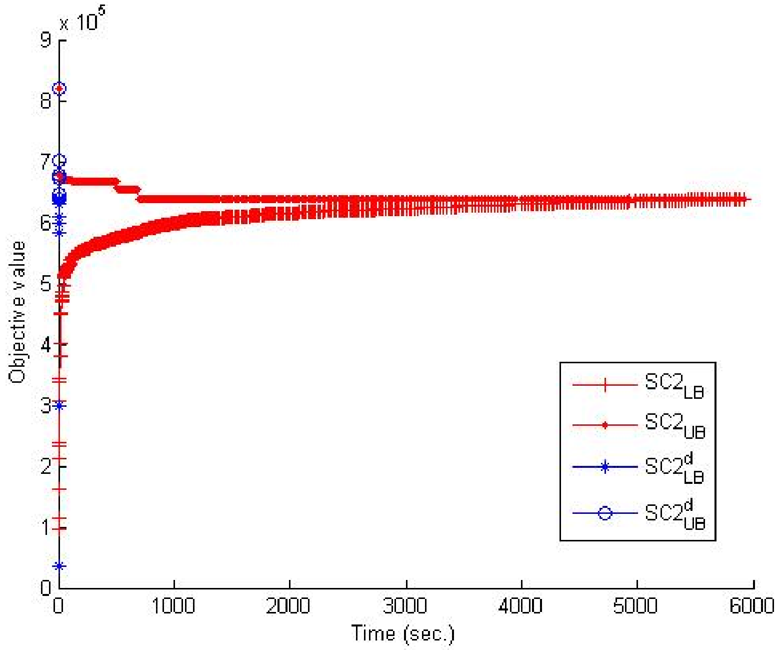


Figure 6.4: Convergence of SC2 vs. SC2^d.

6.5.1 Computational Results: Comparison Between SCs

Here, the computational results of all four algorithms are reported. As is depicted in Table 6.2, the approaches with pareto-optimal cuts (SC1^d and SC2^d), in general, are

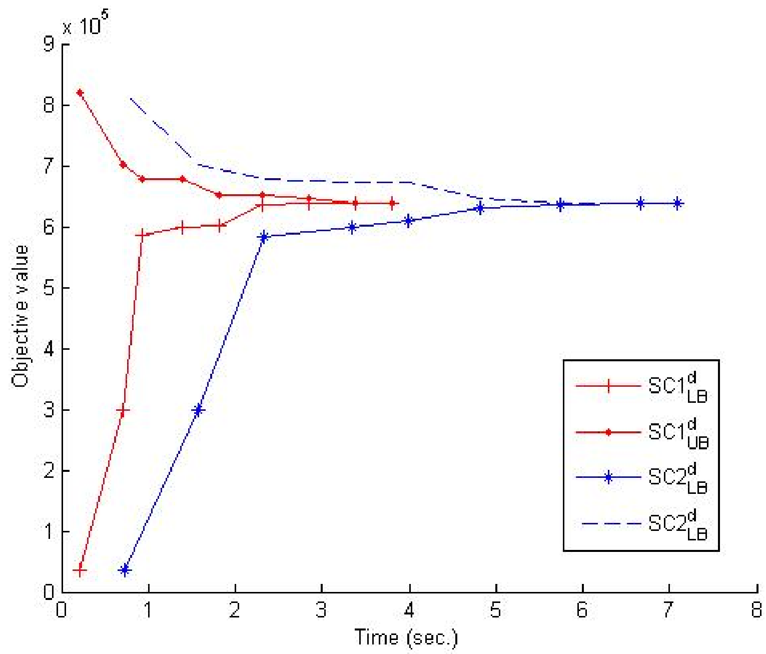


Figure 6.5: Convergence of SC1^d vs. SC2^d.

superior to the other ones. From now on, whenever we used the sign '≫' it means that the computational time of the instance was at least two time higher than the worst result in that row of the table and by '>' we mean that the it is just more than the best result in the row. For example, in the case of AP15.6, the SC2 could not even solve the instance in two times higer than what SC1 as the worst one does.

The superscripts in Table 6.2 indicate the number of iterations required until con-

Table 6.2: Comparison between SCs.

Instance	**SC1s** (sec.)		**SC2s** (sec.)	
	SC1	SC1^d	SC2	SC2^d
AP5.2	0.31$^{(8)}$	0.11$^{(4)}$	2.72$^{(16)}$	0.33$^{(4)}$
AP10.4	42.42$^{(30)}$	3.52$^{(8)}$	6529.39$^{(367)}$	6.02$^{(8)}$
AP15.6	14155.78$^{(97)}$	48.48$^{(8)}$	≫	47.50$^{(9)}$
AP20.8	> 1d	854.08$^{(14)}$	≫	820.83$^{(12)}$
AP25.10	≫	55902.82^{24}	≫	45801.05$^{(22)}$

vergence in the process of algorithm. One can say that the cuts generated from the pareto-optimal sub-cuts are strong enough compared to the instantaneously generated single cuts, especially if there be a trade-off between strength of cuts, efficiency and solvability of SPD regarding the hardware restriction. As the problem size increases $SC2^d$ outperforms $SC1^d$. Howwever, so far the Benders algorithm did not perform as we expected.

6.6 Multiple Cuts vs. Single Cut

,

In the classical Benders decomposition approach, at each iteration a single cut is added to the MP. It is well-known that the computational efficiency of the algorithm strongly depends on the following issues: (i) the number of iterations to reach a global convergence, (ii) the time needed to solve SP or SPD in each iteration, and (iii) the time and computational efforts to solve the MP [74].

Having more than one cut in each iteration (if they are not dominated by each other) can cut off more inferior parts of the feasible space not containing the optimal solution of the MP. Therefore, it results in a better approximation of the feasible set containing an optimal solution of the original problem. However, adding more cuts can make the MP dramatically difficult to solve, as iterations proceeds.

Due to the special structure of our SPD, each separated part of the SPD can be solved easier than solving the whole SPD once. Also, more than one cut (sub-cuts) per each iteration can be generated. Moreover, this set of cuts is stronger than a single cut as a linear combinations of them if the appropriate values of the SDP optimal solutions are selected.

As mentioned earlier, the sub-problem is separable to $\frac{n(n-1)}{2}$ problems. Therefore, we can have two scenarios: the first one, which we will call *Multi Cut1 (MC1)*, aggregates $n-i$ cuts for each i into a single aggregated cut. As a result, $n-1$ cuts per iteration are obtained. The second one, *Multi Cut2 (MC2)*, uses $\frac{n(n-1)}{2}$ cuts per each iteration. The computational experiences reported in Table 6.3 substantiate the superiority of MC2 to MC1.

The pareto-optimal cut strategy was again applied to MC1 and MC2.

6.6.1 Computational Results: Comparison Between MCs

Computational results of two different multi-cut approaches are reported in Table 6.3.

Table 6.3: Comparison between MCs.

Instance	MC1s(sec.)		MC2s(sec.)	
	MC1	$MC1^d$	MC2	$MC2^d$
AP5.2	$0.34^{(5)}$	$0.25^{(3)}$	$0.22^{(4)}$	$0.25^{(4)}$
AP10.4	$9.61^{(11)}$	$4.30^{(6)}$	$5.47^{(7)}$	$3.77^{(5)}$
AP15.6	$97.30^{(10)}$	$16.11^{(4)}$	$21.34^{(5)}$	$12.00^{(4)}$
AP20.8	$933.88^{(11)}$	$240.17^{(8)}$	$223.05^{(8)}$	$153.61^{(6)}$
AP25.10	$26332.39^{(15)}$	$3351.60^{(9)}$	$2672.11^{(7)}$	$1855.69^{(6)}$
AP30.12	$\gg$	$8209.14^{(9)}$	$9083.13^{(10)}$	$3350.86^{(6)}$
AP35.14	$\gg$	$\gg$	$\gg$	$120206.82^{(7)\,a}$
AP40.16	$\gg$	$\gg$	$\gg$	> 6 d

[a] $\simeq$ 33 hours.

As one can see, the pareto-optimal cut approach works much better with respect to the computational time. Yet, $MC2^d$ is absolutely superior to $MC1^d$ with respect to the efficiency of the resolution of the problem. The number of iterations after which optimality is reached are considerably reduced and computational times are significantly less. Still, within a given time limit of 48 hours, $MC2^d$ can solve larger instances.

6.7 Numerical Results

In this section we report the results of `CPLEX 9.1`, the best of single cut and the best of multiple cut Benders algorithms of our study, so far. According to the results of Table 6.2 and Table 6.3 they are compared to the results of `CPLEX 9.1` (see Table 3.4) .

Table 6.4 reports our computational experiences. The standard solver of `CPLEX 9.1` is only able to solve problem instances for up to 30 nodes. Actually, problems of size $n \geq 35$ could not even fit in memory.

Clearly, $MC2^d$ performs absolutely better than $SC2^d$ in terms of computational time and is better than both `CPLEX 9.1` and $SC2^d$ with respect to the instance size that can be solved within a given time limit.

Table 6.4: Overall Comparison.

Instance	CPLEX 9.1	SC2^d	MC2^d
AP5.2	0.03	$0.33^{(4)}$	$0.25^{(4)}$
AP10.4	0.09	$6.02^{(8)}$	$3.77^{(5)}$
AP15.6	2.3	$47.50^{(9)}$	$12.00^{(4)}$
AP20.8	62.02	$820.83^{(12)}$	$153.61^{(6)}$
AP25.10	900.50	$45801.05^{(22)}$	$1855.69^{(6)}$
AP30.12	5530.00	$\gg$	$3350.86^{(6)}$
AP35.14	N.A.	$\gg$	$120206.82^{(7)}$
AP40.16	N.A.	$\gg$	$>$ 6 d

6.8 Accelerating Benders Approaches by Fractional Cuts

We have shown in Chapter 5 that the model can be tightened so that even some instances can be solved to integer optimality by solving them as an LP. Therefore, we can use our Benders algorithm to solve the LP relaxation of our HLPPT, although this problem is slightly different.

Once the LP solutions are found, first of all are useful to perform as lower bounds to measure the quality of solutions of heuristics. Secondly, we can switch to an MIP master problem from a sufficiently small gap at some point quite close to the optimal solution of HLPPT.

In each iteration of Benders resolution of the LP relaxation, fractional values of MP variables are used to generate cuts (we will call them *fractional cuts*) from the solution to the SPD. These cuts are added to the relaxed MP, one per iteration. This approach is very efficient because it limits the frequency of solving an MIP master problem during the Benders algorithm process. For most of our instances, an MIP has been solved only a few times (even less than 3) during the whole solution procedure.

Again, as a rule of thumb, we suggest to select the following relative interior point:

$$
\begin{aligned}
h_k^o &= 0.9, && \forall k, \\
y_{kl}^o &= 0.1, && \forall k, l > k.
\end{aligned}
$$

The new master problem that we will call it *Acceleration Master Problem* (AMP) is depicted in the sequel:

(AMP)

$$
\begin{aligned}
Min \quad & \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}, \\
s.t. \quad & (6.26), (6.27), (6.28), (6.29), (6.30) \\
& (6.31), (6.32), (6.33), (6.34), \\
& \sum_k y_{k0} = 1, && (6.45) \\
& \sum_{k,l>k} y_{kl} \geq 1, && (6.46) \\
& \sum_k h_k \geq 2, && (6.47) \\
& y_{ij} \in (0,1), \{i,j\} \in E_0, h_k \in (0,1), k \in V. && (6.48)
\end{aligned}
$$

We call this approach AMC2^d and computational results are depicted in Table 6.5.

Table 6.5: Computational Results.

Instance	**CPLEX 9.1**(sec.)	**SC2**^{d}(sec.)	**MC2**^{d}(sec.)	**AMC2**^{d}(sec.)
AP5.2	0.03	$0.33^{(4)}$	$0.25^{(4)}$	$0.53^{(6)}$
AP10.4	0.09	$6.02^{(8)}$	$3.77^{(5)}$	$3.30^{(6)}$
AP15.6	2.3	$47.50^{(9)}$	$12.00^{(4)}$	$12.19^{(7)}$
AP20.8	62.02	$820.83^{(12)}$	$153.61^{(6)}$	$56.47^{(8)}$
AP25.10	900.50	$45801.05^{(22)}$	$1855.69^{(6)}$	$172.89^{(8)}$
AP30.12	5530.00	$\gg$	$3350.86^{(6)}$	$675.09^{(13)}$
AP35.14	N.A.	$\gg$	$120206.82^{(7)}$	$4238.14^{(12)}$
AP40.16	N.A.	$\gg$	$>$ 6 d	$25676.55^{(14)}$
AP45.18	N.A.	$\gg$	$\gg$	$87130.57^{(15)}$

A trivial question that might arise here is whether it is necessary to solve the relaxed problems to optimality. Here, we propose two approaches. The first one (which is referred to by AMC2^d in Table 6.5) solves that LP to optimality before un-relaxing the AMP and the second one to a gap of less than 0.5% between lower and upper bounds of Benders algorithm. The first one is be referred by OptLP and the latter by nonOptLP. The results are reported in Table 6.6.

As depicted in Table 6.6, in general, Benders approaches are capable of solving larger

Table 6.6: Final Comparison.

Instance	**CPLEX 9.1** (sec.)	**SC2^d** (sec.)	**MC2^d** (sec.)	**AMC2^d**(sec.) OptLP	nonOptLP
AP5.2	0.03	0.33$^{(4)}$	0.25$^{(4)}$	0.53$^{(6)}$	0.52$^{(6)}$
AP10.4	0.09	6.02$^{(8)}$	3.77$^{(5)}$	3.30$^{(6)}$	4.23$^{(6)}$
AP15.6	2.3	47.50$^{(9)}$	12.00$^{(4)}$	12.19$^{(7)}$	13.09$^{(6)}$
AP20.8	62.02	820.83$^{(12)}$	153.61$^{(6)}$	56.47$^{(8)}$	40.34$^{(7)}$
AP25.10	900.50	45801.05$^{(22)}$	1855.69$^{(6)}$	172.89$^{(8)}$	134.19$^{(7)}$
AP30.12	5530.00	$\gg$	3350.86$^{(6)}$	675.09$^{(13)}$	534.56$^{(8)}$
AP35.14	N.A.	$\gg$	120206.82$^{(7)}$	4238.14$^{(12)}$	2771.99$^{(8)}$
AP40.16	N.A.	$\gg$	$>$ 6 d	25676.55$^{(14)}$	14181.73$^{(8)}$
AP45.18	N.A.	$\gg$	$\gg$	87130.57$^{(15)}$	99483.25$^{(10)}$
AP50.20	N.A.	$\gg$	$\gg$	566360.33$^{(14)}$	528663.42$^{(10)\,a}$

^{a} $\simeq$ 146.85 hrs (approximately 6 days).

instances while `CPLEX 9.1` fails. Yet, multiple cut approaches (MC2^d and AMC2^{d}s) and especially the accelerated multiple cut schemes, AMC2^{d}s, are capable of solving much larger instances. With respect to the computational time, obviously MC2^d and AMC2^{d}s are superior. For some instances like AP30.12, AMC2^d, in the second variant, solves the instance more than 10 times faster than `CPLEX 9.1`.

Among the multiple cut approaches, with respect to the problem instance size which is solved, AMC2^{d}s outperform MC2^d. While it takes more than 6 days to solve AP40.16 with MC2^d, the larger instance of AP50.20 can be solved by AMC2^d (the second variant) in such an amount of time (approximately 146 hrs). However, the absolute superiority of AMC2^{d}s to other methods both in terms of computational time and the instance size that can be solved is obvious in Table 6.6. Between AMC2^{d}s, as Table 6.6 shows, the second variant (namely, nonOptLP) is superior, as the instance size grows. For some instances, its computational time is even half of the one of OptLP.

Similar conslusion and results are also obtained from extensive expriments with many randomly generated instances.

6.9 Algorithms

The implementation of all the proposed Benders algorithms are presented here.

6.9.1 Classical Single Cut Benders Algorithm (SC1)

The algorithm SC1 is the classical Benders algorithm which uses only one cut per iteration. This cut is generated as a result of solving one complete SPD.

Algorithm 1: SC1

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    Solve SPD;
    Add corresponding cut to the MP;
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

The algorithm SC1 equipped with the pareto-optimal cut strategy is depicted below.

Algorithm 2: SC1^d

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    Solve SPD;
    Find the pareto-optimal cut according to (6.41)-(6.44);
    Add corresponding cut to the MP;
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

6.9.2 Modified Classical Single Cut Benders Algorithm (SC2)

Although in this algorithm a single cut is added in each iteration, this cut is an aggregation of sub-cuts achieved from $\frac{n(n-1)}{2}$ separated parts of the problem (see Algorithm 3).

Algorithm 3: SC2

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach $i, j > i$ **do**
        Solve SPD_{ij};
        Add ij-th part of the single cut;
    end
    Add aggregated corresponding cut to the MP;
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

SC2 equipped with the pareto-optimal cut is depicted in Algorithm 4.

6.9.3 Multi-Cut Benders Algorithm (MC1)

In MC1, $\frac{n(n-1)}{2}$ LPs are solved. For a given i, sub-cuts of all $j_{j>i}$-th sub-SPDs are aggregated. As a result $n-1$ cuts are added to MP at each iteration (see Algorithm 5).

MC1 equipped with the pareto-optimal cut strategy is depicted in Algorithm 6.

6.9.4 Modified Multi-Cut Benders Algorithm (MC2)

In $MC2$, at each iteration, $\frac{n(n-1)}{2}$ such cuts are added to MP (see Algorithm 7).

MC2 equipped with the pareto-optimal cut strategy is depicted in Algorithm 8.

Algorithm 4: SC2^d

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach $i, j > i$ **do**
        Solve SPD_{ij};
        Find the pareto-optimal cut according to (6.41)-(6.44);
        Add ij-th part of the single cut;
    end
    Add aggregated corresponding cut to the MP;
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

Algorithm 5: MC1

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach i **do**
        foreach $j > i$ **do**
            Solve SPD_{ij};
            Add j-th part of i-th cut;
        end
        Add corresponding cut (i-th) to the MP;
    end
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

Algorithm 6: MC1^d

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach i **do**
        foreach $j > i$ **do**
            Solve SPD_{ij};
            Find the pareto-optimal cut according to (6.41)-(6.44);
            Add j-th part of i-th cut;
        end
        Add corresponding cut(i-th) to the MP;
    end
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

Algorithm 7: MC2

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach $i, j > i$ **do**
        Solve SPD_{ij};
        Add corresponding cut to the MP;
    end
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
stop. The optimal solution of original problem is obtained;

Algorithm 8: MC2^d

Input: Problem instance
Output: Optimal solution of the original problem
Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
  Solve MP to optimality;
  Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
  foreach $i, j > i$ **do**
    Solve SPD_{ij};
    Find the pareto-optimal cut according to (6.41)-(6.44);
    Add corresponding cut to the MP;
  end
  if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
    Set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
  end
end
stop. The optimal solution of original problem is obtained;

6.9.5 Accelerated Multi-Cut Benders Algorithm (AMC2^d)

The algorithm AMC2^d that reduces the frequency of the MIP master resolution is given by Algorithm 9.

Algorithm 9: AMC2^d

Input: Problem instance
Output: Optimal solution of the original problem
1:Set $UB = +\infty$, $LB = 0$;
while $LB \neq UB$ **do**
    Solve a relaxed MP to optimality;
    Set $LB = z^*_{MP}$ and update h_k and y_{kl} in a new dual problem;
    foreach $i, j > i$ **do**
        Solve SPD_{ij};
        Find the pareto-optimal cut according to (6.41)-(6.44);
        Add corresponding cut to the MP;
    end
    if $z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl} < UB$ **then**
        set $UB = z^*_{SP} + \sum_k F_k h_k + \sum_k \sum_{l>k} I_{kl} y_{kl}$;
    end
end
if *MP is an LP* **then**
    Un-relax the MP;
    go to 1;
end
stop. The optimal solution of original problem is obtained;

Chapter 7

Heuristic Solution Methods for HLPPT

In the preceding chapter we explained that some restrictions prevent us from solving even moderate size instances of HLPPT in a reasonable amount of time. In fact, our limitation is not only the time, but hardware restrictions are an obstacle as well. A possibility is sacrificing the optimality in favor of finding a good solution for each problem instance. Moreover, these solutions need to be a good approximations of the optimum and also need to be achieved in a reasonable amount of time.

7.1 Heuristic Algorithms

A *heuristic* is an algorithm designed and developed to achieve one or both of following goals:

i) finding a pretty good solution,

ii) in a reasonable amount of time.

Of course, there is no proof the solutions could not get arbitrarily bad; or it usually runs reasonably quickly, however there is no argument that this will always be the case.

7.1.1 Metaheuristic Algorithm

A *metaheuristic* is a heuristic method for solving a very general class of computational problems by combining user-given-black-box procedures that are usually again heuristics. Where no problem-specific algorithm or heuristic for a problem exists or when it exists and it is not practically implementable, a metaheuristic can be applied. Combinatorial Optimization Problems (COPs) are among those type of problems. Some of the most well-known metaheuristics are:

- Random search,
- Local search,
- Greedy algorithms and hill climbing,
- Best-first search,
- Genetic algorithms,

- Simulated annealing,
- Tabu search,
- Ant colony optimization, and,
- Greedy randomized adaptive search procedure (GRASP).

In the case of HLPPT, one looks for a metaheuristic algorithm capable of finding good solutions for as many instances as possible in a reasonable amount of time.

What we expect from our metaheuristic is that:

- it is capable of finding optimal solutions for almost all (or at least most) of the problem instances with known optimal solutions,
- if it failed to find the optimal solution, the reported gap should be quite small.

It is assumed that such a metaheuristic that can satisfy these conditions is trustable. In other words, we speculate that the solutions found by this algorithm for all other problem instances where their optima are not known, are close enough to optimality.

7.2 Greedy Algorithms

A greedy algorithm is a metaheuristic algorithm for solving a problem by finding a local optimum in each stage of algorithm, hoping to find the global optimum, eventually.

7.2.1 Strategy and Elements

In a more detailed form, a *set of candidates* which a solution is created out of that is subjected to a *selection function* in order to choose the best of candidates. A *feasibility function* checks whether the selected candidate can contribute to the current solution. Moreover, the quality of this partial solution is evaluated by means of an *objective function*.

7.2.2 Greedy Choice Priority

The choice of a candidate that contributes to the current partial solution can be accomplished through a prioritization process. The prioritization can either be applied at the very beginning of algorithm and remain the same until the end of the process or it can be updated dynamically during the stages of the algorithm.

7.2.3 Well-Known Greedy Algorithms

The greedy algorithms belong to an important class of algorithms in computer science and metaheuristic optimization methods. Examples of these algorithms which are proven to be able to find the global optimum and are much faster than other optimization methods are:

- *Kruskal*'s algorithm and *Prim*'s algorithm for finding minimum spanning trees,
- *Dijkstra*'s algorithm for finding single-source shortest paths and,
- the algorithm for finding optimum *Huffman trees*.

7.3 Greedy Algorithm for HLPPT

As mentioned earlier, our problem can also be restated as a problem of finding a connected hub-level network followed by a minimum flow cost problem. Obviously, the second part is a function of the first part. This means, how the flow should be transferred is induced by the hub-level network configuration. Therefore, without loss of generality we concentrate on the search for the best (or as good as possible) hub-level network. Since we have seen that the well-known greedy algorithms (see Subsection 7.2.3) already performed extremely good on some of the graph problems, we are motivated to employ a greedy approach for the case of HLPPT.

Now, we translate our problem onto the necessary components of a greedy algorithm:

- set of all edges as the *set of candidates*,
- $\Delta = f^{new} - f^{cur}$ as the *selection function*,
- a functionality for checking the connectivity, to act as a *feasibility function*,
- and the objective function of HLPPT (to count the total cost: the hub-level network setup cost plus the flow cost) as the *objective function*.

Therefore, these four fundamental components of a greedy algorithm are already introduced for our problem.

Before closing this section, it may be worth to remind how we evaluate the objective function. We use Dijkstra's single-source shortest path algorithm which itself is a greedy algorithm to find the shortest paths and evaluate the flow transfer cost (this algorithm is efficiently implemented in the `LEDA`[1] library). Finally, the objective

[1] Algorithmic Solutions Software GmbH

function is evaluated (as mentioned earlier).

7.3.1 Strategy

In this section a basic version of our heuristic is described. For each iteration of the external loop (while improvement proceeds), the best edge from among all edges is chosen. Then the search moves to that neighbor and restarts from there (the best move is a move with most negative Δf). As long as such a move exists the algorithm proceeds.

Definition 7.3.1 (Hamming Distance)**.** The Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. In other words, it counts the number of substitutions required to change one into the other.

Definition 7.3.2 (Edge Vector)**.** An edge vector $\mathbf{a}$, is an array of $\frac{n(n-1)}{2}$ binary values, where $\mathbf{a}_i = 1$ if the edge corresponding to i receives a hub edge and 0, otherwise.

Example 7.3.1. Let $a = (1, 1, 0, 1, 1)$ and $b = (1, 0, 0, 1, 0)$. The Hamming distance of a and b, $d_H(a, b) = \sum_i(|a_i - b_i|) = 2$.

Taking a closer look at our greedy algorithm, one can see that it is actually a Hill Climbing algorithm on the neighborhood induced by the Hamming metric on the set of *edge vectors*. This algorithm iteratively looks for the best neighbor with a distance of 1.

In this algorithm we merge the *feasibility function* and the *objective function* and let the function *Eval* to return ∞ if the resulted trial point is infeasible and the objective function value in any other case. That is, in order to improve the performance of algorithm, our concern would not be to move from feasible solution to other feasible ones and examining them to find the best. Instead, it is the objective function that controls whether it is feasible or not, and thus returns the corresponding value.

Note. By flattening out a 2D edge array (like y) and taking into account the undirectedness of the hub-level graph, the hub edge (k, l)(i.e. y_{kl}) corresponds to the $(k \times n - k \times (k-1)/2 + l - k - 1)$-th entry of a the linear edge vector for its implementation in C++. From now on, we will always refer to this 1D vector as the edge vector.

Although, the size of this neighborhood is $\frac{n \times (n-1)}{2}$, but not all of them can result in a connected hub-level network.

Theorem 7.3.1 (Cardinality of Feasible Neighbors Set). *The number of feasible neighbors, $NrNhbrs$, in the neighborhood (induced by Hamming metric) of a given hub structure is $NrNhbrs \leq \frac{n(n-1)}{2} - \frac{(n-NrH)(n-NrH-1)}{2}$, where n is the number of locations and NrH is the number of hubs in the current solution.*

Proof. For a given hub-level configuration (solid edges in Figure 7.1), none of the edges in the complete graph of spoke-level network can be added to the hub-level network and result in a connected hub-level graph. One can observe in the Figure 7.1 that changing the status of none of the dotted edges in the complete spoke-level network can result in a new connected hub-level network.

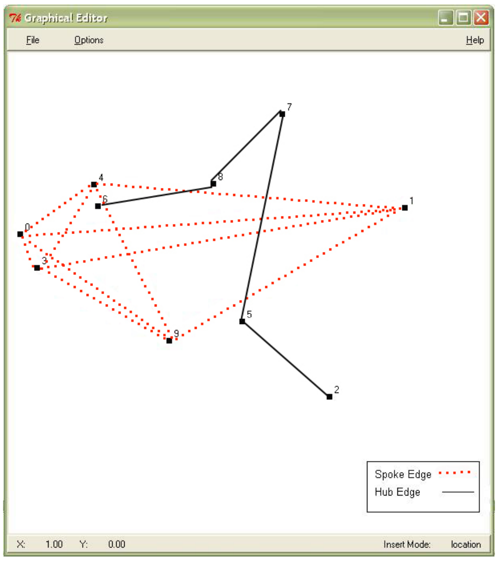


Figure 7.1: Theorem 7.3.1

These dotted edges will remain spoke in any neighbor with distance of one from this configuration. Because, either a spoke node changes to a hub node and a new hub edge will be established between the new hub node and current hub-level network or this change happens inside the hub-level network. Thus, in both case nothing is to be done with the edges between the spokes. Therefore $\frac{(n-NrH)(n-NrH-1)}{2}$ choices are canceled from among our choices, which completes the proof. □

Obviously, $\frac{n(n-1)}{2} - \frac{(n-NrH)(n-NrH-1)}{2}$ is not a tight upper bound because it might happen that the change is the removal of a hub edge and the edge to be removed (like 5-7 or 7-8) is chosen in such a way that its cancelation results in a hub-level graph with two components. Another case may happen if the change is the installation of a hub edge and both end-points are selected within the current hub-level network (like 5-8).

The greedy algorithm is depicted in Algorithm 10.

Algorithm 10: A simple greedy algorithm for HLPPT

```
Input: HLPPT instance
Output: x*
x̄:=Create_initial_solution();
min := Eval(x̄);
last_min := ∞;
repeated_min := 0;
while (repeated_min = 0) do
    f̄ := Eval(x̄);
    if f̄ ≤ min then
        min := f̄;
        x* := x̄;
    end
    foreach i = 1  to  nrLocations * (nrLocations − 1)/2 do
        Δf := 0;
        x' := x̄;
        x'_i := 1 − x'_i;
        if  is_not_feasible(x') then
            Δf := ∞;
        else
            Δf := Eval(x') − min;
        end
        if Δf < 0 then
            x* := x' ;
            min := Eval(x');
        end

    end
    if min = last_min then
        repeated_min := repeated_min + 1;
    end
    last_min := min;
    x̄ = x*;
end
stop.
```

7.3.2 Initial Solution

As we can see in Algorithm 10, an initial solution is generated by our algorithm to work as a starting point. Our experiments have revealed that starting with a random initial solution is not the best idea. Actually, this is the nature of COPs that have many local optima working as whirlpools to capture the search into one of them. Therefore, it is worthwhile to have more prudent strategies for creating an initial solution.

From our experiences in instances of HLPPT, we observed that:

- the number of hubs in the optimal solution is an unknown function of the discount factor. That is, the number of hubs has a direct relationship with the discount considered for using hub edges; The higher the discount, the higher the tendency to have more hub edges and subsequently hub nodes,
- it is more likely for the most center oriented and busiest (in terms of the total flow arriving to and departing from) locations to receive hub (according to our experience, there was at least one hub node in a set composed of the $(n \times 0.2)$ most central nodes in union with the $(n \times 0.2)$ busiest nodes)

For example, if $\alpha = 0.5$, we select $max(n \times 0.2, 2)$ number of the most central nodes and $max\{n \times 0.2, 2\}$ number of busiest locations as initial hubs. Preferably, the hub level network should be a complete graph of these selected locations. This initial solution is passed to the main procedure of algorithm.

7.3.3 Complexity and Neighborhood Size

In this subsection we discuss the complexity of the algorithms and the size of the neighborhood.

Since the hub-level network is an undirected graph, we have $\frac{n\times(n-1)}{2}$ possible hub edges. Two configurations are assumed to be neighbors if their Hamming distance is equal to 1. As a result the cardinality of the set of neighbors of a given feasible configuration is in general $\frac{n\times(n-1)}{2}$, i.e. $O(n^2)$. As we have seen in Theorem 7.3.1, the actual size is strictly less than this, except if removing or adding any hub edge does not harm the connectivity. Therefore, the size of the neighborhood in the worst case is $\frac{n(n-1)}{2}$, that is $O(n^2)$.

Complexity of Algorithm

At each iteration of the external loop, the internal loop looks for the best feasible move from among a maximum of $\frac{n\times(n-1)}{2}$ moves. Therefore, in each iteration, at most one move can take place. The overall iterations of the external loop is not known in anticipation. However, from our experience, it is much less than neighborhood size. Therefore we cannot say that the complexity is $O(n^2)$, as we cannot say it is $O(n)$, either. On the other hand, for each *feasible* neighbor (a feasible neighbor is a neighbor with a connected hub-level graph) and for each pair of origin-destination $i - j_{(j>i)}$ Dijkstra's shortest path algorithm is applied. We use $j > i$ because the shortest paths are symmetric. The complexity of each Dijkstra's algorithm is $O(|E| + |V|log|V|)$, where $|E| \leq q + p(n-p) \leq \frac{n\times(n-1)}{2}$ and $|V| = n$, where q is the number of hub edges and p is the number of hub nodes in the feasible neighbor. This complexity should be considered for $n-1$ nodes and as a result is of $O(|V|(|E| + |V|log|V|)) \leq O((n-1)(\frac{n\times(n-1)}{2}) + n\log n) = O(n^3)$.

7.3.4 Computational Results

Benders algorithm successfully found the optimal solution of HLPPT instances in a smaller amount of time compared to `CPLEX 9.1`, and it could also solve larger size instances within a specific time limit. Table 7.1 reports the computational time for Benders algorithm and that of our greedy heuristic.

As Table 7.1 shows, insofar as the optimal solution for HLPPT instances (i.e. sizes of 5...50) are known, our heuristic except for one case either reached to the optimal solutions or for a few cases to gaps of less than 0.01%.

Table 7.1: Comparison between the greedy algorithm and AMC2^d.

Instance	AMC2^d (sec.)	Greedy Algorithm (sec.)	Gap (%)
AP5.2	0.52	0.00	0_{opt}
AP10.4	4.23	0.01	0_{opt}
AP15.6	13.09	0.08	0_{opt}
AP20.8	40.34	0.44	0_{opt}
AP25.10	134.19	1.33	0_{opt}
AP30.12	534.56	4.64	0_{opt}
AP35.14	2771.99	7.91	0.01
AP40.16	14181.73	20.72	0.01
AP45.18	99483.25	31.89	0.01
AP50.20	528663.42	135.99	2.67

Furthermore, the heuristic could find them in a fraction of the CPU time that Benders algorithm would require.

7.4 Improvement by Local Neighborhood Search

Due to the metaheuristic nature of the method and the myopic characteristics of greedy algorithms, the possibility always exists that the search process gets stuck in a local optimum. This happens for example in the case of the problems of AP35.14 until AP50.20 for which the optimal results are known. This always makes it worth to employ a prudent diversification or intensification of the search, while hoping to reach to a new and better solution.

In order to obtain better solutions given their existence, we compare and explore the existing facts and evidences for both the available optimal solution and best solution of the greedy algorithm. Based on that, we develop some strategies capable of obtaining better solutions.

Our first observations out of visualizing and comparing the optimal solution with that of a greedy heuristic for many problem instances including AP, CAB and also random instances, revealed that the output of the greedy algorithm (Algorithm 10), almost always has exactly the same number of hubs which appear in the optimal solutions.

The second observation states that the spatial layout of the hub-level network in the best solution of the basic greedy heuristic has only a very slight deviation from that of the optimal solution. In other words, the Hamming distance of the optimal hub vector and that of the greedy algorithm output is quite small (as a rule of thumb it is less than the number of common hubs between them $d_H(H^{opt}, H^{greedy}) < \sum_{i=1}^{n}(1 - |h_i^{opt} - h_i^{greedy}|)$.

The third observation points out the fact that spoke nodes having the highest flow transition and at the same time are closer to the current hubs, are more susceptible to be replaced by one in the current solution and thus lead to a better solution (after becoming subjected to a re-started neighborhood search).

Therefore, we will define more neighborhood structures in order to virtually speaking "clashing" to each other. Hopefully this can detect new crevices and narrows containing better solutions than the previously faded crevices as a result of these clashes and establishes a back-and-forth process. In this way, we expect achieving an even better solution or perhaps a global optimum.

7.4.1 Neighborhood Search I

As it is reported in Table 7.1, for the case of AP35.14 we did not reach optimality. Although, the gap was quite small, it still exists. In order to study the ways to escape from the local optimum, the optimal hub-level network is visualized in the `LOLA Graph`[2] software.

Definition 7.4.1 (Neighborhood I)**.** For a given hub-level structure and for a given hub node, i, the new structure resulted by replacing the given hub with a non-hub node having p-th ($p = 1 \dots 3$) highest value of $\frac{\sum_{j \neq k} W_{kj}}{\sum_{j \neq k} C_{kj}}, k \neq i$ and switching the assignment of all the incoming and outgoing edges of i to this p-th closest non-hub is called the p-th level neighbor of the given hub node with respect to the Neighborhood I for the given hub-level structure.

This neighborhood was used in order to investigate more possibilities that may lead to better solutions. Figure 7.2 depicts the best known hub-level structure of our greedy algorithm.

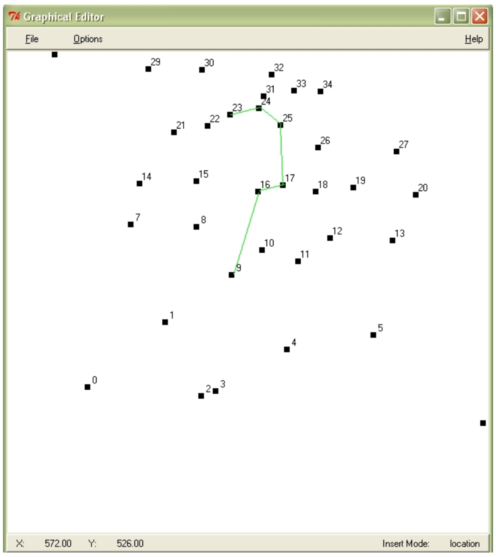


Figure 7.2: Best known solution of main neighborhood (AP35.14)

Figure 7.3 displays the result of searching in cooperation with the structure of new neighborhood (i.e. Neighborhood I). In fact, Figure 7.3 depicts the optimal solution.

The process continues in the following way: when no improvement is observed in the results of our greedy algorithm (Algorithm 10) there is a spoke node as the *best* of

[2]www.mathematik.uni-kl.de\ ~lola

p-th level best spoke nodes corresponding to any of the hubs that can be replaced by it (resulted configuration may have degraded objective function value but is a locally best choice which imitates the diversification process; if the new trial point is worse, maybe it is standing on a previously non-explored peak which can topple into a deeper narrow if become subjected to a search with respect to the original neighborhood). By moving to this neighbor regarding the idea of neighborhood I and clashing this new structure to the original neighborhood structure (by delivering this new trial point to the greedy search) we may have a new better hub-level structure. That is, the greedy search may remove some components in favor of other ones. The same occurrence appears here, when the hub node #10 was replaced by #16.

This new structure was delivered to the greedy algorithm and it restarts with this new solution as the initial solution hoping to find a better one. Ultimately, it found a new structure which coincides with the optimal hub-level structure. In this case we just set $p = 1$.

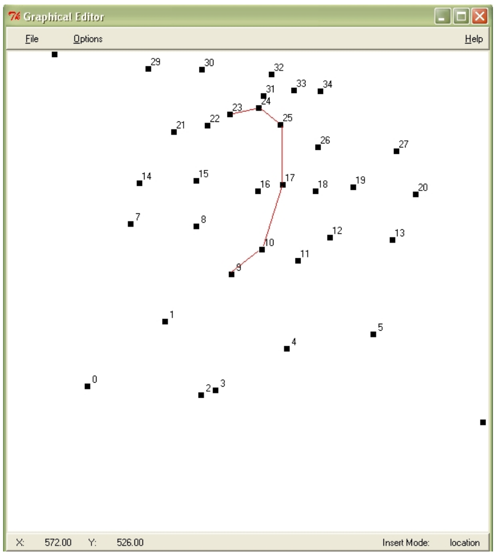


Figure 7.3: Optimal solution found after search on the Neighborhood I (AP35.14)

This strategy with $p = 1$, is successfully applied on problems which their best known solution has not been optimal, like AP35.14 until AP50.20. All reached the optimality.

In fact, what happened here was more or less avoiding *freezing* in the local search solution. That is, sometimes due to the greediness of the algorithm, a component appears (enters) in the hub-level structure with respect to the local optimality and in an improvident way. But, then it plays the role of an obstacle to reach the global optimum. This phenomena is called *freeze* phenomena and this new neighborhood structure is responsible to discover better solutions insofar as it is possible. We denote this algorithm by *greedy*$^+$ algorithm.

So far, even without considering the search on a second neighborhood structure, the greedy algorithm both in terms of run-time and the expected solution quality in this amount of time performs very well. Since HLPPT instances are not solved to optimality for larger size problems, the optimal solution is not available for reference purposes. Therefore, we again define another strategy to reduce the risk of premature convergence of our *greedy*$^+$ algorithm even more. Hence, we propose another neighborhood structure and call it *Neighborhood II*.

7.4.2 Neighborhood Search II

Alternatively, there can be another neighborhood structure to be used in the case that the Neighborhood I gets stuck in a local optimum, or even when we were not able to find a better solution than that of the basic greedy algorithm.

Definition 7.4.2 (Neighborhood II)**.** For a given hub-level structure and for a given hub node i, the new structure which results by replacing the given hub with the p-th ($p = 1 \dots 3$) closest non-hub node to i and switching the assignment of all the incoming and outgoing edges of the given hub to this p-th closest non-hub is called the p-th level neighbor of i with respect to the Neighborhood II for the existing hub-level structure.

In some cases, this neighborhood is dominated by Neighborhood I.

Our observations revealed that in the case of AP problems, the first neighborhood suffices to find optimal solutions. However, the second neighborhood could improve the solutions of some instances which have been solved by *greedy*$^+$ and their optimal solutions are unknown.
In the case of the AP instances, these two neighborhoods dominate all other neighborhoods that we implemented with respect to the quality of solution and also computational time. Many different and some times pretty complicated neighborhood structures have been tried without any improvement.

We denote the *greedy*$^+$ equipped with *Neighborhood II* as *greedy**.

7.4.3 Neighborhood Size and Complexity

Both types of neighborhood structures (I and II) have similar size. Moreover, the procedures applied for employing this neighborhood structures to the result of search on the original neighborhood are exactly the same. In this subsection we shed more light on details and size of such neighborhoods and procedures.

For a given network configuration and a non-empty set of spoke nodes resulted by partitioning the nodes into two sets of hub nodes and spoke nodes, there exists a single spoke node (if a tie exists an arbitrary selection clears it) corresponding to each of hub nodes as p-th level neighbor. Therefore, the size of each p-th level neighborhood is $n-1$, if the spoke node set is a singleton. That is, when there is a single spoke node, that is the p-th level neighbor to each of the hub nodes. In an extreme case, there exist $\frac{n}{2}$ distinct nodes as the neighbors of the remaining $n-\frac{n}{2}$ nodes, where n is even ($\frac{n+1}{2}$ or $\frac{n-1}{2}$ hub nodes for an odd n and $n-\frac{n+1}{2}$ and $n-\frac{n-1}{2}-1$, p-th level neighbors, respectively). For each hub node, the p-th neighbor should be selected from among $\frac{n}{2}$ ($\frac{n-1}{2}$ or $\frac{n+1}{2}$) spoke nodes. Each of the neighbors is evaluated by Dijkstra's algorithm and subjected to a neighborhood search based on the main neighborhood structure. The complexity of each step has been explained before.

7.4.4 Quality of Solutions

Here, we examine the quality of our heuristic by comparing its best-known solutions with those of the LP relaxation of HLPPT (these LP solutions are achieved from Algorithm 9). In Table 7.2, the second column reports the run-time of heuristic and the third one the run-time of Algorithm 9 to solve LP relaxation. The gap between the solution of $greedy^*$ and the LP relaxation is measured by $\frac{UB-LB}{LB}\times 100$ and is depicted in the column titled by Gap^{LP} (%). The gap between the optimal solution of AMC2^d and that of our heuristic is depicted in the last column (Gap^{opt} (%)).

Table 7.2: Final Comparison.

Instance	$greedy^*$ (sec.)	HLPPT^{LP}	Gap^{LP} (%)	Gap^{opt} (%)
AP5.2	0.03	0.41	0	0_{opt}
AP10.4	0.14	2.44	0	0_{opt}
AP15.6	0.27	11.50	0	0_{opt}
AP20.8	0.95	32.98	0.1	0_{opt}
AP25.10	2.53	96.16	1.8	0_{opt}
AP30.12	7.13	451.06	1.4	0_{opt}
AP35.14	18.39	1024.25	3.9	0_{opt}
AP40.16	34.66	2475.16	3.9	0_{opt}
AP45.18	70.22	4098.70	4.8	0_{opt}
AP50.20	176.13	6188.42	5.9	0_{opt}
AP55.22	264.05	12070.44	6.9	-
AP60.24	565.91	28774.47	7.7	-
AP65.26	663.86	54109.12	12.0	-

Table 7.3: *greedy** Algorithm run-time report

Instance	T. Cpu(s)	Instance	T. Cpu(s)
AP5.2	0.03	AP55.22	264.05
AP10.4	0.14	AP60.24	565.91
AP15.6	0.27	AP65.26	663.86
AP20.8	0.95	AP70.28	1407.46
AP25.10	2.53	AP75.30	1785.86
AP30.12	7.13	AP80.32	3142.24
AP35.14	18.39	AP85.34	4274.56
AP40.16	34.66	AP90.36	6934.25
AP45.18	70.22	AP95.38	8601.04
AP50.20	176.13	AP100.40	15370.72

We can conclude that our heuristic is extremely efficient. Moreover, the LP relaxation is a good approximation of optimal solution of HLPPT. For example, in the case of AP 50.20, our heuristic found the optimal solution in 176.13 seconds while an LP solution with gap of 5.9% is achieved in 6188.42 seconds (35.13 times higher computational time).

7.4.5 Computational Results

Table 7.3 reports the computational results for some instances of AP. When the optimal solution is known the algorithm was able to report it.

7.4.6 Algorithm

Algorithm 11 details the *greedy*$^+$ algorithm.

7.5 Heuristic for the DHLPPT

Consideration of delays which occur at each switch from one train to another one in the first glance seems to be complicated. However, it can be handled in an easy way. Focusing on the structure of this variant reveals that, in fact, we are faced with the same problem as HLPPT. The delay cost incurred at the hub node, where a flow from spoke node waits for the first possible high-speed facility (or gets off from it and waits for the spoke-level facility) is actually due to making use of spoke edges. That is, this cost can be added to the spoke edge flow costs, if the corresponding flow is supposed

Algorithm 11: Improved greedy algorithm for HLPPT ($greedy^+$)

```
Input: HLPPT instance
Output: x*
x̄:=Create_initial_solution();
min = Eval(x̄);
last_min := ∞;
repeated_min := 0;
while (repeated_min = 0) do
    f̄ := Eval(x̄);
    if f̄ ≤ min then
        min := f̄;
        x* := x̄;
    end
    foreach i := 1  to  nrLocations ∗ (nrLocations − 1)/2 do
        Δf := 0;
        x' := x̄;
        x'_i := 1 − x'_i;
        if  is_not_feasible(x') then
            Δf := ∞;
        else
            Δf := Eval(x') − min;
        end
        if Δf < 0 then
            x* := x' ;
            min := Eval(x');
        end

    end
    if min = last_min then
        repeated_min := repeated_min + 1;
        if repeated_min = 1 then
            x̄ := Search_Neighborhood_I(x*);
            if Eval(x̄) < Eval(x*) then
                x* := x̄;
                repeated_min := 0;
                goto 6;
            end
        end
        if repeated_min ≥ 2 then
            return x*;
        end
    end
    last_min := min;
    x̄ := x*;
end
stop.
```

to go to destination through some hub edges.

Let i be the origin and j the destination:

- If both i and j are hubs, since all the flows between pairs of hub nodes are sent through hub edges and no switch to any spoke edges takes place, there can not be any delay time (see Figure 7.4).

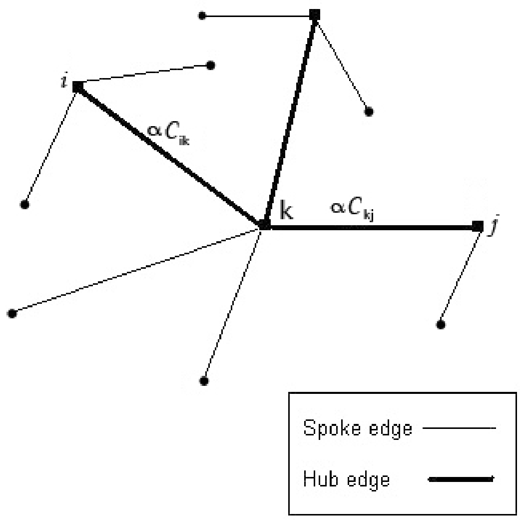


Figure 7.4: No delay occurs in a hub-to-hub path.

- If i is a spoke node and j is a hub node, the only direct possible spoke edge does not incur any delay cost (see Figure 7.5). In any other path containing at least one hub-edge connection where the first hub node of this path is $k \neq i, j$, switching from a spoke edge to a hub edge leads to a delay cost at the hub node k. This cost is added to the spoke edge $i - k$ and again results in a normal HLPPT where delay is implicitly considered (see Figure 7.6). Analogously, this holds when i is a hub node and j is a spoke node.

- If both, i and j are spoke nodes, either these two are connected via a single hub node or through a hub-edge-path.
  - if they are connected by a single hub node, say k, there occurs no waiting time due to any unforseen delay anywhere in the hub-level network. Therefore, no waiting time is imposed by the hub-level network (see Figure 7.7).

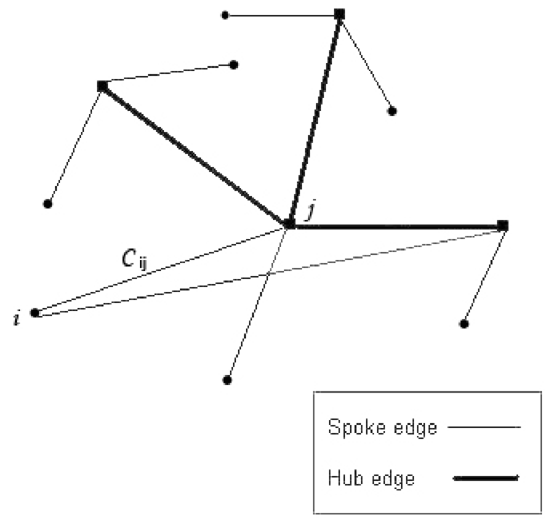


Figure 7.5: No delay occurs in a direct connection.

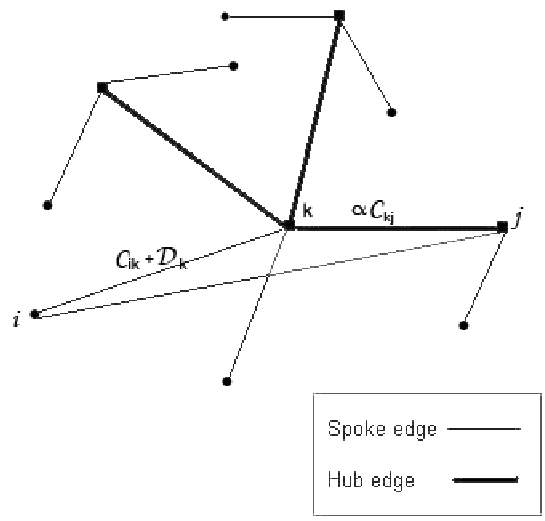


Figure 7.6: Delay occurs at the switch-point hub node.

– If the path contains at least one hub edge, then there are two points where delays occur. The first delays occur at the first hub node and the second one at the last hub node. Therefore, the first delay cost is added to the first spoke edge and the latter to the last one (see Figure 7.8).

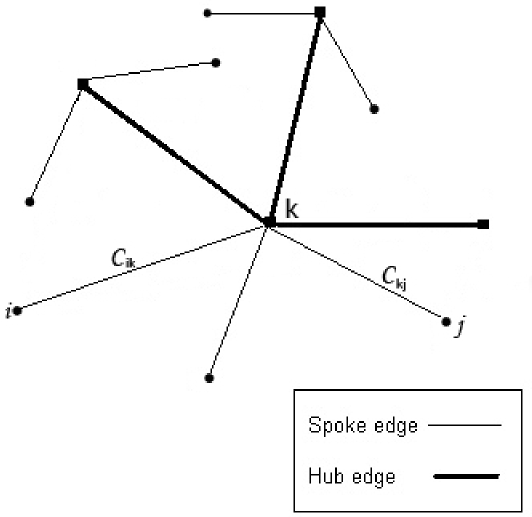


Figure 7.7: No delay cost is imposed on the transportation.

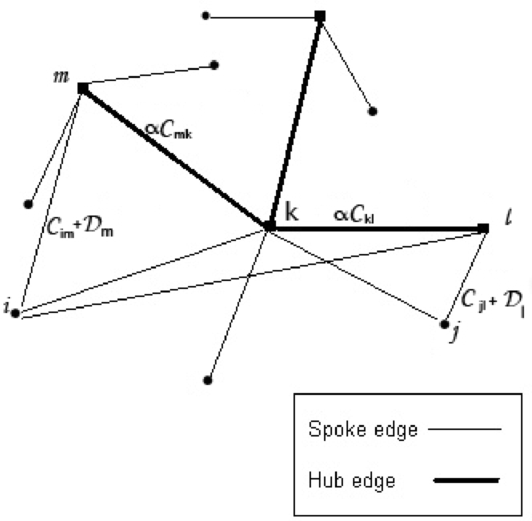


Figure 7.8: Delay occurs at the first and last hub nodes.

Algorithm

The algorithm *greedy** with a slightly modified strategy for measuring the shortest paths is applicable. That is, the weights of the graph should be modified and also all the possible paths not incurring any delay must be examined. Then, the cheapest among the possible paths is selected.

Chapter 8

A Generalized Model for Multi-Period Planning

In reality, hub location systems for public transportation encompass more aspects than those that we have considered so far. In order to achieve a more practical model, at least some of these aspects should be taken into account. At the same time, to avoid uncontrollable complexities in the model, a trade-off between the reliability and complexity should be drawn. However, the model is expected to be a good approximation of the real-life problem.

In this chapter, we extend our proposed basic HLPPT model to a more general one wherein more facts and features of the nature of the application are incorporated.

In 2006, Melo *et al.* [60], proposed a mathematical modeling framework for strategic supply chain planning. This work deals with the dynamic multi-commodity capacitated facility location problem where decisions are made for a planning horizon. They showed that their model incorporates and generalizes all the different aspects considered by other authors in literature.

8.1 Multi-Period Model for HLPPT (MPHLPPT)

In a multi-period HLPPT, the transport network evolves in time over the planning horizon. Decisions about how the network should evolve are not made just in an improvident and myopic manner, that is, the configuration of the system in each individual period may not be optimal for that period. Instead, it sacrifices the individual period optimality in favor of its contribution to the optimality of the whole planning horizon (global optimality). In this model, without loss of generality we assume that there exists an initial configuration prior to the starting period.

Exploiting the full performance of an *a priori* established facility at period t in the subsequent periods $(t+1, t+2, \dots)$ is only possible if we invest on maintenance activities in order to overcome the depreciation of facilities. Of course, when a facility is no longer used in the system, some additional removal costs are incured. This concerns both hub nodes and hub edges. Regarding the hub nodes, it amounts to the costs of re-training or discharge of employees, ceasing costs, etc. Concerning the hub edges, it is related to the removal of elements of fast-lines needed to be paid to maintain (e.g. special navigation system, high-speed trains, particular type of vehicles, specific service-support centers, etc.).

In our extended model, in addition to the existing assumptions for the single period problem, the following assumptions are also taken into consideration:

- the transport network includes an initial configuration before the starting period,
- the status of each hub node or hub edge can change at most once. That is,
  - if a facility exists in the initial configuration it may become closed afterwards,
  - if a facility become closed at a period in the planning horizon, it remains closed until the end of planning horizon and,
  - if a facility become open at a period over the planing horizon it will not be subject to removal.
- a fixed maintenance cost incurred for using a hub node or hub edge in each period (that means, an amount of budget is considered periodically for maintenance. The vehicles, roads and rails, stations, buildings and many more items are subject to inspection, control and renewal) and,
- a fixed ceasing (removal) cost is incurred in order to degrade a hub node or hub edge to a spoke one.

All these aspects of real-life are incorporated into our new model.

8.2 Mathematical Model of MPHLPPT

New parameters and variables are introduced and the re-interpretation of the already existing variables are reviewed to use them in our new model.

8.2.1 Parameters

In order to extend the model to a more general case, which assumes setup, maintenance and removal costs for both hubs nodes and hub edges in each period, the following parameters are introduced:

- HMC_k^t: The maintenance cost incurred by k-th hub node at the t-th period,
- EMC_e^t: The maintenance cost incurred by e-th hub edge at the t-th period,
- HCC_k^t: The removal cost incurred by k-th hub at the t-th period and,

- ECC_e^t: The hub edge removal cost incurred by e-th hub edge at the t-th period,

where $e \in E = \{(k,l)|k,l = 1,\ldots,n, l > k\}$.

8.2.2 Variables

In order to reflect the requirement of our new model, the definition of variables should be revised. According to the assumptions of model concerning the existence of an initial configuration, two sets of facilities are imagined.

Two index sets for facilities are H, keeping track of indices of potential hub nodes and E which analogously is defined for potential hub edges. Each set is partitioned into two subsets. The subset composed of indices of facilities which can be opened (say *openable*) at any $t \in \mathcal{T}$ during the planning horizon labeled with a superscript ^{o}, like H^o and E^o; and the sets of those which are active in initial configuration or, in the other words, that can be closed later on (say *closable*), are labeled by a superscript ^{c}, like H^c and E^c. This partitioning implies:

$$E^c \cap E^o = \emptyset, \quad E^c \cup E^o = E, \tag{8.1}$$
$$H^c \cap H^o = \emptyset, \quad H^c \cup H^o = H. \tag{8.2}$$

Therefore, we revise the definition of the following variables:

For all $k \in H^o$ and $t \in \mathcal{T} = \{1\ldots T\}$,

$$h_k^t = \begin{cases} 1 & \text{if hub node } k \text{ is established at the begining of the time period } t, \\ 0 & \text{otherwise.} \end{cases}$$

For all $k \in H^c$ and $t \in \mathcal{T} - \{T\}$,

$$h_k^t = \begin{cases} 1 & \text{if hub node } k \text{ is removed at the end of time period } t, \\ 0 & \text{otherwise.} \end{cases}$$

For all $k \in H^c$:

$$h_k^T = \begin{cases} 1 & \text{if hub node remains active until the end of the planning horizon,} \\ 0 & \text{otherwise,} \end{cases}$$

where T is the length of the planning horizon. In addition, y_{kl}^ts are analogously defined for the set of edges.

Note. Each edge is represented by $e \in E$ and corresponds to the edge (k,l). That means, k is the head and l is the tail node of that edge. Moreover, due to the undirected nature of the hub-level graph, it is always assumed that $l > k$.

8.2.3 Mathematical Formulation

The generality of the model is very similar to the single period case (i.e. HLPPT). Additionally, we have to establish links between periods and also express the single period model in terms of the newly defined sets and variables ((8.1) and (8.2)). Since the model becomes involved, therefore we explain the constraints and make some labels (A, B, ...) step by step to reach the final model by bringing these labels together.

(A): A closable facility should either be closed before the last period or corresponding variable takes 1 at the last period indicating that it has been open over the planning horizon. In Contrary, an openable facility can be opened just once. The following are the constraints taking care of these facts for both type of facilities.

$$\sum_{t \in \mathcal{T}} h_k^t = 1, \qquad \forall k \in H^c, \quad (8.3)$$

$$\sum_{t \in \mathcal{T}} y_e^t = 1, \qquad \forall e \in E^c, \quad (8.4)$$

$$\sum_{t \in \mathcal{T}} h_k^t \leq 1, \qquad \forall k \in H^o, \quad (8.5)$$

$$\sum_{t \in \mathcal{T}} y_e^t \leq 1, \qquad \forall e \in E^o. \quad (8.6)$$

(B): In our model we assume that there exists just a limited amount of resources (budget, workforce, etc.) which only suffices to establish a limited number of facilities. For instance, for given $q1, q2 \in \mathbb{N}$, $q1$ hub nodes and $q2$ hub edges from among all the openable ones in each time period can be established.

$$\sum_{e \in E^o} y_e^t \leq q1, \qquad \forall t \in \mathcal{T}, \quad (8.7)$$

$$\sum_{k \in h^o} h_k^t \leq q2, \qquad \forall t \in \mathcal{T}. \quad (8.8)$$

(C): An openable hub edge between two end-points of k and l can have its end-points belonging to different sets of the partition.

For all $t \in \mathcal{T}, l > k$,

- if k and l both are openable, an openable edge e can be opened if both end-point have been opened until now.

$$y_e^t \leq \sum_{t'=1}^{t} h_k^{t'}, \qquad \forall e \in E^o, k, l \in H^o,$$

$$y_e^t \leq \sum_{t'=1}^{t} h_l^{t'}, \qquad \forall e \in E^o, k, l \in H^o.$$

- if $k(l)$ is closable and $l(k)$ is openable, an openable edge e can be opened if the closable endpoint is not closed yet and will not be closed afterwards; and the openable end-point is opened until the beginning of this period.

$$y_e^t \leq 1 - \sum_{t'=1}^{T-1} h_k^{t'}, \qquad \forall e \in E^o, k \in H^c, l \in H^o, \quad (8.9)$$

$$y_e^t \leq \sum_{t'=1}^{t} h_l^{t'}, \qquad \forall e \in E^o, k \in H^c, l \in H^o, \quad (8.10)$$

$$y_e^t \leq \sum_{t'=1}^{t} h_k^{t'}, \qquad \forall e \in E^o, k \in H^o, l \in H^c, \quad (8.11)$$

$$y_e^t \leq 1 - \sum_{t'=1}^{T-1} h_l^{t'}, \qquad \forall e \in E^o, k \in H^o, l \in H^c. \quad (8.12)$$

- if k and l are closable an openable edge e can be opened if both end-points are not closed yet and will not be closed until the end of the $(T-1)$-th period.

$$y_e^t \leq 1 - \sum_{t'=1}^{T-1} h_k^{t'}, \qquad \forall e \in E^o, k, l \in H^c, \quad (8.13)$$

$$y_e^t \leq 1 - \sum_{t'=1}^{T-1} h_l^{t'}, \qquad \forall e \in E^o, k, l \in H^c. \quad (8.14)$$

- for a closable edge $e \in E^c$, where definitely both end-points are closable, e can remain open as long as both end-points remain open.

$$y_e^t \geq h_k^{t'}, \qquad \forall e \in E^c, k, l \in H^c,$$

$$y_e^t \geq h_l^{t'}, \qquad \forall e \in E^c, k, l \in H^c.$$

(D): For a given origin i and destination j, the flow from i to j can pass through the hub edge (k, l):

- if (k, l) is openable, it should have been opened so far,

$$x_{ijkl}^t + x_{ijlk}^t \leq \sum_{t'=1}^{t} y_e^{t'}, \qquad \forall i, j > i, e \in E^o, t \in \mathcal{T}.$$

- if (k,l) was closable, it should have not been closed yet.

$$x_{ijkl}^t + x_{ijlk}^t \leq 1 - \sum_{t'=1}^{t-1} y_e^{t'}, \qquad \forall i, j > i, e \in E^c, t \in \mathcal{T}.$$

(E): For a given flow between two nodes when only one of them is a hub:

- if that hub node is openable, it should have been opened in a period since the beginning of the planning horizon,

$$\sum_{l \neq k} x_{kjkl}^t \leq \sum_{t'=1}^{t} h_k^{t'}, \qquad \forall j, k \in H^o, k < j, t \in \mathcal{T},$$

$$\sum_{k \neq l} x_{ilkl}^t \leq \sum_{t'=1}^{t} h_l^{t'}, \qquad \forall i, l \in H^o, l > i, t \in \mathcal{T}.$$

- if that hub node was closable, it should have not been closed yet.

$$\sum_{l \neq k} x_{kjkl}^t \leq 1 - \sum_{t'=1}^{t-1} h_k^{t'}, \qquad \forall j, k \in H^c, k < j, t \in \mathcal{T},$$

$$\sum_{l \neq k} x_{ilkl}^t \leq 1 - \sum_{t'=1}^{t-1} h_l^{t'}, \qquad \forall i, l \in H^c, l > i, t \in \mathcal{T}.$$

(F): The following constraints correspond to those of the single period model with the same modification as carried out in the other constraints.

For all $t \in \mathcal{T}$,

$$e_{ij}^t \leq \sum_{t'=1}^{t} |h_i^{t'} - h_j^{t'}|, \qquad \forall i, j \in H^o, j > i,$$

$$e_{ij}^t \leq 1 - |\sum_{t'=1}^{t} h_i^{t'} - \sum_{t'=1}^{t-1} h_j^{t'}|, \qquad \forall i \in H^o, j \in H^c, j > i,$$

$$e_{ij}^t \leq 1 - |\sum_{t'=1}^{t} h_i^{t'-1} - \sum_{t'=1}^{t} h_j^{t'}|, \qquad \forall i \in H^c, j \in H^o, j > i,$$

$$e_{ij}^t \leq \sum_{t'=1}^{t-1} |h_i^{t'} - h_j^{t'}|, \qquad \forall i, j \in H^c, j > i.$$

However, these constraints are not linear. In order to linearize them, we define more variables namely, $\delta_{ij}^+ \geq 0$ and $\delta_{ij}^- \geq 0$.

For all $t \in \mathcal{T}$,

$$\sum_{t'=1}^{t}(h_i^{t'} - h_j^{t'}) = \delta_{ij}^{t+} - \delta_{ij}^{t-}, \qquad \forall i, j \in H^o, j > i,$$

$$1 - \sum_{t'=1}^{t} h_i^{t'} - \sum_{t'=1}^{t-1} h_j^{t'} = \delta_{ij}^{t+} - \delta_{ij}^{t-}, \qquad \forall i \in H^o, j \in H^c, j > i,$$

$$1 - \sum_{t'=1}^{t-1} h_i^{t'} - \sum_{t'=1}^{t} h_j^{t'} = \delta_{ij}^{t+} - \delta_{ij}^{t-}, \qquad \forall i \in H^c, j \in H^o, j > i,$$

$$\sum_{t'=1}^{t-1}(h_i^{t'} - h_j^{t'}) = \delta_{ij}^{t+} - \delta_{ij}^{t-}, \qquad \forall i, j \in H^c, j > i,$$

and finally for all $t \in \mathcal{T}$, i and $j > i$ we add,

$$e_{ij}^t \leq \delta_{ij}^{t+} + \delta_{ij}^{t-} \leq 1.$$

(G): By definition, the variable a_{ijk} (b_{ijk}) is used to show whether there is a flow sent from (to) a spoke origin (destination) to (from) an arbitrary destination (origin) j (i) via a hub node k.

For all $t \in \mathcal{T}$,

$$a_{ijk}^t \leq 1 - \sum_{t'=1}^{t} h_i^{t'}, \qquad \forall i \in H^o, j > i, k \neq i, j,$$

$$a_{ijk}^t \leq \sum_{t'=1}^{t-1} h_i^{t'}, \qquad \forall i \in H^c, j > i, k \neq i, j,$$

$$b_{ijk}^t \leq 1 - \sum_{t'=1}^{t} h_j^{t'}, \qquad \forall j \in H^o, i < j, k \neq i, j,$$

$$b_{ijk}^t \leq \sum_{t'=1}^{t-1} h_j^{t'}, \qquad \forall j \in H^c, i < j, k \neq i, j.$$

(H): The flow emanated from a node i is received by a node j based on the status of i and j:

For all $t \in \mathcal{T}$,

$$a_{ijk}^t + \sum_{l \neq j,k} x_{ijlk}^t \leq \sum_{t'=1}^{t} h_k^{t'}, \qquad \forall i, j > i, k \in H^o, k \neq i, j,$$

$$a_{ijk}^t + \sum_{l \neq j,k} x_{ijlk}^t \leq 1 - \sum_{t'=1}^{t-1} h_k^{t'}, \qquad \forall i, j > i, k \in H^c, k \neq i, j,$$

$$b_{ijk}^t + \sum_{l \neq k,i} x_{ijkl}^t \leq \sum_{t'=1}^{t} h_k^{t'}, \qquad \forall i, j > i, k \in H^o, k \neq i, j,$$

$$b_{ijk}^t + \sum_{l \neq k,i} x_{ijkl}^t \leq 1 - \sum_{t'=1}^{t} h_k^{t'-1}, \qquad \forall i, j > i, k \in H^c, k \neq i, j,$$

$$e_{ij}^t + 2x_{ijij}^t + \sum_{l \neq j,i} (x_{ijil}^t + x_{ijlj}^t) \leq$$

$$\begin{cases} \sum_{t'=1}^{t} (h_i^{t'} + h_j^{t'}), & \forall i, j \in H^o, j > i, \\ \sum_{t'=1}^{t} h_i^{t'} + (1 - \sum_{t'=1}^{t-1} h_j^{t'}) & \forall i \in H^o, j \in H^c, j > i, \\ (1 - \sum_{t'=1}^{t-1} h_i^{t'}) + \sum_{t'=1}^{t} h_j^{t'}, & \forall i \in H^c, j \in H^o, j > i, \\ 2 - (\sum_{t'=1}^{t-1} h_i^{t'} + \sum_{t'=1}^{t-1} h_j^{t'}) & \forall i, j \in H^c, j > i. \end{cases}$$

Now, we can state MPHLPPT:

(MPHLPPT)

$$\begin{aligned}
Min \quad & \sum_{t \in \mathcal{T}} \Bigg(\sum_i \sum_{j>i} \sum_k \sum_{l \neq k} \alpha^t (W_{ij}^t + W_{ji}^t) C_{kl}^t x_{ijkl}^t + \sum_i \sum_{j>i} (W_{ij}^t + W_{ji}^t) C_{ij}^t e_{ij}^t + \\
& \sum_i \sum_{j>i} \sum_{k \neq i,j} (W_{ij}^t + W_{ji}^t) C_{ik}^t a_{ijk} + \sum_i \sum_{j>i} \sum_{k \neq i,j} (W_{ij}^t + W_{ji}^t) C_{kj}^t b_{ijk}^t + \\
& \sum_{t \in \mathcal{T}} \sum_{k \in H^o} \left(F_k^t + \sum_{t'=t}^{T} HMC_k^{t'} \right) h_k^t + \sum_{t \in \mathcal{T}} \sum_{k \in H^c} \left(\sum_{t'=1}^{t} HMC_k^{t'} + HCC_k^t \right) {h_k}^t + \\
& \sum_{t \in \mathcal{T}} \sum_{e \in E^o} \left(I_e^t + \sum_{t'=t}^{\mathcal{T}} EMC_k^{t'} \right) y_e^t + \sum_{t \in \mathcal{T}} \sum_{k \in E^c} \left(\sum_{t'=1}^{t} EMC_e^{t'} + ECC_e^t \right) {y_e}^t \Bigg) \\
& - \sum_{k \in H^c} HCC_k^t h_k^T - \sum_{e \in E^c} ECC_e^t y_e^T \qquad (8.15) \\
s.t. \quad & \sum_{l \neq i} x_{ijil}^t + \sum_{l \neq i,j} a_{ijl}^t + e_{ij}^t = 1, \qquad \forall t, i, j > i, \\
& \sum_{l \neq j} x_{ijlj}^t + \sum_{l \neq i,j} b_{ijl}^t + e_{ij}^t = 1, \qquad \forall t, i, j > i, \\
& \sum_{l \neq k,i} x_{ijkl}^t + b_{ijk}^t - \sum_{l \neq k,j} x_{ijlk}^t - a_{ijk}^t = 0, \qquad \forall t, i, j > i, k \neq i, j, \\
& \mathbf{A, B, C, E, F, G, H}, \\
& x_{ijkl}^t, a_{ijk}^t, b_{ijk}^t, e_{ij}^t \in (0,1), y_{kl}^t, h_k^t \in \{0,1\}.
\end{aligned}$$

The objective function describes the flow cost plus the setup, maintenance and closing costs. Since, the problem is a minimization problem and positive ceasing costs are considered, no closable facility can be closed in the last period, i.e. $t = T$. Therefore, we do not add the corresponding terms to the objective function.

8.3 Graphical Demonstration

To give a clear understanding and illustrate the MPHLPPT network evolution, we depict the evolution process, period by period and explain it in more detail. This clarifies the ideas of some constraints of the model. We start our explanation using a randomly generated instance with 10 nodes and a planning horizon of 3 periods.

8.3.1 Initial Configuration

A random initial configuration is generated (R10). According to the Figure 8.1 and the definition of the partitions, we have:

$$H^c = \{1, 4\}, \qquad H^o = H \setminus H^c, \tag{8.16}$$

$$E^c = \{(1, 4)\}, \qquad E^o = E \setminus E^c. \tag{8.17}$$

8.3.2 Configuration of First Period

In the first period, some hub edges and hub nodes are established. According to our constraints, there cannot be more than $q1$ hub nodes and $q2$ hub edges (here, we set $q1 = q2 = 3$). Therefore, in the first period we have,

$$h_0^1 = 1,\ h_5^1 = 1,\ h_8^1 = 1,\ y_{04}^1 = 1,\ y_{48}^1 = 1 \text{ and } y_{58}^1 = 1.$$

Figure 8.2 depicts the result of first period. The graph composed of this combined configuration is a connected hub-level network graph.

As an example of paths in the optimal solution we pick up $3-9$. In the first period, flow emanated from 3 and destined to 9 transits through the hub node of 5. That is,

$$\text{3-9} : a_{395}^1 = 1, b_{395}^1 = 1. \tag{8.18}$$

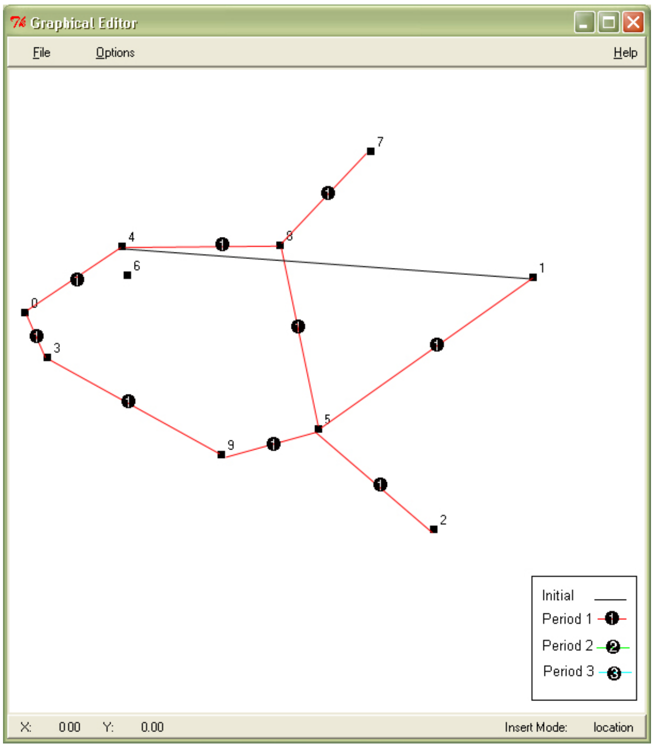


Figure 8.1: Random initial configuration of our instance.

Therefore, just one hub node is used in this transportation path. Moreover, none of the closable facilities are closed yet.

8.3.3 Configuration of Second Period

Again the evolution process result is depicted in Figure 8.3.

The same as in the previous period, at most three hub facilities can be established. As one can see in the Figure 8.3, we have,

$$h_1^2 = 2,\ h_9^2 = 1,\ y_{15}^2 = 1,\ y_{25}^2 = 1 \text{ and } y_{59}^2.$$

Two hub nodes and three hub edges are added in such a way that again the extended hub level network is connected.

$$3\text{-}9 : e_{39}^2 = 1. \tag{8.19}$$

The closable facilities are still operating.

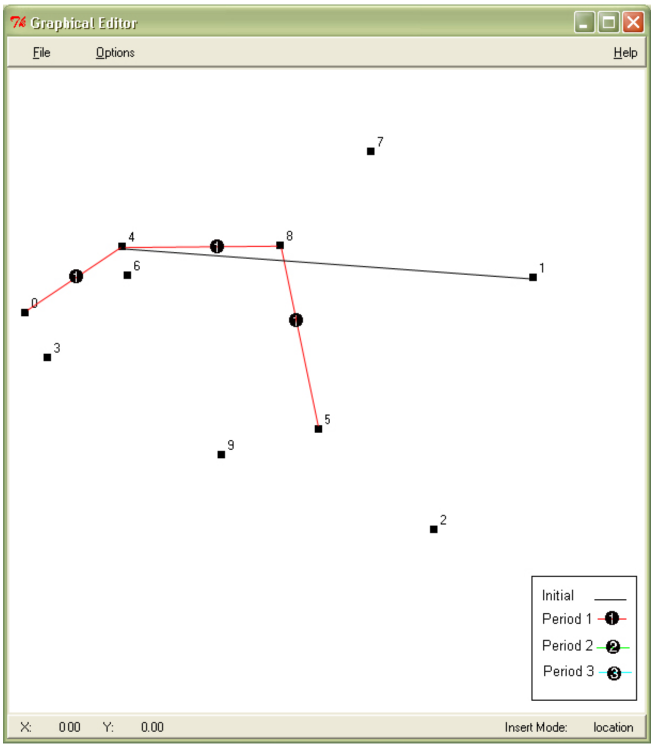


Figure 8.2: The configuration in the first period in the optimal solution

8.3.4 Configuration of Third Period

The final configuration is the result of the last period where the network is completely evolved. The result is displayed in Fig 8.4.

In the same way as with the last two periods, newly added facilities evolve the configuration to a new one with a connected hub level network. It results in,

$$h_3^3 = 1,\ h_7^3 = 1,\ y_{03}^3 = 1,\ y_{39}^3 = 1 \text{ and } y_{78}^3 = 1,$$

and therefore,

$$3\text{-}9 : x_{3939}^3 = 1. \tag{8.20}$$

As the optimal solution shows, the initial configuration elements were always beneficial to keep rather than become closed. Removal of a hub facility is more likely to happen if the planing horizon is quite large and the reduction of a hub facility to a spoke one can have enough time to show its benefits.

Obviously, if there would be no constraints on the number of facilities to be estab-

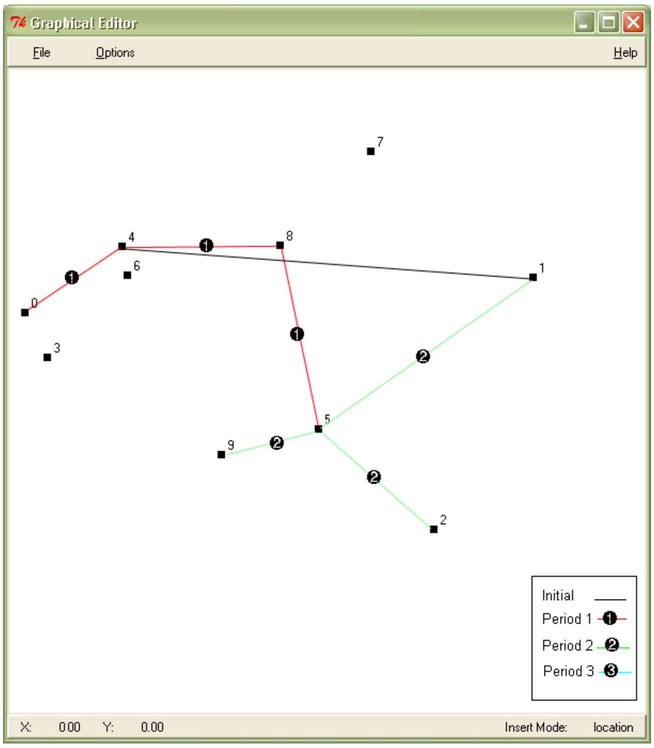


Figure 8.3: The configuration in the second period in the optimal solution

lished at each period and, benefits of using the facilities can dominate the cost incurred for setup and maintenance, most of hub facilities which are going to be opened will preferably be opened at the earlier periods. In this way, as soon as possible the economy of scale can be exploited. This tendency, is more sensible if the flow mass is homogeneously and monotonically increasing at each period. That is in each stage there is more motivation to establish hub-level facilities. But, if the flow in an area has a very low density and dramatically increases in the last periods, it is less likely that this area receives a hub facility in the earlier periods.

Figure 8.5 depicts the optimal solution of the model, where the constraint set **B** is relaxed. As one can see, all the facilities are established in the first period.

8.4 Computational Results

A set of 10 initial configurations is generated for a randomly generated instance with 10 nodes (R10) so that their spatial layout are homogenously distributed and are as scattered as possible with as few as possible of intersections.

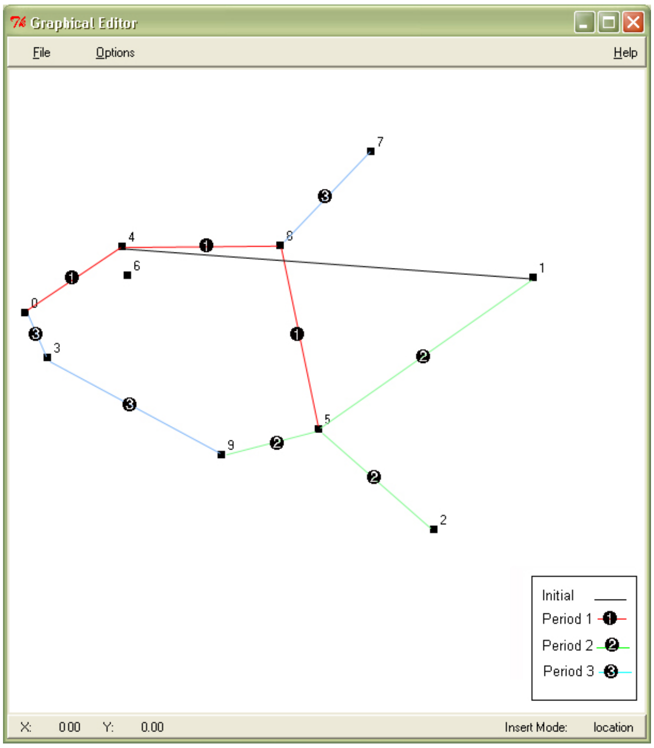


Figure 8.4: The configuration in the third period in the optimal solution

In Figure 8.6 the layout of the this instance is depicted. We divided the area into four quarters and tried to choose the initial configurations as fair as possible. That is, every node appeared at least once as a hub node in an initial configuration and also we selected the edges from almost all the parts of the layout.

The flow and cost structures are also randomly generated. Therefore, the cost as well as flow, in general, does not follow any structure.
The parameters of these instances are defined as follows,

$$
\begin{aligned}
&F_i = 5000 && \forall i,\\
&HCC_i = 1000 && \forall i,\\
&HMC_i = 500 && \forall i,\\
&I_{ij} = 200 \times C_{ij} && \forall i, j > i,\\
&ECC_{ij} = 200 \times C_{ij} && \forall i, j > i,\\
&EMC_{ij} = 20 \times C_{ij} && \forall i, j > i.
\end{aligned}
$$

8.4.1 Constrained by Number of Facilities (CN)

The restriction on the number of facilities among the potential ones (hub nodes and hub edges) to be opened in each period can be drawn by the constraint set **B** ((8.7)-

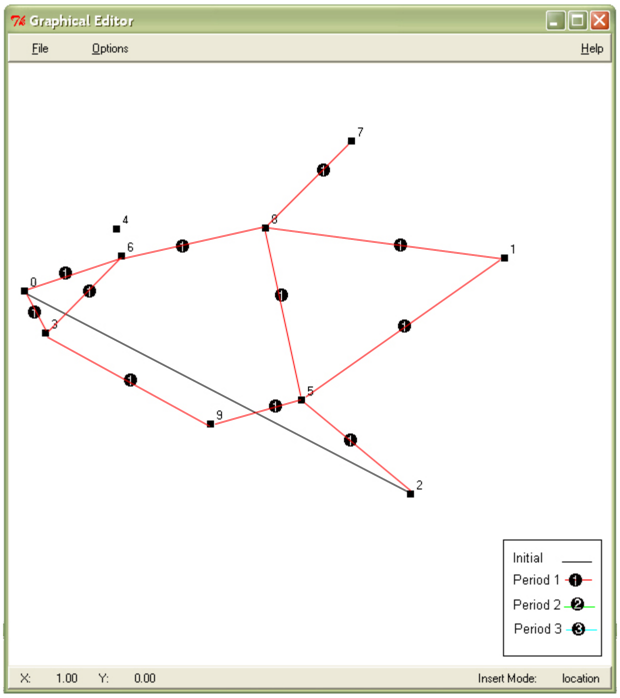


Figure 8.5: Optimal solution without any constraint on the number of facilities or amount of available budget at each period

(8.8)).

Example

From our experiences, `CPLEX 9.1` is not capable of solving instances larger than 10 with $T = 3$ in less than one week of computational effort before the variable reduction (before using the final version of HLPPT). But afterward, although the computational time is considerably reduced, it was not possible to solve them in a reasonable amount of time (i.e. less than half a day for an instance of size 15 and $T = 3$). Therefore we report our results for the randomly generated instance of size 10 with 10 distinctly generated initial configurations. Computational results are reported in Table 8.1.

The computations are carried out on a Intel(R)Xeo, n(TM)CPU 2.60 GHz and 1 GB of RAM.

Table 8.1, states that the root node gaps are small enough and in average the computational time is less than 2 minutes.

An interesting behavior of this problem is that, although the instances of size 10 with $T = 3$ (at least for our initial configurations) are mostly solved in less than 5

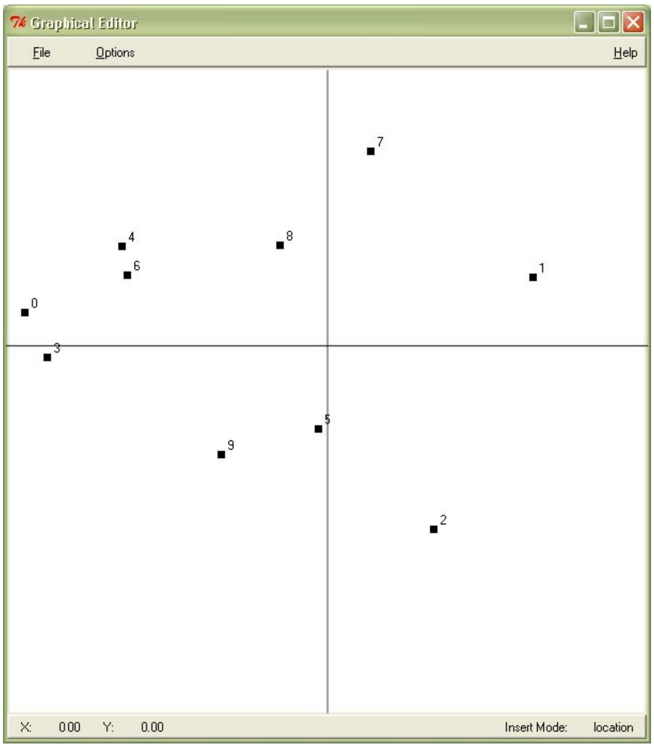


Figure 8.6: Spatial layout of R10.

Table 8.1: Constraints on number of nodes (CN).

CN	T_1. Cpu(s)	Root Node Gap(%)
{1-4}	168.27	2.21
{3-5},{5-8}	55.05	0.39
{1-6},{6-9}	93.24	2.49
{0-4},{4-8}	31.88	1.31
{1-5},{1-8}	66.31	3.47
{3-5},{5-7},{7-8}	70.36	0.45
{2-7},{0-7},{2-9}	161.86	4.68
{0-3},{3-5},{1-5}	47.11	0.62
{1-4},{1-2},{4-9}	116.53	2.10
{1-7},{3-7},{3-9}	95.44	4.56
Avg.	90.61	2.23

minutes. However, as the problem size grows from 10 to 15 the computational time dramatically increases. This clearly indicates the high complexity of the problem.

R15

In Figure 8.7, an optimally solved randomly generated instance of size 15, R15, is depicted. Here we let $q1 = q1 = 3$. The computational time for solving this instance to optimality using `CPLEX 9.1` is 95513.47 seconds (more than 26 hours). The root node gap is 17.34% .

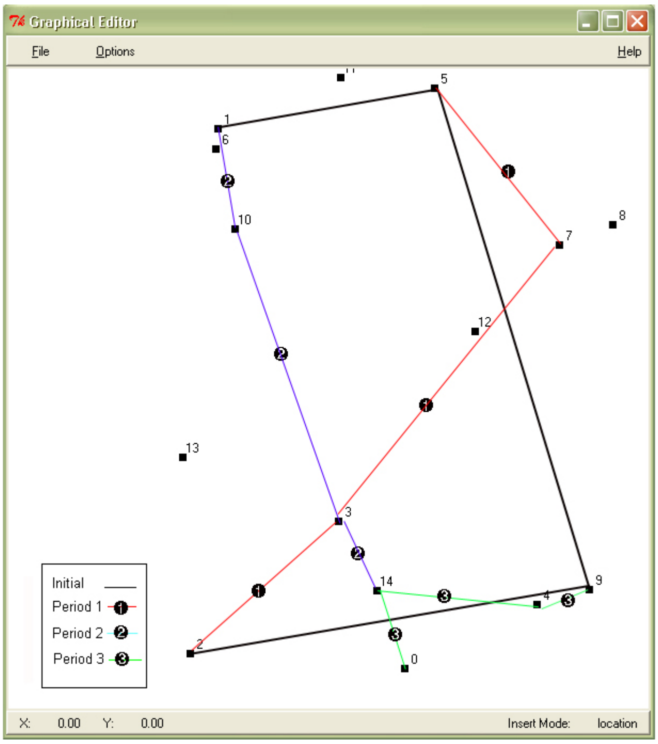


Figure 8.7: CN: Solution to the R15.

8.4.2 Constrained by Budgets for Activities (CB)

It has been assumed in (8.7)-(8.8) that there are limited number of facilities that can be established at each iteration. Rather than directly restricting the number of facilities to be established, these constraints can be expressed in terms of amount of available budget for the activities of project in each period which is more realistic.

It may not be only the financial aspects to prevent us. However, this can also be due to some other factors (like resources and machines, geographical or political issues, etc.). But, in our study only the financial issues are taken into account.

At each period, there exists a fixed amount of budget that can be invested for the facility establishment, maintenance and ceasing. Any capital available in a period but not invested then is subject to an interest rate and the returned value can be used in subsequent periods. This amount of return from the preceding period plus the fixed amount of budget available for the current period sums up to the whole amount of available budget for this period.

New variables and parameters are defined as follows. For all $t \in T$:

B^t : the amount of initially available budget at the beginning of period t,
ρ^t : unit return factor on capital not invested in period t ($\rho^t > 1$),
η^t : remaining amount of budget at the end of a period t.

The amount of budget available for example in the second period would amount to $B^2 + \rho^1\eta^1$.

According to the definition of these new variables, the following constraints are added to the model.

$$\begin{aligned}
&\sum_{k\in H^o}(F_k^1 h_k^1) + \sum_{k\in H^o}(h_k^1 \times HMC_k) + \sum_{k\in h^c}(HCC_k h_k^1) + \sum_{k\in h^c}(h_k^1 HMC_k)\\
&+\sum_{e\in E^o}(I_e^1 y_e^1) + \sum_{e\in E^o}(y_e^1 \times EMC_e) + \sum_{e\in y^c}(YCC_e y_e^1) + \sum_{e\in y^c}(y_e^1 YMC_e)\\
&+\eta^1 = B^1\\
&\sum_{k\in H^o}(F_k^t h_k^t) + \sum_{k\in H^o}((\sum_{t'=1}^{t} h_k^{t'}) \times HMC_k) + \sum_{k\in h^c}(HCC_k h_k^t)\\
&+\sum_{e\in E^o}(I_e^t y_e^t) + \sum_{e\in E^o}((\sum_{t'=1}^{t} y_e^{t'}) \times EMC_e) + \sum_{e\in h^c}(YCC_k y_e^t)\\
&+\sum_{k\in h^c}((1-\sum_{t'=1}^{t-1} h_k^{t'}) \times HMC_k) + \sum_{e\in E^c}((1-\sum_{t'=1}^{t-1} y_e^{t'}) \times EMC_e)\\
&+\eta_t = B^t + (\rho^{t-1}\eta^{t-1}) \qquad\qquad \forall t = 2\ldots T-1\\
&\sum_{k\in H^o}(F_k^T h_k^T) + \sum_{k\in H^o}((\sum_{t=1}^{T} h_k^t) \times HMC_k) + \sum_{k\in h^c}((1-\sum_{t'=1}^{T-1} h_k^{t'}) \times HMC_k)\\
&+\sum_{e\in E^o}(I_e^T y_e^T) + \sum_{e\in E^o}((\sum_{t'=1}^{T} y_e^{t'}) \times EMC_e) + \sum_{e\in E^c}((1-\sum_{t'=1}^{T-1} y_e^{t'}) \times EMC_e)\\
&+\eta^T = B^T + (\rho^{T-1}\eta^{T-1})
\end{aligned}$$

Example

Again, we had a similar difficulty for solving instances of larger than 10 nodes even with $T = 3$. Therefore we report our results for the same randomly generated instance with the same initial configurations as CN, $B^t = 200,000$ and $\rho^t = 1.2$. Computational results are reported in Table 8.2.

In average, instances can be solved in less than 3 minutes and root node gaps are

Table 8.2: Constraints on number of nodes (CB).

CB	T_1. *Cpu(s)*	Root Node Gap(%)
{1-4}	801.28	25.46
{3-5},{5-8}	97.66	0.86
{1-6},{6-9}	140.99	1.87
{0-4},{4-8}	119.75	1.77
{1-5},{1-8}	170.49	5.78
{3-5},{5-7},{7-8}	71.09	2.64
{2-7},{0-7},{2-9}	349.91	2.81
{0-3},{3-5},{1-5}	75.69	1.82
{1-4},{1-2},{4-9}	119.85	5.19
{1-7},{3-7},{3-9}	89.00	1.83
Avg.	203.571	5.00

in general small.

R15

In Figure 8.8, R15 is solved for the CB case. Here, we let $T = 3$ and $B^t = 300,000$ which is not a very tight budget capacity. To solve this instance, 47747.75 seconds (more than 13 hours) is the required computational effort. The root node gap is 8.65%.

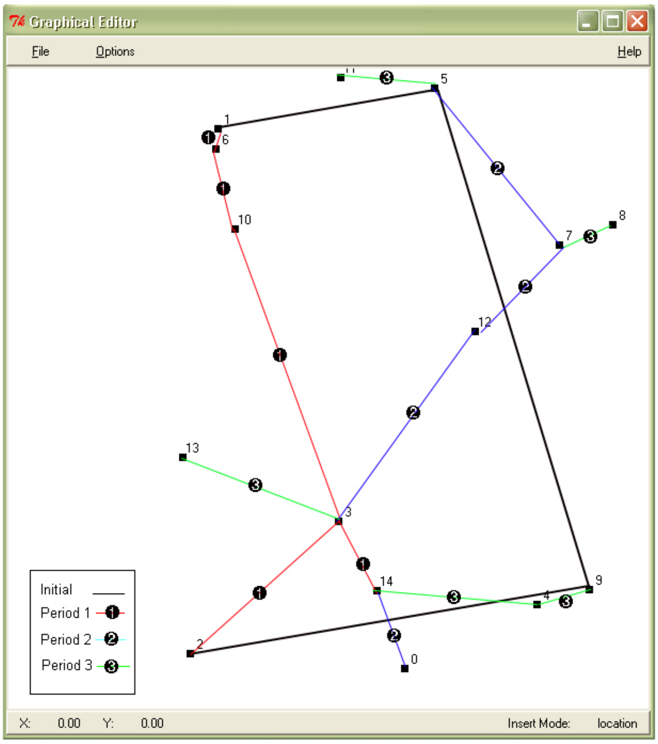


Figure 8.8: CB: Solution to the R15.

Chapter 9

A Heuristic Methods For The Multi-Period Approach

According to the computational results reported in the preceding chapter, even solving instances of quite small size is very time consuming. As an example, for $n = 15$ and a given initial configuration sometimes more than 26 hours of computations (more than one day) using a standard solver is needed.

Usually, in an engineering point of view, one is interested in a good or very good approximation of optimal solution in a much smaller amount of time. Our effort in this section focuses on the design and development of a heuristic capable of reaching as good as possible solutions in smaller amount of computational time compared to what `CPLEX 9.1` would need to find solutions with the similar quality.

9.1 Heuristic for MPHLPPT

The main idea of our heuristic stems from Algorithm 10. However, as one expects, the main difference would be the neighborhood structure. That is, our neighborhood should take care of changes in the status of facilities in periods (like postponing and antedating of the openings and closings to different periods).

9.1.1 Neighborhood Structure

The key element of our algorithm is the neighborhood structure which determines the rules to move from one configuration to another one. We adapt the neighborhood and adopt the idea of the $greedy^+$ algorithm by making it suitable for MPHLPPT. Again, the search process explores the set of edges and the original neighborhood is proposed upon the edge attributes.

Note. We always assume that a facility is closed in a period t, if it will not be available in the successive periods $(t + 1, t + 2, \dots)$, but it still works until the end of period t.

Definition 9.1.1 (Neighborhood Structure for MPHLPPT)**.** For a given period t and for the hub edge (i, j) in a given configuration, the neighborhood of this configuration obeys the following rules:

- if hub edge $i - j$ is a closable hub edge:

  - **"Close it from now on"**: if the hub edge is active now, then this hub is open since the beginning of planning horizon and maybe after the current period. Thus, close it at the end of current period,
  - **"Keep it open until now"**: if the hub edge is closed at this time, then it will be kept open from the beginning of the planning horizon until now and will be closed at the end of the current period until the end of the planning horizon,

- if the hub edge $i - j$ is an openable hub edge:
  - **"Keep it closed until now"**: if the hub edge is active now then it should be kept closed until the end of this period and starts working from the successive period, if it exists,
  - **"Open it from now on"**: if the hub edge is closed at the current period then it becomes open from now on until the end.

In our idea, the prescribed neighborhood structure is comprehensive and concise enough to consider all possibilities of the solution space.

9.1.2 Example

In this section for a given hub-level structure we go through all the neighbors with respect to one of the hub edges. Both types of edges namely, *closable* and *openable* are considered.

Closable Hub Edge Facility

For the closable hub edge $1 \in E^c$, as Table 9.1 shows, it keeps working until the end of period 3.

Table 9.1: A given solution.

Hub edge	$t = 1$	$t = 2$	$t = 3$	$t = 4$
$1 \in E^c$	**1**	**1**	**1**	**0**
$2 \in E^c$	1	0	0	0
$3 \in E^o$	0	1	1	1
$4 \in E^o$	0	0	0	1

In the sequel, we depict the neighbors with respect to this hub edge.

$t = 1$

For the closable hub edge 1, since this facility is active even after the period $t = 1$, we apply the rule *"close it from now on"* as the following table shows. Therefore, it keeps working until the end of period $t = 1$ and stops afterward (see Table 9.2).

Table 9.2: t=1.

Hub edge	$t = 1$	$t = 2$	$t = 3$	$t = 4$
$1 \in E^c$	**1**	**0**	**0**	**0**
$2 \in E^c$	1	0	0	0
$3 \in E^o$	0	1	1	1
$4 \in E^o$	0	0	0	1

$t = 2$

The same rule says that *"close it from now on"*. It stops working at the end of $t = 2$ (see Table 9.3).

Table 9.3: t=2.

Hub edge	$t = 1$	$t = 2$	$t = 3$	$t = 4$
$1 \in E^c$	**1**	**1**	**0**	**0**
$2 \in E^c$	1	0	0	0
$3 \in E^o$	0	1	1	1
$4 \in E^o$	0	0	0	1

$t = 3$

Again the same rule, *"close it from now on"*. This neighbor coincides with the solution itself (everybody, is neighbor of himself; see Table 9.4).

Table 9.4: t=3.

Hub edge	$t=1$	$t=2$	$t=3$	$t=4$
$1 \in E^c$	**1**	**1**	**1**	**0**
$2 \in E^c$	1	0	0	0
$3 \in E^o$	0	1	1	1
$4 \in E^o$	0	0	0	1

$t = 4$

This time because the edge is not active now, another rule says that: *"keep it open until now"* (see Table 9.5).

Table 9.5: t=4.

Hub edge	$t=1$	$t=2$	$t=3$	$t=4$
$1 \in E^c$	**1**	**1**	**1**	**1**
$2 \in E^c$	1	0	0	0
$3 \in E^o$	0	1	1	1
$4 \in E^o$	0	0	0	1

Openable Hub Edge Facility

The hub edge $3 \in E^o$ is an openable hub edge and as the following table shows it is opened at the beginning of the period 2 and remains open until the end of the planning horizon (see Table 9.6).

Table 9.6: A given solution.

Hub edge	$t=1$	$t=2$	$t=3$	$t=4$
$1 \in E^c$	1	1	1	0
$2 \in E^c$	1	0	0	0
$3 \in E^o$	**0**	**1**	**1**	**1**
$4 \in E^o$	0	0	0	1

Again analogously, we have the following neighbors with respect to this hub edge.

$t = 1$

The edge 3 is openable and closed at this period. Thus, the rule is applied and the following table is achieved (see Table 9.7).

Table 9.7: t=1.

Hub edge	$t = 1$	$t = 2$	$t = 3$	$t = 4$
$1 \in E^c$	1	1	1	0
$2 \in E^c$	1	0	0	0
$3 \in E^o$	**1**	**1**	**1**	**1**
$4 \in E^o$	0	0	0	1

$t = 2$

In $t = 2$ since the facility is active now, then the rule *"keep it closed until now"* closes the hub edge (see Table 9.8).

Table 9.8: t=2.

Hub edge	$t = 1$	$t = 2$	$t = 3$	$t = 4$
$1 \in E^c$	1	1	1	0
$2 \in E^c$	1	0	0	0
$3 \in E^o$	**0**	**0**	**1**	**1**
$4 \in E^o$	0	0	0	1

$t = 3$

The facility is opened once before, so according to the rule *"keep it closed until now"*, it should be kept closed until now (see Table 9.9).

Table 9.9: t=3.

Hub edge	$t=1$	$t=2$	$t=3$	$t=4$
$1 \in E^c$	1	1	1	0
$2 \in E^c$	1	0	0	0
$3 \in E^o$	**0**	**0**	**0**	**1**
$4 \in E^o$	0	0	0	1

$t = 4$

Again the same rule as in $t = 3$ is applied, *"keep it closed until now"* (see Table 9.10).

Table 9.10: t=4.

Hub edge	$t=1$	$t=2$	$t=3$	$t=4$
$1 \in E^c$	1	1	1	0
$2 \in E^c$	1	0	0	0
$3 \in E^o$	**0**	**0**	**0**	**0**
$4 \in E^o$	0	0	0	1

9.2 Computational Results

It is observed that for the variants of MPHLPPT, problem instances up to size 10 can be solved to optimality in a reasonable amount of time. For the instances of size 15, almost all initial configurations that we have employed, could not reach small gap solutions in less than half a day. In addition, our heuristic proves to be capable of finding the optimal solution for instances of size 5 for almost any given initial configuration that we have examined.

In this section we are going to solve instances of MPHLPPT for a variety of given and distinct initial configurations by our heuristic and compare results with those of `CPLEX 9.1`. Both variants of restrictions on the number of facilities to be established (CN) and the budget constrained are considered (CB). Table 9.11 and 9.12 report the computational results of CN and CB, respectively.

Instances of MPHLPPT for both CN and CB, are solved by `CPLEX 9.1` to solutions with qualities similar to those of our heuristic.

Table 9.11: Constraints on number of nodes (CN).

CN	CPLEX 9.1	Heuristic	
	T_c. $Cpu(s)$	T_1. $Cpu(s)$	Gap(%)
{1-4}	44.78	1.11	2.3
{3-5},{5-8}	55.52	1.19	0.1
{1-6},{6-9}	93.41	0.95	2.4
{0-4},{4-8}	30.67	1.19	0.7
{1-5},{1-8}	63.99	1.02	1.6
{3-5},{5-7},{7-8}	24.42	1.30	1.1
{2-7},{0-7},{2-9}	133.88	1.19	4
{0-3},{3-5},{1-5}	18.22	1.24	1.4
{1-4},{1-2},{4-9}	114.78	1.09	1.5
{1-7},{3-7},{3-9}	88.39	1.13	3.7
Avg.	66.80	1.14	1.88

From Table 9.11, one observes that, first of all, the quality of the solutions of our basic heuristic is quite good. Moreover, for a given solution quality, our heuristic outperforms `CPLEX 9.1` with respect to the computational time. This is also indicated in the last row Table 9.11. The average gap is 1.88% and in average such gap can be achieved by our heuristic around 59 times faster than `CPLEX 9.1`. This can be visualized in the Figure 9.1.

Again, for *CB*, our heuristic was capable of finding good solutions much faster than what `CPLEX 9.1` needed to find solutions with such quality. The average gap is satisfactory and our heuristic is around 62 times faster than `CPLEX 9.1` as it can be seen in Figure 9.2.

The results of Table 9.11 and 9.12 indicate that for the best-known solutions of our heuristic with the reported gaps in the last column, `CPLEX 9.1` finds solutions with the similar gaps in much higher amount of times.

Although the reported average gaps are satisfactory, they can be improved. In the next section we address this issue.

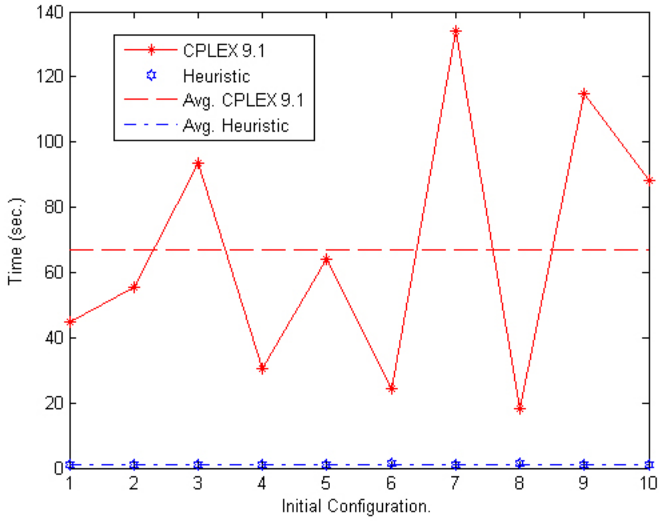


Figure 9.1: CN: Heuristic vs. CPLEX 9.1

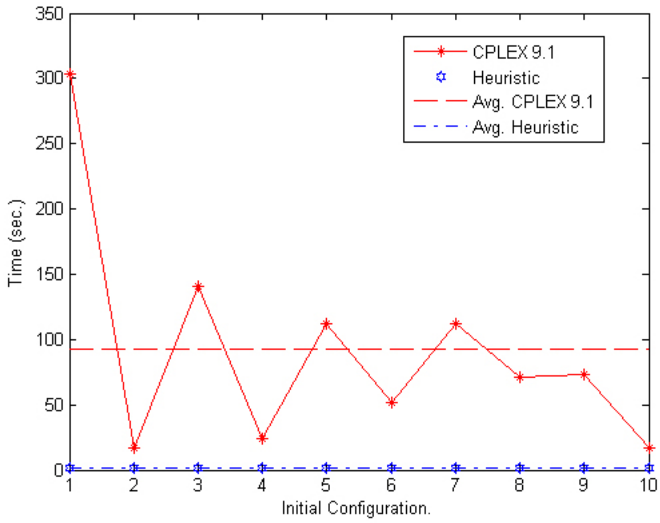


Figure 9.2: CB: Heuristic vs. CPLEX 9.1

9.3 Algorithm

The proposed basic algorithm for the multi-period problem is drawn in this section. We would like to remind that two models have been proposed in the previous chapter: *CB* and *CN*.

Both models share the same heuristic skeleton, except they have different *feasibility*

Table 9.12: Constraints by construction budget (CB).

CB	CPLEX 9.1	Heuristic	
	T_c. $Cpu(s)$	T_1. $Cpu(s)$	Gap(%)
{1-4}	303.93	1.34	5.7
{3-5},{5-8}	16.91	1.48	1.4
{1-6},{6-9}	140.55	1.74	0.1
{0-4},{4-8}	23.86	1.30	2
{1-5},{1-8}	112.13	1.31	1.6
{3-5},{5-7},{7-8}	51.44	1.64	1.3
{2-7},{0-7},{2-9}	111.83	1.53	6.7
{0-3},{3-5},{1-5}	71.63	1.81	0.6
{1-4},{1-2},{4-9}	73.41	1.45	4.4
{1-7},{3-7},{3-9}	16.81	1.42	7.4
Avg.	92.25	1.48	3.02

functions. In the first one, the feasibility function is in charge of checking the connectivity and bounds on the number of established facilities. In the latter case, it controls the connectivity of the hub-level network in each period as well as the violation of budget constraints.

9.4 Local Search

With the same arguments as for the single period planning, our observations from the output of the greedy heuristic; and considering the fact that multi-period approach makes the likelihood of premature convergence to a local optimum in a greedy algorithm much higher than in the single period case; we develop an approach in order to get rid of it, as much as possible.

9.4.1 Alternative Hub Edges

Out of visualizing the results, it has been observed that most of the time this local optimality is caused by the inappropriate establishment of some facilities in periods. That is, even though these facilities already exist in the optimal solution, they are not starting to work in the period which appears in the solution of greedy heuristic. At the same time, the optimal solution cannot be achieved by a single move from the current configuration based on the neighborhood rules and no more improvement is possible

Algorithm 12: A simple greedy algorithm for MPHLPPT

```
Input: instance and init_conf
Output: x*
x̄ := x_initcfg;
min := Eval(x̄);
last_min := ∞;
repeated_min := 0;
while (repeated_min == 0) do
    f̄ := Eval(x̄);
    if f̄ ≤ min then
        min := f̄;
        x* := x̄;
    end
    foreach t := 1  to  nrPeriods do
        foreach i := 1  to  nrLocations ∗ (nrLocations − 1)/2 do
            Δf := 0;
            x' := x̄;
            if i ∈ ClosableEdges then
                switch x'_i^t do
                    case 0:  x'_i^{t'} := 1      ∀t' ≤ t ;
                    case 1:  x'_i^{t'} := 0      ∀t' ≥ t;
                end
            else
                switch x'_i^t do
                    case 0  x'_i^{t'} := 1      ∀t' ≥ t ;
                    case 1  x'_i^{t'} := 0      ∀t' ≤ t;
                end
            end
            if  is_not_feasible(x') then
                Δf := ∞;
            else
                Δf := Eval(x') − min;
            end
            if Δf < 0 then
                x* := x' ;
                min := Eval(x');
            end

        end

    end
    if min = last_min then
        repeated_min := repeated_min + 1;
    end
    last_min := min;
    x̄ := x*;
end
stop.
```

(because it can not be achieved by a single move).

What can be beneficial is to create trajectories in the search space to find better solutions. This can be done by closing those hub edges which have been opened once in any period and trying to substitute them with those that have not been opened so far. That means, we give the chance to the spoke edges to become hub edges and then be subject to the original neighborhood search, hoping to find better solutions. This may help the new configuration to find such a trajectory. A slight difference is that as soon as the first improvement is visited, the search carries on from that point rather than waiting for the best choice.

Procedure

In Algorithm 13, the unfreezing process as explained earlier is displayed.

Algorithm 13: An improvement procedure for MPHLPPT

```
Input: y
Output: local_opt
local_opt := y;
min := ∞;
foreach i = 1  to  nrLocations − 1, j = i + 1  to  nrLocations do
    x := y;
    if hub edge i − j has been active once in a any period then
        x^t_ij := 0        ∀t;
        foreach p = i + 1  to  nrLocations do
            x^t_ip := 1        ∀t;
            x=Neighborhood_Search(x);
            if Eval(x) ≤ min then
                min := Eval(x);
                Local_Opt := x;
                y := x;
            end
        end
    else
        continue for the next hub edge;
    end
end
return local_opt.
```

Improvement Results

After the column of instance names, the first two columns in Table 9.13 and Table 9.14 are the result of heuristic before including additional improvement strategy. The second two are those after improvement and the last one is the `CPLEX 9.1` run-time for solutions with the smaller gaps as reported by heuristic after improvement.

CN-R10

After improvement, the heuristic found the optimal solution some instances. In average, the gap is below one percent which is halved. Figure 9.3 visualizes the results.

Table 9.13: Constraints on number of nodes (CN).

CN	T_1. *Cpu(s)*	Gap(%)	T_1'. *Cpu(s)*	Gap(%)	CPLEX 9.1
{1-4}	1.11	2.3	16.78	0.7	151.66
{3-5},{5-8}	1.19	0.1	11.03	0.1	55.52
{1-6},{6-9}	0.95	2.4	8.13	2.4	93.41
{0-4},{4-8}	1.19	0.7	10.95	0.7	30.67
{1-5},{1-8}	1.02	1.6	25.03	0.9	63.99
{3-5},{5-7},{7-8}	1.30	1.1	24.36	0.00	71.55
{2-7},{0-7},{2-9}	1.19	4	25.19	0_{opt}	161.86
{0-3},{3-5},{1-5}	1.24	1.4	27.41	0_{opt}	47.11
{1-4},{1-2},{4-9}	1.09	1.5	12.47	1.4	114.78
{1-7},{3-7},{3-9}	1.13	3.7	12.69	3.7	88.39
Avg.	1.14	1.88	17.40	0.99	87.89

R15

For the R15, our heuristic is capable of achieving a solution with gap of 2.00% in less than 120 seconds while `CPLEX 9.1` can not find a solution with gap of less than 3.07% in less than 40878.81 seconds (around 340 times later). According to our results, `CPLEX 9.1` needs 95513.47 seconds (more than 26 hours) of computational effort to solve this instance to optimality.

CB-R10

As one can see in the Table 9.14, after improvements one instance reached optimality and most of them possessed very small gaps. The average gap reported in our tables became smaller after improvements (almost halved). From our experience, the best-known solution of our heuristic strongly depends on the deviation between the spatial

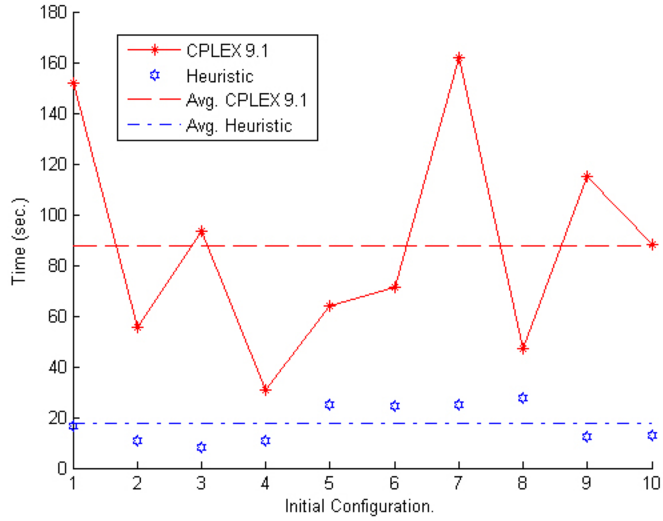


Figure 9.3: CN: Heuristic vs. CPLEX 9.1 after improvement.

layout of the hub-level network in the optimal solution and initial configuration of the instance. Figure 9.4 visualizes the results.

Table 9.14: Constraints on the amount of available budget (CB).

CB	T_1. $Cpu(s)$	Gap(%)	T_1'. $Cpu(s)$	Gap(%)	CPLEX 9.1
{1-4}	1.34	5.7	42.14	0_{opt}	801.28
{3-5},{5-8}	1.48	1.4	20.17	1.4	16.91
{1-6},{6-9}	1.74	0.1	24.03	0.1	140.55
{0-4},{4-8}	1.30	2	16.42	2	23.86
{1-5},{1-8}	1.31	1.6	31.72	1.1	132.57
{3-5},{5-7},{7-8}	1.64	1.3	57.28	1.2	58.17
{2-7},{0-7},{2-9}	1.53	6.7	48.72	2.2	137.08
{0-3},{3-5},{1-5}	1.81	0.6	15.80	0.6	71.63
{1-4},{1-2},{4-9}	1.45	4.4	83.22	0.4	113.85
{1-7},{3-7},{3-9}	1.42	7.4	19.44	7.4	16.81
Avg.	1.48	3.02	35.89	1.64	151.27

R15

Again, for the R15 with the budget constraints, by means of our heuristic we could obtain a solution with the gap of 5.8% after 1104.60 seconds while a solution with such

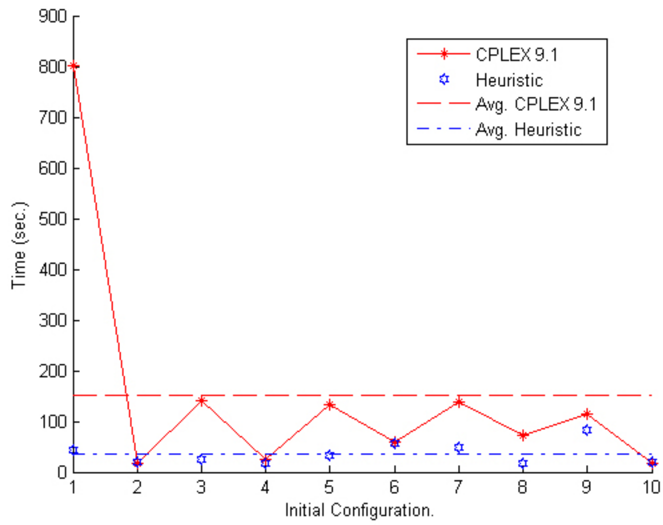


Figure 9.4: CB: Heuristic vs. CPLEX 9.1 after improvement.

a gap could not be found in less than 3383.39 seconds (3.06 times faster) in `CPLEX 9.1`.

9.4.2 Improved Algorithm

Algorithm 14 depicts the improvement of Algorithm 12 by including additional local search strategy.

9.5 A Larger Scale Instance

In this section we are going to solve larger instances of MPHLPPT for which no optimal solution is available. This gives us an imagination of the run-time of our heuristic.

A set of randomly generated instances of size 40 with $T = 3, 6, 9$ and 12 are solved by our heuristic. The maintenance and ceasing costs are also considered in addition to the setup costs. Furthermore, the interest rate is set to $\alpha = 1.2$ in the budget constrained variant and the maximum number of hub facilities that can be setup in each period of *CN* is restricted to 3.

The initial configuration of our random instance of size 40 is depicted in Figure 9.5.

The run-time reported for this instance for both CN and CB are depicted in Figure 9.6

Algorithm 14: Improved greedy algorithm for MPHLPPT

```
Input: instance and init_conf
Output: x*
x̄:= x_initcfg;
min := Eval(x̄);
last_min := ∞;
repeated_min := 0;
while (repeated_min == 0) do
    f̄ := Eval(x̄);
    if f̄ ≤ min then
        min := f̄;
        x* = x̄;
    end
    foreach t = 1 to nrPeriods do
        foreach i = 1 to nrLocations * (nrLocations − 1)/2 do
            Δf := 0;
            x' := x̄;
            if i ∈ ClosableEdges then
                switch x'_i^t do
                    case 0: x'_i^t' = 1          ∀t' ≤ t ;
                    case 1: x'_i^t' = 0          ∀t' ≥ t;
                end
            else
                switch x'_i^t do
                    case 0  x'_i^t' = 1          ∀t' ≥ t ;
                    case 1  x'_i^t' = 0          ∀t' ≤ t;
                end
            end
            if is_not_feasible(x') then
                Δf := ∞;
            else
                Δf := Eval(x') − min;
            end
            if Δf < 0 then
                x* := x' ;
                min := Eval(x');
            end

        end

    end
    if min = last_min then
        repeated_min := repeated_min + 1;
        if repeated_min = 2 then
            goto 61;
        end
        if repeated_min = 1 then
            x̄= Alternate(Local_Opt);
            if Eval(x̄) ≤ min then
                min := Eval(x̄);
                Local_Opt := x;
                repeated_min := 0;
            else
                min := last_min;
            end
            goto 39;
        end
    end
    last_min := min;
    x̄ = x*;
end
stop.
```

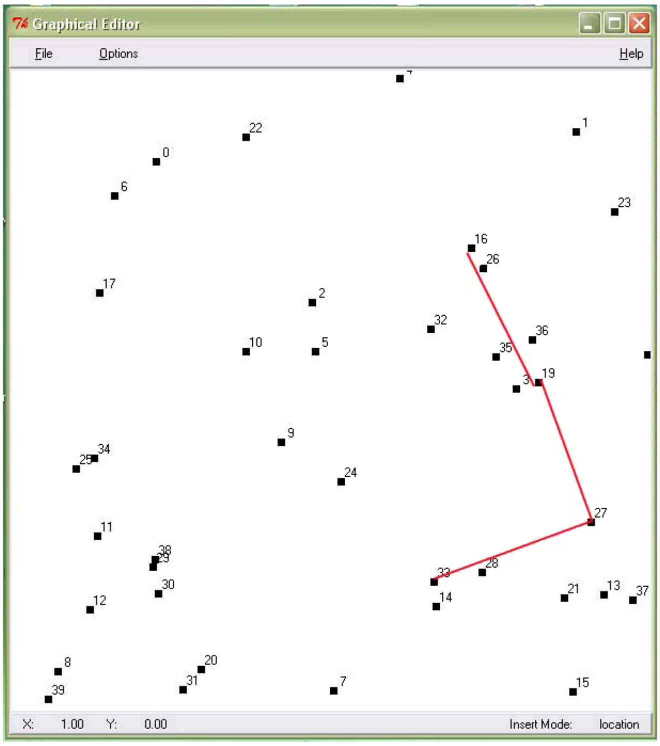


Figure 9.5: Initial configuration of N=40

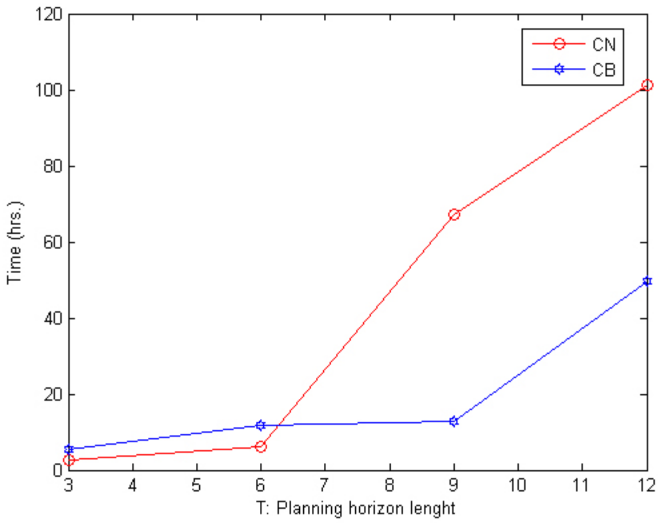


Figure 9.6: Run-time in relation to the number of time periods.

Chapter 10

Summary, Conclusions and Outlook to Future Work

In this chapter we finalize our dissertation by summarizing the work which was carried out and lightening new possible avenues deserving more attention and study.

10.1 Summary and Conclusions

In this dissertation, we started with the single period approach and proposed a mixed integer programming model for the application of Hub Location Problems in Public Transport planning (HLPPT) [34].

This model worked as a basis for our extension to models reflecting more features of real-life applications. The reported computational results in the corresponding chapters substantiate the superiority of our base model to the existing ones in literature for this application.

The problem is known to be NP-hard. Existing standard solvers have faced with some restrictions to solve instances of even small size. For example, for $n \geq 35$, the computer ran out of memory. Therefore, the motivation of applying new solution strategies by exploiting the special structure of the model to solve the instances more efficiently was raised.

Among these solution strategies, we started with the Lagrangian decomposition and showed that for the designated sub-problem the integrality property prevents us from obtaining the desired results.

We proposed some successful variants of the classical Benders decomposition algorithm transcending the standard solvers, like `CPLEX 9.1`, especially if equipped with a pareto-optimal cut strategy. This superiority is not only in terms of instance sizes which can be solved but also regarding the computational time.

By means of some classes of valid inequalities, the formulation of the problem is so tight that the LP relaxation becomes a good approximation of MIP. Using this formulation all instances that are solvable by `CPLEX 9.1` are solved at the root node and in only one case the optimum is found at the first node of branching tree. Therefore, we improved the performance of our Benders algorithm in such a way that it solves larger problems much faster. Specifically, a problem instance which in the earlier versions of our Benders algorithm is solved in more than 6 days, is now solvable (in the final version) in less than 4 hours.

A very successful (fast and efficient) heuristic that we called *greedy**, is proposed which is able to find the optimal solution for all of the instances for which the optimal solutions are known and the exact algorithms are not able to tackle larger instances. That is a greedy neighborhood search algorithm equipped with some improvement mechanisms.

We proposed the first multi-period HLP (MPHLPPT) model for this application. By taking into account the nature of construction projects and since we are faced with long-lasting and finance-demanding projects, multiple phase or multi-period planning models of the problem make more sense and are more practical and reliable. Therefore, we proposed a model that considers an initial configuration of the transportation network and aims to evolve the network according to the changes in the system parameters. This is accomplished by closing non-profitable and opening beneficial facilities and considering certain assumptions that permits us to change the status (open/close) of each facility at most once and incorporates the setup, maintenance and ceasing costs. We also considered the limitation on the facility establishment activities by means of restricting the number of facilities (CN) that can be established in each period or limiting the construction budgets (CB). In the case of CB, we also considered the fact that any capital available in a period but not invested then is subject to an interest rate and the returned value can be used for the construction in the successive periods.

We adapt our successful heuristic by modifying the moving rules and the improvement method to be capable of handling this new model. The results were quite satisfactory and promising. Again, for this problem, instances of even very small size (like 15 or more) could not be solved to optimality in a reasonable amount of time by means of standard solvers.

A very important aspect of any model is its solvability. It has been shown that our HLPPT prepares a good basis for exact decomposition and heuristics methods to be applied on.

The Benders decomposition strategy is shown to be very promising. At the same time, the Lagrangian algorithm is not promising in the case of our model.

The heuristics in both models, single and multi-period, are shown to be very effective and promising.

10.2 Future Work

This work covers the application of HLPs in public transport planning. It proposes a strong, realistic and flexible model which even can be employed in other applications like telecommunication. Some important facts of real-life are incorporated into the model. The model is shown to be a very suitable basis for applying decomposition algorithms as well as heuristics.

However, there are still some open topics to work on. The first one is to incorporate capacity and scheduling policies into the model. This can lead to models which reduce the likehood of congestion in the system. Interaction and trade-off between the economy of scale and congestion caused by the accumulation of flows, sensitivity analysis of system parameters (e.g. rate of economy of scale), making use of a piecewise linear cost functions, service-level and performance analysis, incorporation of stochastic and statistical parameters etc., are among the areas deserving more attention.

In a citywide network, different transportation facilities with different rate of scale economy such as tramway, subways, buses, aeronautic urban transport with helicopters, taxis, etc. can also be considered (intermodal transportation in conjunction with the hub location problems).

Construction of new track, is very expensive and hardly possible in many cities. The capacity of the existing network must therefore be better utilized to meet the customer demand with an enlarged offer. This avenue deserves some studies in an HLP framework in conjunction with multi-period planning.

In a countrywide or continental-wide context, for example in Europe, the rail transportation plays a very important role in comparison with other continents. The railway traffic in these countries has increased considerably for both passenger and freight transportation during the last few years. This trend is expected to continue, especially due to the extension of the Schengen territory. Therefore, revision of the already existing networks to fit in the framework of HLPs can be an interesting area of study.

In addition to the existing fast-line network in the European Union, the newly joined authorities in Schengen territory must connect their networks to this existing fast-line network. This is again an additional source of potential problems and conflicts. This includes cases such as the neighborhood of competing hub nodes and overlapping servicing area for different hub nodes. Therefore, the study of such pairing strategies can be very useful.

Usually, in such a public system the main goal is to offer service rather than making revenue. Therefore, the study of such systems by considering both costs and revenue may have less priority versus other objectives. Such objectives can come from political, economic, cultural and geographical issues that can be taken into account to make a trade-off between them. Thus, the multi-criteria aspects of the real-life application could also be taken into account. Such studies can also consider the inclusion of other objectives such as customer-satisfaction or service-level issues.

More efficient solution procedures (exact and heuristic) should always be studied. In contrast with the general purpose solvers, these approaches exploit special structure of problems (in fact, the models without solution methods except for depicting the complexities of problems almost have no other benefits). Very often, these problems are modeled as MIPs and belong to the class of NP-hard problems. The solution procedures like stand-alone dual ascent which is not using any external LP solvers and at the same time deals with lower and upper bounds are very attractive methods in these areas. However, other decomposition approaches are highly suggested and expected to be very successful.

Efficiency of exact solution approaches highly depends on the knowledge of polyhedral structure of the problems. This avenue can lead to drastic progress in the solution strategies. The problem dependent branch-and-bound, branch-and-cut and relax-and-cut procedures are candidates for receiving the most benefits from such studies, even in presence of hardware restrictions as a permanent obstacle.

Heuristic algorithms which are simultaneously dealing with both the lower and upper bounds deserve more attention. However, other upper bound heuristics are also very welcome.

Due to the complexities of HLPs, very often only concise, strong and simple heuristics are the best. Incorporation of any extra complexity may not lead to a very efficient solution procedure. Our experiences in applying genetic algorithms (GA) on instances of HLPPT revealed that, due to complexity of GAs (regarding the parameter tuning and so on), they have not been more successful than our greedy algorithm. Our experiences with a very simple and basic simulated annealing method was not very successful either.

However, incorporating additional components like tabu lists and different cheap intensification and diversifications can be also interesting. These heuristics use the best encoding of the problems and work only on a very small set as representative.

In our model, the search is only restricted to a set of hub edge arrays. The moving rules should be introduced concisely and at the same time, very easy to implement (regarding the complexity of data structures), because the cardinality of the feasible space in this problem is very large.

Moreover, since solutions with small gap might not be found in a reasonable amount of time in standard solvers, solutions of heuristics can play important roles. These heuristics are capable of finding good upper bounds which can be used in a branch-and-bound, to generate cuts for the standard solvers or even in other solution procedures. This helps to accelerate the resolution by cutting off those parts of feasible space not containing optimal solution. Such solutions may be found by heuristics in a small fraction of the time that the solver or solution procedure might need.

Chapter A

The authors scientific career

Studies

Since May. 2008	Postdoctoral research fellow at Civil Engineering Department of National University of Singapore (NUS), Singapore, Title of the project: *Analysis of Optimal Containership Size and its Impact on Liner Shipping Operations*,
Jly. 2006 - Apr. 2008	PhD study in Mathematics at the Technical University of Kaiserslautern, Germany, Title of the PhD thesis: *Hub Location Models in Public Transport Planning*,
Jan. 2005 - May. 2005	Lecturer in Mathematics at Azad University, Iran,
Sep. 2003 - Apr. 2005	Master study in Applied Mathematics (Operations Research) at the University Sistan and Baluchstan, Iran, Title of the master thesis: *On Solving Quadratic Assignment Problem: Heuristic Methods and Soft-Computing Approaches*,
Sep. 1999 - Sep. 2003	Bachelor study in Applied Mathematics at the Ferdowsi University of Mashhad, Iran, Title of the bachelor thesis: *Web-Based Discrete-Event Simulation of Queues.*

Chapter B

Scientific Carrier

B.1 Publication

The following publications contain parts of this thesis or precursory work:

[33] Gelareh, S. and S. Nickel. Multi-Period Public Transport Design: A Novel Model and Solution Approaches. Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), D 67663 Kaiserslautern, Germany, 2008.

[34] Gelareh, S. and S. Nickel. New Approaches to Hub Location Problems in Public Transport Planning. Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), D 67663 Kaiserslautern, Germany, 2008.

[32] Gelareh, S. and S. Nickel. A Benders Decomposition for Hub Location Problems Arising in Public Transport, *Proceeding of GOR2007*, 2007.

[31] Gelareh, S. and H.M. Nehi. Survey of Meta-Heuristic Solution Methods for the Quadratic Assignment Problem. *Applied Mathematical Sciences*, 1(46), 2293 - 2312, 2007.

B.2 Talks

The following talks are presented in scientific meetings and conferences:

B.2.1 Invited

Université Libre de Bruxelles, November 8, 2007, Brussels, Belgium

Hub Location Workshop 2007, September 8, Trippstadt, German

B.2.2 Conferences

German Operation Research Conference (GOR07), 5^{th} - 7^{th} September 2007, Saarbrücken, Germany

15^{th} *Seminar of Mathematical Analysis*, 10^{th} - 11^{th} March 2005, Zahedan, Iran,

35^{th} *Iranian Mathematical Conference (AIMC35)*, 27^{th} - 29^{th} February 2005, Ahvaz, Iran,

6^{th} *Iranian Conference of Intelligent Systems*, 5^{th} - 7^{th}, December 2004, Kerman, Iran

7^{th} *Iranian Statistical Conference*, 23^{rd} - 25^{th}, August 2004, Tehran, Iran,

Bibliography

[1] S. Abdinnour-Helm. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operational Research*, 106(2–3):489–499, 1998.

[2] S. Abdinnour-Helm. Using simulated annealing to solve the p-hub median problem. *International Journal of Physical Distribution & Logistics Management*, 31 (3):203–220, 2001.

[3] S. Abdinnour-Helm and M.A. Venkataramanan. Solution approaches to hub location problems. *Annals of Operations Research*, 78:31–50, 1998.

[4] S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 2007.

[5] R. Aversa, H.E. Botter, R.C. Haralambides, and H.T.Y. Yoshizaki. A mixed integer programming model on the location of a hub port in the east coast of south america. *Maritime Economics and Logistics*, 7:1–18, 2005.

[6] T. Aykin. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79 (3):501–523, 1994.

[7] N. Boland, M. Krishnamoorthy, A.T. Ernst, and J. Ebery. Preprocessing and cutting for multiple allocation hub location problems. *European Journal of Operational Research*, 155 (3):638–653, 2004.

[8] J.F. Campbell. Location and allocation for distribution systems with transshipments and transportation economies of scale. *Annals of Operations Research*, 40:77–99, 1992.

[9] J.F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72, 1994b.

[10] J.F. Campbell. Hub location and the p-hub median problem. *Operations Research*, 44 (6):1–13, 1996.

[11] J.F. Campbell. *Strategic network design for motor carriers. Logistics Systems: Design and Optimization*, chapter 6. Springer, 2005.

[12] J.F. Campbell, A.T. Ernst, and M. Krishnamoorthy. *Hub location problems*, chapter 12. Springer, 2002.

[13] J.F. Campbell, A.T. Ernst, and M. Krishnamoorthy. Hub arc location problems: Part i – introduction and results. *Management Science*, 51 (10):1540–1555, 2005a.

[14] J.F. Campbell, A.T. Ernst, and M. Krishnamoorthy. Hub arc location problems: Part ii – formulations and optimal algorithms. *Management Science*, 51 (10):1556–1571, 2005b.

[15] J.F. Campbell, G. Stiehr, A.T. Ernst, and M. Krishnamoorthy. Solving hub arc location problems on a cluster of workstations. *Parallel Computing*, 29 (5):555–574, 2003.

[16] L. Cánovas, S. García, and A. Marín. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. *European Journal of Operational Research*, 179:990–1007, 2007.

[17] G. Carello, F. Della Croce, M. Ghirardi, and R. Tadei. Solving the hub location problem in telecommunication network design: A local search approach. *Networks*, 44 (2):94–105, 2004.

[18] S. Cetiner, C. Sepil, and H. Sural. Hubbing and routing in postal delivery systems. Technical report, Industrial Engineering Department, Middle East Technical University, 06532 Ankara, Turkey., 2006.

[19] J.F. Chen. A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega*, 35:211–220, 2007.

[20] J.F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32 (4):380–404, 1998.

[21] M.G. Costa, M.E. Captivo, and J. Climaco. Capacitated single allocation hub location problem – a bi-criteria approach. *Computers and Operations Research*, in press, 2007.

[22] T.G. Crainic and G. Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97:409–438, 1997.

[23] C.B. Cunha and M.R. Silva. A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. *European Journal of Operational Research*, 179:747–758, 2007.

[24] J. Ebery. Solving large single allocation p-hub problems with two or three hubs. *European Journal of Operational Research*, 128 (2):447–458, 2001.

[25] J. Ebery, M. Krishnamoorthy, A. Ernst, and N. Boland. The capacitated multiple allocation hub location problem: Formulations and algorithms. *European Journal of Operational Research*, 120:614–631, 2000.

[26] S. Elhedhli and F.X. Hu. Hub-and-spoke network design with congestion. *Computers and Operations Research*, 32:1615–1632, 2005.

[27] A.T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4 (3):139–154, 1996.

[28] A.T. Ernst and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, 104:100–112, 1998a.

[29] A.T. Ernst and M. Krishnamoorthy. An exact solution approach based on shortest-paths for p-hub median problems. *Informs Journal on Computing*, 10 (2):149–162, 1998b.

[30] A.T. Ernst and M. Krishnamoorthy. Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research*, 86:141–159, 1999.

[31] S. Gelareh and H. M. Nehi. A survey of meta-heuristic solution methods for the quadratic assignment problem. *Applied Mathematical Sciences*, 1(46):2293 – 2312, 2007.

[32] S. Gelareh and S. Nickel. A benders decomposition for hub location problems arising in public transport. In *Proceeding of GOR2007*, 2007.

[33] S. Gelareh and S. Nickel. Multi-period public transport planning: A model and greedy neighborhood heuristic approaches. Technical report, Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), D 67663 Kaiserslautern, Germany, 2008.

[34] S. Gelareh and S. Nickel. New approaches to hub location problems in public transport planning. Technical report, Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), D 67663 Kaiserslautern, Germany, 2008.

[35] A.J. Goldman. Optimal location for centers in a network. *Transportation Science*, 3:352–360, 1969.

[36] M. Guignard. Lagrangian relaxation. *TOP*, 11(2):151–228, 2003.

[37] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.

[38] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.

[39] R.W. Hall. Configuration of an overnight package air network. *Transportation Research A*, 23:139–149, 1989.

[40] H.W. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn. Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, 145 (1):104–116, 2004.

[41] M.W. Horner and M.E. O'Kelly. Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, 9(4):255–265, 2001.

[42] A.V. Iyer and H.D. Ratliff. Accumulation point location on tree networks for guaranteed time distribution. *Management Science*, 36(8):958–969, 1990.

[43] P. Jaillet, G. Song, and G. Yu. Airline network design and hub location problems. *Location Science*, 4 (3):195–212, 1996.

[44] S.J. Jeong, C.G. Lee, and J.H. Bookbinder. The european freight railway system as a hub-and-spoke network. *Transportation Research Part A*, 41 (6):523–536, 2007.

[45] B.Y. Kara. *Modeling and analysis of issues in hub location problems*. PhD thesis, Bilkent University Industrial Engineering Department, 06800 Bilkent, Ankara, Turkey, 1999.

[46] J.G. Klincewicz. Heuristics for the p-hub location problem. *European Journal of Operational Research*, 53:25–37, 1991.

[47] J.G. Klincewicz. Avoiding local optima in the p-hub location problem using tabu search and grasp. *Annals of Operations Research*, 40:283–302, 1992.

[48] J.G. Klincewicz. A dual algorithm for the uncapacitated hub location problem. *Location Science*, 4 (3):173–184, 1996.

[49] J.G. Klincewicz. Hub location in backbone/tributary network design: A review. *Location Science*, 6:307–335, 1998.

[50] J.G. Klincewicz. Enumeration and search procedures for a hub location problem with economies of scale. *Annals of Operations Research*, 110:107–122, 2002.

[51] M.J. Kuby and R.G. Gray. The hub network design problem with stopovers and feeders: The case of federal express. *Transportation Research A*, 27 (1):1–12, 1993.

[52] M. Labbé and H. Yaman. Projecting flow variables for hub location problems. *Networks*, 44 (2):84–93, 2004.

[53] M. Labbé, H. Yaman, and E. Gourdin. A branch and cut algorithm for the hub location problems with single assignment. *Mathematical Programming*, 102:371–405, 2005.

[54] N. Maculan, G. Plateau, and A. Lisser. Integer linear models with a polynomial number of variables and constraints for some classical combinatorial optimization problems. *Pesquisa Operacional*, 23(1):161 – 168, 01 2003.

[55] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[56] A. Marín. Formulating and solving splittable capacitated multiple allocation hub location problems. *Computers & OR*, 32 (12):3093–3109, 2005a.

[57] A. Marín. Uncapacitated euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. *Journal of Global Optimization*, 33:393–422, 2005b.

[58] A. Marín, L. Cánovas, and M. Landete. New formulations for the uncapacitated multiple allocation hub location problem. *European Journal of Operational Research*, 172 (1):274–292, 2006.

[59] G. Mayer and B. Wagner. Hublocater: An exact solution method for the multiple allocation hub location problem. *Computers & OR*, 29:715–739, 2002.

[60] M. T. Melo, S. Nickel, and F. Saldanha da Gama. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Comput. Oper. Res.*, 33(1):181–208, 2006.

[61] S. Nickel, A. Schöbel, and T. Sonneborn. *Hub location problems in urban traffic networks*, chapter 1. Mathematics Methods and Optimization in Transportation Systems. Kluwer Academic Publishers., 2001.

[62] M.E. O'Kelly. The location of interacting hub facilities. *Transportation Science*, 20 (2):92–105, 1986a.

[63] M.E. O'Kelly. Activity levels at hub facilities in interacting networks. *Geographical Analysis*, 18 (4):343–356, 1986b.

[64] M.E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 1987.

[65] M.E. O'Kelly. Hub facility location with fixed costs. *Papers in Regional Science*, 71 (3):293–306, 1992a.

[66] M.E. O'Kelly, D. Bryan, D. Skorin-Kapov, and J. Skorin-Kapov. Hub network design with single and multiple allocation: A computational study. *Location Science*, 4 (3):125–138, 1996.

[67] M.E. O'Kelly and D.L. Bryan. Hub location with flow economies of scale. *Transportation Research B*, 32 (8):605–616, 1998.

[68] M.E. O'Kelly and Y. Lao. Mode choice in a hub-and-spoke network: A zero-one linear programming approach. *Geographical Analysis*, 23:283–297, 1991.

[69] M.E. O'Kelly, D. Skorin-Kapov, and J. Skorin-Kapov. Lower bounds for the hub location problem. *Management Science*, 41 (4):713–721, 1995.

[70] H. Pirkul and D.A. Schilling. An efficient procedure for designing single allocation hub and spoke systems. *Management Science*, 44(12):235–242, 1998.

[71] W.B. Powell and Y. Sheffi. The load planning problem of motor carriers: Problem description and a proposed solution approach. *Transportation Research A*, 17:471–480, 1983.

[72] I. Racunica and L. Wynter. Optimal location of intermodal freight hubs. *Transportation Research Part B*, 39 (5):453–477, 2005.

[73] I. Rodriguez-Martin and J. J. Salazar-Gonzalez. Solving a capacitated hub location problem. *European Journal of Operational Research*, In Press.

[74] Jr. R.S. de Camargo, G. Miranda, and H.P. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research*, In Press.

[75] M. Sasaki and M. Fukushima. On the hub-and-spoke model with arc capacity constraints. *Journal of the Operations Research Society of Japan*, 46 (4):409–428, 2003.

[76] M. Sasaki, A. Suzuki, and Z. Drezner. On the selection of hub airports for the airline hub-and-spoke system. *Computers & OR*, 26:1411–1422, 1999.

[77] D. Skorin-Kapov and J. Skorin-Kapov. On tabu search for the location of interacting hub facilities. *European Journal of Operational Research*, 73:502–509, 1994.

[78] D. Skorin-Kapov, J. Skorin-Kapov, and M. O'Kelly. Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, 94:582–593, 1996.

[79] K. Smith, M. Krishnamoorthy, and M. Palaniswami. Neural versus traditional approaches to the location of interacting hub facilities. *Location Science*, 4 (3):155–171, 1996.

[80] J. Sohn and S. Park. Efficient solution procedure and reduced size formulations for p-hub location problems. *European Journal of Operational Research*, 108:118–126, 1998.

[81] C.S. Sung and H.W. Jin. Dual-based approach for a hub network design problem under non-restrictive policy. *European Journal of Operational Research*, 132:88–105, 2001.

[82] H. Topcuoglu, F. Corut, M. Ermis, and G. Yilmaz. Solving the uncapacitated hub location problem using genetic algorithms. *Computers & OR*, 32 (4):967–984, 2005.

[83] B. Wagner. An exact solution procedure for a cluster hub location problem. *European Journal of Operational Research*, 178:391–401, 2007.

[84] H. Yaman. Polyhedral analysis for the uncapacitated hub location problem with modular arc capacities. *SIAM Journal on Discrete Mathematics*, 19 (2):501–522, 2005.

[85] H. Yaman and G. Carello. Solving the hub location problem with modular link capacities. *Computers & OR*, 32 (12):3227–3245, 2005.

List of Algorithms

Index