

**MODELING AND HEURISTIC APPROACHES FOR THE HUB
COVERING PROBLEM OVER INCOMPLETE HUB
NETWORKS**

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Hatice Çalık

January, 2009

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

Assoc. Prof. Bahar Yetiş Kara (Advisor)

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

Assoc. Prof. Oya Ekin Karaşan (Co-advisor)

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

Asst. Prof. Mehmet Rüştü Taner

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

Dr. Banu Yüksel Özkaya

Approved for the Institute of Engineering and Science

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

ABSTRACT

MODELING AND HEURISTIC APPROACHES FOR THE HUB COVERING PROBLEM OVER INCOMPLETE HUB NETWORKS

Hatice Çalık

M.S. in Industrial Engineering

Advisors: Assoc. Prof. Bahar Yetiş Kara, Assoc. Prof. Oya Ekin Karaşan

January, 2009

Hubs are the accumulation points within the transportation and the telecommunication networks that collect and distribute the flow or data, which is originated from a starting point and needs to be transferred to a destination point. The main application areas of the hub location problem are airline systems, telecommunication network design and cargo delivery systems. In the literature, a common treatment of hub location problems is under the classification dating back to the location literature. In this classification, four different types are identified. Namely, the p-hub median problem, the hub location problem with fixed costs, the p-hub center problem, and the hub covering problem in the literature. In most of the hub location studies, the hub networks are assumed to be complete; however, the observations on the real life cases showed that this may not be the case. Therefore, in this thesis, we relax this assumption and focus on the single allocation version of the hub covering problem over incomplete hub networks. We propose two new mathematical formulations and a tabu search based heuristic algorithm for this problem. We perform several computational experiments on the formulations with the CAB data set from the literature and a larger scale network corresponding to the cities in Turkey. The results we obtained from our experimentations reveals that designing incomplete hub networks to provide service within a given service time bound is cost effective in accordance with designing complete hub networks.

Keywords: Hub location problem, hub covering problem, network design

ÖZET

EKSİKLİ ANA DAĞITIM ÜSSÜ (ADÜ) AĞLARINDA ADÜ KAPLAMA PROBLEMİ İÇİN MODELLEME VE SEZGİSEL YAKLAŞIMLAR

Hatice Çalık

Endüstri Mühendisliği Yüksek Lisans

Tez Yöneticileri: Doç. Dr. Bahar Yetiş Kara ve Doç. Dr. Oya Ekin Karaşan

Ocak, 2009

ADÜ'ler akışların toplandığı ve yayıldığı çoklu dağıtım sistemlerindeki özel tipteki merkezlerdir. ADÜ yer seçimi probleminin temel uygulama alanları havayolu sistemleri, haberleşme ağları tasarımı ve kargo taşıma sistemleridir. Literatürde ADÜ yer seçimi problemi dört farklı türde çalışılmıştır. Bunlar p-ADÜ ortanca, sabit maliyetli ADÜ yer seçimi, p-ADÜ merkez ve ADÜ kaplama problemleridir. ADÜ yer seçimi problemlerinin büyük bir bölümünde, tüm ADÜ'lerin birbirlerine doğrudan bağlı oldukları varsayılmaktadır. Ancak gözlemlerimiz bunun çoğu zaman doğru olmadığını göstermiştir. Bunun üzerine, biz bu varsayımı kaldırdık ve eksikli ADÜ ağlarında tekli atama kuralına bağlı ADÜ kaplama problemine odaklandık. Problemimiz için iki tam sayılı programlama modeli ve bir sezgisel algoritma önerdik. Sonrasında, literatürde sıkça kullanılan CAB ve daha büyük bir ağ olan Türkiye verilerini kullanarak modellerimizin ve algoritmamızın performansını test ettik. Elde ettiğimiz sonuçlarda, eksikli ADÜ ağları tasarlamamanın doğrudan bağlı ADÜ ağlarına oranla daha az maliyetli olduğunu gözlemledik.

Anahtar Kelimeler: ADÜ yer seçimi problemi, ADÜ kaplama problemi, ağ tasarımı

Acknowledgement

I would like to express my sincere gratitude to Assoc. Prof. Bahar Yetiş Kara and Assoc. Prof. Oya Ekin Karaşan for their attention, support, and valuable guidance throughout my study, as well as for their patience and insight.

I am indebted to my dissertation committee, Asst. Prof. Mehmet Rüştü Taner and Dr. Banu Yüksel Özkaya, for accepting to read and review this thesis and for their suggestions.

I would like to express my deepest gratitude to my family for their precious and perpetual love and support. I feel very lucky to have such a wonderful family.

I would like to send special thanks to my intimate friends Ece Demirci, Esra Koca, and Emre Uzun for their everlasting support and friendship. Furthermore, I want to thank to my office mates Efe Burak Bozkaya, Can Öz, Işık Öztürkeri, and Ersin Körpeoğlu for their patience and support during my study. I also wish to thank to my former office mates Tuğçe Akbaş and Sıtkı Gülten for their assistance and friendship.

I am grateful to all my friends from the department for providing me such a friendly environment and for their valuable friendship.

I am also thankful to TÜBİTAK for the financial support they provided during my research.

Finally, I would like to thank to all my friends who never stopped encouraging me on my way.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION.....	1
CHAPTER 2: LITERATURE REVIEW.....	8
2.1 The p-hub Median Problem.....	9
2.2 The Hub Location Problems with Fixed Costs.....	11
2.3 The p-hub Center Problem	13
2.4 The Hub Covering Problem.....	14
CHAPTER 3: MATHEMATICAL FORMULATIONS	16
3.1 New Model for the Hub Covering Problem over Incomplete Hub Networks.....	18
3.2 The Model with Path Variables Restricted to the Hub Network.....	23
CHATER 4: TABU SEARCH BASED HEURISTIC ALGORITHM	28
4.1. A Brief Description of Tabu Search.....	28
4.2. General Structure of the Proposed Heuristic Algorithm	29
4.3. Illustration of the Algorithm with Examples.....	38
CHAPTER 5: COMPUTATIONAL RESULTS	48
CHAPTER 6: CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS	60
BIBLIOGRAPHY	63

LIST OF FIGURES

Figure 1: The illustration of complete and incomplete hub networks	6
Figure 2: The illustration of the flow on the resulting network	7
Figure 3: Parameters of the models	18
Figure 4: Variables of the first model	19
Figure 5: Link variations on paths	22
Figure 6: The resulting network of a solution	26
Figure 7: The flowchart of the heuristic algorithm	37
Figure 8: The coverage of the hubs while obtaining hub set $\{1, 2, 7, 4, 8\}$	40
Figure 9: The coverage of the hubs while obtaining hub set $\{6, 7, 8\}$	41
Figure 10: The solution obtained after the nearest allocation	42
Figure 11: The solution with incomplete hub network	43
Figure 12: Infeasible solution obtained before the second process of Type I allocation	44
Figure 13: Feasible solution obtained after the second process of Type I allocation	44
Figure 14: The solution obtained after all non-hub nodes are allocated to hub node 3	45
Figure 15: The solution obtained after node 10 is allocated to hub node 5.....	46
Figure 16: The solution obtained after all non-hub nodes are allocated to hub node 5	46
Figure 17: The resulting hub network on TR data for $T=1840, 1820$, and 1770	59

LIST OF TABLES

Table 1: Test bed for the CAB data set	50
Table 2: Computational comparison of Model 1, Model 2 and the heuristic for $n=10$	52
Table 3: Computational comparison of Model 2 and the heuristic for $n=15$	54
Table 4: Computational comparison of Model 2 and the heuristic for $n=20$	56
Table 5: Computational results of the heuristic on the Turkish network	58

Chapter 1

INTRODUCTION

The hub location problem is a rather new research area, which is originally introduced by O'Kelly (1986a) together with real life examples. This problem arises especially in the transportation and telecommunication systems. Hubs are the accumulation points on the transportation and the telecommunication systems that collect and distribute the flow, data or demand, which is originated from a starting point and needs to be transferred to a destination point. In the hubs, the flows that originate from different origins are collected and they are grouped according to their destinations. The flows that have the same destination are put together and distributed to their destinations. Therefore, direct flows between origin and destination pairs are not allowed. This collection and distribution of the flow is referred to as 'hubbing' in the literature.

The main application areas of the hub location problem are airline systems, telecommunication network design and cargo delivery systems.

In the airline systems hubbing is commonly applied to the flight services. An airline company that provides service between certain number of cities cannot provide direct flights between each city pair since it requires a highly expensive airline network and generates traffic congestion. Instead, the airline companies allow indirect flights between the cities that

are too far from each other or have low volume of traffic. Therefore, service can be provided for all cities by using a smaller number of aircraft.

The telecommunication networks generally consist of two levels: the backbone network and the local access network. The components in the backbone network can be considered as the hubs. Each local access point sends and receives its messages via the backbone network. The message emanating from a local access point is initially sent to a device in the backbone network. After the destination of the message is characterized, it is sent to the destination point in the local access network. Therefore, the congestion and the high cost that will be caused by direct connections are eliminated.

In the cargo delivery systems, the packages taken from the customers are sent to the operation centers. In these operation centers, the parcels are combined according to their destinations. Then, each group of parcels is sent to its destination point from these operation centers. The operation centers in the cargo delivery systems are considered as hubs.

The first integer programming formulation of the hub location problem, which is a quadratic model, is presented by O’Kelly (1987). In this model, p nodes from given n nodes should be selected as hubs and the remaining nodes should be allocated to these selected hubs so that the total transportation cost is minimized. There are three main assumptions for this problem. The first one of these assumptions is that each node should be allocated to exactly one hub, which is also known as single allocation in the literature. The hub location problem can be classified into two types according to how the demand nodes are allocated to the hubs: Single allocation and multiple allocation. If a demand node is restricted to send its flow via a single hub, this version of the allocation is called single allocation. On the other hand, if any demand node may send its flow via more than one hub, then the multiple allocation comes into the picture. Note that the direct flow between non-hub nodes is generally not allowed in the hub location problems.

Moreover, in this model, it is assumed that there exists a discount factor $\alpha \in [0,1]$, that provides economies of scale on the inter-hub connections. The real life motivation of this assumption follows from the fact that the flow between hubs is sustained with faster and larger vehicles or aircrafts. Therefore, the transportation between hubs is less costly and requires relatively small amount of time.

Another assumption in the standard hub location problem is the completeness of the hub network, that is, each hub node has a direct connection to every other hub node.

The hub location problem is modeled as the p -hub median problem, the hub location problem with fixed costs, the p -hub center problem, and the hub covering problem in the literature. These models differ mainly on their objectives. In the p -hub median problem, p nodes from a given node set are chosen as hubs and the remaining nodes are allocated to these hubs and the objective of the problem is to minimize the total transportation cost. This problem is presented by O’Kelly (1987). O’Kelly (1992) introduces the hub location problem with fixed costs to the literature. In this problem, in addition to the transportation costs, fixed costs of operating hub facilities are also included in the objective function; the remaining restrictions and requirements are the same as the p -hub median problem. The p -hub center and the hub covering problems are proposed by Campbell (1994). The objective of the p -hub center problem is either to minimize the maximum transportation cost or the maximum travelling time between origin-destination pairs by selecting p hubs from a given node set and allocating the remaining nodes to these hubs. The hub covering problem has an inverse relationship to the p -hub center problem. The objective of the hub covering problem is to minimize the number of hubs to open such that the total transportation cost or the travelling time between each origin-destination pair does not exceed a specified value. The objective of the hub covering problems might also be to minimize the total cost of transportation and operating hub facilities or to maximize the number of nodes covered with a given number of hubs. The details of these problems can be found in the next chapter.

In most of the hub location studies the hub networks are assumed to be complete; however, the observations on the real life cases showed that this may not be the case. In almost all the networks, in order to reach some hub point from another hub point, usage of another hub point might be required. In this case another decision is called for, namely, the location of the links between hubs. When we think of the cargo delivery systems, sending separate trucks from an operation center to all other operation centers is quite costly in terms of transportation. Instead, forcing some trucks to visit more than one operation center decreases the total transportation cost considerably.

Incomplete hub network is a more obvious concern in airline and telecommunication systems. An airline company may schedule several flights from an airport to a large number of destinations. Assigning separate aircraft, and accordingly separate air staff for each hub not only causes congestion in airports and air networks, but also high operating and transportation costs to the company. As a result of this fact, most flights are over some other hubs (airports). Also in telecommunication systems, connecting all terminals directly is probably the most expensive way of providing good service to the users. Therefore, incomplete networks are the matters of real life problems, thus, of network design problems.

In this thesis, we study the single allocation hub covering problem over incomplete hub networks. In order to analyze the problem, we focused on the cargo delivery systems. In the cargo delivery systems, customer satisfaction is an important concern. Providing high quality service with minimum cost is crucial for customer satisfaction in the sector. The cargo delivery firms want to minimize their total operating and transportation costs. Other than the cost matter, in order to achieve customer satisfaction, the delivery time issue should also be considered by the cargo delivery firms. If the delivery time of the parcel is more essential for a customer, the customer will most probably choose the service provider that assures the shortest delivery time for the parcel even if that service is the most expensive one. Therefore, providing quick service for any distance is highly advantageous for the cargo delivery firms. With this motivation, the objective of the problem is to provide a high service level such that

each parcel is delivered within a specified amount of time with the minimum total transportation and operating costs.

The hub covering problem over incomplete hub networks involves the decision on the number of hubs, their locations, the network between them, and the allocation of the origins to these hubs so that the travelling time from any origin to any destination is within a specified bound. A pair of nodes is considered as covered if the travelling time or the travelling cost between them on the network constructed is within a specified value. The hub covering problem aims to find the minimum number of hubs to open from the node set so that each pair of nodes in the set is covered. When we consider covering in terms of travelling time, after deciding on the hubs to open, the travelling time from origin i to destination j will be the summation of the travelling time from origin i to the hub it is allocated to, discounted travelling time from the hub of origin i to the hub of destination j , and the travelling time between destination j and its hub. Since we require this summation to be within a certain value, we need to choose the hubs to open accordingly.

Let us consider a network with eight nodes; some of the nodes in this network will be chosen as hubs, say nodes 3, 4, 5, and 8 in Figure 1a. Then, the links to be established between these hubs will be decided up on and the remaining nodes will be assigned to these hubs. Since the aim is to keep the travelling time between any origin-destination pair within a specified value, all above decisions need to be taken accordingly. An illustration of the incomplete hub network can be seen in Figure 1b. When there exist direct connections between each hub pair, then the resulting hub network will be complete as in Figure 1a: however, if some hubs are not directly connected, then, the resulting hub network will be incomplete as in Figure 1b.

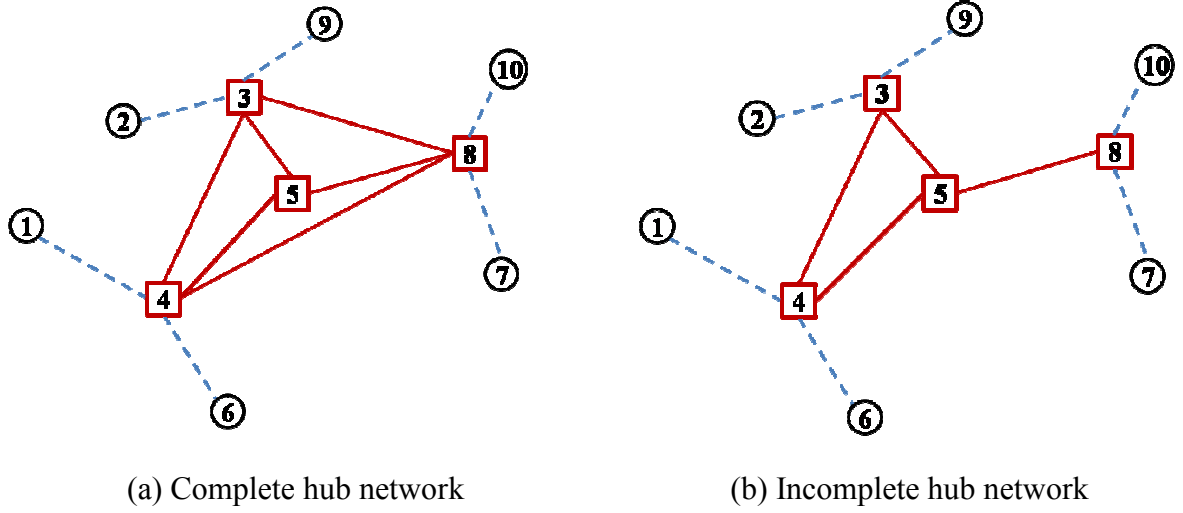


Figure 1: The illustration of complete and incomplete hub networks

When we consider the resulting network of a sample design in Figure 1b, the flow that originates from node 1 is initially sent to its operation center, node 4, and then depending on its destination point, it is either sent to another operation center or the destination point, which is assigned to the same hub with node 1. In Figure 2, one can see the path of the flow that originates from node 1 and arrives at node 7.

In this thesis, we propose two new mathematical formulations and a tabu search based heuristic algorithm for the single allocation hub covering problem over incomplete hub networks. We perform several computational experiments on the formulations with CAB data set and on the algorithm with both CAB data set and the Turkish Network.

In Chapter 3, we present a detailed formal description of the single allocation hub covering problem over incomplete hub network and we propose two new mathematical models for this problem.

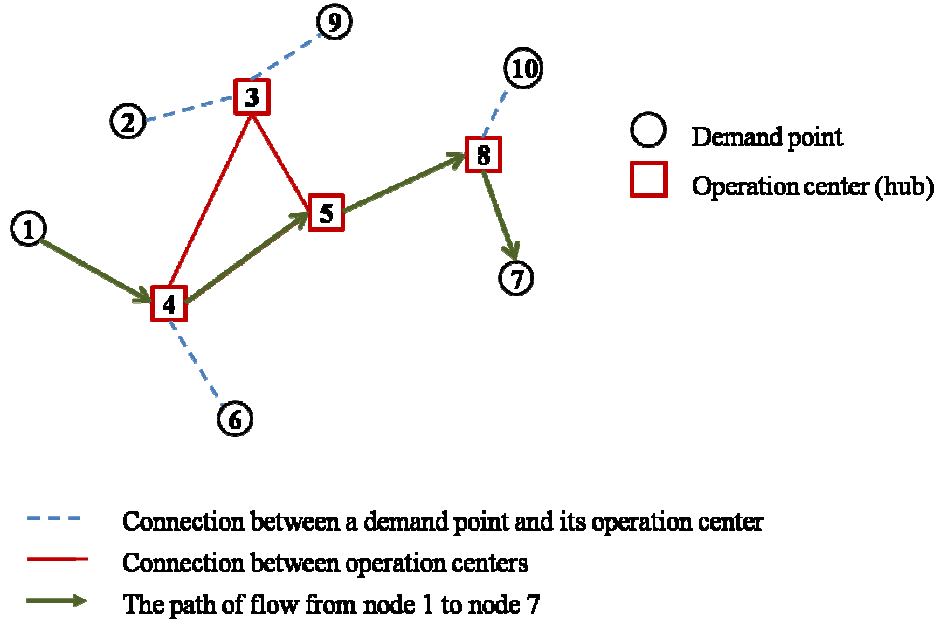


Figure 2: The illustration of the flow on the resulting network

The hub covering problem is an NP-Hard problem (Kara and Tansel, 2000); therefore, solving this problem to optimality for realistic sized instances is quite hard. In fact, even finding a feasible solution for this problem is rather challenging. Therefore, in order to be able to solve larger problems, we develop a tabu search based heuristic. Chapter 4 describes the heuristic algorithm we proposed in detail. In this chapter, we also provide a brief description of the tabu search algorithm and related literature.

Chapter 5 includes the results we obtained from the computational experiments conducted on the models and the algorithm by using the CAB data, as well as the results of the implementation of our algorithm on TR data which has 81 nodes. A detailed analysis of the results obtained from these experiments is also included in this chapter.

Finally, we discuss the results we obtained and the further possible improvement directions and research topics in the last chapter. In the next chapter, the hub location literature is analyzed comprehensively.

Chapter 2

LITERATURE REVIEW

The hub location problem involves the selection of the hubs and assignment of the origins to the selected hubs. The hub location problem is originally introduced by O'Kelly (1986a) together with real life examples. After this study, the single assignment hub location problem is described by O'Kelly (1986b). In a single assignment hub location problem, every node is required to be assigned to exactly one of the hubs chosen; thus, for a node, which is not a hub, the only emanating edge is the one that connects this node with its single hub.

The first integer programming formulation proposed for the hub location problem is a quadratic model (O'Kelly, 1987). Quite a while, the literature focused on the linearization of the quadratic model proposed in this paper. (Aykin (1995); Campbell (1996); A.T. Ernst and M. Krishnamoorthy (1996); O'Kelly et al. (1996); Skorin-Kapov et al. (1996)). In addition to the integer programming formulation, two heuristic approaches (HEUR1 and HEUR2) are presented O'Kelly (1987). HEUR1, which is also called as 'nearest hub allocation rule', basically investigates allocating each node to the nearest hub while HEUR2 investigates the idea of assigning each non-hub node to either its first or second nearest hub. These heuristics are generally quite effective in providing good upper bounds during complete enumeration of hub locations. One can refer to Campbell (1994b), Klineciewicz (1998), Campbell et al. (2002), and Alumur and Kara (2008) as the surveys on hub location problems.

O'Kelly (1987) introduces the Civil Aeronautics Board (CAB) data set data, which is based on airline passenger interaction between 25 U.S. cities, to the literature and uses the CAB data to test the performance of the heuristics proposed in this study. Ernst and Krishnamoorthy (1996) introduce the Australia Post (AP) data set, which consists of 200 nodes that represent postcode districts in Sydney, Australia. Unlike the CAB data set, the flow matrix of the AP data set is not symmetrical.

The hub location models studied in the literature commonly have analogous location versions. Throughout this thesis, we classify the hub location problems into four groups: the p-hub median problem, the hub location problem with fixed costs, the p-hub center problem and the hub covering problem according to their objectives. Campbell (1994a) provides the first linear integer programming formulation for the p-hub median problem together with mathematical formulations for the hub location problem with fixed costs, the p-hub center and the hub covering problem. Both single and multiple allocation cases are studied in this paper.

In the following sections, the hub location literature is analyzed in detail under the p-hub median problem, the hub location problem with fixed cost, the p-hub center problem, and the hub covering problem titles together with the definitions of the related problems.

2.1 The p-hub Median Problem

Considering the general setup of the hub location problems, n demand points (origins and destinations), the flow w_{ij} and the per unit transportation cost c_{ij} from origin i to destination j , and the discount factor α for hub-to-hub transportation are given. The per unit transportation cost from origin i to destination j via hubs k and m is denoted by $c_{ik} + c_{mj} + \alpha c_{km}$. Then the total transportation cost from origin i to destination j via hubs k and m is $w_{ij} \times (c_{ik} + c_{mj} + \alpha c_{km})$. The objective of the p-hub median problem is to minimize the total transportation

cost under specified hub location constraints, which force each node to be allocated to exactly one hub, restrict the number of hubs to be located to p , and avoid the direct connections between non-hub nodes. Therefore, p hubs need to be selected from given n nodes and the remaining nodes should be allocated to these selected hubs so that the total transportation cost is minimized. In this problem, the hub network is assumed to be complete.

The first linear integer programming formulation for the p-hub median problem is presented by Campbell (1994a). Both single and multiple allocation cases are studied in this paper. Skorin-Kapov et al. (1996) present new formulations for both single and multiple allocation p-hub median problems with tighter LP relaxations. The comparisons of these formulations with Campbell (1994a) show that for the single allocation case, less number of variables were used while for the multiple allocation case, less number of constraints were used to obtain tight LP relaxations.

Klincewicz (1991) develops the single exchange and the double exchange heuristics (exchange one or two hubs with one or two non-hub nodes) for the problem. Based on comparisons of these heuristics with clustering heuristics and enumeration heuristics on previous works in the literature, the double exchange heuristic seems quite promising for the solution of the p-hub median problem.

Campbell (1996) proposes mathematical formulations for both single allocation and multiple allocation p-hub median problems and examines the ways of solving single allocation problem by using the solutions obtained from multiple allocation problem. Inherent in this approach is the idea that the solution of multiple allocation is a lower bound for the single allocation problem. For this purpose, initially a greedy-interchange heuristic is developed to obtain multiple allocation solutions and then by using multiple allocation solution, two other heuristics are proposed to obtain solutions for the single allocation problem. The computational comparisons with Klincewicz (1991) show that these algorithms outperform the ones in Klincewicz (1991).

Skorin-Kapov and Skorin-Kapov (1994) propose a new heuristic based on tabu search algorithm. The computational comparisons in this study show that this algorithm performs better than heuristics provided by O'Kelly (1987).

Ernst and Krishnamoorthy (1996) provide a new formulation for single allocation p-hub median problem which requires less number of variables and constraints; thus, is able to solve the problems with larger size in shorter durations. The problem is formulated as a multicommodity flow problem, but no computational experiments are conducted for the model. They develop a heuristic based on simulated annealing which produces solutions, which are comparable in terms of both the quality and the computation time with the tabu search algorithm proposed by Skorin-Kapov and Skorin-Kapov (1994). By using the upperbounds obtained from this heuristic, they also develop a branch and bound algorithm. Both algorithms are tested on CAB data set and Australian Post (AP) data set.

Ernst and Krishnamoorthy (1998) develop an efficient heuristic algorithm based on shortest path, an explicit enumeration algorithm, and a linear mixed integer programming (MIP) for multiple allocation p-hub median problem. The algorithms use the idea that once the hub locations are decided, each pair sends its flow from the shortest path over the selected hubs. They also propose an LP based branch and bound algorithm and strengthen the lower bound by distinguishing valid inequalities and adding them to the LP.

2.2 The Hub Location Problems with Fixed Costs

The p-hub median problem aims to minimize only the transportation costs and does not take fixed cost of opening hub facilities into consideration. However, these fixed costs might be included in the objective function by defining a decision variable that represents the decision of opening hub facilities. In the p-hub median problem the number of hubs to open is fixed and given. However, in the hub location problems with fixed costs, the number of hubs

to open is not specified, it is decided by the model depending on the fixed costs of selecting a node as hub. Therefore, the model will decide the number of hubs to open, which nodes to choose as hubs, and the allocation of the non-hub nodes to the selected hubs such that the total transportation cost and the operating cost (cost of opening a hub) is minimized. O'Kelly (1992) introduces the single allocation version of this problem to the literature and develops a quadratic integer programming formulation. Campbell (1994a) provides the first linear programming formulations for both single and multiple allocation types of the problem as well as capacitated and uncapacitated versions. In the capacitated version of this problem, the capacity restrictions are on the flow carried by the links.

Abdinnour-Helm and Verkataramanan (1998) provide a new quadratic integer programming model which uses the idea of multicommodity flows and a branch and bound algorithm, in which the bounds are obtained by using the underlying network of the problem. In addition to the branch and bound algorithm, they also develop a genetic algorithm, which finds solutions quickly and efficiently. Abdinnour-Helm (1998) presents a new heuristic algorithm, which is a combination of genetic algorithm and tabu search. The algorithm locates the hubs by using genetic algorithm and construct solutions by nearest allocation, then; these solutions are improved by tabu search. The comparisons with genetic algorithm in Abdinnour-Helm and Verkataramanan (1998) show that this algorithm outperforms the one in Abdinnour-Helm and Verkataramanan (1998). Topçuoğlu et al. (2005) consider the single allocation hub location problem with fixed costs and develop a genetic algorithm for the problem. This algorithm performs better than the one in Abdinnour-Helm (1998) on the tests conducted over the CAB and the AP data sets.

Ernst and Krishnamoorthy (1999) study the capacitated version of the single allocation hub location problem with fixed costs and provide two new formulations for the problem. The capacity restrictions are on the amount of flows passing through hubs. In addition to the formulations for the problem, two heuristics based on single allocation and random descent are developed for obtaining upper bounds to be used in branch and bound method. The

computational experiments show that random descent based algorithm performs better in small and medium sized problems.

2.3 The p-hub Center Problem

The p-hub center problem is a minimax type problem and the objective might be either minimizing the maximum cost or the maximum travelling time between any origin destination pair. The center problems have important applications such as locating emergency service facilities and vehicles. When the objective of the p-hub center problem might be to minimize the maximum travelling time between each origin destination pair, the decisions of the problem are the locations of p hubs and the assignment of other nodes to these hubs so that the maximum travelling time between origin-destination pairs is minimized.

The first formulation for the p-hub center problem is proposed by Campbell (1994a). Although the original formulation is quadratic, a linearization of this model is also presented in the paper.

Kara and Tansel (2000) focus on the single allocation p-hub center problem and provide three different linearizations of the formulation in Campbell (1994a). They also include a new formulation for the p-hub center problem and the linearization of this formulation outperforms all the linearizations of the previous model of Campbell (1994a).

Pamuk and Sepil (2001) develop a single-relocation heuristic for the single allocation p-hub center problem and to avoid getting stuck at the local optima they adapt tabu search to their algorithm.

Ernst et al. (2002) focus on multiple allocation p-hub center problem and propose two new formulations together with a heuristic for both single and multiple allocation versions. In

addition, a branch and bound algorithm based on shortest path is developed for the multiple allocation p-hub center problem, and this algorithm is quite similar to the one in Ernst and Krishnamoorthy (1998).

2.4 The Hub Covering Problem

In covering problems, some cost or time parameters are restricted to a specified value due to the resource limitations or for customer satisfaction. Some variations of the hub covering problem might be minimizing the total cost under the restriction of the travelling time for any origin-destination pair, or minimizing the number of facilities opened by restricting the travelling cost of each origin-destination pair. The objective of the hub covering problem might be to minimize the number of hubs to open so that the total transportation cost is within a specific value. Then the model needs to decide the number and location of the hubs together with the allocation of non-hub nodes to these selected hubs. Moreover, the objective of the hub covering problem might also be to minimize the total cost as well. When we consider the cargo delivery systems, the firms might want to establish a network structure that will enable service for each origin-destination pair in certain time period, say 24 hours, with minimum total cost (cost of transportation and operating hubs).

The first MIP formulation for the hub covering problem is developed by Campbell (1994a), which mainly studies the hub set-covering problem and the maximal hub-covering problem. The hub set-covering problem locates hubs to cover all nodes such that the cost for operating hubs is minimized while the maximal hub-covering problem aims to maximize the number of nodes covered with given number of hubs. A node is covered if it is close enough to the hubs so that hubs can serve these nodes within specified parameters such as maximum delivery time.

Campbell (1994a) introduces the hub covering problem to the literature with a quadratic IP model, then he also develops linear models for the problem, but the first

computational results are presented by Kara and Tansel (2003). Kara and Tansel (2003) focus on the single allocation hub set-covering problem. They provide three different linearizations of the original quadratic model together with a new linear model and test the performance of these models on the CAB data. The comparisons on these models show that the new model outperforms the other linear models.

Wagner (2004) considers the single and multiple allocation hub covering problem and proposes new formulations for both of them as well as some preprocessing techniques. Since these techniques eliminate some hub assignments, the models require less number of variables and constraints than the one in Kara and Tansel (2003).

Ernst et al. (2005) studies the single and the multiple allocation hub set covering problem and provide a new formulation for the single allocation problem which is similar to the one presented in Ernst et al. (2002), and two new formulations for the multiple allocation together with an implicit enumeration method. The comparisons of the model for the single allocation hub set-covering problem with the one proposed by Kara and Tansel (2003) indicates that the formulation of Ernst et al. (2005) performs better in terms of CPU time.

Hamacher and Meyer (2006) consider different formulations of the hub covering problem and pinpoint some facet-defining valid inequalities. They also provide a solution methodology for the p-hub center problem that solves the hub set-covering problem for a cover radius T (T is the time bound for the hub-set covering problem) by iteratively reducing it.

In this thesis, we analyze the single allocation hub covering problem by relaxing the complete hub network assumption and we provide two new mathematical formulations, of which the objective function includes the fixed costs of opening hubs. Moreover, in order to be able to solve realistically sized problems, we develop an effective tabu search based heuristic algorithm and conduct several computational experiments on both the formulations and the algorithm.

Chapter 3

MATHEMATICAL FORMULATIONS

Most of the studies in the hub location literature concentrate on the p-hub median problem and there are only a few studies that focus on the hub covering problem. The application area of the hub covering problem is especially the cargo delivery systems and the objective of the problem is to select the minimum number of operation centers (hubs) such that the transportation time between each origin destination pair is within the specified time bound.

In this study, we focus on a different version of the hub covering problem. In our problem, not necessarily all the operation centers have direct connections between each other; therefore, some of the connections between operation centers can be established via other operation centers. Thus, the network between the hubs can be incomplete or complete depending on the need. We consider the single allocation case of the problem. Therefore, no direct connections between two demand points (non-hub nodes) are allowed and one demand point can be connected to exactly one operation center.

The motivation follows from the fact that for most of the customers of the cargo delivery systems rather than the cost of the carriage, the delivery time of the cargo is also important especially when the perishable items are considered. Therefore, from the cargo companies' point of view, establishing a transportation network, which aims to deliver any

item in certain amount of time period, is welcome since it increases the customer satisfaction and revenue accordingly. Furthermore, unlike the completeness assumption of the hub networks, the cargo companies prefer not to connect each operation center directly, but link some of them over other operation centers. Therefore, investigating the problem over incomplete hub networks is meaningful as well.

The mathematical description of the problem is as follows:

Let a node set N with n nodes and a potential hub set H (subset of N) with h nodes be given. The model selects some number of hubs from the potential hub set H , constructs the links between the hub nodes and allocates the remaining nodes in N to these selected hubs based on the single allocation requirements so that the time required to transport a unit flow from any origin to any destination is less than a predetermined time bound T and the total cost of opening hubs and establishing hub links is minimized.

We modeled the problem with two different approaches. The main difference of these two approaches is the way we define the four indexed decision variable of the models. In the second model, we restrict the four indexed decision variable defined on the hub network only while we do not impose such a restriction for the first model. Before observing the details of the models in the forthcoming sections, we define the parameters of the models as follows:

f_{km} : Fixed cost of opening a hub link between nodes $k \in H$ and $m \in H$

fh_k : Fixed cost of opening a hub at node $k \in H$

t_{ij} Travel time from node $i \in N$ to node $j \in N$

T : Given time bound

α : Economies of scale time discount factor for hub-to-hub connections

An important note here is that the time discount factor, $\alpha \in [0,1]$, is different than the cost discount factor; and it is expected to be close to 1. One can see the illustration of the parameters in Figure 3.

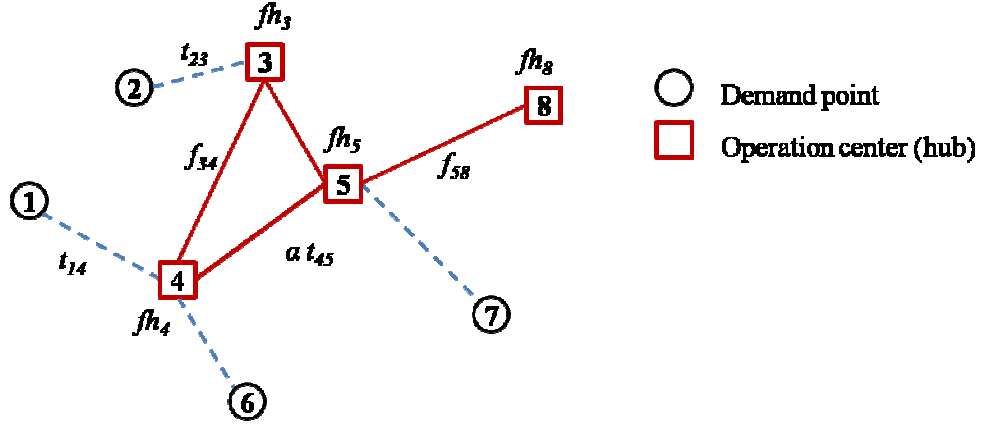


Figure 3: Parameters of the models

3.1. New Model for the Hub Covering Problem over Incomplete Hub Networks

We define the decision variables of the first model as follows:

$X_{ik} = 1$ if node $i \in N$ is allocated to a hub at node $k \in H$; 0 otherwise

$Y_{km} = 1$ if there is a hub arc between hub $k \in H$ and hub $m \in H$ ($k \neq m$); 0 otherwise

$Z_{km}^{ij} = 1$ if the arc (k,m) is used on the path from node $i \in N$ to node $j \in N$; 0

otherwise

Observe that if the variable $X_{kk}=1$ for some $k \in H$, then, it means that node k is a hub. See also Figure 4 for the illustration of the variables.

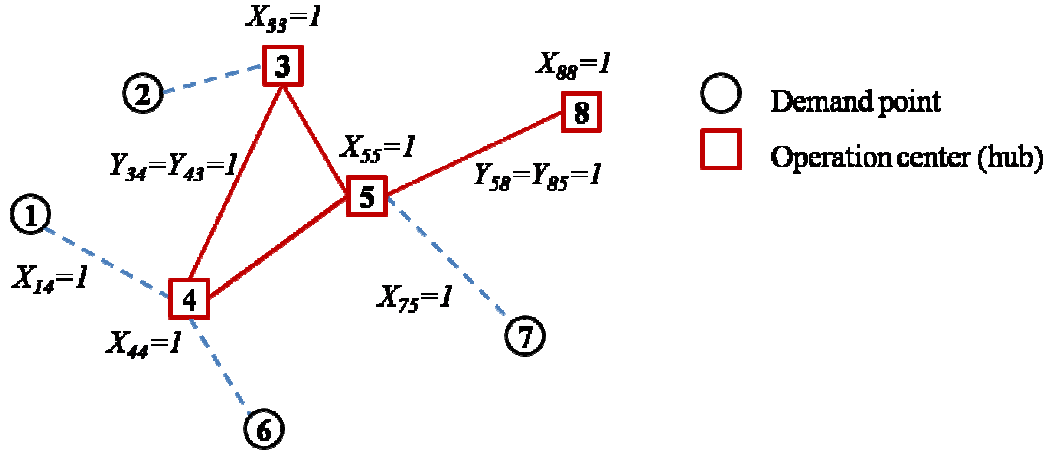


Figure 4: Variables of the first model

The objective function of our mathematical model minimizes the total cost of establishing hubs and hub links, and with the previously defined parameters and decision variables, our model can be expressed as follows:

$$\text{Minimize} \quad \sum_{k \in H} \sum_{m \in H: m > k} f_{km} Y_{km} + \sum_{k \in H} f h_k X_{kk} \quad (1)$$

Subject to

$$\sum_{k \in H} X_{ik} = 1 \quad \forall i \in N \quad (2)$$

$$X_{ik} \leq X_{kk} \quad \forall i \in N, k \in H \quad (3)$$

$$Y_{km} \leq X_{kk} \quad \forall k, m \in H, k \neq m \quad (4)$$

$$Y_{km} \leq X_{mm} \quad \forall k, m \in H, k \neq m \quad (5)$$

$$Y_{km} = Y_{mk} \quad \forall k, m \in H, k \neq m \quad (6)$$

$$\sum_{m|m \neq i} Z_{im}^{ij} = 1 \quad \forall i, j \in N, i \neq j \quad (7)$$

$$\sum_{m|m \neq j} Z_{mj}^{ij} = 1 \quad \forall i, j \in N, i \neq j \quad (8)$$

$$\sum_{m|m \neq i} Z_{km}^{ij} = \sum_{m|m \neq j} Z_{mk}^{ij} \quad \forall i, j, k \in N, i \neq j, k; k \neq j \quad (9)$$

$$Z_{km}^{ij} \leq Y_{km} \quad \forall i, j, k, m \in N, i \neq j, k, m; k \neq j, m; j \neq m \quad (10)$$

$$Z_{im}^{ij} \leq Y_{im} + X_{im} \quad \forall i, j, m \in N, i \neq j, m; m \neq j \quad (11)$$

$$Z_{mj}^{ij} \leq Y_{mj} + X_{jm} \quad \forall i, j, m \in N, i \neq j, m; m \neq j \quad (12)$$

$$Z_{ij}^{ij} \leq Y_{ij} + X_{ij} + X_{ji} \quad \forall i, j \in N, i \neq j \quad (13)$$

$$\sum_{k \in N} \sum_{m \in N} \alpha t_{km} Z_{km}^{ij} + (1 - \alpha) \left[\sum_{k \in N | k \neq i} t_{ik} X_{ik} + \sum_{m \in N | m \neq j} t_{jm} X_{jm} \right] \leq T \quad \forall i, j \in N, i \neq j \quad (14)$$

$$X_{ik} \in \{0, 1\} \quad \forall i \in N, k \in H \quad (15)$$

$$Y_{km} \in \{0, 1\} \quad \forall k, m \in H \quad (16)$$

$$Z_{km}^{ij} \in \{0, 1\} \quad \forall i, j, k, m \in N \quad (17)$$

In the objective function, the first term represents the sum of the individual fixed costs of establishing hub links; and the second term represents the total cost of establishing hubs.

Since we model the single allocation case of the problem, constraints (2) and (15) ensure that every node is allocated to exactly one hub node. By constraint (3), if node $i \in N$ is allocated to node $k \in H$, then, node $k \in H$ is forced to be a hub node; therefore, a node cannot be allocated to another node unless it is a hub node.

Constraints (4), (5), and (16) force the nodes that have a direct hub arc between each other to be hubs and constraint (6) implies that once a link between two hubs is constructed, it will serve for both directions.

When travelling on a path from origin $i \in N$ to destination $j \in N$, the number of outgoing arcs from the origin and the number of incoming arcs to destination is restricted to 1 by constraints (7) and (8), respectively. On the other hand, the flow balance on the nodes rather than origin and destination is satisfied by constraint (9), by equating the incoming flow to such a node to outgoing flow from that node.

While travelling from an origin $i \in N$ to a destination $j \in N$, if an arc (k, m) , of which neither i nor j is an end point, is traversed, then, this arc should be a hub arc (10). If instead an additional node is traversed, this node should be a hub and needs to be directly connected to origin in the case of constraint (11) and to destination in the case of constraint (12). However, if a direct arc between the origin and the destination is travelled, then, either both of them should be a hub with a direct hub, or at least one of them should be a hub and the other one should be allocated to this one (13). For the pictorial explanation of the constraints (10), (11) and (13) see Figure 5. One can easily see that the related picture for constraint (12) is quite similar to the one for constraint (11).

In the first term of constraint (14), the travelling time of all the links traversed while going from origin $i \in N$ to destination $j \in N$ is multiplied with the discount factor α and summed, then, the ones which are not hub links are subtracted and added as normal links to the summation; consequently, this summation is restricted to be within the time bound T .

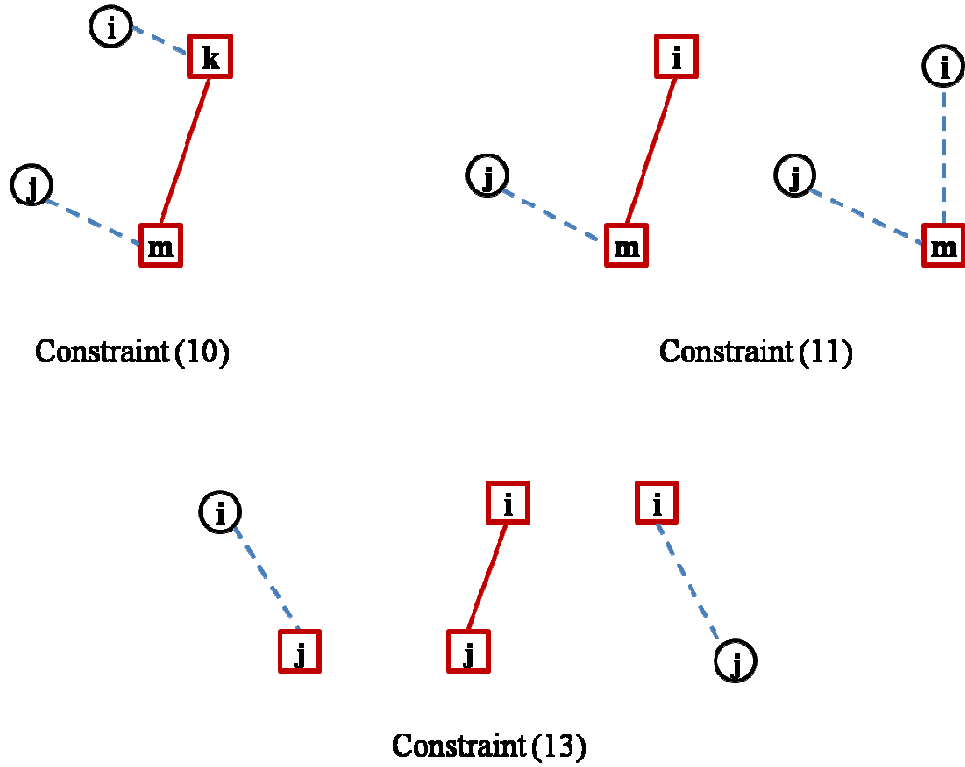


Figure 5: Link variations on paths

We additionally construct three valid inequalities for our model as follows:

$$Y_{km} \leq Z_{km}^{km} \quad \forall k, m \in N, k \neq m \quad (18)$$

$$X_{ik} \leq Z_{ik}^{ik} \quad \forall l, k \in N, l \neq k \quad (19)$$

$$X_{ik} \leq Z_{ki}^{ki} \quad \forall l, k \in N, l \neq k \quad (20)$$

The first valid inequality implies that, if there exists a direct hub arc between two nodes, then, this arc has to be used while going from one of these nodes to the other one. Similar implications are obtained for the arcs between non-hub nodes and hubs by constraints (19) and (20).

Our first mathematical model consists of the objective function (1) and the constraints (2)–(20). Assuming that $h = n$, the model has $O(n^4)$ binary variables and $O(n^4)$ constraints.

3.2. The Model with Path Variables Restricted to the Hub Network

After constructing the first model, we noticed that we do not need to keep the information of the whole path from an origin to a destination. Since the information of starting and the ending nodes are known by the allocation variables, the only lacking information is the path traversed in the hub network. In order to use the advantage of this fact, we define the path variables, \mathbf{z}_{km}^{ij} , over the hub network in our second model. We also make a slight change in the description of the Y_{km} variable, by defining them for $k < m$. The X_{ik} and X_{kk} variables are defined in the same way as the first model. The formal definition of Y_{km} and \mathbf{z}_{km}^{ij} are as follows:

$Y_{km} = 1$ if there is a hub link between hub k and hub m ($k < m$); 0 otherwise

$\mathbf{z}_{km}^{ij} = 1$ if the hub link $\{k, m\}$ is used on the path from hub i to hub j in the direction from k to m ; 0 otherwise
 $r_k = \text{radius of hub } k$

We define the decision variable, radius r , similarly to the one used in Ernst et al. (2005). The radius of a hub represents the time value that is greater than or equal to the travelling time from that hub to the any node allocated to it. We need radius values of the hubs in this model because we define the path variables restricted to hubs. Thus, in order to check if travelling time between any origin and destination pair is within the time bound, instead of considering each node allocated to a certain hub, checking only the radius values of the hubs provides the sufficient information.

The objective function of this model is exactly the same with the previous model and the first set of constraints model involves the constraints (2), (3), and (15) from the previous model. The other constraints of this mathematical model can be described as follows:

Minimize (1)

Subject to

(2), (3), (15)

$$Y_{km} \leq X_{kk} \quad \forall k, m \in H, k < m \quad (4')$$

$$Y_{km} \leq X_{mm} \quad \forall k, m \in H, k < m \quad (5')$$

$$Z_{km}^{ij} + Z_{mk}^{ij} \leq Y_{km} \quad \forall i, j, k, m \in H, k < m, i \neq j \quad (21)$$

$$\sum_{m \in H, m \neq k} Z_{km}^{kj} \geq X_{kk} + X_{jj} - 1 \quad \forall k, j \in H, k \neq j \quad (22)$$

$$\sum_{m \in H, m \neq k} Z_{mk}^{ik} \geq X_{kk} + X_{ii} - 1 \quad \forall i, k \in H, i \neq k \quad (23)$$

$$\sum_{m \in H, m \neq k} z_{mk}^{ij} - \sum_{m \in H, m \neq k} z_{km}^{ij} = 0 \quad \forall i, k, j \in H, i \neq k, i \neq j, k \neq j \quad (24)$$

$$z_{km}^{ij} + z_{mk}^{ij} \leq X_{ii} \quad \forall i, j, k, m \in H, k < m, i \neq j \quad (25)$$

$$z_{km}^{ij} + z_{mk}^{ij} \leq X_{jj} \quad \forall i, j, k, m \in H, k < m, i \neq j \quad (26)$$

$$r_k \geq t_{ik} X_{ik} \quad \forall i \in N, k \in H \quad (27)$$

$$\sum_{k \in H} \sum_{m \in H, k \neq m} \alpha t_{km} z_{km}^{ij} + r_i + r_j \leq T \quad \forall i, j \in H \quad (28)$$

$$Y_{km} \in \{0, 1\} \quad \forall k, m \in H, k < m \quad (16')$$

$$z_{km}^{ij} \in \{0, 1\} \quad \forall i, j, k, m \in H, i \neq j, k \neq m \quad (17')$$

This time we define constraints (4') and (5') in accordance with the new definition of the Y_{km} variable. By constraint (21) if a hub link is to be used for a given origin-destination hub pair, that hub link has to be established. This constraint also ensures that at most one of z_{km}^{ij} and z_{mk}^{ij} can be one since they need to provide a simple directed path. By constraints (16') and (17') path variables and the hub link decision variables are required to be binary.

Constraints (22), (23) and (24) are the flow balance constraints in the hub network. These constraints ensure that every hub node sends and receives one unit of flow, and the connectivity in the hub network is established. By constraint (22), if both nodes k and j are hubs, then the origin hub k sends one unit of flow to the destination hub j in the hub network. By constraint (23), if nodes i and k are both hub nodes, then the destination hub k receives one unit of flow from the origin hub i in the hub network. In the case where hub node k is neither

the origin nor the destination, the incoming flow must equal the outgoing flow, by constraint (24). In this model, we route the flow only in the hub network; thus, constraints (25) and (26) ensure that the origin and destination nodes can only be hub nodes.

For each hub, the maximum travel time between that hub and the nodes that are allocated to it is calculated by the r variable in constraint (27). By the time bound constraint (28), for each pair of hubs, the radii of these hubs plus the discounted total travel time between these hubs using the established hub links must not be greater than the given time bound, T .

In order to explain this mathematical model thoroughly, let us consider the example illustrated in Figure 6.

Let Figure 6 represent the resulting network of a potential solution. According to this solution, $X_{33} = X_{44} = X_{66} = X_{77} = 1$ implying that nodes 3, 4, 6, and 7 are chosen as hub nodes. The variables $X_{13} = X_{23} = X_{54} = X_{86} = X_{97} = 1$ represent the allocations of the non-hub nodes to the hub nodes. Moreover, $Y_{34} = Y_{36} = Y_{47} = Y_{67} = 1$ indicate the constructed hub links. All other X and Y variables have zero values at this solution.

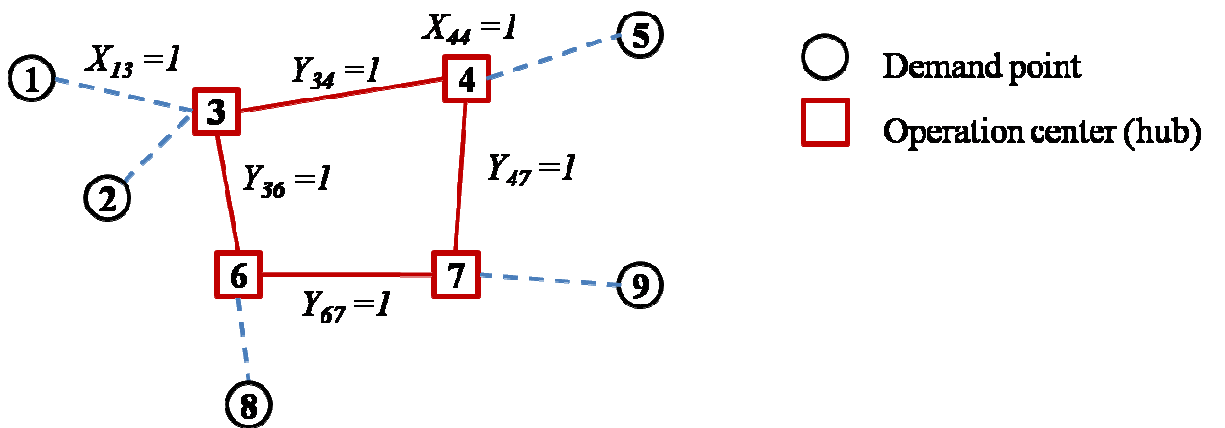


Figure 6: The resulting network of a solution

All of the Z variables associated with the nodes 1, 2, 5, 8, and 9 are forced to be zero by constraint (25) and (26). By constraints (22), (23), and (24) each of the hub nodes sends one unit of flow to all other hub nodes. Consider the flow from hub node 3 to hub node 7. Hub node 3 sends one unit of flow to hub node 7 in the network by constraint (22). Thus either Z_{34}^{37} or Z_{36}^{37} must be equal to 1. Similarly, hub node 7 must receive one unit of flow from hub node 3 by constraint (23). Thus, either Z_{47}^{37} or Z_{67}^{37} must be equal to 1. By constraint (24), the incoming flow must be equal to the outgoing flow for rest of the Z_{km}^{37} variables. Note that by constraint (21), for some hub link $\{i,j\}$, either one of Z_{km}^{37} and Z_{mk}^{37} can take on the value one. Thus, there exist exactly 2 possible paths from hub node 3 to hub node 7. First one is by using hub arcs (3,6) and (6,7) and the second one is by using hub arcs (3,4) and (4,7). The model decides which path to choose by using the time bound constraint (28). Assume that $\alpha t_{34}Z_{34}^{37} + \alpha t_{47}Z_{47}^{37} + r_3 + r_7 > T$ and $\alpha t_{36}Z_{36}^{37} + \alpha t_{67}Z_{67}^{37} + r_3 + r_7 \leq T$. Then the model lets $Z_{36}^{37} = Z_{67}^{37} = 1$ and all other Z_{km}^{37} variables to be zero.

The second mathematical model consists of the objective function (1) and the constraints (2), (3), (4'), (5'), (21)-(30), (15), (15') and (17'). Assuming that $h = n$, the model has $O(n^4)$ binary variables and $O(n^4)$ constraints.

As expected, for greater values of n , the models result in a large number of variables and constraints, which will make them harder to solve to optimality. Thus, we propose a tabu-based heuristic for our model in the next section.

Chapter 4

TABU SEARCH BASED HEURISTIC ALGORITHM

It is a well known fact that most of the NP-Complete problems cannot be solved to optimality for realistically sized instances. For the problem that we are studying, even finding a feasible solution is challenging. Therefore, we decided to develop a heuristic algorithm for our problem to be able to solve large problems. In order to avoid getting stuck at local optima, we employed ideas from the well-known tabu search heuristic methodology. A brief description of the tabu search methodology and related literature can be found in the following section.

4.1 A Brief Description of Tabu Search

Tabu search is a meta-heuristic, which avoids getting stuck at the local optima despite it explores the solutions with a local search procedure. The term *tabu search* is originated by Glover (1986) which introduces the term *meta-heuristic*. Tabu search incorporates the adaptive memory and the responsive exploration in solving problems. The adaptive memory

feature allows searching the solution space efficiently and effectively since the movements are performed according to the information collected during the search. The prominence of the responsive exploration arises from the idea that bad movements might produce better results than the good ones. A broad investigation of Tabu Search methodology can be found in the book by Glover and Laguna (1997).

The TS methodology is widely used in solving hard optimization problems such as scheduling, telecommunication, network design, and location problems. Since the hub location is a rather new research area, only a few implementations of TS algorithm for the hub location problems are available in the literature. Skorin-Kapov and Skorin Kapov (1994) propose a heuristic method based on tabu search for the p-hub median problem. Abdinour-Helm (1998) presents a heuristic algorithm, which is a combination of genetic algorithm and tabu search for the hub location problem with fixed costs. Pamuk and Sepil (2001) develop a single-relocation algorithm with tabu search for the p-hub center problem. There exists no implementation of tabu search methodology for the hub covering problem in the literature. Thus, our study is the pioneering work in this respect. In the following section, we explain the general structure of our heuristic algorithm.

4.2 General Structure of the Proposed Heuristic Algorithm

The tabu search based heuristic algorithm we proposed mainly consists of the construction and the improvement phases. Initially, the algorithm aims to construct several feasible solutions for the problem and after obtaining these feasible solutions, it looks for possible improvements on these solutions. The improvement on a feasible solution basically means a decrement on the total cost by removing some hub links from this solution without violating the feasibility. In fact, removing some links from some solution may create a new solution; therefore, we can also define improvement as searching for new feasible solutions that brings smaller cost values.

The algorithm starts with choosing a set of initial hub locations. For this purpose, a strategy, which is based on a relationship between the time distances of the nodes to each other and the specified time bound value, is developed. After a set of initial hub locations is chosen, the algorithm allocates the rest of the nodes to these hubs according to the allocation strategies that we develop and it constructs the hub network. The algorithm does not start with a complete solution but with a partial solution that can be either feasible or infeasible. This partial solution is a set of hubs and it does not guarantee a feasible allocation construction for further steps. In order to construct complete solutions from this partial solution, three different construction methods, which are based on different allocation strategies, are used. In each of these allocation strategies, initially a complete network is constructed between the selected hubs and feasible allocations are searched over this complete network. As mentioned earlier, when a feasible solution is obtained, the search for better feasible solutions starts in the improvement phase. For this purpose, the hub links that do not lead to infeasibility when removed are taken off from the hub network.

The construction phase is performed for a specified number (*NeighIteration*) of random neighbors of a set of hub locations. These neighbors are obtained by random exchange of a hub node with a non-hub node. The neighbor that provides a feasible solution with the best objective function value is selected for the next move. If no feasible solution can be obtained within $2 * NeighIteration$ neighbor traversals, a neighbor is selected randomly for the next move even if it is infeasible. Thus, our algorithm allows moving to infeasible neighbors of a solution.

The construction and improvement stages are performed at each move and the best among all the feasible solutions produced by the algorithm is reported. To sum up, the algorithm tries to find new solutions initially by changing the hub set, then by constructing the allocations, and, finally, by reducing the hub connectivity.

In order to avoid getting stuck at a local optimum solution, both worse and better feasible solutions are accepted in moving to a neighbor unless they constitute a tabu move

(movement to the forbidden neighborhood). Incidentally, a list of tabu moves is kept to prevent cycling.

We provide formal and detailed descriptions of the steps of the algorithm below.

Solution: In a solution, all hub locations and allocations are determined, and the hub network is constructed, but its feasibility is not guaranteed.

Feasible solution: A solution is feasible if it guarantees that the shortest travel time from each origin to each destination is within the time bound.

Move: At each iteration of the tabu search, a base hub set is chosen to identify the solutions that might be obtained from its neighborhoods. At the beginning, this hub set is chosen from the initial hub locations set. During the following iterations, this hub set is chosen from the neighborhood of the base hub set of the previous iteration, with some criteria. The selection of the base hub set represents a move in the algorithm.

Initial hub locations: Finding a starting feasible solution for the hub covering problem is difficult when the time bound value is tight. Therefore, instead of starting the algorithm with a feasible solution, initially we construct several initial hub sets via a strategy we developed and then, by selecting one of these hub sets we try to generate feasible solutions by traversing neighbors of this base hub set.

Neighborhood: We define neighborhoods of solutions over the hub sets. A neighbor of a hub set H_i is another hub set H_j obtained by exchanging the role of exactly one of the hubs of H_i with a non-hub node. At each iteration of the algorithm, a specified number (*NeighIteration*) of random neighbors of the related hub set are generated as candidates for the next move.

Construction of the initial hub sets:

During the formation of initial hub sets, initially an $n \times n$ covering matrix $= [a_{ij}]$, is constructed as follows:

$$a_{ij} = \begin{cases} 0 & \text{if } t_{ij} > \frac{T}{2} \\ 1 & \text{otherwise} \end{cases}$$

where T is the time bound.

For each node $i \in N$, the following steps are repeated: node i is selected as a hub and the nodes that are not covered by node i are collected in a set. Among the remaining nodes, the node that covers the elements of this set the most is selected as another hub (if more than one such node exists, the remaining steps are followed for each case) and the covered elements are removed from the set.

Until all nodes are covered, the node that covers the uncovered set the most is selected as the hub and the elements covered by this hub are removed from the set. At the end of this process, at least one or more hub sets, which possibly contain different number of nodes, are created. Among the constructed hub sets, the ones that contains the minimum number of hubs, are selected and included in the initial hub locations set, say *locationsSet*, of the algorithm. Note that, for each of the elements of this set, although the allocations to hubs are within the time bound, the locations are not guaranteed to provide feasible solutions.

Selection of the base hub sets:

While the time limit is not exceeded, each hub set in *locationsSet* is chosen as the base hub set of the algorithm. If the tabu iteration limit is reached for a base hub set with no

feasible solution, a randomly selected node is added to this base hub set, increasing the set size by one, and the same steps are repeated until a feasible solution is obtained unless the time limit is exceeded. If all the elements of *locationsSet* are traversed before the time limit, algorithm continues to select them for one more time.

Solution Construction:

When the locations of the hubs are determined, in order to construct feasible solutions with this given set of hubs, we perform three different allocation strategies: Type I, Type II, and Type III allocation. During all the allocation strategies, the feasibility of the solutions is determined by checking the time bound constraints of the problem. At the beginning of each allocation strategy, the hub network is assumed to be complete. As soon as a feasible solution is obtained at the end of any allocation strategy, the algorithm focuses on the hub network. To obtain better feasible solutions with the given allocations, hub links are removed randomly from the complete hub network. If the removal of a link leads to infeasibility, that link is added back to the solution, and another hub link is chosen, again randomly, to be removed. We describe our three allocation strategies as follows:

Type I allocation

In this strategy, as a starting point, the allocations are decided using the nearest allocation heuristic HEUR1 of O’Kelly (1987), i.e., every non-hub node is allocated to its nearest hub node. We then concentrate on the hub with the largest radius, with respect to the nearest allocation strategy. We allocate all non-hub nodes to the hub with the largest radius, as long as this allocation does not increase the radius of this hub. In this way, the radii of some hubs may decrease, while the largest radius in the network stays constant, and the chance of reaching feasible solutions increases. If no feasible solution can be obtained with this allocation strategy, two additional processes are performed respectively: Initially, the radii of

all hubs are calculated, and, for each hub, the nodes allocated to it are distributed to the other hubs, as long as their radii do not increase. In the second process, all the nodes are first allocated to their nearest hub. Then, for each hub h , the following steps are repeated: the node i that determines the radius value of h is removed from the network. Now, since we removed node i from the network, another node, say node j , will determine the radius of hub h . Afterwards, the feasibility of the network is checked and in this network, if the shortest path from j to any other node is not within time bound, then, j is also removed from the network and the feasibility is checked for the new network with $n-2$ nodes. When the network is feasible, the search of hubs for nodes i and j to be allocated begins. We need to allocate nodes i and j to some hub nodes since we are to obtain a feasible solution for the problem with all nodes. Note that i and j are allowed to be allocated to different hubs. If one of them cannot be allocated to a new hub k such that the feasibility is not violated, then, another hub h is chosen for the same process starting from nearest allocation of all nodes. In this process, we remove at most two nodes, which are allocated to hub h , from the network.

Type II allocation

In this strategy, first, we calculate a value that we call the *potential radius* for each hub. The potential radius is the maximum possible radius value for a hub node that will not exceed the given service time bound, T . In the beginning, without any allocations, since we constructed a complete hub network, the potential radius value of any hub is $T - \alpha *$ (the maximum travel time from that hub to another hub). We allocate each non-hub node, starting from the non-hub node with the smallest index, to a randomly chosen hub to which the travel time is no more than the calculated potential radius. When a non-hub node is allocated to a hub, the potential radii of all other hubs are updated accordingly. If, at some point there is no feasible allocation for a non-hub node, we discontinue this strategy.

Type III allocation

In this strategy, all non-hub nodes are first allocated to only one hub, say h_1 , in the hub set. If this allocation is not feasible, the non-hub node that determines the radius of h_1 is allocated to another hub, h_2 (for the beginning h_1 is the hub with the smallest index and h_2 is the hub with the second smallest index). All non-hub nodes allocated to h_1 are selected one by one, in decreasing order of travel time to hub h_1 , until the feasibility is reached or until all non-hub nodes are allocated to h_2 . If feasibility is not achieved by any allocation from h_1 to h_2 , allocations from h_1 to h_3, h_4, \dots , from h_2 to h_3, h_4, \dots and all other combinations are checked, that is, all non-hub nodes will be allocated to h_1 and then from h_1 to h_3 , one by one, from h_1 to h_4 , one by one. Then, all non-hub nodes will be allocated to h_2 and then from h_2 to h_3 , one by one, from h_2 to h_4 , one by one, and so on. Note that the Type III allocation strategy restricts all allocations to, at most, two hubs.

During the experiments we conducted on the individual allocation strategies, we observed that the least time-consuming strategy is Type I allocation, while the most time-consuming one is Type II allocation. On the other hand, Type II and Type III allocations may produce feasible solutions when no feasible solution can be obtained via Type I allocation. Therefore, in order to obtain good solutions in reasonable amounts of time, we primarily perform Type I allocation for each hub set. If no feasible solution can be obtained by this strategy, Type II allocation is applied. The Type III allocation strategy is required, only if feasibility is not achieved with the first two strategies.

Feasibility Check:

The feasibility of a constructed solution is checked as follows: initially a configuration matrix corresponding to the hub network of the solution is constructed. In this matrix, the indices corresponding to the links opened between the hubs have their time value multiplied by the discount factor, and the other indices have an infinite value. Then, the radius values of

the hubs are calculated. Two conditions must be satisfied for a feasible solution: i) traversing any radius twice should take no more than the time bound, and ii) for each hub pair h_i and h_j , the summation of $r(h_i)$, $r(h_j)$, and the shortest path between h_i and h_j times α should be no greater than the time bound T , where $r(h_i)$ is the radius value of hub h_i . The shortest path between h_i and h_j is calculated by using the configuration matrix as the distance matrix in Dijkstra's algorithm (Ahuja et al., 1993).

Tabu search iterations:

The search starts with an initial hub set. At any iteration, the algorithm moves to a neighbor hub set with the best feasible solution. If no feasible solution is found within *NeighIteration* neighbors of a hub set, another subset of *NeighIteration* neighbors is generated and searched. If still no feasible solution is found within these neighbors, a random infeasible neighbor is chosen for the next hub set. In order to prevent cycling, the same node exchanges are avoided for a certain number of iterations, which is called tabu tenure in the literature (Glover and Laguna, 1997). If no feasible solution is found within the specified number of tabu iterations, another hub is randomly added to the hub set, and the same steps are followed. The algorithm continues to randomly select base hub sets and traverse their neighbors until a specified time limit is reached.

We present a flow chart of the algorithm in Figure 7.

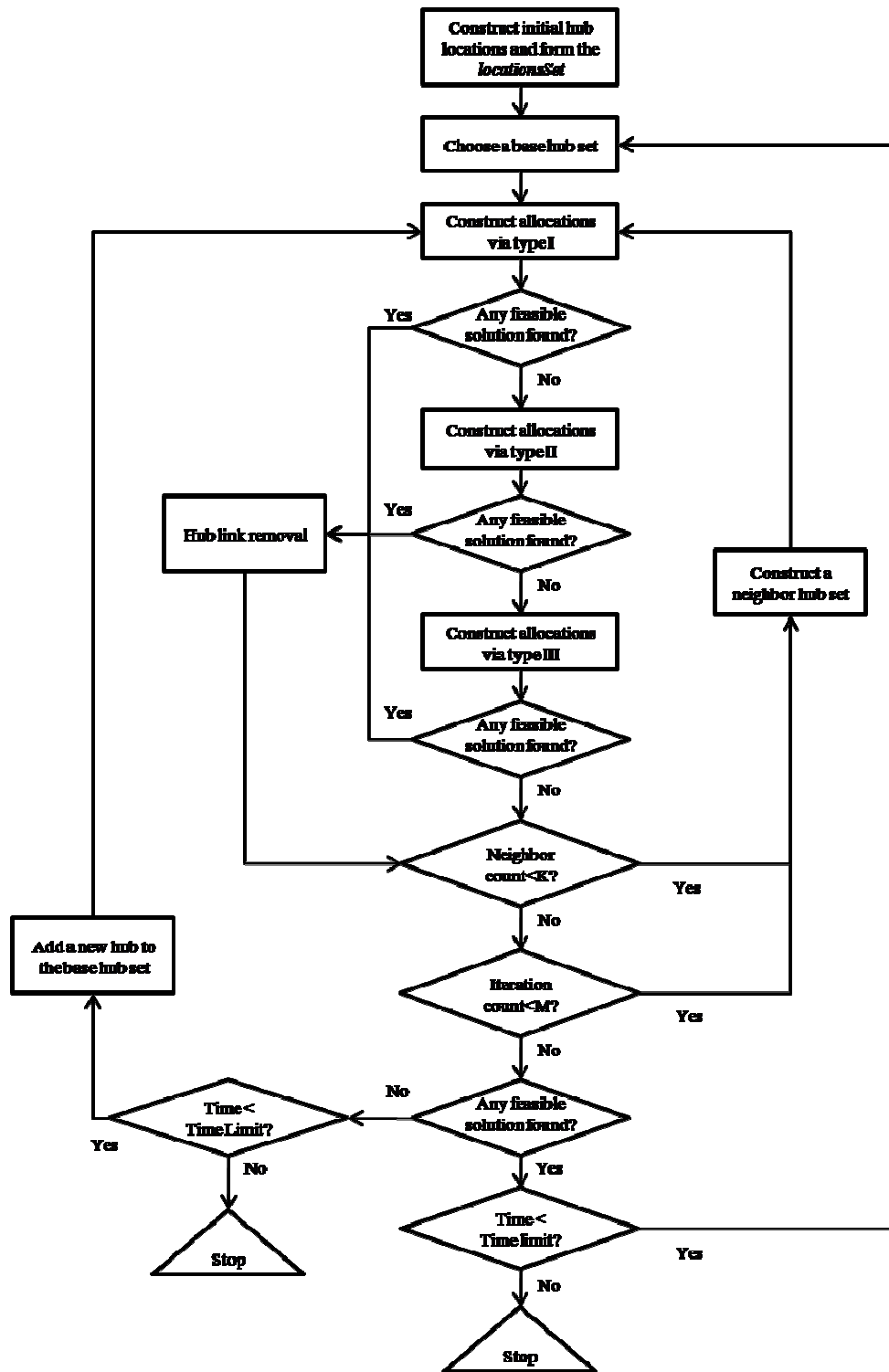


Figure 7: The flowchart of the heuristic algorithm

4.3 Illustration of the Algorithm with Examples

In order to describe the steps of the algorithm more practically, we provide explanations on a sample instance of the problem from the CAB data where $n=10$, $\alpha=0.2$, $T=1118$ and the time matrix is as follows;

	1	2	3	4	5	6	7	8	9	10
1	0	576	946	597	373	559	707	1208	602	692
2	576	0	370	613	430	312	1196	1503	404	1240
3	946	370	0	860	752	558	1543	1766	623	1603
4	597	613	860	0	254	312	790	907	238	932
5	373	430	752	254	0	226	793	1081	238	879
6	559	312	558	312	226	0	1010	1217	94	1104
7	707	1196	1543	790	793	1010	0	664	983	222
8	1208	1503	1766	907	1081	1217	664	0	1144	875
9	602	404	623	238	238	94	983	1144	0	1095
10	692	1240	1603	932	879	1104	222	875	1095	0

Construction of the initial hub sets:

The related *covering matrix* for this instance of the problem will be as follows:

c_{ij} :		1	2	3	4	5	6	7	8	9	10
1		1	0	0	0	1	1	0	0	0	0
2		0	1	1	0	1	1	0	0	1	0
3		0	1	1	0	0	1	0	0	0	0
4		0	0	0	1	1	1	0	0	1	0
5		1	1	0	1	1	1	0	0	1	0
6		1	1	1	1	1	1	0	0	1	0
7		0	0	0	0	0	0	1	0	0	1
8		0	0	0	0	0	0	0	1	0	0
9		0	1	0	1	1	1	0	0	1	0
10		0	0	0	0	0	0	1	0	0	1
Coverage		3	5	3	4	6	7	2	1	5	2

Selection of the initial hub node:

In order to produce more initial hub sets, for each node we will repeat the same initial hub sets construction steps, which start with choosing that node as the initial hub. When we choose the initial hub as node 1, then, the hub set = {1} and the uncovered nodes set={2, 3, 4, 7, 8, 9, 10}.

		2	3	4	7	8	9	10
2		1	1	0	0	0	1	0
3		1	1	0	0	0	0	0
4		0	0	1	0	0	1	0
7		0	0	0	1	0	0	1
8		0	0	0	0	1	0	0
9		1	0	1	0	0	1	0
10		0	0	0	1	0	0	1
Coverage		3	2	2	2	1	3	2

Now, node 2 and node 9 are the ones that cover the remaining elements the most. Then, we select node 2 as the second hub: hub set = {1, 2}, uncovered nodes set = {4, 7, 8, 10}

	4	7	8	10
4	1	0	0	0
7	0	1	0	1
8	0	0	1	0
10	0	1	0	1
Coverage	1	2	1	2

At the end, starting with node 1 and choosing node 2 as the second provides the hub set {1, 2, 7, 4, 8}. The way that nodes are covered by the hub set {1, 2, 7, 4, 8} can be seen in Figure 8.

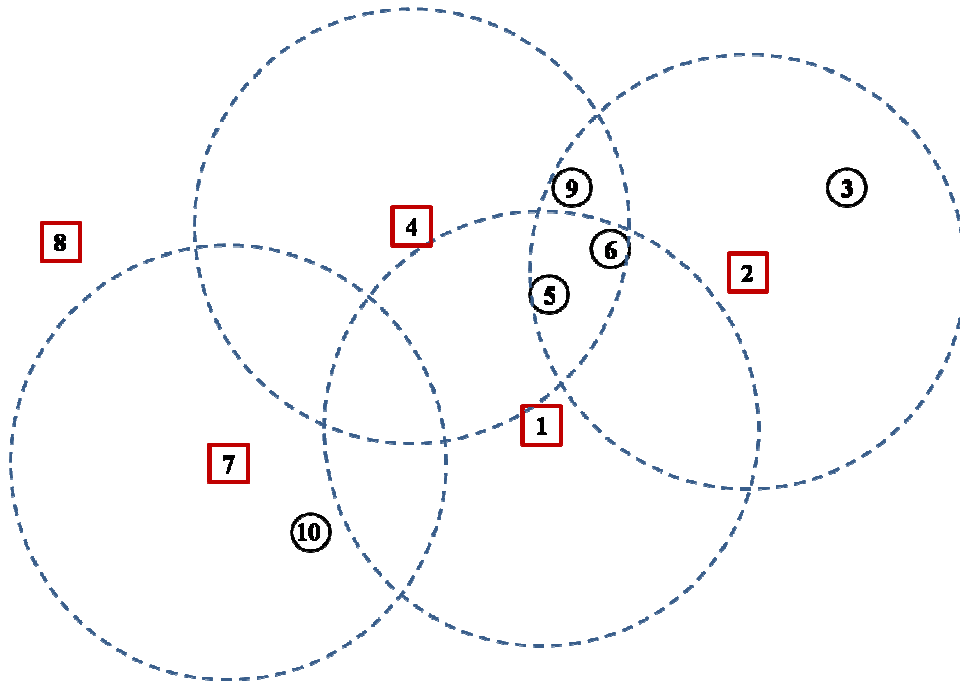


Figure 8: The coverage of the hubs while obtaining hub set {1, 2, 7, 4, 8}

We will repeat the same procedures by choosing nodes 2, 3,..., 10 as the initial hub, respectively. When we start with node 6 as the initial hub and pick node 7 as the second hub, the hub set $\{6, 7, 8\}$ will be constructed while picking node 10 as the second hub will give the hub set $\{6, 8, 10\}$ (Observe how the nodes are covered by the hub set $\{6, 7, 8\}$ from Figure 9). No other choice establishes a hub set with less number of elements. Therefore, among all the hub sets produced, the ones with three hubs will be selected for the further steps (*selection of the base hub set*).

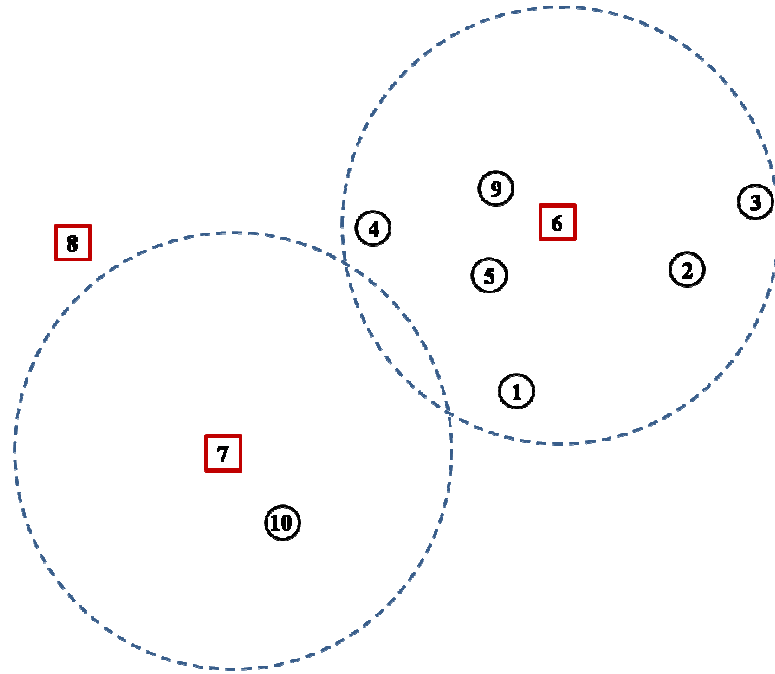


Figure 9: The coverage of the hubs while obtaining hub set $\{6, 7, 8\}$

When we apply *Type I allocation* to the hub set $\{6, 7, 8\}$ that we obtained from the previous steps, it results as follows: after the nearest allocation, nodes 1,2,3,4,5 and 9 are allocated to hub node 6 and node 10 is allocated to hub node 7. Then, the radius values $r_6=559$, $r_7=222$, and $r_8=0$ (See Figure 10).

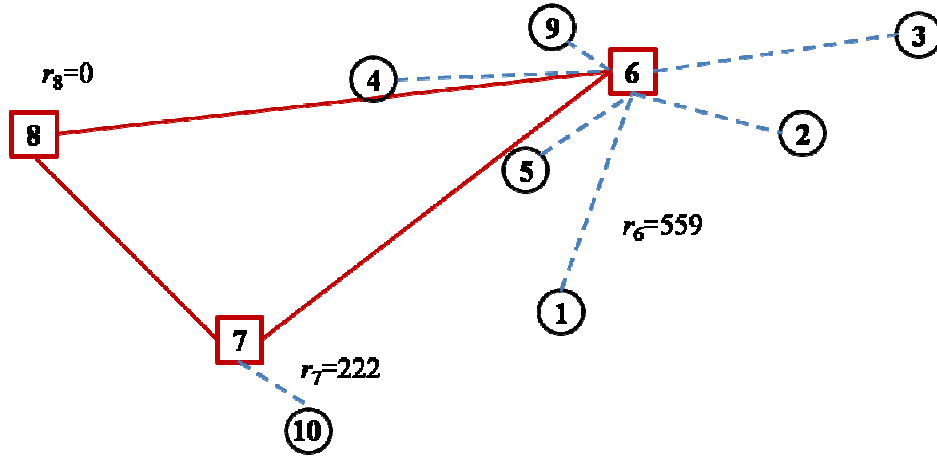


Figure 10: The solution obtained after the nearest allocation

The largest radius value belongs to hub node 6. When we try to allocate node 10, which is currently assigned to hub node 7, to hub node 6, we see that this action increases the radius of hub node 6. Therefore, we do not change the allocation of node 10. Now, we check the feasibility of the obtained solution. The complete hub network with discounted time values is as follows:

$$\begin{array}{c}
 \begin{array}{ccc}
 & 6 & 7 & 8 \\
 \begin{array}{c} 6 \\ 7 \\ 8 \end{array} & \begin{bmatrix} 0 & 202 & 243.4 \\ 202 & 0 & 132.8 \\ 243.4 & 132.8 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

Then,

$$r_6 + r_7 + \alpha t_{67} = 559 + 222 + 202 = 983$$

$$r_6 + r_8 + \alpha t_{68} = 559 + 0 + 243.4 = 802.4$$

$$r_7 + r_8 + \alpha t_{78} = 222 + 0 + 132.8 = 354.8$$

which means that the obtained solution is feasible. Since we obtained a feasible solution, we start searching for hub links that do not lead to infeasibility when removed from the network

and we do not consider Type II and Type III allocation strategies for this base hub set. If we remove link $\{7,8\}$ from the solution, then the total travelling time from node 8 to node 10 is $r_7 + \alpha t_{67} + \alpha t_{68} = 222 + 202 + 243.4 = 667.4$. Since the other pairs are already feasible, this solution is also a feasible solution (see Figure 11 for the new solution).

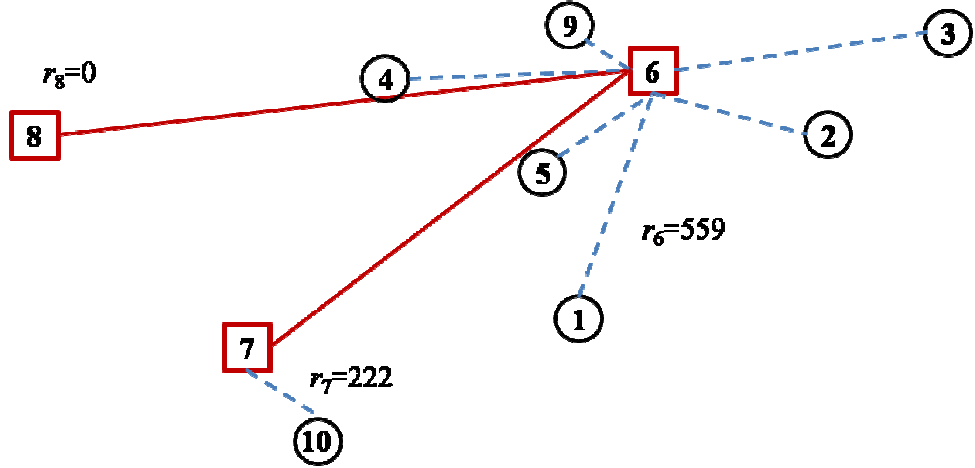


Figure 11: The solution with incomplete hub network

In order to illustrate the steps of the second process of *Type I allocation*, let us consider the instance with the same set of nodes with $\alpha=0.4$ and $T=970$. When we construct the initial hub sets, the initial hub sets with minimum size will have four elements and $\{3, 5, 7, 8\}$ will be one of them. When we apply the same Type I allocation procedure to this hub set, before moving to the second process of Type I allocation, we obtain the following infeasible solution on hand: $H_3=\{2, 3\}$, $H_5=\{1, 4, 5, 6, 9\}$, $H_7=\{7, 10\}$ and $H_8=\{8\}$, where H_i represents the set of nodes allocated to hub i . Then, $r_3=370$, $r_5=373$, $r_7=222$ and $r_8=0$. Furthermore, we recognize that none of the non-hub nodes can be allocated to another hub node without increasing the radius value of that hub (See Figure 12). Therefore, we continue with the second process and remove node 2 from the network. Since the removal of node 2 achieves the feasibility in the network, we stop removing nodes and start looking for a feasible allocation for node 2. If we allocate node 2 to hub node 5, r_5 increases to 430 but the solution remains feasible. Thus, we obtain a feasible solution via Type I allocation (See Figure 13 for the feasible solution).

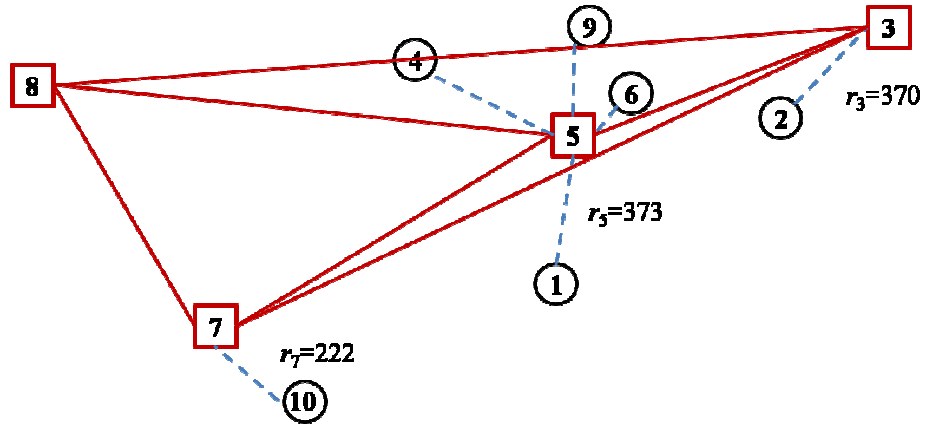


Figure 12: Infeasible solution obtained before the second process of Type I allocation

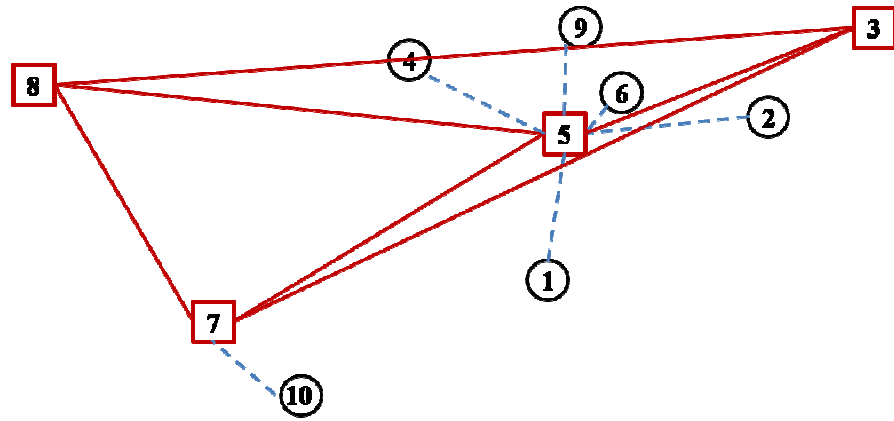


Figure 13: Feasible solution obtained after the second process of Type I allocation

Now, let us apply *Type II allocation* to hub set $\{3, 5, 7, 8\}$ for this instance.

	α_{ij}				potential radius
	3	5	7	8	
3	0.0	300.8	617.2	706.4	263.6
5	300.8	0.0	317.2	432.4	537.6
7	617.2	317.2	0.0	265.6	352.8
8	706.4	432.4	265.6	0.0	263.6

We start allocating the non-hub nodes from the non-hub node with the smallest index, thus, we pick node 1, allocate it to node 5 and update the potential radius values accordingly.

	potential radius
3	263,6
5	537,6
7	279,8
8	164,6

All other non-hub nodes except node 10 will be assigned to hub node 5 and node 10 will be assigned to node 7 at the end of this allocation strategy.

Finally, in order to illustrate Type III allocation strategy, we are going to examine the same hub set of the same instance. Initially, we allocate all the non-hub nodes to hub node 3. In this case, node 10 determines the radius of hub node 3 (See Figure 14).

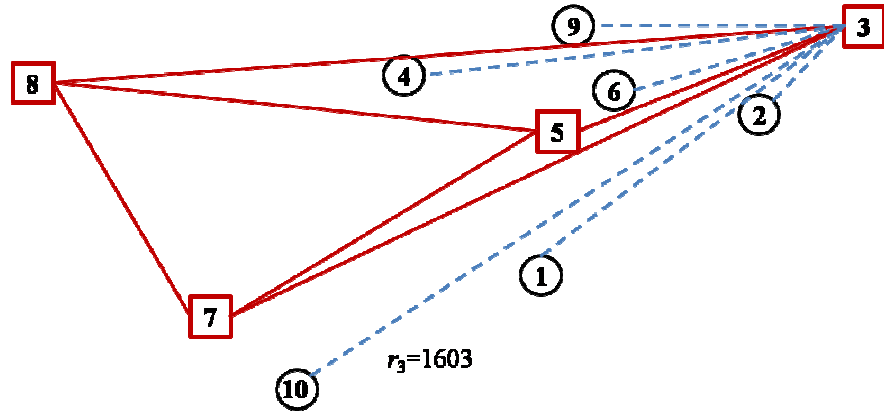


Figure 14: The solution obtained after all non-hub nodes are allocated to hub node 3

Since this solution is infeasible, we allocate node 10 to hub node 5. Then, the radius of hub node 3 is determined by node 1 (See Figure 15); however, we still cannot obtain a feasible solution. Thus, we allocate also node 1 to hub node 5. By repeating this procedure,

we allocate all non-hub nodes to hub node 5 (Figure 16). Since we cannot achieve feasibility, we reallocate the non-hub nodes to hub node 3, and then allocate them from hub node 3 to hub node 7, one by one; then to hub node 8.

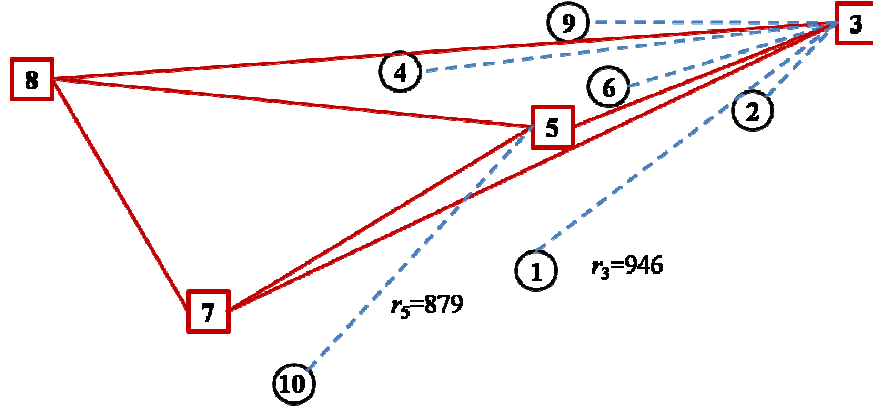


Figure 15: The solution obtained after node 10 is allocated to hub node 5

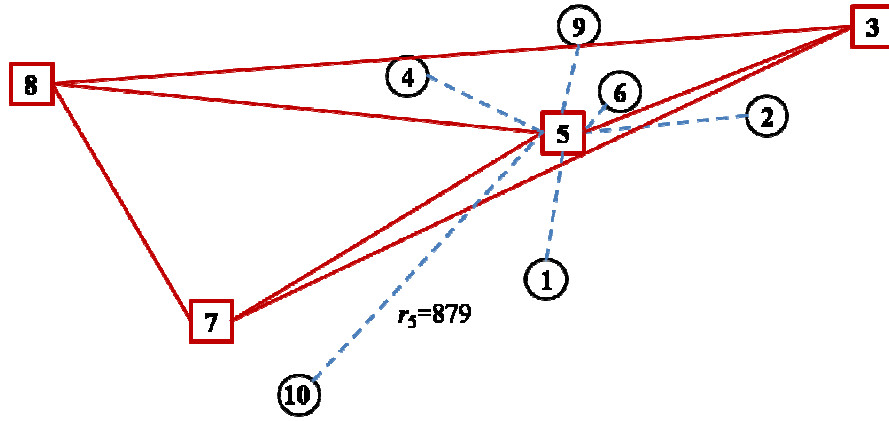


Figure 16: The solution obtained after all non-hub nodes are allocated to hub node 5

Then, we turn back to the beginning and assign all non-hub nodes to hub node 5. In this case, assigning node 10 to hub node 7 produces the feasible solution in Figure 13. Note that all three solutions we obtained from different allocation strategies for the same instance are exactly the same. However, since our algorithm moves to another hub set after obtaining a

feasible solution via Type I allocation, it does not consume extra time for searching this solution with other allocation strategies.

Chapter 5

COMPUTATIONAL RESULTS

In order to test the performance of our models and heuristic, we performed several experiments primarily over the CAB data. Since no real time data is provided for the CAB data set, similar to other hub covering studies in the literature, we used distance values for time data in our computations by setting the time value between two points directly equal to the distance value between them ($t_{ij} = d_{ij}$ where d_{ij} is the distance between node i and node j). In addition, in order to see the performance of our heuristic for larger data, we conducted additional experiments on TR data, which consists of 81 cities in Turkey. Furthermore, we took the fixed cost of opening hub facilities $fh_k=100$ and fixed for all nodes (O’Kelly, 1992) for the experiments on the CAB data.

In addition to the fixed cost of opening hub facilities, in order to be able to establish incomplete hub networks, we also need fixed costs for hub links to include in the objective function. The fixed cost value of a hub link between two nodes depends on both the travel distance and the flow between these nodes in several applications of the hub location problem. By including flow in the fixed cost value, operational costs can also be reflected to the solution. In reality, the distance between two nodes directly affects the cost of establishing a link between them while the flow value between them inversely affects this cost value. Following from this fact, we introduced a more realistic data set to the literature by calculating the fixed cost of a hub link between node i and node j as follows:

$$f_{ij} = \frac{\frac{d_{ij}}{w_{ij}}}{\max_{i,j} \left\{ \frac{d_{ij}}{w_{ij}} \right\}} \times 100 \quad \text{for all } i, j \neq t$$

where d_{ij} is the distance between nodes i and j , and w_{ij} is the flow between nodes i and j .

For our computations, we assumed $H=N$ in all of the test instances. For the rest of the parameters, we used the values reported in Table 1. As can be seen from Table 1, during our computations we used all α values from the hub literature for the CAB data set.

Each value in the last five columns of Table 1 represents the time bound value for an instance. The instances in Table 1 include both tight and loose time bound values. Kara and Tansel provided the optimal values of p-hub center model over complete hub networks for the 15 instances for $p=2, 3, 4$ and $\alpha= 0.2, 0.4, 0.6, 0.8, 1.0$ combinations. These values correspond to the highest possible bounds for our problem since they correspond to complete hub networks. However, since our problem is a covering problem we take these 15 instances as our time bounds. In our instances, for each n and α combination we provide one tightest bound that represents the optimal value of the p-hub center problem where $p=4$ for that n and α combination. We also included the bounds that will provide solutions with $p=2$ and 3. Then, since we wanted to create 5 instances for each n and α combination, we created two additional bounds, which are calculated as follows: we take the average of the tight bounds for two consecutive p values for a specified α , and we obtain a loose bound for the larger p value for that α . For instance, for $n=10$ and $\alpha= 0.2$, 1425 is the optimal value when $p=2$ and 1118 is the optimal value when $p=3$ for p-hub center problem. By taking the average of 1425 and 1118, we obtain a new time bound value equal to 1271.5. Since 1271.5 is smaller than the tight bound for $p=2$, no feasible solution with $p=2$ can be obtained with this bound, however, since it is greater than 1118, it represents a loose bound for the solution with $p=3$. In the same manner, 975 is a loose bound for $p=4$.

We solved our integer programming models by using CPLEX 10.1 on a personal computer with a 2.00 GHz Intel Core 2 Duo processor and 2 GB of RAM. We solved our first model for all the CAB instances for $n=10$ and $n=15$ and second model for all instances listed in Table 1. While solving the models, we limited the CPU time to two hours on CPLEX. In order to test the performance of our heuristic algorithm we applied it to the same CAB instances.

In our computations with the CAB data set we used a tabu list with a size of 5, the number of tabu iterations was 500, the number of neighbors to be detected at each iteration (*NeighIteration*) was 50 and the time limit was 100 seconds for the instance with 10 nodes and 600 seconds for the instances with 15 and 20 nodes. Table 2 reports the results obtained from the first model, the second model and the heuristic with 10 nodes.

n	α	T				
10	0.2	1425	1271.5	1118	975	832
	0.4	1627	1406	1185	1077.5	970
	0.6	1758	1572.5	1387	1267.5	1148
	0.8	1758	1673.5	1589	1523	1457
	1	1839	1815	1791	1778.5	1766
15	0.2	2004	1877	1750	1546	1342
	0.4	2162	1961	1760	1598	1436
	0.6	2214	2029	1844	1800	1756
	0.8	2424	2294.5	2165	2122.5	2080
	1	2611	2605.5	2600	2600	2600
20	0.2	1892	1720.5	1549	1452.5	1356
	0.4	2162	1961	1760	1616.5	1473
	0.6	2278	2137	1996	1915.5	1835
	0.8	2508	2386	2264	2209	2154
	1	2611	2605.5	2600	2600	2600

Table 1: Test bed for the CAB data set.

In Table 2, the third column presents the optimum objective function value and the columns under “Model 1” and “Model 2” present the CPU time requirement in seconds for the first model and the second model, respectively. The CPU values are the ones reported by CPLEX. The columns under Model 1 represent the CPU time requirement for the first model without valid inequalities and with valid inequalities, respectively. The objective function value and the gap of the heuristic from the best known solution are reported under “Tabu Heuristic”. The column labeled Bcpu reports the CPU time when the best solution is obtained by the heuristic algorithm for the first time. In the last column of this table, the number of hub links opened in the best solution found by the heuristic is reported in order to observe the solutions with incomplete hub network. The average and maximum CPU time requirements in seconds, for both the models and our heuristic are listed in the last two rows of Table 2 as well.

The experimentations we performed on the individual valid inequalities for the first model revealed that by including all three valid inequalities in the first model we obtain better results in terms CPU time when compared to including each inequality individually. Furthermore, when the valid inequalities (18) and (19) or (18) and (20) are inserted in the first model, the model performs as good as the one with all valid inequalities. On the other hand, including only the constraints (19) and (20) in the model performs worse than the case with three valid inequalities. Therefore, we decided to include all valid inequalities in the first model.

Note from Table 2 that CPLEX solved all the instances with 10 nodes optimally in an average of 52 seconds of CPU time requirement for the first model without inequalities and thirty-eight seconds of CPU time requirement for the first model with valid inequalities while it took a little less than six seconds of CPU time requirement for the second model. When we compare the results for the first model with inequalities and without inequalities, we observe that on the average the model with valid inequalities perform better in terms of CPU time

Test Bed			Model 1		Model 2	Tabu Heuristic			
			without v.e.	with v.e.					
α	T	Obj.	CPU Time (sec)	CPU Time (sec)	CPU Time (sec)	Obj.	Bcpu (sec)	Gap	Number of Hub Links
0.2	1425	206.374	1.612	1.564	1.892	206.374	0.141	0	1
0.2	1271.5	306.631	3.990	1.085	2.422	306.631	0.752	0	2
0.2	1118	308.195	0.260	0.691	1.070	308.195	0.327	0	2
0.2	975	413.927	27.710	1.532	1.827	413.927	1.805	0	3
0.2	832	423.804	69.066	3.772	2.380	423.804	0.248	0	4
0.4	1627	206.374	3.473	1.985	3.725	206.374	0.147	0	1
0.4	1406	306.631	32.376	8.341	2.576	306.631	2.152	0	2
0.4	1185	317.359	1.881	14.255	1.746	317.359	0.189	0	2
0.4	1077.5	413.927	39.141	1.800	1.836	413.927	3.431	0	3
0.4	970	435.865	12.328	19.670	2.637	435.865	1.961	0	4
0.6	1758	221.938	85.587	33.573	5.331	221.938	0.508	0	1
0.6	1572.5	308.195	45.378	21.166	4.151	308.195	2.857	0	2
0.6	1387	319.180	71.260	25.679	4.800	319.180	1.916	0	3
0.6	1267.5	435.865	100.286	104.176	15.409	435.865	6.345	0	4
0.6	1148	444.084	37.712	11.435	9.061	444.084	4.534	0	5
0.8	1758	221.938	16.415	25.548	2.838	221.938	0.147	0	1
0.8	1673.5	313.089	23.489	27.352	9.677	313.089	2.102	0	3
0.8	1589	319.180	26.514	52.791	8.314	319.180	1.977	0	3
0.8	1523	413.037	163.259	80.705	14.646	413.037	5.741	0	4
0.8	1457	453.790	166.612	168.035	19.630	453.790	5.586	0	6
1	1839	201.867	6.519	2.812	1.605	201.867	0.649	0	1
1	1815	306.828	75.028	75.491	6.466	306.828	1.929	0	3
1	1791	319.180	57.171	36.827	5.360	319.180	2.046	0	3
1	1778.5	413.037	99.607	114.166	11.694	413.037	5.987	0	5
1	1766	422.742	75.664	69.780	11.857	422.742	5.639	0	6
Average			51.753	37.648	5.879		2.365	0	
Maximum			166.612	168.035	19.630		6.345	0	

Table 2: Computational comparison of Model 1, Model 2 and the heuristic for n=10.

requirements. Although the first model performed slightly better in terms of CPU time for some of the instances, the second model outperformed the first model in terms of CPU time on the average. The maximum CPU time requirement for this network by Model 1 was around 168 seconds while it was less than 20 seconds for Model 2. Thus, Model 2 performs better in terms of maximum CPU time requirement as well. Furthermore, our heuristic was able to solve all 10 node instances optimally. Observe from Table 2 that on the average the heuristic is able to find the optimal solutions in less than 3 seconds.

For the instances with 15 and 20 nodes we only report the results obtained from the second model and the heuristic since the first model performs considerably worse than the second model.

The comparison of the results obtained from the second model and the heuristic with 15 nodes is reported in Table 3. As it can be seen from Table 3, when the number of nodes becomes 15, the average CPU time requirement of the second model goes up to nine minutes, with a maximum value of about 49 minutes. These considerably large CPU time values actually indicate that, even with a small number of nodes, it is hard to solve this problem to optimality with the models proposed. On the other hand, our heuristic was able to obtain the optimal solutions at 21 out of the 23 instances with 15 nodes. At the other instances, in which our heuristic was not able to obtain the optimal solutions, the gap of the heuristic from the optimal was 0.49 % on the average. In addition, our heuristic was able to find the best objective values in less than seven CPU seconds on the average and with a maximum CPU time requirement of less than fifty-eight seconds.

From Table 2 and 3, we observe that the CPU time requirements for the instances with $\alpha=0.2$ and 0.4 are generally smaller when compared to the instances with larger α values for both models. This can be explained as follows: since the time bound values in our experimentation are relatively small for $\alpha=0.2$ and 0.4, most of the allocation possibilities are infeasible and thus, are readily eliminated hence decreasing the remaining feasible search space. As a justification of this, note that in most of the solutions corresponding to small α

Test Bed		Model 2		Tabu Heuristic			
α	T	Obj.	CPU Time (sec)	Obj.	Bcpu (sec)	Gap	Number of Hub Links
0.2	2004	221.938	46.122	221.938	0.251	0	1
0.2	1877	301.698	29.151	301.698	0.736	0	2
0.2	1750	307.941	48.007	307.941	4.264	0	2
0.2	1546	405.821	144.628	406.225	3.921	0.10	3
0.2	1342	413.026	108.052	413.026	57.378	0	3
0.4	2162	210.057	43.626	210.057	0.176	0	1
0.4	1961	301.265	26.515	301.265	5.263	0	2
0.4	1760	313.450	55.393	313.450	2.402	0	2
0.4	1598	408.281	202.490	408.281	3.185	0	3
0.4	1436	424.826	524.280	424.826	1.426	0	3
0.6	2214	210.057	39.990	210.057	0.179	0	1
0.6	2029	301.698	38.293	301.698	3.299	0	2
0.6	1844	323.578	728.059	323.578	1.079	0	3
0.6	1800	417.946	1.148.998	417.946	6.618	0	4
0.6	1756	419.459	1.118.715	423.162	6.433	0.88	5
0.8	2424	209.334	168.332	209.334	0.191	0	1
0.8	2294.5	311.362	223.915	311.362	3.290	0	3
0.8	2165	332.650	690.407	332.650	3.085	0	3
0.8	2122.5	424.225	2.943.998	424.225	10.193	0	3
0.8	2080	427.929	1.904.888	427.929	20.465	0	4
1	2611	202.814	49.559	202.814	0.184	0	1
1	2605.5	304.110	607.270	304.110	3.982	0	3
1	2600	304.110	1220.41	304.110	3.729	0	3
Average			526.569		6.162	0.04	
Maximum			2.943.998		57.378	0.88	

Table 3: Computational comparison of Model 2 and the heuristic for n=15.

values, the critical path is determined by the tours from an origin to its hub and then back to the origin. Another observation is that when the time bound value is small, for some of the nodes, two times the undiscounted travelling time to any other node does not remain within the time bound. Therefore, that node cannot be allocated to any node and any node cannot be allocated to that node. Then, this node has to be selected as hub and all combinations that do not include this hub need to be eliminated. For instance, for $n=10$ in the CAB data, the travelling time between node 8 and the closest node to node 8 is 552. Therefore, for any instance with time bound value less than 1104, node 8 has to be selected as hub. Similarly, when $n=15$, the travelling time to the closest node from node 12 is 839; thus, for any instance with time bound value less than 1678, node 12 has to be selected as hub.

The results obtained from the tests on the instances with 20 nodes from the CAB data set are reported in Table 4. In 13 out of the 23 instances with 20 nodes the model was not able to obtain optimal solutions within two hours of CPU time. Since the model even could not find an initial feasible solution for some of the instances, we included lower bound values for each instance as well together with the value of the best integer solution found and its gap reported by CPLEX. For the instances that no feasible solution is obtained, the gap of our heuristic was calculated from the lower bounds reported. These estimated gaps, indicated with an asterisk (*) in Table 4, are naturally expected to be much higher than the actual optimality gaps of the heuristic.

Observe from Table 4 that the model was able to obtain optimal solutions in 10 out of the 23 instances for 20 nodes within the time limit we imposed. Our heuristic was able to solve nine of these ten instances optimally and in the other single instance, the gap of our heuristic from the optimal value was 0.80%. In the other instances that we could not obtain the optimal solution, the gap of our heuristic from the lower bound was 31.30% on the average.

Note from Tables 2, 3, and 4 that our heuristic solved 55 of the 71 CAB instances optimally. From the remaining instances, in only three of them the model was able to find the

Test Bed		Model 2				Tabu Heuristic			
α	T	Obj.	CPU Time (sec)	Gap (%)	Lower Bound	Obj.	Bcpu (sec)	Gap (%)	Number of Hub Links
0.2	1892	236.454	532.03	0	236.454	236.454	0.213	0	1
0.2	1720.5	338.609	515.991	0	338.609	338.609	5.580	0	2
0.2	1549	365.989	3.196.707	0	365.989	365.989	0.405	0	3
0.2	1452.5	405.031	4.281.201	0	405.031	408.281	47.784	0.80	3
0.2	1356	513.635	7200	21.42	403.624	415.621	131.648	2.89*	3
0.4	2162	247.777	2.373.811	0	247.777	247.777	0.227	0	1
0.4	1961	303.825	917.251	0	303.825	303.825	13.463	0	2
0.4	1760	355.593	2.494.244	0	355.593	355.593	2.516	0	2
0.4	1616.5	619.423	7200	44.85	341.644	416.408	9.529	17.95*	4
0.4	1473	858.353	7200	53.97	395.067	521.085	494.676	24.18*	6
0.6	2278	252.748	4.552.373	0	252.748	252.748	0.216	0	1
0.6	2137	340.369	7200	22.40	264.115	340.369	7.208	22.40*	3
0.6	1996	497.862	7200	50.60	245.938	355.593	6.224	30.84*	2
0.6	1915.5	4.699.047	7200	36.62	297.811	427.223	28.833	30.29*	2
0.6	1835	6.148.686	7200	51.21	300.000	427.946	27.645	29.90*	4
0.8	2508	2.028.139	2.867.445	0	202.814	202.814	0.452	0	1
0.8	2386	3.065.502	7200	34.51	200.753	302.896	9.750	33.72*	2
0.8	2264	560.239	7200	63.61	203.844	361.692	5.005	43.64*	3
0.8	2209	N/A	7200	N/A	200.000	406.064	13.579	50.75*	5
0.8	2154	N/A	7200	N/A	200.000	415.162	62.126	51.83*	4
1	2611	202.814	6.714.423	0	202.814	202.814	0.212	0	1
1	2605.5	N/A	7200	N/A	200.000	304.110	5.266	34.23*	3
1	2600	N/A	7200	N/A	200.000	304.110	5.660	34.23*	3
Average			5.306.325	19.96			38.851	17.72	
Maximum			7200	63.61			494.676	51.83*	

Table 4: Computational comparison of Model 2 and the heuristic for n=20.

optimal solution and, the average gap of our heuristic in these three instances was 0.59%. In the remaining 13 instances we do not know how close our heuristic is to the optimal.

In order to be able to observe the incomplete hub network solutions, we included the number of hub links in the solutions in the last columns of the Table 2, 3, and 4. Excluding the five instances with $p = 2$, where constructing an incomplete hub network solution is impossible, we obtained incomplete hub networks at 39 out of 71 instances. In general, the factor that forces the completeness of the hub network in the solutions was the tightness of the time bound values. From these results one can easily observe that designing complete hub networks to provide service within a given service time bound is not cost effective, in many cases.

Besides the experimentations on the CAB data set, we tested our heuristic on the Turkish network as well to observe its performance on larger networks. The Turkish network has 81 nodes, and all nodes are chosen as candidate hub nodes for our computations. The time discount factor on the Turkish network with ground transportation was found to be 0.9 (Tan and Kara, 2007). Thus, we assumed $\alpha = 0.9$ for all of the Turkish network instances. The fixed costs for opening hubs and hub links in the Turkish network were also obtained from Tan and Kara (2007).

For the Turkish network, we set the size of the tabu list as 5, the number of tabu iterations as 200, and the number of neighborhoods to be detected at each iteration (*NeighIteration*) as 100. All other parameter values of the test problems can be found in Table 5 together with the corresponding solutions.

Observe from Table 5 that we increased the time limit as the time bound got tighter and the tighter time bound values resulted in larger number of hubs and hub links in the solutions we obtained. On the average, we were able to obtain best solutions (the solutions

retuned by the algorithm) on the Turkish network in 25 minutes. For the last instance in Table 5, the CPU time that the best solution is obtained for the first time is larger than the time limit we impose to the algorithm. The reason for this is that we do not interrupt the neighborhood search at any iteration, therefore, when we stop at some iteration of the algorithm, the time consumed might be slightly larger than the time limit. Excluding the ones with $p=2$, the solution we obtained provided incomplete hub network in 6 out of the 13 instances. Even though we do not know much about the quality of our solutions on the Turkish network, this network is the largest data set that has been tested with incomplete hub network design problems. In reality, obtaining optimal solutions on such a large network is difficult even with complete hub networks.

T	CPU Time (min.)	Bcpu (min.)	Number of Hubs	Number of Hub Links
1880	10	1.065	2	1
1870	10	1.115	2	1
1860	30	16.554	3	3
1850	30	3.266	3	2
1840	30	8.899	3	2
1830	30	28.751	3	2
1820	30	1.825	3	3
1810	60	23.244	3	3
1800	60	4.753	4	6
1790	60	52.824	5	7
1780	90	84.073	5	10
1770	90	7.356	5	8
1760	90	91.828	7	15
Average CPU Time (min.)	47.692	25.043		

Table 5: Computational results of the heuristic on the Turkish network.

When we consider the networks obtained from the instances in Table 5, we observe that as the time bound gets smaller, the cities that are relatively far from each other are selected as hubs. The reason for this might be that when hubs are chosen in this manner, the total discounted travelling time increases and the possibility of decreasing the travelling time between the pairs that are far from each other increases.

When we consider the solutions obtained for $T=1840$ and $T=1820$ in Figure 17, we observe that Ankara, Sivas, and Hakkari are selected as hubs. Note that the hub network is incomplete when $T=1840$; however, this solution with incomplete hub network does not provide a feasible allocation when $T=1820$, instead a feasible solution with the complete hub network of the same hubs can be obtained for $T=1820$.

For the instances with time bound larger than or equal to 1820, the hubs are generally located in the central and eastern regions of Turkey; however, when T is less than or equal to 1800, some of the hubs are selected from the western regions.

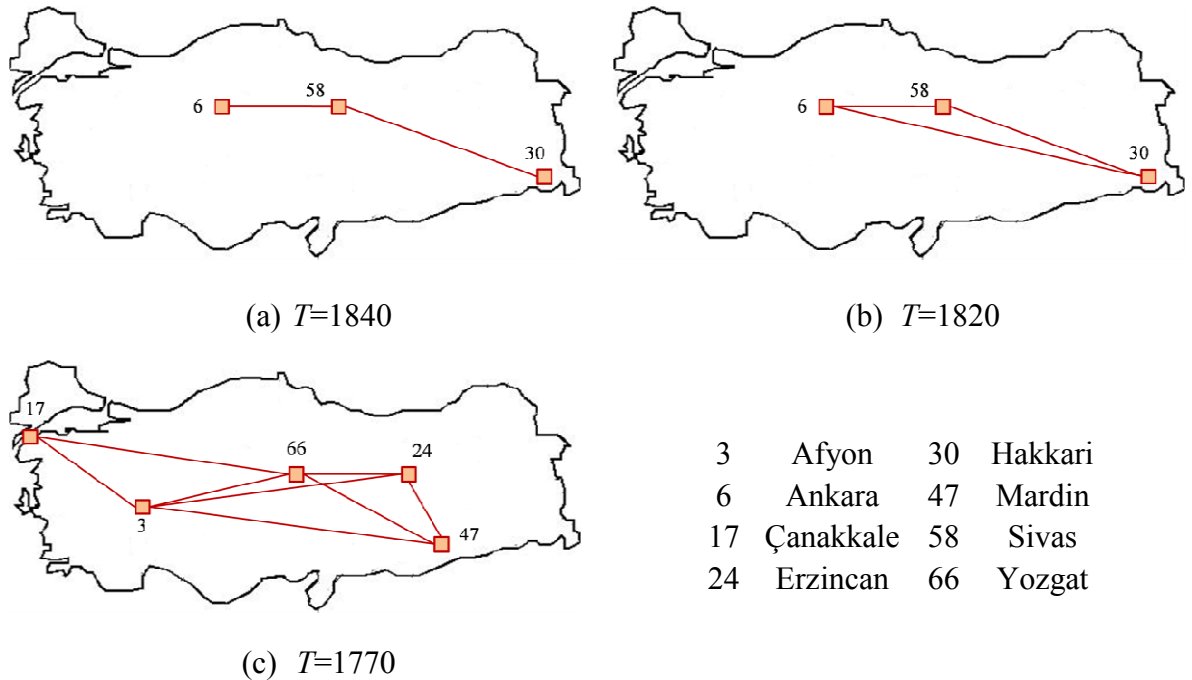


Figure 17: The resulting hub network on TR data for $T=1840$, 1820, and 1770.

Chapter 6

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this thesis, we studied the single allocation hub covering problem over incomplete hub networks. We presented two $O(n^4)$ integer programming formulations for the problem. In the latter one of these models, we defined the four-indexed variable over hub links while in the former one we did not impose such a restriction.

In order to solve realistically sized instances, we proposed a tabu search based heuristic algorithm. Considering the fact that in contrast to other hub location problems, constructing feasible solutions for the hub covering problem, especially with tight time bounds, is challenging, we proposed and tested three different allocation strategies for constructing feasible solutions. In order to decide the size of the starting hub locations set, we developed a preliminary construction stage that is employed at the very beginning of the algorithm. To the best of our knowledge, the heuristic we developed is the first one in the literature that is proposed for the hub covering problem.

We tested both models and the heuristic with CAB data set. We reported only the results with 10 nodes for the first model and compared these results with the ones obtained

from the second model and the heuristic. We observed that the second model outperforms the first model in terms average CPU time requirement. Moreover, the heuristic was able to find optimal solutions in short CPU times on the average; however, we let the heuristic run for longer time in order to achieve stability in the results.

For the instances with 15 and 20 nodes we just performed tests on the second model and the heuristic. Although the heuristic was able to find the best solutions in much shorter time than the model on the average for 15 nodes, due to the same reasoning for stability, we let the algorithm run for longer time. When $n=20$, the model was not able to obtain optimal solutions within two hours for several instances. For those instances we compared our heuristic with the lower bound obtained from the CPLEX.

We tested our heuristic algorithm on the Turkish network as well. The computational times of our heuristic on the Turkish network were reasonable for such a large network, even with tight time bounds. Since we did not know the optimal solutions for the test instances in the Turkish Network, we were not able to comment on the quality of the solutions of our heuristic. One can note that the Turkish network, with 81 nodes, is the largest data set in the literature that is to be tested with incomplete hub network design problems.

In most of the instances that performed experimentations, the hub networks constructed in the best solutions were incomplete. Thus, the results we obtained from these experimentations revealed that designing complete hub networks is not cost effective in many cases.

As a future direction, the multiple allocation case of this problem can be considered in the similar scope of this study. In addition to developing integer programming formulations, the heuristic algorithm we proposed can be adapted to the multiple allocation version of the problem. One can also try to find optimal solutions for larger problems by finding valid inequalities and developing new modeling approaches.

Furthermore, different heuristic and exact algorithms might be adapted for the problem we studied. In the hub center literature, genetic algorithm approach is commonly used in recent studies. The genetic algorithm approach as well as other metaheuristics might be adjusted for the hub covering problem over incomplete hub networks.

BIBLIOGRAPHY

1. Abdinnour-Helm S. (1998), “A hybrid heuristic for the uncapacitated hub location problem”, *European Journal of Operational Research*, 106 (2–3), 489–499.
2. Abdinnour-Helm S. and Venkataramanan M.A. (1998), “Solution approaches to hub location problems”, *Annals of Operations Research*, 78, 31–50.
3. Ahuja R.K., Magnanti T.L., and Orlin J.B. (1993), *Network Flows: Theory, Algorithms and Applications*, Englewood Cliffs, NC: Prentice Hall.
4. Alumur S. and Kara B.Y. (2008), “Network hub location problems: The state of the art”, *European Journal of Operational Research*, 190, 1–21.
5. Aykin T. (1995), “Networking policies for hub and spoke systems with application to the air transportation system”, *Transportation Science*, 29, 201–221.
6. Campbell J.F. (1994a), “Integer programming formulations of discrete hub location problems”, *European Journal of Operational Research*, 72, 387–405.
7. Campbell J.F. (1994b), “A survey of network hub location”, *Studies in Locational Analysis*, 6, 31–49.
8. Campbell J.F. (1996), “Hub location and the p-hub median problem”, *Operations Research*, 44, 923–935.

9. Campbell J.F., Ernst A.T. and Krishnamoorthy M. (2002), "Hub location problems", In: Drezner Z. and Hamacher H.W. editors. *Facility location: applications and theory*, New York: Springer, 373-407.
10. Ernst A.T. and Krishnamoorthy M. (1996), "Efficient Algorithms for the uncapacitated single allocation p-hub median problem", *Location Science*, 4 (3), 139–154.
11. Ernst A.T. and Krishnamoorthy M. (1998), "Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem", *European Journal of Operational Research*, 104, 100-112.
12. Ernst A.T. and Krishnamoorthy M. (1999), "Solution algorithms for the capacitated single allocation hub location problem", *Annals of Operations Research*, 86, 141-159.
13. Ernst A., Hamacher H., Jiang H., Krishnamoorthy M. and Woeginger G. (2002), "Uncapacitated single and multiple allocation p-hub center problems", Unpublished Report, CSIRO Mathematical and Information Sciences, Australia.
14. Ernst A.T., Jiang H., and Krishnamoorthy M. (2005), "Reformulations and computational results for uncapacitated single and multiple allocation hub covering problems", Unpublished Report, CSIRO Mathematical and Information Sciences, Australia.
15. Glover F. (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research*, 13, 533-549.
16. Glover F. and Laguna M. (1997), *Tabu Search*, Kluwer Academic Publishers, Boston.
17. Hamacher H.W. and Meyer T., (2006), "Hub cover and hub center problems", Working paper, Department of Mathematics, University of Kaiserslautern, Gottlieb-Daimler-Strasse, 67663 Kaiserslautern, Germany.

18. Kara B.Y. and Tansel, B.Ç. (2000), "On the single-assignment p-hub center problem", *European Journal of Operational Research*, 125, 648-655.
19. Kara B.Y. and Tansel, B.Ç. (2001), "The Latest Arrival Hub Location Problem", *Management Science*, 125, 648-655.
20. Kara B.Y. and Tansel B.Ç. (2003), "The single assignment hub covering problem: Models and linearizations", *Journal of the Operational Research Society*, 54, 59–64.
21. Klincewicz, J.G. (1991), "Heuristics for the p-hub location problem", *European Journal of Operational Research*, 53, 25-37.
22. Klincewicz, J.G. (1998), "Hub location in backbone/tributary network design: a review", *Location Science*, 6, 307-335.
23. O’Kelly M.E. (1986a), "The location of interacting hub facilities", *Transportation Science*, 20 (2), 92–105.
24. O’Kelly M.E. (1986b), "Activity levels at hub facilities in interacting networks:", *Geographical Analysis*, 18 (4), 343-356.
25. O’Kelly M.E. (1987), "A quadratic integer program for the location of interacting hub facilities", *European Journal of Operational Research*, 32, 393–404.
26. O’Kelly M.E. (1992), "Hub facility location with fixed costs", *Papers in Regional Science*, 71 (3), 293–306.
27. O’Kelly M.E., Bryan D., Skorin-Kapov D., and Skorin-Kapov J. (1996), "Hub network design with single and multiple allocation: A computational study", *Location Science*, 4:125–138.

28. Pamuk F.S. And Sepil C. (2001), "A solution to the hub center problem via a single-relocation algorithm with tabu search", *IIE Transactions*, 33 (5), 399-411.
29. Skorin-Kapov D. and Skorin-Kapov J. (1994), "On tabu search for the location of interacting hub facilities", *European Journal of Operational Research*, 73, 502-509.
30. Skorin-Kapov D., Skorin-Kapov J., and O'Kelly M (1996), "Tight linear programming relaxations of uncapacitated p-hub median problems", *European Journal of Operational Research*, 94, 582-593.
31. Tan P.Z. and Kara B.Y. (2007), "A hub covering model for cargo delivery systems", *Networks*, 49, 28-39.
32. Topcuoglu H., Corut F., Ermis M., Yilmaz G., (2005), "Solving the uncapacitated hub location problem using genetic algorithms", *Computers & OR*, 32 (4), 967-984.
33. Wagner B. (2004), "Model formulations for Hub Covering Problems", *Schriften zur Quantitativen Betriebswirtschaftslehre*, Unpublished.