THE HUB COVERING PROBLEM OVER INCOMPLETE HUB NETWORKS

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

by
Murat Kalaycılar
April 2006

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

Asst. Prof. Bahar Y. Kara (Principal Advisor)

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

Assoc. Prof. Oya Ekin Karaşan

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

Asst. Prof. Mehmet Rüştü Taner

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray

Director of Institute of Engineering and Sciences

ABSTRACT

THE HUB COVERING PROBLEM OVER
INCOMPLETE HUB NETWORKS

Murat Kalaycılar
M.S. in Industrial Engineering
Supervisor: Assist. Prof. Bahar Y. Kara
April 2006

The rising trend in the transportation and telecommunication systems increases the importance of hub location studies in recent years. Hubs are special types of facilities in many-to-many distribution systems where flows are consolidated and disseminated. Analogous to location models, p-hub median, p-hub center and hub covering problems have been studied in the literature. In this thesis, we focus on a special type of hub covering problem which we call as "Hub Covering Problem over Incomplete Hub Networks". Most of the studies in the hub location literature assume that the hub nodes are fully interconnected. We observe that, especially in cargo delivery systems, hub network is not complete. Thus, in this study we relax this fundamental assumption and propose integer programming models for single and multi allocation cases of the hub covering problem. We also propose three heuristics for both single and multi allocation cases of the problem. During the computational performance of proposed models and heuristics, CAB data was used. Results and comparisons of these heuristics will also be discussed.

Keywords: Hub Location, Covering, Mixed Integer Programming, Heuristic

ÖZET

EKSİKLİ ANA DAĞITIM ÜSSÜ AĞLARINDA
KAPLAMA PROBLEMİ

Murat Kalaycılar
Endüstri Mühendisliği Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Bahar Y. Kara
Nisan 2006

Taşımacılık ve telekomünikasyon sistemlerindeki yükselen trend, son yıllardaki Ana Dağıtım Üssü (ADÜ) yerleştirme çalışmalarının önemini artırmaktadır. ADÜ'ler akışların toplandığı ve yayıldığı çoklu dağıtım sistemlerindeki özel tipteki merkezlerdir. Literatürde, p-ADÜ ortanca, p-ADÜ merkez ve ADÜ kaplama problemleri çalışılmıştır. Bu tezde, ADÜ kaplama problemlerinin özel bir durumu olan, "Eksikli ADÜ Ağlarında ADÜ Kaplama Problem"i olarak adlandırdığımız, problem üzerine odaklandık. Literatürdeki ADÜ yerleştirme çalışmalarının çoğunda, tüm ADÜ'lerin birbirlerine tam bağlı oldukları varsayılmaktadır. Bizim gözlemlerimize göre, özellikle kargo dağıtım sistemlerinde, tüm ADÜ'ler arasında bağlantılar bulunmamaktadır. Bunun üzerine, bu temel varsayımı kaldırdık ve problemimizin, tekli ve çoklu atama durumları için tamsayılı programlama modelleri önerdik. Bir de, problemimizin tekli ve çoklu atama durumları için üçer sezgisel çözüm yöntemi önerdik. Önerilen modellerin ve sezgisellerin çözüm performanslarında, CAB verisi kullanılmıştır. Sezgisellerin sonuçları ve karşılaştırmaları da tartışılacaktır.

Anahtar Kelimeler: ADÜ Yerleştirme, Kaplama, Karışık Tamsayı Programlama, Sezgiseller

To my family

ACKOWLEDGEMENT

I would like to express my sincere gratitude to Asst. Prof. Bahar Yetiş Kara for her supervision, guidance, suggestions, encouragement and insight throughout the research.

I would like to extend my special appreciation and gratitude to my family.

I would like to thank to my friends in Bilkent University and my colleagues in ASELSAN Inc. for their continuous support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1. INTRODUCTION

The rising trend in the transportation and telecommunication systems increases the importance of hub location studies in the recent years. Hub location problems arise when it is desirable to consolidate and disseminate flows in many-to-many distribution systems. In hub based systems direct flows between origin and destination pairs are not allowed and the service between these origin/destination pairs is provided by using set of node-hub and hub-hub links. Major application areas of hub locations are airline systems, cargo delivery systems and telecommunication systems.

In standard hub location problems given demand centers and known cross traffic, the problem is finding the location of hub nodes and the allocation of demand nodes. There are three fundamental hub location models namely the p-hub median, the p-hub center and the hub covering problems. Single and multiple allocation versions of these problems are defined in the hub location literature. In single allocation, each demand center should be allocated to only one hub and each demand center should send and receive all its flow through exactly one hub. On the other hand, in multiple allocation version, each demand center can be allocated to many number of hubs and the traffic between an origin and different destinations can be routed through different hubs. Not allowing direct service among an origin and a destination pair is one of the basic assumptions inherent in all the hub location problems. It is assumed that the flow exchange between demand centers are performed via hubs. When the flow departs from the origin, it firstly arrives at origin's assigned hub node. If origin and destination are assigned to the same hub, the flow is sent to its destination point directly from this hub. If they are assigned to different hubs, first of all flow is sent to destination's assigned hub from origin's assigned hub and then from there, it arrives at its destination point. In the hub location problems, it is also assumed that there is a hub-to-hub link between all hub pairs. Therefore, the resulting hub network is a complete network.

In this thesis, we focus on a special type of hub covering problem which is called "The Hub Covering Problem over Incomplete Hub Networks". Different than the classical hub covering problem, the assumption of fully interconnected hub network is relaxed in this study. In the classical hub covering problem, the aim is to find the location of hub nodes and the allocation of demand centers to these hub nodes in such a way that transportation time between any origin destination pair does not exceed a predefined time bound. In addition to these decisions, relaxation of the fully interconnected hub network assumption forces our problem to decide hub-to-hub links between hub nodes. Our problem aims at deciding the location of hub nodes, the allocation of demand centers to these hub nodes and connection of hub nodes where these decisions ensure that transportation time between any origin destination pair does not exceed a predefined time bound, say β. Hub nodes are interconnected by hub-hub links. In order to take into account the economies of scale on these hub-hub links resulting from mass transportation, transportation time on these links are discounted by a factor of α, where $0 \leq \alpha \leq 1$.

This study is motivated by applications arising from three large scale competitive cargo delivery companies in Turkey namely Yurtiçi Cargo, Aras Cargo and MNG Cargo. Before starting this study, we interviewed with these three cargo companies and learned some critical information on cargo delivery. They talked about lots of important concepts about cargo but especially two of them gave hints to us while defining our problem. These concepts are time and money. There is a competition between cargo companies and customers have many different alternatives for choosing the cargo delivery company to send their parcels. The most important criteria that affect the customers are time and money while choosing the company. So, these companies should consider time issue while constructing their delivery network.

After obtaining the basic information about cargo delivery, we started analyzing the network structure of these companies. In their terminology, hubs are called as transfer centers. Each demand center is allocated to at least one transfer center and direct service between two demand centers is not allowed. All incoming and outgoing items are consolidated in transfer centers and then rerouted to destinations via transfer centers. Among the cargo delivery companies, the oldest one is Yurtiçi Cargo and it performs their service via 28 transfer centers. Aras Cargo performs service via 26 transfer centers since 1989. Lastly, MNG Cargo performs service via 22 transfer centers. In the service network of MNG Cargo and Aras Cargo, each demand center is allocated to exactly one transfer center whereas in the service network of Yurtiçi Cargo, demand centers are allocated to more than one transfer center. When we analyze the structure of links between the transfer centers of Yurtiçi Cargo, Aras Cargo and MNG Cargo, we observe that not all transfer centers are connected to each other because establishing a fully interconnected system is costly and also complex. Opening a hub-to-hub link between transfer centers is an investment because you should allocate new trucks on this link and recruit new employees. In cargo delivery system, it is also important to keep the record of the parcels, through their paths. The parcels should be sent to their consignee in the earliest possible time without being lost. So cargo companies avoid establishing complex network structures in order to decrease the number of lost parcels. Thus, they prefer incomplete transfer center networks.

Incomplete hub network structure preference of Turkish cargo delivery companies is the main observation motivating our study because in standard hub location problems, it is assumed that the hub nodes are fully interconnected. According to this critical information, we relax this fundamental assumption and we define the "Hub Covering Problem Over Incomplete Network". In the next chapter, relevant literature about hub location problems will be given. Studies in literature about p-hub median, p-hub center and hub covering problems will be explained in this chapter. Then in Chapter 3, integer programming formulations of single allocation and multiple allocation versions of the problem will be given. Computational study of the proposed models will be given in Chapter 4. This chapter provides computational performance of the models via CPLEX 9.1. The computational performances of the mixed integer models are not very promising. Hence, we propose heuristic solution techniques which are detailed in Chapter 5. Computational performance of these heuristics and comparison of the results of these heuristics can also be found in Chapter 5. The thesis ends with concluding remarks in Chapter 6.

Chapter 2

2. LITERATURE REVIEW

The hub location problem was first posed by O'Kelly(1986). In this paper, the author considered the organization of a single hub network and the organization of systems with two hubs. The author provided some real world examples operating under hub-and-spoke systems. He presented a valuable discussion on cost savings of hub-and-spoke systems.

O'Kelly(1987) developed a quadratic integer program for a general hub location model. In this model, he used decision variables which represent the assignment of the nodes to hubs. The parameters are number of units of flow between nodes and the transportation cost of a unit of flow between nodes. In this paper the author extended his previous model by suggesting a quadratic formulation for the p-hub location problem. In this formulation, there are quadratic cost terms which arise from the inter-hub transactions. Also, inter-hub costs are multiplied with a parameter ≤ 1 to account for economies of scale. Because flows between hubs have a discounted transportation rate arising from bulk transportation. In this model, the assignment of demands to hubs is not a by-product of the location of the hubs. Even if the hub locations are given, the allocation of demand nodes to these hubs still require the quadratic function. So nearest allocation approach does not work here. Since the model is quadratic, the author developed two heuristics called HEUR 1 and HEUR 2. In the first heuristic, by assuming each node is allocated to its nearest hub an upper bound on the objective function can be found with complete enumeration of the locational configurations. In the second heuristic, for each hub location, allocations are determined by evaluating the first or second nearest hub for each node. As a computational study, he used Civil Aeronautics Board(CAB) data which is based on the airline passenger interactions between top 25 U.S cities in 1970 as evaluated by Civil Aeronautics Board. Finally, the author concluded that heuristics are the practical approach for solving the quadratic hub location problems.

O'Kelly(1992) included the fixed costs of opening hubs. He developed a model scheme for hub network planning in his recent researches but he ignored the fixed costs of opening facilities. So by including the fixed facility costs, the number of hubs became a decision variable. The problem is finding the optimal number and locations of hubs, and the assignments of demand nodes to hubs. He formulated the problem and for solving the problem he analyzed four special cases of the model. In the first case, he ignored the quadratic components of the systems interactions costs and the problem became a simple plant location problem. In the second case, he took very large fixed cost and the problem became 1-median problem. Then in the third case, he ignored the fixed costs and hubs are opened at all potential locations. Finally in the last case, he analyzed the general problem and devised a two-step procedure to solve the problem. In the first step, by using heuristics, a good upper bound was found. And then in the second step, a lower bound is developed by underestimating the quadratic contributions to the objective. Finally, the author gave computational results by using CAB data.

The first review and synthesis paper was introduced by O'Kelly and Miller(1994). In this paper, they reviewed some analytical research papers and gave some empirical examples. Authors refer to hubs as major sorting or switching centers in many-to-many distribution systems. According to their explanations, a hub location problem involves the decisions of: i) finding optimal locations of hubs, ii) assignment of nodes to the hubs, iii) determining the linkages between hubs and iv) routing the traffic between origin-destination pairs. Then they mentioned the properties of standard hub models. These properties can be summarized as; i) all nodes are assigned to a single hub, ii) hub nodes are fully interconnected(complete network) and iii)non-hub to non-hub linkages is not allowed. After giving the properties, they classify a hub network system. They classified the system as single/multi allocation; full/partial hub interconnection; allowed/not allowed direct service between demand centers. By combining these classes, they obtained eight different network design protocols and gave brief empirical examples of these network design protocols.

Campbell(1994) proposed mathematical programming formulations for four standard discrete hub location problems which are the p-hub median problem, the uncapacitated hub location problem(hub location with fixed costs), p-hub center problem and hub covering problem. The author considered both single allocation and multiple allocation cases of these problems.

Campbell firstly formulated the p-hub median problem. The author observed that if there are no capacity constraints on the links, then an optimal solution will have all X_{ijkm}(fraction of flow from origin i to destination j that is routed via hubs at locations k and m in that order) equal to zero or one because each origin-destination pair will select the shortest route. Then the author incorporated flow thresholds and fixed costs for the spoke links to his formulation. He showed that when the spoke flow thresholds and fixed costs are taken as zero, problem reduces to p-hub median problem. He also showed that if the spoke flow thresholds and fixed costs are large, the result will be single allocation. For the p-hub median problem, he finally mentioned that when p=1, multiple allocation is not possible and the 1-hub median problem is identical to the 1-median problem. Secondly, the author gave the basic formulation of the uncapacitated hub location problem which differs from the p-hub median problem in that the number of hubs is not specified and non-negative fixed costs are associated with each potential hub location. He incorporated flow thresholds and fixed costs for the spoke links to his formulation and he showed that with maximum flow thresholds for the spokes is the single allocation uncapacitated hub location problem. Being different than O'Kelly(1992), the author stated that for each node, flow threshold is total flow. By using this information, the author proposed a linear IP for the single allocation uncapacitated hub location problem whereas O'Kelly(1992) proposed a quadratic IP for the same problem.

The author also explained p-hub center problem in three types. In the first type, the aim is to find a set of hubs such that the maximum cost for any o-d pair is minimized. A second type aims to select a set of hubs that minimizes the maximum cost for movement on any single link. Finally, the last type involves selecting a set of hubs that minimizes the maximum cost for movement between a hub and an origin/destination. Then he gave the basic formulations of these types and incorporated flow thresholds and fixed costs for the spoke links to these formulations.

The author finally defined the hub covering problem in three different versions as in p-hub center problem. First version stated that o-d pair (i,j) is covered by hubs *k* and *m* if the cost from *i* to *j* via *k* and *m* does not exceed the predefined bound. Second version stated that o-d pair (i,j) is covered by hubs k and m if the cost for each link in the path from *i* to *j* via k and m does not exceed the predefined bound. Final version is similar with second version but considers spokes only. In this version, o-d pair (i,j) is covered by hubs *k* and *m* if each of the hub-origin/destination links meets the predefined bound. This one corresponds to the notion of coverage in the facility location literature. He developed basic formulations in the two ways. In the first formulation he defined non-negative penalty cost for uncovered pairs and in the second formulation he maximized the demand covered with a given number of hub facilities. Finally, he incorporated flow thresholds and fixed costs for the spoke links to these formulations.

So far, we explained the pioneering research in the literature. These studies are the fundamentals of the hub location studies. In the next sections, hub location literature is analyzed under three main subjects: *p-Hub Median* and *Hub Location Problems with Fixed Cost (Uncapacitated Hub Location Problem), p-Hub Center* and *Hub Covering*.

2.1 The p-Hub Median Problem and The Hub Location Problem with Fixed Costs

Klincewicz(1991) provided additional heuristics for the p-hub median problems. The author mentioned that the p-hub median problems and the problem of assigning demand nodes to hubs are in the class of NP-Complete problems. Because of that reason, he tried to develop efficient heuristics for the p-hub median problem rather than optimal algorithms. He developed two types of exchange heuristics which are called single exchange(one-at-a-time) and double-exchange(two-at-a-time). These heuristics work with an incumbent set of hubs and systematically substitute other nodes for the incumbents based on local improvement measures. The author compared these heuristics with clustering heuristics and enumeration heuristics based on previous work in the literature and concluded that the double-exchange heuristic showed great promise as a solution technique for the p-hub median problems.

Skorin-Kapov and Skorin-Kapov(1994) provided a tabu search algorithm (TABUHUB) for the problem. By using the tabu search approach, the authors obtained a new heuristic method which weighs equally the locational as well as allocational decisions of the problem. The authors performed computational studies and compared their method with heuristics, HEUR1 and HEUR2, which were developed by O'Kelly(1987). Finally, they concluded that TABUHUB is better than these heuristics.

O'Kelly, Skorin-Kapov and Skorin-Kapov(1995) found a lower bound for the p-hub median problem with fixed costs utilizing O'Kelly's(1992) formulation. The authors assume that distances satisfy the triangle inequality. Their lower bound is based on a linearization of the problem and its modification obtained by incorporating a known heuristic solution. Instead of ignoring the quadratic term completely, they developed a lower bound by adding underestimate of the costs of the interfacility flows. The authors used CAB data in their computational analysis. The novel approach of using a known heuristic solution to derive a lower bound in all cases reduced the difference between the upper and lower bounds.

Skorin-Kapov, Skorin-Kapov and O'Kelly(1996) presented new mathematical formulations for multiple and single allocation p-hub median problems. The authors developed tighter linear programming relaxations. After solving their own formulation, authors compare the results with Campbell(1994). Comparisons show that the tight LP relaxations are achieved without increasing the number of variables for single allocation. For the multiple allocation case, the number of constraints actually decreased. Authors tested their LP relaxations by using CAB data and obtained integer LP solutions in 96% of the instances. For the LP solutions which are not integer, solutions are less than 1% below the optimal in single allocation and less than 0.1% below optimal in multi allocation.

Campbell(1996) mainly focused on the p-hub median problem. The author presented the mathematical programming formulations for single and multiple allocation p-hub median problems and he tried to solve single allocation problems by using the solution of multiple allocation problem. He observed that solving multiple allocation hub location problems provides new opportunities for solving single allocation problems and solution of multiple allocation gives a lower bound for single allocation. In order to solve multi allocation problem, he first developed a greedy-interchange heuristic. In this heuristic, optimal hub pairs are found by enumeration. After finding the solution for multiple allocation, he developed two heuristics, MAXFLO and ALLFLO, which derive solutions to the single allocation p-hub median problem from the solution of greedy-interchange heuristic. Lastly, he compared the results of these heuristics with the heuristics in Klincewicz(1991) and concluded that the author's proposed heuristics generally perform well in comparison with Klincewicz(1991).

O'Kelly, Bryan, Skorin-Kapov and Skorin-Kapov(1996) considered the exact solutions for hub network design with single and multiple allocation. This paper is a computational follow-up to Skorin-Kapov et al.(1996) and their formulations were similar as in Skorin-Kapov et al.(1996). But for multiple allocation there were two major changes. Authors used symmetric flow data and the ranges of summations can be trimmed to avoid impractical routes. After formulating the problems, they performed numerical studies, which include a discussion of the role of the interhub discount factor(α) and the relaxation of the strict single hub allocation policy to give a more general multiple hub location rule, by using CAB data. Using various values of p and variations in α between zero and one, single and multiple allocation problems were solved to optimality. As a result, they showed that for a given number of hubs, total cost for single allocation is greater than total cost for multi allocation. Also, as α decreases, total cost for the two models decreases.

Ernst and Krishnamoorthy(1996) studied the uncapacitated single allocation p-hub median problem whose mathematical formulation requires fewer variables and constraints than the formulations in the literature. Since the authors did not keep track of every flow of traffic between pairs of nodes separately, their proposed formulation decreases the problem size. They formulated their problem as a multi-commodity flow problem. In this paper, their aim was finding efficient algorithms for this problem. The authors firstly developed a heuristic based on simulated annealing. The authors only considered feasible solutions and started with a randomly generated initial solution. Then they defined a cluster as the set of nodes allocated to the same hub and generated a neighbor solution. In order to generate neighborhood solutions, they used three types of transitions:

- Change the allocation of a randomly chosen non-hub node to a different cluster.

- Change the location of the hub within a randomly chosen cluster to a different node in the cluster

- In the special case where a cluster contains only a single node, pick a non-hub node at random and make it hub then allocate the previous single hub-node to a randomly chosen cluster.

The authors selected one of these transitions according to initially set probabilities. If at the end objective is improved, they accepted the solution. If not, they might accept according to Boltzmann's probability which depends on the temperature that is periodically updated. Then they used the solution of simulated annealing as an upper bound to develop an LP-based branch-and-bound solution method. Authors tested their algorithms with data set from Australia Post(AP). It consists of 200 nodes representing postcode districts. The authors obtained optimal solutions in a small amount of computational time by using SA for most of the test problems by using smaller instances of AP data set.

Ernst and Krishnamoorthy(1998) studied on exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. Authors described an efficient heuristic algorithm based on shortest path and explicit enumeration algorithm. Then, they presented a new mixed integer linear program for the multiple allocation p-hub median problem. For solving the problem, they used an LP based branch-and-bound method. Authors strengthened the LP lower bound by adding valid inequalities. Lastly, they obtained results by using CAB data and concluded that heuristics found solutions in a reasonable time.

Ernst and Krishnamoorthy(1999) focused on the capacitated single allocation hub location problem with fixed costs. This is the first study that consider capacity restriction on hubs. This study was motivated by real-world application in postal delivery network design. There are capacity restrictions on hubs, because in a postal delivery, in order to meet time constraints, only a limited amount of mail can be sorted at each sorting center(hub).The authors allowed their model to choose the number and location of hubs based on the fixed costs of establishing them. Cost function consists of collection, transfer and distribution of mails. After formulating the problem, the authors developed two heuristics based on simulated annealing and random descent, for obtaining upper bounds. They also obtained optimal solutions by using branch-and-bound method with initial upper bound provided by the heuristics. Finally, authors performed some numerical studies and found that random descent based algorithm is preferable on small to medium sized problems because it is easier to implement and it provides better performance than the simulated annealing based algorithm.

Pirkul and Schiling(1998) worked on finding an efficient procedure for designing single allocation p-hub median problem. The authors focused on the model of Skorin-Kapov et al. 1996. Their aim was finding a method that produces bounds that help measure the quality of the solutions obtained. They began with applying standard lagrangian relaxation. First of all they chose constraint sets to relax and by using the lagrangian multipliers they separated the problem into two subproblems(SUB-1 and SUB-2). Then they used subgradient optimization to obtain a good set of multipliers and bounds to problem. They found p hubs from SUB-1 and assigned each node to its nearest hub. By solving SUB-2, quality of the bound derived in SUB-1 improved. This allowed them to find optimal solutions in 83 out of the 84 test problems and reduced average gaps to nearly zero.

Sasaki, Suzuki and Drezner(1999) focused on the 1-stop multiple allocation p-hub median problem. 1-stop model is the special case of 2-stop model which is the airline hub-and-spoke system. Authors formulated this problem as multiple allocation p-hub median problem. Firstly they proposed a branch-and-bound algorithm that uses lagrangian relaxation by dualizing the constraint on the number of hubs to solve the problem. The authors used depth-first search rule. Since branch-and-bound algorithm is an implicit enumeration algorithm, it takes a lot of time. This situation motivated them to develop a greedy-type heuristic. They tested these algorithms with CAB data and some random data. Authors concluded that their algorithms work better than the nested-dual algorithm, particularly for relatively small problems.

Ebery, Krishnamoorthy, Ernst and Boland(2000) focused on the capacitated multiple allocation hub location problem. Authors developed mixed integer linear programming formulation for the problem and constructed an efficient heuristic algorithm which is based on shortest path algorithms. At the beginning of the algorithm, there is a set of uncapacitated hubs. Then by using shortest path algorithm, allocations of demand nodes to hub nodes are made optimally without considering the capacities. If any feasible solution, which satisfies the capacities, is obtained, the solution is recorded. If the solution does not satisfy the capacity constraints, they reroute excess flow using a heuristic procedure. The procedure is repeated with a different hub set and an upper bound is obtained. Then, the upper bound is incorporated into a linear programming based branch-and-bound solution procedure. Finally, authors performed computational study with the heuristic and exact methods.

Mayer and Wagner(2002) considered a special type of uncapacitated multiple allocation hub location problem and developed a technique which is called HubLocator, to find an optimal solution for this problem. Their technique was based on branch-and-bound and was performed in two steps. In the first step, the dual ascent procedure was used to solve the dual of the LP relaxation of disaggregated model for uncapacitated multiple allocation hub location problem. Then in the second step, by using the dual solution obtained in the first step, a dual solution of the relaxed aggregated formulation is determined. Additionally, specially tailored dual ascent technique was used to find tighter lower bounds. Upper bounds are found by applying some complementary slackness conditions. Authors tested HubLocator on CAB an AP data sets. Also for comparing their technique with CPLEX, they used Ernst and Krishnamoorthy's(1996) formulation. Finally, they concluded that in a reasonable amount of time optimal solutions for problems with up to 40 nodes can be found.

Elhedhli and Hu(2005) studied on a special type of p-hub median problem where congestion effects at a certain hub are taken into account. Firstly, the authors proposed a nonlinear large scale mixed integer congestion model. By using piecewise linear functions, authors linearized the problem. During the linearization process, they approximated the nonlinear cost function with a set of tangent hyperplanes. Then Lagrangian approach was applied where the lower bound is calculated using a subgradient algorithm and use the solution from one of the subproblems to find a heuristic solution. CAB data was used to test the proposed model and the algorithm gave the solutions within 1% optimality in reasonable time.

Campbell et al.(2005a, 2005b) considered the hub arc location problems which generalize the p-hub median problem. They view a hub location and network design problem from a hub arc location perspective. In this problem, instead of deciding the hub locations, they focused on locating hub arcs and access arcs. Two end points of these hub arcs are considered as hubs. They also mentioned that when there is a lot of hubs in the network, there is no need to connect each hub to all of the other hubs. So, the authors relaxed fully interconnected hub network assumption. They introduced two new concepts to hub location problems which are hub arcs and bridge arcs. In this problem the assumption of using discount factor between all hubs was also relaxed. Discount factor is equal to one on a bridge arc between two hubs. Unit flow cost was only discounted on hub arcs. They provided a mixed integer linear program for the hub arc location problem which is to locate q hub arcs to minimize the total flow cost. Different variations of the hub arc location model are described in Campbell et al.(2005a, 2005b) They used two different optimal solution approaches which are solving directly with CPLEX 6.6 and solving by using enumeration-based algorithm. For most of the instances, the enumeration-based algorithm gave better solutions than integer programming approach.

Sohn and Park(1997) and (1998) focus on the allocation problem only. Sohn and Park(1997) considered the two-hub location problem(p=2). In this problem, locations of demand nodes and hub locations are known, and aim is allocating each demand node to one of the two hubs. The authors developed a quadratic 0-1 integer program of this problem. Then they transformed quadratic formulation into a linear program. Firstly, quadratic integer program was transformed into a 0-1 mixed integer program. Then authors showed that the linear programming relaxation of mixed integer program gives a polytope whose extreme points are all integral. Because of that they can solve linear program to find an optimal solution to mixed integer program and quadratic integer program. They also showed that this problem can be transformed into a minimum cut problem.

Sohn and Park(1998) considered the general case of the allocation problem. The authors studied on the uncapacitated multiple and single allocation p-hub median problems. Real life situations motivated them to focus on methods to find optimal solutions for the allocation problems with fixed hub locations. Because in real situations, the hub locations are fixed for some time interval as a result of long term lease contract and cost of moving hubs. First of all, they worked on multiple allocation p-hub median problems. The authors used the formulation of Skorin-Kapov et al. 1996. They showed that this problem can be solved in polynomial time by the shortest path algorithm when number of hubs are fixed. Once hub locations are fixed without any capacity restrictions, then each pair will be routed from its shortest path via the hub node. Secondly, they worked on single allocation p-hub location problem. They modified the formulation of Skorin-Kapov et al. (1996) by fixing hub locations and hub costs. They also reduced the number of variables and constraints of the formulation. During the computational performance of the proposed model, in 74240 out of 74260 cases they obtained integer solutions.

To conclude, we can say that most of the studies in the p-hub median and hub location problem with fixed cost literature aim to develop heuristics for larger problems and achieve closer results to optimum in a reasonable time. Remaining ones aim to linearize the quadratic integer program which is developed by O'Kelly(1987). The authors tried to obtain closer results to optimum by using heuristics: Greedy-interchange, local neighborhood search, tabu serach and simulated annealing. Among these heuristics, tabu search and simulated annealing seem better than the others. Among the linearizations, Ernst and Krishnamoorthy's(1996) linearization is better than the others according to CPU time requirements.

2.2 The p-Hub Center Problem

O'Kelly and Miller(1991) considered the single facility minimax hub location problem which arises when the most costly interaction through the system is required to be as inexpensive as possible. The authors reviewed some solution strategies such as discrete locational evaluation, Helly's Theorem, a graphical approach, linear programming feasibility and Drezner's round trip location algorithm[Drezner(1982)]. Among these solution strategies, Drezner's round trip location algorithm is the best one according to solution accuracy and computational cost. Finally, the authors applied this solution strategy to hub location problems for air passenger flows between US cities.

Kara and Tansel(2000) considered the single assignment p-hub center problem. Firstly, the authors developed a combinatorial formulation of this problem and proved that it is NP-Hard. Then they gave the mathematical formulation of Campbell(1994). The authors presented three different linearizations of Campbell's model. First one is LIN1 which is the Campbell's original linearization. Second one is LIN2 which is the adaption of Skorin-Kapov's(1996) linearization to the p-hub center problem. Last one is the authors' own linearization which is called LIN3. They tested all of these linearizations with CAB data and they showed among LIN1, LIN2 and LIN3, LIN3 is the best but cannot solve all CAB instances. Then they reformulated the p-hub center problem from a different perspective. Authors obtained better performance from the linearization of the new formulation.

Kara and Tansel(2001) studied on the minmax version of the latest arrival hub location problem. Unloading, sorting, handling and reloading take a lot of time for cargo delivery systems. There is also an additional waiting time depending on how late the latest arriving unit is. This study took into account the transient times at hubs in addition to the flight times. The authors referred this problem as the latest arrival hub location problem. First of all the authors gave a combinatorial formulation of the problem. Then authors proved the NP-Hardness of the problem. After giving the complexity of the problem, they developed nonlinear mixed integer programming formulations for the minimax version of the latest arrival hub location problem and linearized the problem. After that they considered time zones and modified their model to capture the effects of different time zones. Finally, they tested the linear integer model using CPLEX 5.0 based on 60 instances of the standard CAB data set.

2.3 The Hub Covering Problem

Kara and Tansel(2003) studied on the single-assignment hub covering problem after the first proposition of the problem by Campbell(1994). In this problem, p is a variable and it is to be minimized while making sure that all trip times between origin-destination pairs are within predetermined bounds. The authors presented the combinatorial formulation of the hub covering problem and developed a nonlinear integer programming model. Then they gave linearization of their model. This was a strong linearization in the sense that there is no change in the dimension of the space, the feasible sets are exactly the same and the optimal sets are the same. After this linearization, the authors provided three different linearizations for Campbell's(1994) model. Finally, they performed computational study with the CAB data set using CPLEX 5.0 and concluded that the linear version of the proposed model performed better than the most successful linearization of the previous model both in terms of average and maximum CPU times as well as in core storage requirements.

Wagner(2004) proposed model formulations for single and multiple allocation cases of hub covering problems. First of all, he proposed single and multiple allocation hub covering models with quantity-independent transport times and then for single allocation case, he proposed a model which includes quantity-dependent time functions. He tested his models on CAB and AP data set. Finally, he showed that non-increasing function for transport times can be used in hub covering problems.

Chapter 3

3. MODEL FORMULATION

When we look at the hub location literature, we see that most of the authors have worked on the p-Hub Median problem. There are only a few studies considering the Hub Covering Problems. As it is mentioned before, hub covering problems arise especially in cargo delivery applications and the aim of this problem is to select a minimum number of hubs such that the transportation time between any origin destination pair does not exceed a predefined time bound.

In this thesis, we focus on a different version of hub covering problem that is not found in the literature. Most of the studies in the hub location literature assume that the hub nodes are fully interconnected. Observations on network structures of Turkish cargo delivery companies show that hub network is not complete. For example MNG Cargo has 22 transfer centers and 50 hub-to-hub links between these transfer centers. If they use fully interconnected network structure, they should have 231 hub-to-hub links. They prefer to construct an incomplete hub network because establishing this kind of system is costly and also complex. Also, MNG Cargo and Aras Cargo prefer single allocation in their service network whereas Yurtiçi Cargo prefers multiple allocation in its service network. Thus, in this study we relax this fundamental assumption and propose integer programming models for single and multi allocation cases of the hub covering problem over incomplete hub networks.

The problem is posed on a graph $G=(N,A)$ with node set $N=\{1,...,n\}$ where each node represents origins, destinations and potential locations of hubs; and A is the set of arcs between the nodes of the given network.

The arcs between hubs are called as hub-to-hub links. Transportation time on these links are discounted by a factor and transportation time between any origin destination pair shoould not exceed a predefined time bound. Opening hubs and hub-to-hub links brings cost to the system. According to these explanation, the parameters are:

α**:** discount factor for hub-to-hub transportation

β**:** predetermined bound that imposes a deadline for travel time between any pair of nodes

δ**:** cost of opening a hub

γ**:** cost of having a connection between two hubs

T_{ik}**:** the travel time between node i and k, $i,k \in N$

3.1 Single Allocation Hub Covering Model Over Incomplete Network

In single allocation version of the problem, following decision variables are used:

$$X_{ik}:\begin{cases}1 & \textit{if} \text{ node } i \text{ is served from a hub at node } k\\ 0 & \text{otherwise}\end{cases}$$

$$Z_{kl}:\begin{cases}1 & \textit{if} \text{ there is a link between hubs } k \text{ and } l\\ 0 & \text{otherwise}\end{cases}$$

$$M_{kl}^{ij}:\begin{cases}1 & \textit{if} \text{ hub arc } k\text{-}l \text{ is used on the path from } i \text{ to } j \text{ where } k \text{ and } l \text{ are hubs}\\ 0 & \text{otherwise}\end{cases}$$

Observe that $X_{kk}=1$ when node k is a hub.

And the resulting model is: (P-S)

$$\min \quad \sum_{i} \delta X_{ii} + \sum_{k<l} \gamma Z_{kl} \tag{1}$$

s.t

$$\sum_{j} X_{ij} = 1 \qquad \forall i \tag{2}$$

$$X_{ij} \leq X_{jj} \qquad \forall i \neq j \tag{3}$$

$$2Z_{kl} \leq X_{kk} + X_{ll} \qquad \forall k \neq l \tag{4}$$

$$M_{kl}^{ij} \leq Z_{kl} \qquad \forall i \neq j, k \neq l, i \neq l, j \neq k \tag{5}$$

$$Z_{ij} \leq M_{ij}^{ij} \qquad \forall i \neq j \tag{6}$$

$$X_{ii} = \sum_{k:k\neq i} M_{ik}^{ij} + X_{ji} \qquad \forall i \neq j \tag{7}$$

$$X_{ik} + \sum_{l:l\neq k, l\neq j} M_{lk}^{ij} = X_{jk} + \sum_{l:l\neq k, l\neq i} M_{kl}^{ij} \qquad \forall i \neq j, i \neq k, j \neq k \tag{8}$$

$$\sum_{k} T_{ik} X_{ik} + \sum_{k:k \neq j} \sum_{l:l \neq k, l \neq i} \alpha T_{kl} M_{kl}^{ij} + \sum_{l} T_{jl} X_{jl} \leq \beta \quad \forall i \neq j \tag{9}$$

$$X_{ij} \in \{0,1\} \quad \forall i, j \tag{10}$$

$$Z_{kl} \in \{0,1\} \quad \forall k \neq l \tag{11}$$

$$M_{kl}^{ij} \in \{0,1\} \quad \forall i \neq j, k \neq l, i \neq l, j \neq k \tag{12}$$

Objective is minimizing the total cost of opening hubs and cost of establishing the hub-to-hub links. Constraints (2) and (10) ensure that each node is assigned to exactly one hub. Constraint (3) states that any node can be assigned to a hub node only. Hub-to-hub links take place between two hub nodes due to Constraint (4). Constraint (5) states that while sending a flow between origin/destination pair *i-j*, if flow passes through a link *k-l*, this link must be a hub-to-hub link. Constraint (6) states that if there is a hub-to-hub link between two hubs, there should be a direct transportation between these two hubs. Constraints (7) and (8) are flow balance constraints. Constraint (7) is written for a fixed i-j pair. If node *i* is a hub node, then the left hand side of Constraint (7) is 1 which implies that either the destination node *j* is assigned to hub *i* (X_{ji}=1) or it triggers one of the M_{kl}^{ij} values. So that the flow from *i* to *j* is sent to another hub *k*. Constraint (8) states that if there is an incoming flow to any of the hub(say hub *k*) from demand center or another hub, there should be an outgoing flow from that hub(hub *k*) to another hub or a demand center. Next is a cover constraint which ensures that transportation time between any origin destination pair does not exceed the predefined time bound (β) for a fixed *i-j* pair. First part of the covering constraint represents the time between origin *i* and the first hub *k*. Then second part represents the total transportation time of the flow between hubs that are used on the path from origin *i* to destination *j*. Recall that due to

Constraints (7) and (8), only these M_{kl}^{ij}s are 1 which are the hub-to-hub links on the path from i to j. Last part represents the time between the last hub l and destination j. Finally, last three constraints are for binary restrictions. In this model, there are $2n^2+n^4$ binary variables and the number of constraints are in the order of $n^4+n^3+5n^2+n$ where n is number of nodes.

3.2 Multiple Allocation Hub Covering Model Over Incomplete Network

In multiple allocation version of the problem, following decision variables are also used in addition to Z_{kl} and M_{kl}^{ij} defined in 3.1:

$$Y_k : \begin{cases} 1 & \textit{if } k \text{ is a hub} \\ 0 & \text{otherwise} \end{cases}$$

$$X_{ikj} : \begin{cases} 1 & \textit{if } k \text{ is the first hub on the path from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

and the resulting model is: (P-M)

$$\min \quad \sum_k \delta Y_k + \sum_{k<l} \gamma Z_{kl} \qquad (13)$$

s.t (5), (6), (11) and (12)

$$X_{ikj} \le Y_k \qquad \forall i \ne j, \forall k \qquad (14)$$

$$X_{ikj} + Y_i \le 1 \qquad \forall i \ne j, i \ne k \qquad (15)$$

$$2Z_{ij} \leq Y_i + Y_j \qquad \forall i \neq j \qquad (16)$$

$$\sum_{l:l\neq i} M_{il}^{ij} + \sum_{l} X_{ilj} = 1 \qquad \forall i \neq j \qquad (17)$$

$$\sum_{l:l\neq j} M_{lj}^{ij} + \sum_{l} X_{jli} = 1 \qquad \forall i \neq j \qquad (18)$$

$$\sum_{k:k\neq l,k\neq j} M_{kl}^{ij} + X_{ilj} = \sum_{k:k\neq l,k\neq i} M_{lk}^{ij} + X_{jli} \qquad \forall i \neq j, i \neq l, j \neq l \qquad (19)$$

$$\sum_{k:k\neq j} \sum_{l:l\neq k,l\neq i} \alpha T_{kl} M_{kl}^{ij} + \sum_{k:k\neq i} (T_{ik} X_{ikj}) + \sum_{l:l\neq j} (T_{jl} X_{jli}) \qquad \forall i \neq j \qquad (20)$$

$$+ T_{ij}(X_{iij} + X_{ijj}) \leq \beta$$

$$Y_i \in \{0,1\} \qquad \forall i \qquad (21)$$

$$X_{ikj} \in \{0,1\} \qquad \forall i \neq j, \forall k \qquad (22)$$

Objective is the same as in the single allocation version of the problem which is minimizing the total cost of opening hubs and cost of establishing the hub-to-hub links. Constraint (14) states that if node i is assigned to node k, then node k must be a hub node. Constraint (15) guarantees that if origin is a hub node, another hub node cannot be a first hub between origin/destination pair i-j. Constraint (16) ensures that if there is a hub-to-hub link between any two nodes, these nodes must be hub nodes. Constraints (17), (18) and (19) are the flow balance constraints. First one represents the departure of the flow from origin (node i). There must be a departure from each origin and this origin node is either a demand center or a hub node. Similarly, constraint (18) represents the arrival of the flow to destination (node j). The flow must reach each destination node and this destination node is either a demand center or a hub node. Constraint (19) provides a flow balance between ingoing and outgoing links. Constraint (20) is the cover constraint which ensures that transportation time between any origin destination pair does not exceed a predefined time bound (β) for a fixed *i-j* pair. First part of the covering constraint represents the total transportation time of the flow between hubs that are used on the path from origin *i* to destination *j*. Second and third parts of the covering constraint represent the time between origin *i* and the first hub *k* and the time between the last hub *l* and destination *j*. The forth term represents the transportation time between origin/destination pair *i* and *j* where either node *i* or node *j* is a hub node and non-hub one is assigned to the other one. Constraints (21) and (22) are for the binary restrictions of the additional variables. In this model, there are $n^4+n^3+n^2+n$ binary variables and the number of constraints are in the order of $n^4+3n^3+5n^2$ where *n* is number of nodes.

Chapter 4

4. COMPUTATIONAL PERFORMANCE

We test the computational performance of the two proposed models by using CAB data set which is generated from the Civil Aeronaustics Board Survey of 1970 passenger data in the United States [O'Kelly(1987)]. The cities in the CAB data set are listed in Table 1.

01	Atlanta	**10**	Houston	**19**	Phoenix
02	Baltimore	**11**	Kansas City	**20**	Pittsburgh
03	Boston	**12**	Los Angeles	**21**	St. Louis
04	Chicago	**13**	Memphis	**22**	San Fransisco
05	Cincinnati	**14**	Miami	**23**	Seattle
06	Cleveland	**15**	Minneapolis	**24**	Tampa
07	Dallas-FW	**16**	New Orleans	**25**	Washington DC
08	Denver	**17**	New York		
09	Detroit	**18**	Philadelphia		

Table 1: Cities in CAB Data

Customarily the northwest 10, 15, 20 and 25 nodes of this set are taken for different values of *n*. The discount factor α is taken from the set {0.2, 0.4, 0.6, 0.8 and 1.0}. For both single and multiple allocation versions of the problem, 80 instances are used to test the performance of the models. In the tests, β values, which were generated by Kara and Tansel(2003), are used. They generate these β values by taking optimal objective function values of the p-hub center problem. For each value of α and the number of nodes *n*, β values given in the following table are used:

n	**α**				
	0.2	**0.4**	**0.6**	**0.8**	**1.0**
10	1425	1627	1671	1744	1839
	1117	1185	1387	1589	1791
	811	970	1148	1457	1770
	736	863	1079	1413	1766
15	2004	2019	2103	2424	2611
	1638	1741	1844	2165	2610
	1324	1436	1756	2100	2605
	1149	1287	1560	2080	2600
20	1851	2067	2255	2493	2611
	1549	1744	1996	2264	2605
	1356	1473	1835	2154	2601
	1162	1386	1663	2118	2600
25	2136	2401	2557	2713	2826
	1913	2099	2336	2552	2762
	1617	1881	2184	2457	2726
	1346	1597	2002	2307	2725

Table 2: β values used

We present optimal hub locations, optimal hub arcs and CPU times reported by CPLEX 9.1 running on a computer which has 12*400 MHz speed and 3 GByte memory. Optimal results for single and multiple allocation versions of the problem for n=10 are given in Tables 3 and 4.

	β	SINGLE-ALLOCATION		
		# of hubs	# of hub arcs	CPU time(sec)
α=0.2	1425	2	1	169,1
	1117	3	2	722,4
	811	4	3	403,38
	736	5	4	15940,43
α=0.4	1627	2	1	657,09
	1185	3	2	4362,42
	970	4	4	14159,73
	863	5	4	1992,13
α=0.6	1671	2	1	776,44
	1387	3	3	8242,43
	1148	4	5	2147,93
	1079	5	6	2391,12
α=0.8	1744	2	1	430,74
	1589	3	3	16447,63
	1457	5	5	10688,67
	1413	5	7	12410,29
α=1.0	1839	1	0	3,06
	1791	3	3	10234,52
	1770	4	5	75954,63
	1766	4	8	35270,76

Table 3: Optimal Results of P-S for n=10

	β	MULTI-ALLOCATION		
		# of hubs	# of hub arcs	CPU time(sec)
α=0.2	1425	2	1	1207,05
	1117	3	2	1489,47
	811	4	3	2474,22
	736	5	4	60614,24
α=0.4	1627	2	1	801,45
	1185	3	2	1192,54
	970	4	3	4324,47
	863	5	4	1874,48
α=0.6	1671	2	1	444,2
	1387	3	2	3214,78
	1148	4	4	714,13
	1079	5	5	1318,08
α=0.8	1744	2	1	937,02
	1589	3	2	906,37
	1457	4	3	678,79
	1413	4	4	995,24
α=1.0	1839	1	0	4,09
	1791	2	1	13,43
	1770	2	1	37,73
	1766	2	1	119,61

Table 4: Optimal Results of P-M for n=10

For n=10, in single allocation case, CPU times are in the range between 3.06 seconds and 21.1 hours. When we compare the CPU times of P-S and P-M, in 13 out of 20 cases P-M generates faster solutions. But for tight β values, its performance usually goes down. For example, when α=0.2 and β=736, CPU time reaches 16.8 hours.

As it can be understood from the CPU times of P-S and P-M, we could not manage to generate optimum results with CPLEX in reasonable time for the instances which have more than 10 nodes. Hence, in order to find good quality solutions for larger instances, heuristics are developed.

In the next chapter, detailed explanations of heuristics, results of these heuristics and comparisons of them will be given.

Chapter 5

5. HEURISTICS

We propose heuristics for both single and multiple allocation versions of the problem. Detailed explanations and comparisons of these heuristics are given in Section 5.1 for single allocation and in Section 5.2 for multi allocation. The pseudo codes of these heuristics are presented in the Appendix.

5.1 Heuristics for Single Allocation

Three heuristics are proposed: S_HEUR1, S_HEUR2 and S_HEUR3. S_HEUR1 and S_HEUR2 were coded in C programming language. Computational complexities of S_HEUR1 and S_HEUR2 are $O(n^2)$ and $O(n^3)$, respectively. S_HEUR3 is an optimization solver, like CPLEX, based heuristic. Computational complexity of S_HEUR3 is exponential. At the beginning of each heuristic, feasibility of the problem is checked.

5.1.1 S_HEUR1

As it is mentioned before, it is assumed that our problem is posed over a node set $N=\{1,...,n\}$. S_HEUR1 starts with identifying two nodes from the node set N which are farthest apart. In the first step of this heuristic; these two nodes, say Hub 1 and Hub 2, are considered as hubs and a hub-to-hub link is established in between them. Then Hub 1 and Hub 2 are thought as the center of the circles whose radius is β/2 and the nodes which are inside of these circles are assigned to these hubs. In other words, after identifying Hub 1 and Hub 2, the nodes, which are at most β/2 distance away from Hub 1, are assigned to this hub and among the remaining nodes, those which are at most β/2 distance away from Hub 2, are assigned to Hub 2. By this way, it is guaranteed that transportation time between the nodes which are assigned to Hub 1 cannot exceed the β bound. Same situation also holds for Hub 2.

After this assignment process, the nodes which are not yet hubs and are not assigned to any of the hubs are identified and these nodes are called as *outer nodes*. Then, the process of connecting the outer nodes to the nodes which are assigned to Hub 1 starts. This connection process aims to check if the flow exchange between outer nodes and the nodes assigned to Hub 1 is within the β bound. Connection can be done by:

C1. Assigning the outer node to Hub 1

C2. Assigning the outer node to Hub 2

C3. Considering the outer node as a hub node and setting hub-to-hub link with Hub 1

C4. Considering the outer node and one of the nodes assigned to Hub 1 as hub nodes, and setting hub-to-hub links between Hub 1 and outer node and between Hub 1 and the assigned node

These steps are tried in the order from 1 to 4 for each assigned node and outer node pair and the action corresponding to selected process, for which β bound is verified, is taken.

During the process of connecting the outer nodes to the nodes assigned to Hub 1, outer nodes are either assigned to Hub 1 or Hub 2, or considered as hub nodes by themselves. So after the connection process, there are no outer nodes which are not assigned to any of the hubs and it is guaranteed that transportation time between nodes assigned to Hub 1 and outer nodes are performed within the β bound. Because of that reason, after checking the β bound between the outer nodes and the nodes which are assigned to Hub 1, we proceed to check the bound between the nodes assigned to Hub 1 and the nodes assigned to Hub 2. If the β bound is exceeded, either one of these nodes is set as a hub node or these two nodes are set as hub nodes and connected to each other with hub-to-hub links. Finally, transportation time between any two hubs is checked and if it exceeds β, hub-to-hub links are set between the two hubs. Pseudo code of S_HEUR1 can be seen in Appendix A-1.

Let us illustrate S_HEUR1 on an example. Suppose the following network in Figure 1 is given:

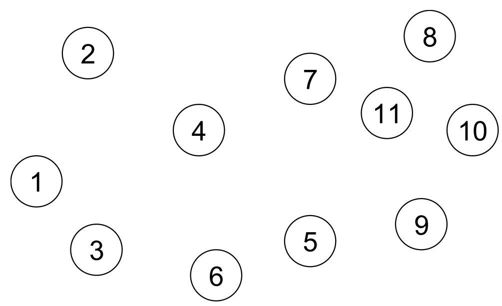

Figure 1: Nodes for example of S_HEUR1

In the first step of the heuristic, distances between each node pairs are calculated and then the two farthest away nodes are identified. In this example, say, Node 1 and Node 10 are considered as the farthest away nodes. Nodes 1*(Hub 1)* and 10*(Hub 2)* are considered as hub nodes and hub-to-hub link between these nodes are activated. Suppose Node 2 and Node 3 are at most β/2 distance away from Hub 1; and Node 8, Node 9 and Node 11 are at most β/2 distance away from Hub 2. In the next step, nodes 2 and 3 are assigned to Hub 1 ($T_{2,Hub1}+T_{Hub1,3}\leq\beta$), and nodes 8, 9 and 11 are assigned to Hub 2 ($T_{8,Hub2}+T_{Hub2,9}\leq\beta$, $T_{8,Hub2}+T_{Hub2,11}\leq\beta$, $T_{9,Hub2}+T_{Hub2,11}\leq\beta$). All of these steps can be seen in Figure 2.

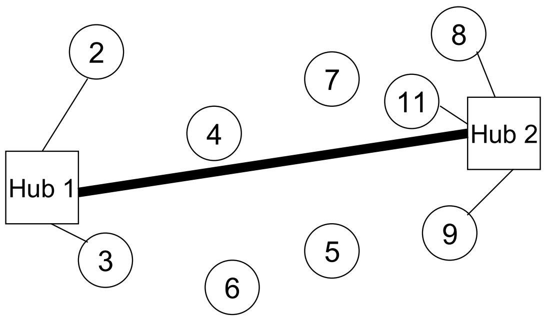

Figure 2: Example of S_HEUR1 after the first assignments

As it can be seen from Figure 2, nodes 4, 5, 6 and 7 are the *outer nodes*. Then, we apply connection steps (**C1**, **C2**, **C3** and **C4**) to Nodes 4, 5, 6 and 7 respectively. Say,

C1: Node 4 is assigned to Hub 1;

Since $T_{4,Hub1}+T_{Hub1,2}\leq\beta$ and $T_{4,Hub1}+T_{Hub1,3}\leq\beta$.

C2: Node 5 is assigned to Hub 2;

Since $T_{5,Hub1}+T_{Hub1,4}>\beta$ and $T_{5,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,2}\leq\beta$,

$T_{5,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,3}\leq\beta$ and $T_{5,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,4}\leq\beta$

C3: Node 6 is considered as Hub 3 and hub-to-hub link between Hub 1 and Hub 3 is activated;

Since $T_{6,Hub1}+T_{Hub1,4}>\beta$ and $T_{6,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,4}>\beta$;

$\alpha T_{6,Hub1}+T_{Hub1,2}\leq\beta$, $\alpha T_{6,Hub1}+T_{Hub1,3}\leq\beta$ and $\alpha T_{6,Hub1}+T_{Hub1,4}\leq\beta$

C4: Node 7 and Node 2 are considered as Hub 4 and Hub 5, respectively; and hub-to-hub link between Hub 4 and Hub 5; Hub 1 and Hub 4; Hub 1 and Hub 5 are activated;

Since $T_{7,Hub1}+T_{Hub1,2}>\beta$, $T_{7,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,2}>\beta$ and $\alpha T_{7,Hub1}+T_{Hub1,2}>\beta$; $\alpha T_{7,Hub1}+T_{Hub1,3}\leq\beta$, $\alpha T_{7,Hub1}+T_{Hub1,4}\leq\beta$ and $\alpha T_{7,2}\leq\beta$

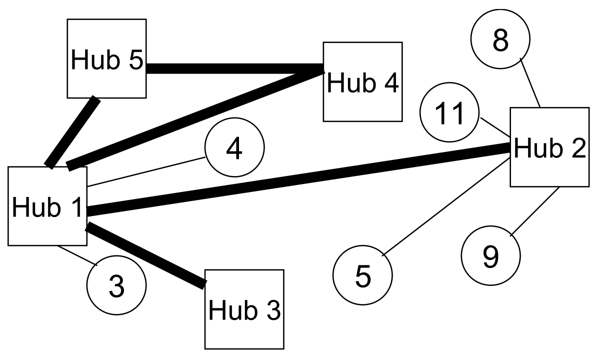

Figure 3: Example of S_HEUR1 after connecting the outer nodes and the nodes assigned to Hub1

Now, due to the construction, the nodes assigned to Hub 1 do satisfy the β bound. Thus we proceed to check the β bound between the nodes assigned to Hub 1 and Hub 2; and between hubs. Suppose β bound is exceeded between Node 3 and Node 8. For satisfying the β bound, Node 3 and Node 8 are considered as Hub 6 and Hub 7, respectively and hub-to-hub links between Hub 1 and Node 3; Hub 2 and Node 8 are activated. Finally, transportation times between any two hubs are checked and suppose transportation time between Hub 5 and Hub 7 exceeds β bound. In order to satisfy the β bound between Hub 5 and Hub 7, hub-to-hub link between these two hubs are activated. The resulting hub network is given in Figure 4.

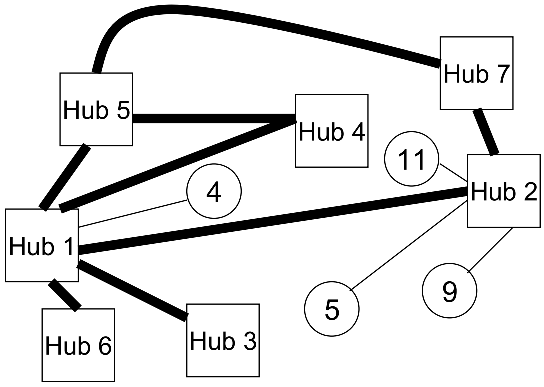

Figure 4: Final hub and hub-to-hub link structure of example of S_HEUR1

As it is expressed at the beginning of the heuristic, S_HEUR1 starts with defining two farthest apart nodes as hubs. That is for all α and β values, each feasible solution includes at least two hubs. There is no problem for smaller β values but for larger β values, flow exchange between nodes can be provided with single hub. To overcome this deficiency, we propose the following heuristic.

5.1.2 S_HEUR2

In the beginning of S_HEUR2, for each node, the nodes which are at most β/2 distance away are counted and the one which contains maximum number of nodes is considered as the first hub. Then the nodes, which are at most β/2 distance away from this hub, are assigned to this hub. Alternatively, as in S_HEUR1, in the beginning of the heuristic, each node can be thought as the center of a circle whose radius is β/2. Then, the nodes that are inside of the circles are counted. The one which contains maximum number of nodes is considered as Hub 1 and the nodes that are inside of Hub 1's circle are assigned to Hub 1. By this way, it is guaranteed that transportation time between the nodes which are assigned to Hub 1 cannot exceed β bound.

Then, the distances between the remaining nodes and the first hub are checked. If the distance is greater than β, this node should be a hub node and we should establish a hub-to-hub link between the first hub so that these two hubs are served within β time bound. Then, the nodes, which are not selected as hubs and are not assigned to any hub, are identified (*outer nodes*) and connection between outer nodes and nodes assigned to Hub 1 are done according to the steps that are defined in S_HEUR1 except **C2**. In S_HEUR2, C2 is modified as;

> **C2'.** Assigning the outer node to another opened hub and setting hub-to-hub link between this selected hub and Hub 1

Lastly, the transportation time between each node and hub is checked. If β bound is exceeded, these nodes are assigned as hub and hub-to-hub links are set between them. Also, transportation time between any two hubs is checked and if it exceeds β, hub-to-hub links are set between two hubs. Pseudo code of S_HEUR2 can be seen in Appendix A-2.

Let us illustrate S_HEUR2 on an example. Suppose the following network in Figure 5 is given:

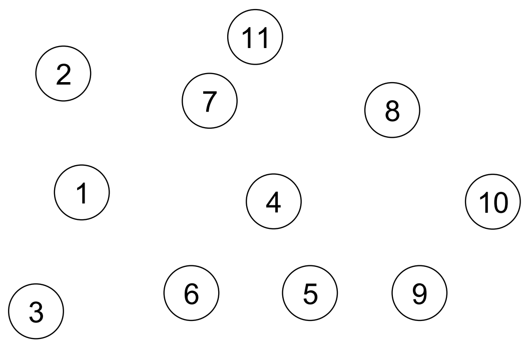

Figure 5: Nodes for example of S_HEUR2

In the first step of the heuristic, each node is thought as the center of a circle whose radius is β/2. Then among the remaining nodes, which are inside of the circle, are counted and the one which contains maximum number of nodes is considered as the first hub. For example, when we draw a circle with radius β/2 by considering Node 10 as the center of the circle, Node 8 and Node 9 are take place inside of this circle. In this example, Node 4's circle contains maximum number of nodes (Nodes 5, 6, 7 and 8) and Node 4 is considered as Hub 1. Nodes 5, 6, 7 and 8 are assigned to Hub 1. Also, the distances between Hub1 and Node 2; and Hub 1 and Node 3 are greater than β. In order to perform transportation between Hub1 and Node 2; and Hub 1 and Node 3 in β bound, Node 2 and Node 3 are considered as Hub 2 and Hub 3, respectively. Hub-to-hub link between these nodes (Node 2 and Node 3) and Hub 1 are activated. These steps can be seen in Figure 6.

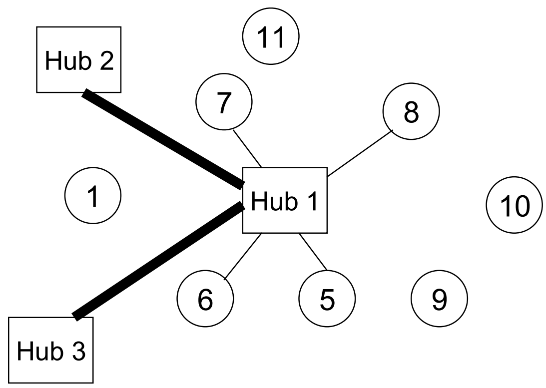

Figure 6: Example of S_HEUR2 after the first assignments

Nodes 1, 9, 10 and 11 are defined as outer nodes. Connection between these nodes and nodes assigned to Hub 1 (Nodes 5, 6, 7 and 8) are done according to the connection steps. (**C1**, **C2'**, **C3** and **C4**) Then we apply these steps to outer nodes.

C1: Node 1 is assigned to Hub 1

C2': Node 11 is assigned to Hub 2

C3: Node 9 is considered as Hub 4 and hub-to-hub link between Hub 1 and Hub 4 is activated

C4: Node 10 and Node 6 are considered as Hub 5 and Hub 6, respectively; and hub-to-hub link between Hub 5 and Hub 6; Hub 1 and Hub 5; Hub 1 and Hub 6 are activated

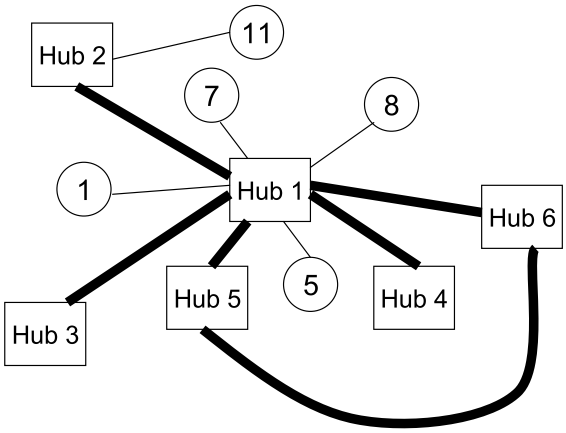

Figure 7: Example of S_HEUR2 after connecting the outer nodes and the nodes assigned to Hub1

Then, the transportation time between each node (Nodes 1, 5, 7, 8 and 11) and hub (Hubs 1, 2, 3, 4, 5 and 6) are checked. Suppose transportation time between Node 8 and Hub 3 exceeds β bound, so Node 8 is considered as Hub 7 and hub-to-hub link between Hub 7 and Hub 1 is activated. Finally, transportation times between any two hubs are checked and suppose transportation time between Hub 2 and Hub 4 exceeds β bound. In order to satisfy the β bound between these hubs, hub-to-hub link between these two hubs are activated. The resulting hub network is given in Figure 8.

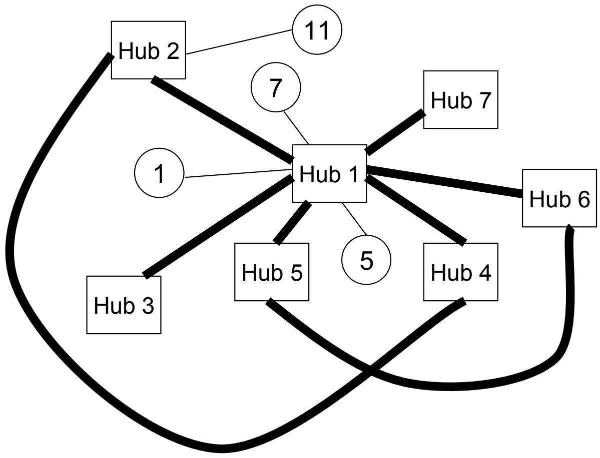

Figure 8: Final hub and hub-to-hub link
structure of example of S_HEUR2

S_HEUR1 and S_HEUR2 have similar structures. Both of them start with opening initial hubs by checking the β bound. The last heuristic that we propose has a completely different structure.

5.1.3 S_HEUR3

S_HEUR3 has two phases. In the first phase, we aim to obtain optimal hub locations in complete hub network structure by solving the single allocation hub covering model given by Kara and Tansel(2000). Then in the second phase, we aim to find necessary hub-to-hub links in order to connect the hubs that are found in Phase I. The optimal locations of hub nodes are used as an input to our original single allocation model and the model (P-S) is solved to find hub-to-hub links. As a result, by solving two consecutive models, we obtain locations of hubs and the hub-to-hub links.

5.1.4 Comparison of the Proposed Heuristics for Single Allocation

Number of hubs, number of hub arcs and CPU times obtained from S_HEUR1, S_HEUR2 and S_HEUR3 for n=10, 15, 20 and 25 are given in Tables 5, 6, 7 and 8, respectively.

	β	S_HEUR1			S_HEUR2			S_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	1425	6	6	0,059	3	2	0,024	2	1	10,09
	1117	8	7	0,066	5	4	0,039	3	2	5,06
	811	6	5	0,05	6	5	0,052	5	5	3,2
	736	7	6	0,062	6	5	0,049	6	7	8,53
α=0.4	1627	5	5	0,046	3	2	0,023	2	1	5,94
	1185	9	11	0,082	4	3	0,03	3	2	3,06
	970	10	18	0,1	5	4	0,043	4	4	1,78
	863	8	18	0,104	5	4	0,04	5	8	2,21
α=0.6	1671	6	6	0,055	3	2	0,023	2	1	10,55
	1387	7	13	0,083	5	4	0,042	3	3	25,4
	1148	9	26	0,115	7	9	0,068	4	6	1,56
	1079	10	31	0,131	5	7	0,058	5	8	1,63
α=0.8	1744	5	8	0,064	3	2	0,024	2	1	8,71
	1589	6	11	0,078	5	5	0,048	3	3	30,16
	1457	7	14	0,085	10	12	0,09	5	5	2,08
	1413	7	15	0,097	10	13	0,086	5	7	3,1
α=1.0	1839	8	20	0,101	2	1	0,015	1	0	48,62
	1791	6	12	0,081	5	7	0,048	3	3	34,62
	1770	6	12	0,082	5	7	0,046	4	5	25,49
	1766	6	12	0,09	5	7	0,049	4	10	28,86

Table 5: Results of S_HEUR1, S_HEUR2 and S_HEUR3 for n=10

As it is mentioned before, for Single Allocation Hub Covering Problem over Incomplete Network, CPLEX can only generate optimal results for smaller instances. Thus, we can only compare the optimal results and the solutions of the proposed heuristics for n=10.

Among the heuristics, for most of the cases S_HEUR3 gives closer results to the optimum. Results in Tables 3 and 5 show that, in 14 out of 20 cases, S_HEUR3 finds the optimum solutions. The average gap between S_HEUR3's results and optimal results is %22.8.

Starting with opening two hubs is a main disadvantage of the S_HEUR1. Because of that reason, in 17 out of 20 cases S_HEUR2 outperforms S_HEUR1. That situation occurs because of the structure of the data used in computational performance and also the structure of the S_HEUR1. Especially during the connection process, S_HEUR1 tends to open hubs in order to perform flow exchange between the nodes assigned to Hub 1 and the outer nodes within the β bound. It also tends to open hubs while connecting the nodes assigned to Hub 1 and the nodes assigned to Hub 2. For example let us take the bound β=1079, for α=0.6. β=1079 is a tight bound according to the transportation times in CAB data. So, for connecting the assigned nodes, S_HEUR1 needs to open all nodes as hub and most of the hub-to-hub links in order to satisfy the β bound.

The performance of S_HEUR1 is also related with the initial hubs. Another disadvantage of the S_HEUR1 is that it starts with the same initial hubs for all instances of n=10. Since the two nodes which are farthest apart are the same for all α and β values of n=10, S_HEUR1 always starts with the same hubs. All of the remaining steps in the S_HEUR1, are done according to the initial hubs which are found in Step 1. So, initial hubs are very effective on the remaining steps of S_HEUR1. All of the assignments and connections are done according to these hubs. Because of that reason it gives worse results than S_HEUR2 and S_HEUR3.

	β	**S_HEUR1**			**S_HEUR2**			**S_HEUR3**		
		# of hubs	**# of hub arcs**	**CPU time (sec)**	**# of hubs**	**# of hub arcs**	**CPU time (sec)**	**# of hubs**	**# of hub arcs**	**CPU time (sec)**
α=0.2	**2004**	4	4	0,037	3	2	0,022	2	1	255,77
	1638	11	10	0,082	5	4	0,041	4	3	56,65
	1324	12	11	0,086	6	5	0,048	5	3	365,91
	1149	13	12	0,098	8	7	0.072	5	8	31,87
α=0.4	**2019**	4	4	0,037	3	2	0.022	3	2	192,65
	1741	5	5	0,059	4	3	0,032	4	3	174,68
	1436	12	18	0,101	8	7	0,065	4	3	203,72
	1287	13	25	0,124	11	10	0,085	5	8	18,84
α=0.6	**2103**	4	5	0,061	4	3	0,032	3	3	430
	1844	6	10	0,076	3	2	0,023	3	2	89,23
	1756	5	7	0,06	5	4	0,042	4	4	23,17
	1560	12	38	0,152	13	15	0,103	5	8	31,93
α=0.8	**2424**	5	8	0,064	5	5	0,046	2	1	318,36
	2165	4	5	0,045	7	7	0,065	3	2	14,32
	2100	4	5	0,045	13	14	0,093	4	3	15,4
	2080	4	5	0,043	13	14	0,101	5	8	42,56
α=1.0	**2611**	6	12	0,085	8	8	0,073	2	1	281,97
	2610	6	12	0,079	8	8	0,074	3	3	1546,5
	2605	6	12	0,084	8	8	0,076	3	3	1510,3
	2600	6	12	0,076	8	8	0,072	3	4	1401,1

Table 6: Results of S_HEUR1, S_HEUR2 and S_HEUR3 for n=15

For n=15, in all but 4 of cases, S_HEUR3 outperforms S_HEUR1 and S_HEUR2 according to the number of hubs and number of hub-to-hub links. S_HEUR2 generates closer results to other heuristics for loose β values of α

0.2, 0.4 and 0.6. That situation also occurs because of the structure of S_HEUR2 and the structure of CAB data. S_HEUR2 starts with counting the nodes which are at most β/2 distance away and considering the one which contains maximum number of nodes, as the first hub. Then the nodes, which are at most β/2 distance away from this hub, are assigned to this hub. So for loose β values, most of the nodes in the node set are assigned to first hub. Similarly, most of the remaining nodes' distances according to first hub are in the range of β value. So they are connected to the nodes assigned to Hub 1 without being considered as hubs. This situation decreases the needed number of hubs and hub arcs.

Interestingly, S_HEUR1 shows good performance for the tightest β value of α=0.8. As in n=10, the structure of the S_HEUR1 affects the results. When the number of nodes is increased from 10 to 15, initial hubs change. Node 3 and node 8 are initial hubs when the number of nodes is equal to 10, when the number of nodes is increased to 15, then node 3 and node 12 are selected as the initial hubs. Rest of the steps are done according to these initial hubs and because of the structure of 15 node instances of the CAB data, S_HEUR1 generates better results than S_HEUR2 and S_HEUR3 for α=0.8.

	β	S_HEUR1			S_HEUR2			S_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	1851	5	5	0,049	5	4	0,04	3	2	2035,4
	1549	13	13	0,093	6	5	0,05	3	3	246,54
	1356	14	14	0,094	8	7	0,069	4	3	814,23
	1162	15	15	0,098	10	9	0,087	5	10	233,59
α=0.4	2067	5	6	0,054	4	3	0,032	3	2	2928,72
	1744	6	7	0,082	6	5	0,05	4	5	2244,74
	1473	15	25	0,126	8	7	0,069	4	3	5471,36
	1386	15	30	0,128	11	11	0,093	6	9	11438,8
α=0.6	2255	6	10	0,095	3	2	0,023	3	3	4595,79
	1996	7	13	0,087	6	6	0,056	3	3	5994,26
	1835	8	18	0,097	10	11	0,084	4	4	9703,58
	1663	13	37	0,143	17	21	0,119	5	10	7054,69
α=0.8	2493	7	15	0,099	5	6	0,053	3	2	2971,18
	2264	6	11	0,083	8	10	0,074	3	3	5871,09
	2154	6	11	0,079	12	16	0,093	4	4	15702,4
	2118	6	11	0,079	16	20	0,111	5	4	407,32
α=1.0	2611	8	22	0,114	2	1	0,014	2	1	2103,8
	2605	8	22	0,109	3	3	0,029	3	3	3614,36
	2601	8	22	0,106	3	3	0,03	3	3	39397,2
	2600	8	22	0,104	3	3	0.029	3	6	7607

Table 7: Results of S_HEUR1, S_HEUR2 and S_HEUR3 for n=20

	β	S_HEUR1			S_HEUR2			S_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	2136	9	11	0,095	5	4	0,044	2	1	4828,65
	1913	18	18	0,111	6	5	0,049	3	2	27538,1
	1617	21	21	0,118	8	7	0,068	4	4	21894,1
	1346	22	22	0,119	11	10	0,08	6	6	87473,2
α=0.4	2401	7	10	0,093	5	4	0,044	2	1	30262,6
	2099	9	11	0,102	6	5	0,049	3	2	23691,7
	1881	21	22	0,12	7	6	0,059	4	3	33076,6
	1597	21	26	0,127	9	11	0,083	5	4	37195,8
α=0.6	2557	7	15	0,124	4	3	0,031	2	1	49941,7
	2336	7	16	0,121	5	6	0,056	3	3	46625
	2184	10	24	0,118	8	10	0,074	4	3	145173
	2002	13	39	0,151	10	17	0,099	5	9	53350,9
α=0.8	2713	9	25	0.131	4	3	0,031	2	1	66588,6
	2552	10	30	0,126	4	3	0,034	3	3	67883,6
	2457	11	35	0,135	12	29	0,102	4	5	163402
	2307	12	44	0,162	15	32	0,11	6	7	212811
α=1.0	2826	11	40	0.168	5	9	0,055	2	1	8733,24
	2762	10	35	0,136	5	9	0,051	3	3	76787,8
	2726	10	35	0,134	7	11	0,07	4	6	112321,19
	2725	10	35	0,226	7	11	0,075	6	7	225951,33

Table 8: Results of S_HEUR1, S_HEUR2 and S_HEUR3 for n=25

Similar observations can be derived for n=20 and 25 and S_HEUR3 outperforms S_HEUR1 and S_HEUR2 in terms of solution quality. When we compare the CPU times of the heuristics, we see that S_HEUR1 and S_HEUR2 generate faster results than S_HEUR3. Solving two consecutive models with CPLEX, increases the CPU time of S_HEUR3 but on the other hand using CPLEX based solution approach, improves the quality of the solutions.

Although S_HEUR3 shows worse performance than S_HEUR1 and S_HEUR2 according to CPU times, in 14 out of 20 cases for n=10, S_HEUR3 finds optimums. Besides, in 79 out of 80 instances, it generates better or the same solutions in terms of the number of hubs and the number of hub-to-hub links than the others. So, we can say that for single allocation case, S_HEUR3 is the best one.

5.2 Heuristics for Multiple Allocation

Three heuristics are developed for multi allocation version of the problem. (M_HEUR1, M_HEUR2 and M_HEUR3) M_HEUR1 and M_HEUR2 are modifications of S_HEUR1 and S_HEUR2. They include additional steps that allow assignments of demand centers to more than one hub. Computational complexities of M_HEUR1 and M_HEUR2 are $O(n^3)$ and $O(n^3)$, respectively. M_HEUR3 is also a CPLEX based heuristic. Computational complexity of M_HEUR3 is exponential. At the beginning of each heuristic, feasibility of the problem is checked.

5.2.1 M_HEUR1

Similar to S_HEUR1, M_HEUR1 starts with selecting two hubs. For the allocation of outer nodes, there are additional steps for allowing multiple allocation. In addition to connection steps defined in S_HEUR1, connections can also be done by;

> **C2+.** Assigning outer nodes and nodes that are assigned to Hub 1 to another opened hub (After **C2** in **S_HEUR1**)
>
> **C3+.** Setting outer node as hub node and assigning the node assigned to Hub 1 to outer node (After **C3** in **S_HEUR1**)

The nodes which are not hubs and are not assigned to Hub 1 and Hub 2 are identified. The process of connection steps that are defined in S_HEUR1 and the additional steps(**C2+** and **C3+**) are tried for each node pair in order to connect the outer nodes to the nodes which are assigned to Hub 1. The action corresponding to selected process, for which β bound is verified, is taken.

Differently from S_HEUR1, after checking the β bound between the outer nodes and the nodes which are assigned to Hub 1, again the nodes which are not hubs and are not assigned to Hub 1 and Hub 2 are identified*(outer nodes)*. Due to multi allocation, this time there may be unassigned nodes. Then, similarly outer nodes and the nodes assigned to Hub 2 are connected according to connection steps.

Then, the transportation time between the nodes which are assigned only to Hub 1 and the nodes which are assigned only to Hub 2 are checked. If β time is exceeded, either these two nodes are assigned to other opened hubs or one of these nodes is set as hub node. If everything else fails, these two nodes are set as hub nodes and connected to each other with a hub-to-hub links. Pseudo code of M_HEUR1 can be seen in Appendix A-3.

Let us illustrate M_HEUR1 on an example. Suppose the following network in Figure 9 is given:

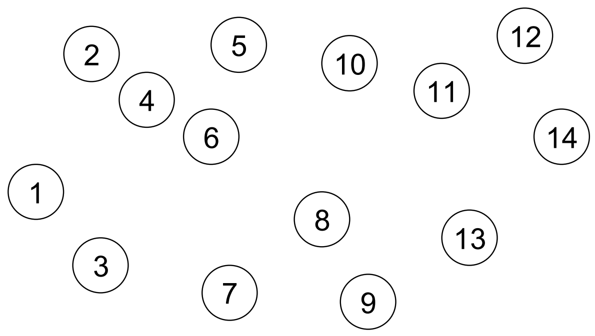

Figure 9: Nodes for example of M_HEUR1

Node 1 and Node 14 are the farthest away nodes. Nodes 1*(Hub 1)* and 14*(Hub 2)* are considered as hub nodes and hub-to-hub link between these nodes are activated. Suppose Nodes 2, 3 and 4 are β/2 distance away from Hub 1; and Nodes 11, 12 and 13 are β/2 distant away from Hub 2. In the next step, Nodes 2, 3 and 4 are assigned to Hub 1, and Nodes 11, 12 and 13 are assigned to Hub 2. All of these steps can be seen in Figure 10.

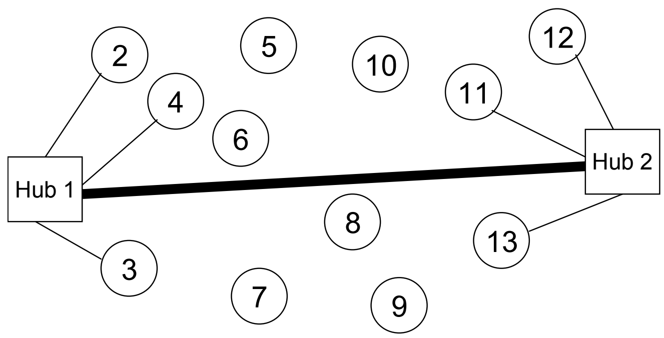

Figure 10: Example of M_HEUR1 after the first assignments

Nodes 5, 6, 7, 8, 9 and 10 are defined as outer nodes. Connection between these nodes and nodes assigned to Hub 1 (Nodes 2, 3 and 4) are done according to the connection steps.

C1: Node 6 is directly assigned to Hub 1;

Since $T_{6,Hub1}+T_{Hub1,2} \leq \beta$, $T_{6,Hub1}+T_{Hub1,3} \leq \beta$ and $T_{6,Hub1}+T_{Hub1,4} \leq \beta$

C3: Node 5 is considered as Hub 3 and hub-to-hub link between Hub 1 and Hub 3 is activated;

Since $T_{5,Hub1}+T_{Hub1,2}>\beta$ and $T_{6,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,2}>\beta$;

$\alpha T_{5,Hub1}+T_{Hub1,2} \leq \beta$, $\alpha T_{5,Hub1}+T_{Hub1,3} \leq \beta$, $\alpha T_{5,Hub1}+T_{Hub1,4} \leq \beta$ and $\alpha T_{5,Hub1}+T_{Hub1,6} \leq \beta$

C3+: Node 7 is considered as Hub 4 and Node 3 is assigned to Hub 4;

Since $T_{7,Hub1}+T_{Hub1,3}>\beta$, $T_{7,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,3}>\beta$ and $T_{7,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub3,3}>\beta$;

$\alpha T_{7,Hub1}+T_{Hub1,2} \leq \beta$, $\alpha T_{7,Hub1}+T_{Hub1,4} \leq \beta$, $\alpha T_{7,Hub1}+T_{Hub1,6} \leq \beta$ and $T_{7,3} \leq \beta$

C2: Node 8 is assigned to Hub 4;

Since $T_{8,Hub1}+T_{Hub1,2}>\beta$;

$T_{8,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,2} \leq \beta$, $T_{8,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,3} \leq \beta$, $T_{8,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,4} \leq \beta$ and $T_{8,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,6} \leq \beta$

C2+: Node 10 and Node 2 are assigned to Hub 3

Since $T_{10,Hub1}+T_{Hub1,2}>\beta$, $T_{10,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,2}>\beta$, $T_{10,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub1,2}>\beta$ and $T_{10,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,2}>\beta$;

$T_{10,Hub3}+T_{Hub3,2} \leq \beta$, $T_{10,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub1,3} \leq \beta$, $T_{10,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub1,4} \leq \beta$ and $T_{10,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub1,6} \leq \beta$

C4: Node 9 and Node 4 are considered as Hub 6 and Hub 5, respectively and hub-to-hub links between Hub 1 and Hub 5; and Hub 1 and Hub 6 are activated;

Since $T_{9,Hub1}+T_{Hub1,4}>\beta$, $T_{9,Hub2}+\alpha T_{Hub2,Hub1}+T_{Hub1,4}>\beta$,

$T_{9,Hub3}+\alpha T_{Hub3,Hub1}+T_{Hub3,4}>\beta$, $T_{9,Hub4}+\alpha T_{Hub4,Hub1}+T_{Hub1,4}>\beta$,
$T_{9,Hub2}+T_{Hub2,4}>\beta$, $T_{9,Hub3}+T_{Hub3,4}>\beta$, $T_{9,Hub4}+T_{Hub4,4}>\beta$,
$\alpha T_{9,Hub1}+T_{Hub1,4}$ and $T_{9,4}>\beta$;
$\alpha T_{9,Hub1}+T_{Hub1,2}\leq\beta$, $\alpha T_{9,Hub1}+T_{Hub1,3}\leq\beta$, $\alpha T_{9,Hub1}+T_{Hub1,6}\leq\beta$ and
$\alpha T_{9,Hub1}+\alpha T_{Hub1,4}\leq\beta$

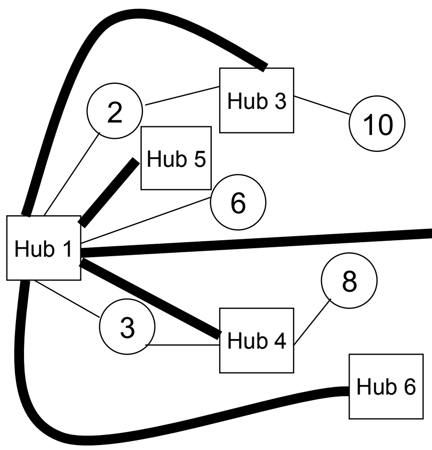

Figure 11: Example of M_HEUR1 after connecting the outer nodes and the nodes assigned to Hub1

Then, the nodes which are not hubs and are not assigned to Hub 1 and Hub 2 are identified. (Nodes 8 and 10) Similarly these nodes and the nodes assigned to Hub 2 are connected according to connection steps.

C1: Node 8 is assigned to Hub 2;

Since $T_{8,Hub2}+T_{Hub2,11}\leq\beta$, $T_{8,Hub2}+T_{Hub2,12}\leq\beta$ and $T_{8,Hub2}+T_{Hub2,13}\leq\beta$

C3: Node 10 is considered as Hub 7 and hub-to-hub links between Hub 2 and Hub 7; Hub 3 and Hub 7 are activated;

Since $T_{10,Hub2}+T_{Hub2,12}>\beta$, $T_{10,Hub1}+T_{Hub1,12}>\beta$, $T_{10,Hub3}+T_{Hub3,12}>\beta$,
$T_{10,Hub4}+T_{Hub4,12}>\beta$, $T_{10,Hub5}+T_{Hub5,12}>\beta$ and $T_{10,Hub6}+T_{Hub6,12}>\beta$;

$\alpha T_{10,Hub2}+T_{Hub2,11} \leq \beta$, $\alpha T_{10,Hub2}+T_{Hub2,12} \leq \beta$ and $\alpha T_{10,Hub2}+T_{Hub2,13} \leq \beta$

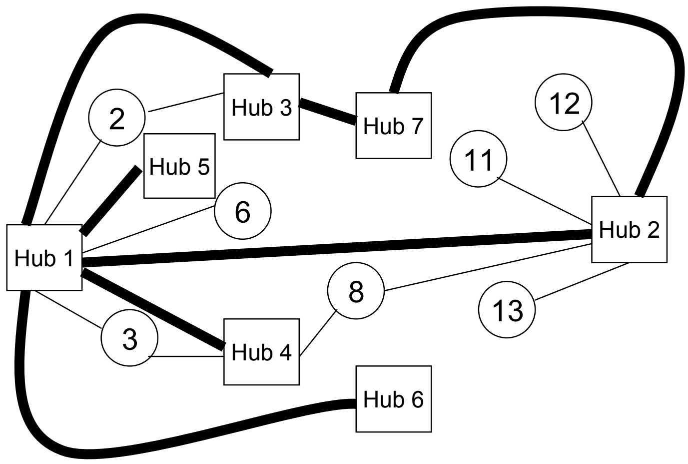

Figure 12: Example of M_HEUR1 after connecting the outer nodes and the nodes assigned to Hub2

After that process, we proceed to check the β bound between the nodes assigned to Hub 1(Nodes 2, 3 and 6) and the nodes assigned to Hub 2 (Nodes 8, 11, 12 and 13). Suppose β bound is exceeded between Node 6 and Node 12. For satisfying the β bound, Node 6 and Node 12 are considered as Hub 8 and Hub 9, respectively and hub-to-hub links between Hub 1 and Hub 8; Hub 2 and Hub 9 are activated. Finally, transportation times between any two hubs are checked and suppose transportation time between Hub 3 and Hub 6 exceeds β bound. In order to satisfy the β bound, hub-to-hub link between Hub 3 and Hub 6 are activated. The resulting hub network is given in Figure 13.

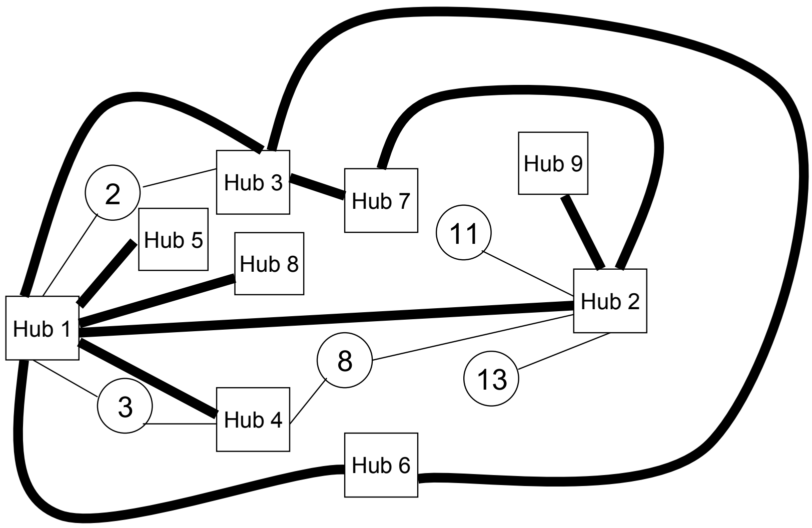

Figure 13: Final hub and hub-to-hub link structure of example of M_HEUR1

Finally as in S_HEUR1, transportation time between any two hubs is checked and if it exceeds β time, these two hubs are connected with hub-to-hub link.

5.2.2 M_HEUR2

Except the connection procedure between the outer nodes and the nodes that are assigned to the initial hub, the underlying idea of S_HEUR2 and M_HEUR2 are the same. For each node, we count the nodes which are at most β/2 distance away and the one which contains maximum number of nodes is selected as the initial hub. Then the nodes, which are at most β/2 distance away from this hub, are assigned to this hub. Similar to S_HEUR2, the nodes, which are >β distance away from the initial hub, are considered as hubs. Differently, in M_HEUR2, after identifying the outer nodes, for connection process, the steps that are defined in M_HEUR1 (**C1, C2, C2+, C3, C3+** and **C4**) are used. These steps are tried for each node pair and the action corresponding to selected process, for which β bound is verified, is taken.

Finally, the transportation time between each node and hub is checked. If β bound is exceeded, these nodes are assigned as hub and hub-to-hub link is set between them. Also, transportation time between any two hubs is checked and if it exceeds β, hub-to-hub links are set between two hubs.

Let us illustrate M_HEUR2 on an example. Suppose the network in Figure 14 is given.

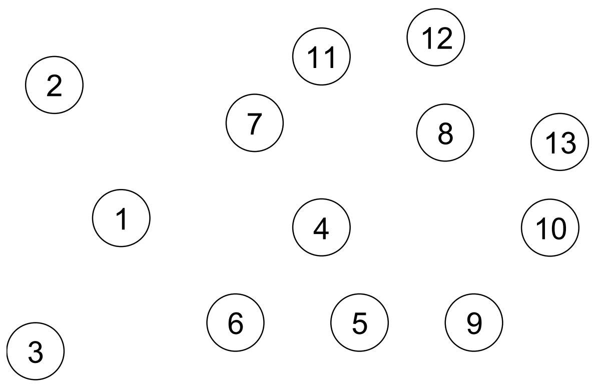

Figure 14: Nodes for example of M_HEUR2

Node 4's circle contains maximum number of nodes (Nodes 5, 6, 7 and 8) and Node 4 is considered as Hub 1. Nodes 5, 6, 7 and 8 are assigned to Hub 1. Also, the distances between Hub1 and Node 2; and Hub 1 and Node 3 are greater than β. In order to perform transportation between Hub1 and Node 2; and Hub 1 and Node 3 in β bound, Node 2 and Node 3 are considered as Hub

2 and Hub 3, respectively. Hub-to-hub link between these nodes (Node 2 and Node 3) and Hub 1 are activated. These steps can be seen in Figure 15.

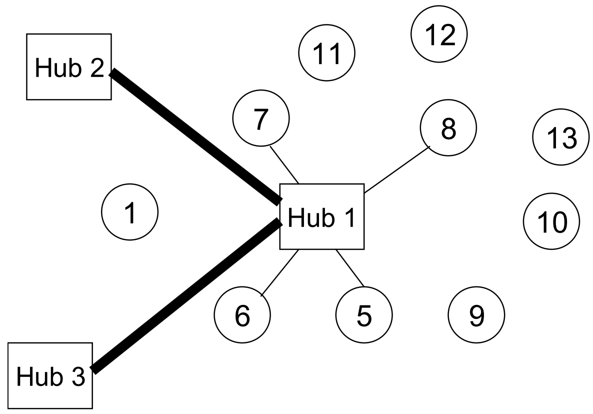

Figure 15: Example of M_HEUR2 after the first assignments

Nodes 1, 9, 10, 11, 12 and 13 are defined as outer nodes. Connection between these nodes and nodes assigned to Hub 1 (Nodes 5, 6, 7 and 8) are done according to the connection steps.

C1: Node 1 is assigned to Hub 1.

C3+: Node 11 is considered as Hub 4 and Node 7 is assigned to Hub 4.

C2: Node 12 is assigned to Hub 4.

C3: Node 10 is considered as Hub 5 and hub-to-hub link between Hub 1 and Hub 5 is activated.

C2+: Node 13 and Node 8 are assigned to Hub 5.

C4: Node 9 and Node 1 are considered as Hub 6 and Hub 7, respectively and hub-to-hub links between Hub 6 and Hub 1; Hub 7 and Hub 1 are activated.

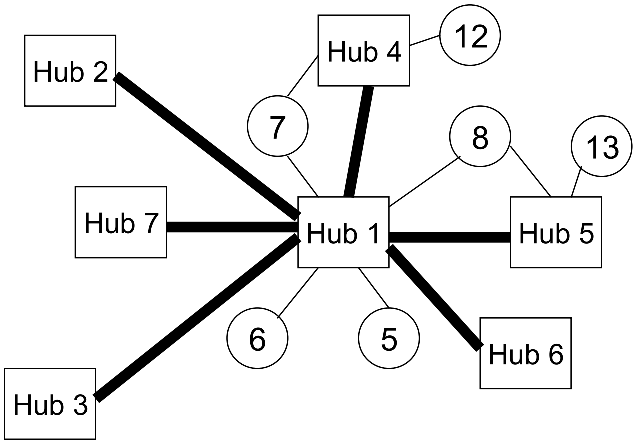

Figure 16: Example of M_HEUR2 after connecting the outer nodes and the nodes assigned to Hub1

Then, the transportation time between each node (Nodes 5, 6, 7, 8, 12 and 13) and hub (Hubs 1, 2, 3, 4, 5, 6 and 7) are checked. Suppose transportation time between Node 13 and Hub 1 exceeds β bound, so Node 13 is considered as Hub 8 and hub-to-hub links between Hub 8 and Hub 1; Hub 8 and Hub 5 are activated. Finally, transportation times between any two hubs are checked and suppose transportation time between Hub 2 and Hub 8 exceeds β bound. In order to satisfy the β bound between these hubs, hub-to-hub link between these two hubs are activated. The resulting hub network is given in Figure 17.

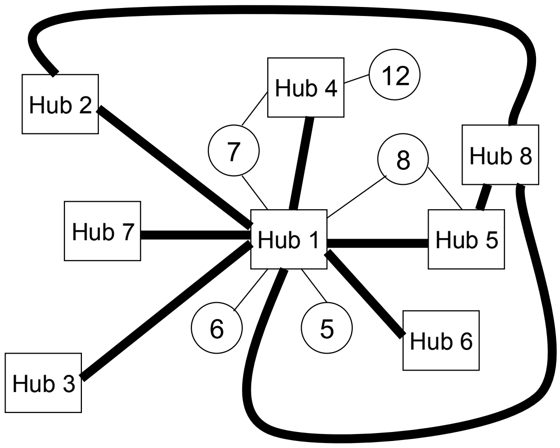

Figure 17: Final hub and hub-to-hub link structure of example of M_HEUR2

5.2.3 M_HEUR3

The third heuristic has two phases as in S_HEUR3. In the first phase, multi allocation hub covering model provided by Campbell(1994) is solved. Solution of this model gives the optimal locations of hub nodes. Then in the second phase, these optimal hub locations are used as an input of our original multi allocation model (P-M) and after an appropriate modification in the objective function, our original model is solved to find hub arcs for connecting the hub nodes, which are found in Phase I. As a result, hub locations and hub arcs between these hubs are obtained by solving two consecutive models.

5.2.4 Comparison of the Proposed Heuristics for Multiple Allocation

Number of hubs, number of hub arcs and CPU times obtained from M_HEUR1, M_HEUR2 and M_HEUR3 for N=10, 15, 20 and 25 are given in Tables 9, 10, 11 and 12, respectively.

	β	M_HEUR1			M_HEUR2			M_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	**1425**	3	2	0,023	3	2	0,023	2	1	7,2
	1117	4	3	0,032	4	3	0,032	4	3	20,57
	811	6	5	0,049	5	4	0,039	4	3	1,91
	736	6	5	0,05	6	5	0,052	6	5	34,1
α=0.4	**1627**	3	2	0,023	3	2	0,023	2	1	5,54
	1185	5	7	0,057	3	2	0,023	3	2	1,8
	970	8	13	0,085	5	4	0,042	4	3	1,07
	863	8	15	0,087	5	4	0,04	5	4	1
α=0.6	**1671**	3	2	0,024	3	2	0,024	3	2	8,36
	1387	4	5	0,042	4	3	0,032	3	3	3,11
	1148	6	12	0,086	5	7	0,059	4	4	0,84
	1079	7	18	0,086	5	7	0,057	5	6	1,11
α=0.8	**1744**	3	3	0,022	3	2	0,024	3	2	4,28
	1589	3	3	0,028	3	3	0,029	3	3	3,15
	1457	4	6	0,031	4	5	0,045	4	6	1,11
	1413	4	6	0,028	4	5	0,045	4	5	1,16
α=1.0	**1839**	3	2	0,022	2	1	0,014	2	1	2,42
	1791	3	2	0,022	2	1	0,015	2	1	2,25
	1770	3	2	0,022	2	1	0,014	2	1	2,3
	1766	3	2	0,022	2	1	0,014	2	1	2,79

Table 9: Results of M_HEUR1, M_HEUR2 and M_HEUR3 for n=10

Like the single allocation version of the problem, with the model (P-M), we could only obtain optimal results for n=10 from CPLEX. As we see from the Tables 3 and 9, M_HEUR2 and M_HEUR3 give closer results to the optimum values. Especially for tighter β values, M_HEUR2 and M_HEUR3 show good performance. In 5 out of 20 cases M_HEUR2 and in 10 out of 20 cases M_HEUR3 generate the optimums. The average gap between the solution found by M_HEUR2 and the optimum results is %29.8. For M_HEUR3, this gap is %27.7.

As in S_HEUR1, the main disadvantage of the M_HEUR1 is opening two hubs at the beginning of the heuristic. Hence, for loose β values, it cannot generate closer results to M_HEUR2 and M_HEUR3 but for tight β values, its performance approaches to M_HEUR2 and M_HEUR3's performances. For some instances such as α=0.2; β=1425, 1117 and 736, α=0.4; β=1627, α=0.6; β=1671 and α=0.8; β=1589, M_HEUR1 generates same results with M_HEUR2. According to CPU times, M_HEUR1 and M_HEUR2 outperform M_HEUR3.

	β	M_HEUR1			M_HEUR2			M_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	**2004**	3	2	0,023	3	2	0,023	2	1	726,75
	1638	4	3	0,033	4	3	0,034	4	3	868,2
	1324	6	5	0,049	5	4	0,04	4	3	95,55
	1149	6	5	0,049	7	6	0,059	5	4	153,07
α=0.4	**2019**	3	2	0,023	3	2	0,024	3	2	279,35
	1741	4	4	0,038	3	2	0,023	3	2	60,8
	1436	6	8	0,066	5	4	0,043	4	3	33,74
	1287	6	10	0,08	5	4	0,04	5	4	12,46
α=0.6	**2103**	3	3	0,029	4	3	0,031	2	1	40,29
	1844	4	5	0,045	3	2	0,022	3	3	23,74
	1756	4	6	0,043	4	3	0,034	4	4	16,94
	1560	4	5	0,047	5	7	0,058	5	7	14,18
α=0.8	**2424**	3	3	0,024	2	1	0,015	2	1	53,67
	2165	3	3	0,029	2	1	0,015	3	3	10,59
	2100	3	2	0,028	4	5	0,043	4	4	12,13
	2080	3	2	0,03	4	5	0,049	4	5	14,89
α=1.0	**2611**	3	4	0,025	2	1	0,014	2	1	16,85
	2610	3	4	0,025	2	1	0,014	2	1	16,84
	2605	3	4	0,026	2	1	0,015	2	1	16,83
	2600	3	4	0,025	2	1	0,014	2	1	24,96

Table 10: Results of M_HEUR1, M_HEUR2 and M_HEUR3 for n=15

Differently from the single allocation, closer results were obtained from M_HEUR2 and M_HEUR3 for most of the cases when n=15. Especially for α values 0.4 and 1.0, except few *β* cases, they give same results. Allowing the assignment of demand nodes to more than one hub, increases the solution quality of M_HEUR2. In single allocation, since each demand node should be allocated to exactly one hub, during connecting the outer nodes to the nodes assigned to first hub, most of the outer nodes are considered as hub in order to satisfy the *β* bound. Allowing multi allocation solves this problem. By this, outer nodes and the nodes assigned to the first hub are connected by alternative hubs. This situation decreases the number of opened hubs and hub-to-hub links. Similar observations can be done for M_HEUR1. Allowing multiple allocation, also increases the solution quality of M_HEUR1. For example, M_HEUR1 gives better results than the others for α=0.6. As expected; M_HEUR1 and M_HEUR2's CPU time is better than M_HEUR3's CPU time.

	β	M_HEUR1			M_HEUR2			M_HEUR3		
		# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)	# of hubs	# of hub arcs	CPU time (sec)
α=0.2	1851	3	2	0,023	4	3	0,032	3	2	35744,4
	1549	5	5	0,045	6	5	0,05	3	3	7495,25
	1356	6	5	0,053	6	5	0,048	4	3	6371,15
	1162	7	6	0,058	8	7	0,071	5	5	791,11
α=0.4	2067	3	2	0,021	4	3	0,032	2	1	2950,96
	1744	7	10	0,079	5	4	0,041	3	2	3977,87
	1473	6	8	0,065	6	5	0,048	4	3	7064,18
	1386	15	28	0,122	7	7	0,068	5	4	2125,97
α=0.6	2255	3	3	0,026	3	2	0,023	2	1	2742,48
	1996	6	9	0,028	4	4	0,038	3	2	926,92
	1835	8	15	0,094	5	6	0,054	4	4	1887,97
	1663	4	4	0,043	5	7	0,057	5	5	221,91
α=0.8	2493	3	3	0,028	3	3	0,029	2	1	222,88
	2264	3	2	0,025	3	3	0,03	3	3	2426,89
	2154	5	8	0,034	5	7	0,048	4	6	176,39
	2118	6	10	0,053	5	7	0,046	5	5	57,21
α=1.0	2611	4	7	0,032	2	1	0,015	2	1	78,15
	2605	4	7	0,032	2	1	0,014	2	1	78,16
	2601	4	7	0,035	2	1	0,014	2	1	78,24
	2600	4	7	0,035	2	1	0,014	2	1	87,71

Table 11: Results of M_HEUR1, M_HEUR2 and M_HEUR3 for n=20

For n= 20, we again observe the same structure. Most of cases for n=20, M_HEUR3 shows great performance than other heuristics. For all β values of α=1.0, M_HEUR2 and M_HEUR3 generates same results.

	β	M_HEUR1			M_HEUR2		
		# of hubs	# of hub arcs	CPU time(sec)	# of hubs	# of hub arcs	CPU time(sec)
α=0.2	2136	10	16	0,091	3	2	0,024
	1913	6	5	0,05	5	4	0,041
	1617	8	7	0,065	7	6	0,06
	1346	18	25	0,129	8	7	0,069
α=0.4	2401	9	14	0,031	4	3	0,031
	2099	14	23	0,044	5	4	0,043
	1881	6	6	0,057	7	6	0,058
	1597	8	11	0,093	8	10	0,081
α=0.6	2557	5	7	0,024	3	2	0.022
	2336	11	19	0,054	5	6	0,056
	2184	13	30	0,117	5	7	0,058
	2002	9	24	0,155	6	11	0,08
α=0.8	2713	5	9	0,029	3	2	0,024
	2552	6	14	0,063	4	3	0,031
	2457	8	21	0,082	6	12	0,078
	2307	8	23	0,099	7	16	0,083
α=1.0	2826	4	7	0,038	2	1	0,014
	2762	5	10	0,041	2	1	0,014
	2726	5	10	0,036	3	3	0,031
	2725	5	10	0,04	3	3	0,029

Table 12: Results of M_HEUR1, M_HEUR2 and M_HEUR3 for n=25

As mentioned before, M_HEUR3 starts with the solution of complete hub network given by Campbell(1994). We cannot generate optimal results for large sized problems in a reasonable time via CPLEX, so for n=25, we could not get optimal solutions for Campbell's model. We only generate feasible results by using M_HEUR1 and M_HEUR2. When we compare these two heuristics, we see that M_HEUR2 shows better performance than M_HEUR1. CPU times of these heuristics are very close to each other.

In 10 out of 20 cases for n=10, M_HEUR3 finds optimal results but it cannot generate solutions for the instances greater than 20 nodes in a reasonable time. On the other hand, M_HEUR1 and M_HEUR2 generates solutions for instances greater than 20 nodes in at most 0,155 seconds. As it is mentioned in the previous paragraph, in terms of the number of hubs and the number of hub-to-hub links, M_HEUR2 generates better results than M_HEUR1. So, we can say that among the heuristics for multi allocation, M_HEUR2 is the best one.

Chapter 6

6. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this thesis, we consider the “Hub Covering Problem over an Incomplete Hub Network”. Main difference of our study from the existing studies in hub covering literature is that we relax the fully interconnected hub network assumption. This thesis is motivated by the applications of three large scale competitive cargo delivery companies operating in Turkey. In the network structure of these companies, not all the transfer centers are connected to each other. In the service network of two of these companies, each demand center is allocated to exactly one transfer center whereas in the service network of the remaining one, demand centers are allocated to more than one transfer center. Thus, we proposed integer programming formulations for both single and multiple allocation versions of the incomplete network hub covering problem.

We obtain optimal results for only N=10 by solving these models directly in CPLEX 9.1. For generating good solutions for larger instances, we develop three heuristics for both single and multiple allocation versions of the problem. CAB data was used for testing the computational performance of the heuristics.

In order to generate feasible solutions for larger instances, we developed three heuristics for both single (S_HEUR1, S_HEUR2 and S_HEUR3) and multiple (M_HEUR1, M_HEUR2 and M_HEUR3) allocation versions of the problem. M_HEUR1 and M_HEUR2 are modifications of S_HEUR1 and S_HEUR2, respectively. Some steps that allow multiple allocation are added to S_HEUR1 and S_HEUR2, and M_HEUR1 and M_HEUR2 are developed. These four heuristics were coded with C programming language. The underlying idea of S_HEUR3 and M_HEUR3 is the same. In both of the heuristics, first of all complete network hub covering problems that are obtained from literature are solved. Then the optimal hub locations are given as an input of two proposed models, and by solving these models in CPLEX, hub-to-hub links between hubs are determined. During the computational performance of CPLEX and heuristics, CAB data is used.

Among the single allocation heuristics, S_HEUR1 and S_HEUR2 produce faster results but yield weaker results than S_HEUR3. In 14 out of 20 cases for N=10, S_HEUR3 finds optimal results. M_HEUR2 is the most preferable heuristic among the multiple allocation heuristics. Although M_HEUR3 finds optimal results in half of the cases for N=10, it cannot generate feasible results for instances greater than 20 nodes in a reasonable time. Because of that, we can say that M_HEUR2 shows better performance than the others for large sized problems.

As a further research direction, it can be tried to obtain optimal results for larger problems. By finding valid inequalities and proposing new modeling approaches, optimal results for larger problems can be obtained. In this thesis, we define and formulate a new problem. For solving this problem, we proposed constructive heuristics for generating good solutions for our problem. In the future, improvement iterations can be applied to these constructive heuristics. By switching the hub nodes, by changing the assignments or by adding/replacing hub-to-hub links, some neighborhoods can be defined. Then by iterating these neighborhoods, the quality of the solutions can be improved. Similarly, in the future some metaheuristics like tabu search can be applied to our problem.

Bibliography

1. Campbell, J.F (1994) "Integer programming formulations of discrete hub location problems", *European Journal of Operational Research,* 72, 387-405.

2. Campbell, J.F. (1996) "Hub Location and The p-Hub Median Problem", *Operations Research,* 44, 1-13.

3. Campbell, J.F.; Ernst, A.T. and Krishnamoorthy, M.(2005a) "Hub arc location problems: Part I - Introduction and Results", *Management Science*, 51, 1540-1555.

4. Campbell, J.F.; Ernst, A.T. and Krishnamoorthy, M.(2005b) "Hub arc location problems: Part II - Formulations and Optimal Algorithms", *Management Science*, 51, 1540-1555.

5. Drezner, Z. (1982) "Fast algorithms for the round trip location problem", *IIE Transactions*, 14, 243-248.

6. Ebery, J., Krishnamoorthy, M., Ernst, A. and Boland, N. (2000) "The capacitated multiple allocation hub location problem: Formulations and algorithms", *European Journal of Operational Research,* 120, 614-631.

7. Elhedhli, S. and Hu, F.X. (2005) "Hub-and-spoke network design with congestion", *Computers and Operations Research*, 32, 1615-1632.

8. Ernst, A.T. and Krishnamoorthy, M. (1996) "Efficient Algorithms for the Unacapacitated Single Allocation p-Hub Median Problem", *Location Science,* 4, 139-154.

9. 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.

10. Ernst, A.T. and Krishnamoorthy, M. (1999) "Solution algorithms for the capacitated single allocation hub location problem", *Annals of Operations Research,* 86, 141-159.

11. Kara, B.Y. and Tansel, B.Ç. (2000) "On the single-assignment p-hub center problem", *European Journal of Operational Research,* 125, 648-655.

12. Kara, B.Y. and Tansel, B.Ç. (2001) "The Latest Arrival Hub Location Problem", *Management Science,* 47, 1408-1420.

13. Kara, B.Y. and Tansel, B.Ç. (2003) "The single-assignment hub covering problem: Models and linearizations", *Journal of Operational Research Society,* 54, 59-64.

14. Klincewicz, J.G. (1991) "Heuristics for the p-hub location problem", *European Journal of Operational Research,* 53, 25-37.

15. Mayer, G. and Wagner, B. (2002) "HubLocator: an exact solution method for the multiple allocation hub location problem", *Computers and Operations Research*, 29, 715-739.

16. O'Kelly, M.E. (1986) "The Location of Interacting Hub Facilities", *Transportation Science,* 20, 92-105.

17. O'Kelly, M.E. (1987) "A quadratic integer program for the location of interacting hub facilities", *European Journal of Operational Research,* 32, 393-404.

18. O'Kelly, M.E. (1992) "Hub Facility Location With Fixed Costs", *Papers in Regional Science,* 71, 293-306.

19. O'Kelly, M.E. and Miller, H.J. (1991) "Solution Strategies for the Single Facility Minimax Hub Location Problem", *Papers in Regional Science,* 70, 367-380.

20. O'Kelly, M.E. and Miller, H.J. (1994) "The hub network design problem: A review and synthesis", *Journal of Transport Geography,* 2, 31-40.

21. O'Kelly, M.E., Skorin-Kapov, D. and Skorin-Kapov, J. (1995) "Lower Bounds for the Hub Location Problem", *Management Science,* 41, 713-721.

22. 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.

23. Pirkul, H. and Schilling, D.A. (1998) "An Efficient Procedure for Designing Single Allocation Hub and Spoke Systems", *Management Science,* 44, 235-242.

24. Sasaki, M., Suzuki, A. and Drezner, Z. (1999) "On the selection of hub airports for an airline hub and spoke system", *Computers and Operations Research,* 26, 1411-1422.

25. 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.

26. Skorin-Kapov, D., Skorin-Kapov, J. and O'Kelly, M.E. (1996) "Tight linear programming relaxations of uncapacitated p-hub median problems", *European Journal of Operational Research,* 94, 582-593.

27. Sohn, J. and Sungsoo, P. (1997) "A linear program for the two-hub location problem", *European Journal of Operational Research,* 100, 617-622.

28. Sohn, J. and Sungsoo, P. (1998) "Efficient solution procedure and reduced size formulations for p-hub location problems", *European Journal of Operational Research,* 108, 118-126.

29. Wagner, B. (2004) "Model formulations for Hub Covering Problems", *Schriften zur Quantitativen Betriebswirtschaftslehre,* Unpublished

APPENDIX

A-1. Pseudo Code of S_HEUR1:

Representations used in the pseudo code are;

Hub[a]: represents the a^{th} hub

out[s]: represents s^{th} outer node

Asgn(Hub[i],b): represents the b^{th} node which is assigned to a^{th} hub

#hubs: represents the number of hubs

#links: represents the number of hub-to-hub links

#assigned_nodes_H[i]: represents the number assigned nodes to i^{th} hub

#outer: represents the number of outer nodes

algorithm *S_HEUR1*

begin

#hubs←0

#links←0

#assigned_nodes_H[i] ←0 $\forall i \in N$

#outer←0

for i =1 **to** n**;** j = 1 **to** n

begin

If $\alpha * T_{i,j} > \beta$ **then** original problem is infeasible and

Either increase β or stop by infeasibility.

end

for i =1 **to** n**;** j = 1 **to** n

begin

Let $T_{rk} = \arg\max_{i,j} T_{ij}$

```
    Hub[1]←r and increment #hubs
    Hub[2]←k and increment #hubs
   activate Hub[1]-Hub[2] link and increment #links
end
for i =1 to n
begin
   If T_{i,Hub[1]}<= β /2 then
   increment #assigned_nodes_H[1] and
  Asgn(Hub[1], #assigned_nodes_H[1])←i
end
for i =1 to n
begin
  If (T_{i,Hub[2]}<= β /2 )
  and (i ≠ Asgn(Hub[1],b) for all b=1,…, #assigned_nodes_H[1])
  then increment  #assigned_nodes_H[2] and
  Asgn(Hub[2], #assigned_nodes_H[2])←i
end
 for i =1 to n
 begin
   If (i ≠ Hub[1] or Hub[2])
   and (i ≠ Asgn(Hub[1],b) for all b=1,…, #assigned_nodes_H[1])
   and (i ≠ Asgn(Hub[2],b) for all b=1,…, #assigned_nodes_H[2])
   then
   increment #outer and out[#outer]←i
end
CONNECTION_FUNCTION_S(out[s],#outer)
 for i =1 to #assigned_nodes_H[1]; j =1 to #_assigned_nodes_H[2]
 begin
```

```
If T_{Asgn(Hub[2],j),Hub[2]} + α*T_{Hub[2],Hub[1]} + T_{Hub[1],Asgn(Hub[1],i)} > β
 begin
 If α *T_{Asgn(Hub[2],j),Hub[1]} + T_{Hub[1],Asgn(Hub[1],i)} <= β
 begin
   increment #hubs and Hub[#hubs]←Asgn(Hub[2],j)
   increment #links and activate Asgn(Hub[2],j)-Hub[1]) link
   increment #links and activate Asgn(Hub[2],j)-Hub[2]) link
 end
      else
 begin
  increment #hubs and Hub[#hubs]←Asgn(Hub[1],i)
 increment #hubs and  Hub[#hubs]←Asgn(Hub[2],j)
 increment #links and activate Asgn(Hub[1],i)-Hub[1]) link
 increment #links and activate Asgn(Hub[2],j)-Hub[2] link
 increment #links and activate Asgn(Hub[1],i)-Asgn(Hub[2],j)
 link
 end
 end
end
 for i=1 to #hubs; j=1 to #hubs
 begin
    If (Hub[i]-Hub[j] is activated)
   and (α*T_{Hub[i],Hub[1]} + α*T_{Hub[1],Hub[j]} > β)
  then  increment #links and activate Hub[i]-Hub[j] link
 end
end
```

```
procedure CONNECTION_FUNCTION_S(out[s],#outer)
begin
        for i =1 to #assigned_nodes_H[1]; s = 1 to #outer
        begin
        If T_{out[s],Hub[1]}+T_{Hub[1],Asgn(Hub[1],i)}<= β
        then increment #assigned_nodes_H[1] and
        Asgn(Hub[1], #assigned_nodes_H[1])←out[s]
                else
         If T_{out[s],Hub[2]}+ α *T_{Hub[2],Hub[1]} +T_{Hub[1],Asgn(Hub[1],i)}<= β
         then increment  #assigned_nodes_H[2] and
         Asgn(Hub[2], #assigned_nodes_H[2])←out[s]
                else
         If α *T_{out[s],Hub[1]}+T_{Hub[1],Asgn(Hub[1],i)}<= β
         begin
         increment #hubs and Hub[#hubs]←out[s]
         increment #links and activate out[s]-Hub[1] link
         end
                else
        begin
         increment #hubs and Hub[#hubs]←out[s]
         increment  #hubs and Hub[#hubs]←Asgn(Hub[1],i),
         increment #links and activate out[s]-Hub[1]) link
         increment #links and activate Asgn(Hub[1],i)-Hub[1]) link
         increment #links and activate out[s]-Asgn(Hub[1],i) link
         end
         end
end
```

A-2. Pseudo Code of S_HEUR2:

Same representations defined for pseudo code of S_HEUR1 hold for S_HEUR2. There is also an additional representation:

count(i): represent the number of nodes

algorithm *S_HEUR2*

begin

#hubs←0

#links←0

#assigned_nodes_H[i] ←0 $\forall i \in N$

#outer←0

count(i)←0 $\forall i \in N$

for i =1 **to** n**;** j = 1 **to** n

begin

If $\alpha * T_{i,j} > \beta$ **then** original problem is infeasible and

Either increase β or stop by infeasibility.

end

for i = 1 **to** n**;** j=1 **to** n

begin

Take T_{ij} and **if** $T_{ij} <= \beta/2$ **then increment** *count(i)*

Let $k = \underset{i}{\text{argmax}}(count(i))$

end

increment #hubs and Hub[#hubs] ←k

for i =1 **to** n

begin

If $T_{i,Hub[1]} <= \beta/2$ **then**

increment #assigned_nodes_H[1] and

```
  Asgn(Hub[1], #assigned_nodes_H[1])←i
end
for i =1 to n
begin
   If T_{i,Hub[1]}>β then
   begin
   increment #hubs and Hub[#hubs]←i
    increment #links and activate i-Hub[1] link
  end
end
for i =1 to n
begin
    If (i≠Hub[t] for all t=1,…,#hubs)
    and (i≠Asgn(Hub[1],b) for all b=1,…, #assigned_nodes_H[1])
   then increment #outer and out[#outer]←i
end
CONNECTION_FUNCTION_S(out[s],#outer)
for i=1 to n; j=1 to #hubs; k=1 to #hubs
begin
    If (i≠Asgn(Hub[j],b) for all b=1,…, #assigned_nodes_H[j])
   and (i=Asgn(Hub[k],b) for one of b=1,…, #assigned_nodes_H[k])
   and (T_{i,Hub[k]}+α*T_{Hub[k],Hub[1]} + α*T_{Hub[1],Hub[j]}>β)  then
   begin
   increment #hubs and Hub[#hubs]←i
   increment #links and activate i-Hub[j] link
   increment #links and activate i-Hub[k]) link
   end
 end
```

for i=1 **to** #hubs**;** j=1 **to** #hubs; k=1 **to** #hubs

begin

If (Hub[i]-Hub[j] link is **activated**) and (Hub[i]-Hub[k] link is **activated** and ($\alpha * T_{Hub[i],Hub[k]} + \alpha * T_{Hub[k],Hub[1]} + \alpha * T_{Hub[1],Hub[j]} > \beta$)

then increment #links and **activate** Hub[i]-Hub[j] link

end

end

C2' (The modified step in *CONNECTION_FUNCTION_S(out[s],#outer)***)**

for t = 1 **to** #hubs

begin

If $T_{out[s],Hub[t]} + \alpha * T_{Hub[t],Hub[1]} + T_{Hub[1],Asgn(Hub[1],i)} <= \beta$

Then increment #assigned_nodes_H[1] and

Asgn(Hub[t], #assigned_nodes_H[t])←out[s]

end

A-3. Pseudo Code of M_HEUR1:

Additional representations;

Out1[s]: represents s^{th} outer node before connecting to nodes assigned to Hub 1

Out2[s]: represents s^{th} outer node before connecting to nodes assigned to Hub 2

#outer1: represents the number of outer nodes before connecting to nodes assigned to Hub 1

#outer2: represents the number of outer nodes before connecting to nodes assigned to Hub 2

algorithm *M_HEUR1*
begin
    #hubs ← 0
    #links ← 0
    #assigned_nodes_H[i] ← 0 $\forall i \in N$
    #outer1 ← 0
    #outer1 ← 0
    for i =1 **to** n**;** j = 1 **to** n
    begin
        If $\alpha * T_{i,j} > \beta$ **then** original problem is infeasible and
        Either increase β or stop by infeasibility.
    end
    for i =1 **to** n**;** j = 1 **to** n
    begin
        Let $T_{rk} = \arg\max_{i,j} T_{ij}$

Hub[1]←r and **increment** #hubs

Hub[2]←k and **increment** #hubs

activate Hub[1]-Hub[2] link and **increment** #links

end

for i =1 **to** n

begin

If $T_{i,Hub[1]}$<= β /2 **then**

increment #assigned_nodes_H[1] and

Asgn(Hub[1], #assigned_nodes_H[1])←i

end

for i =1 **to** n

begin

If ($T_{i,Hub[2]}$<= β /2)**then**

increment #assigned_nodes_H[2] and

Asgn(Hub[2], #assigned_nodes_H[2])←i

end

for i =1 **to** n

begin

If (i ≠ Hub[1] or Hub[2])

and (i ≠ Asgn(Hub[1],b) for all b=1,…, #assigned_nodes_H[1])

and (i ≠ Asgn(Hub[2],b) for all b=1,…, #assigned_nodes_H[2])

then increment #outer1 and out1[#outer1]←i

end

CONNECTION_FUNCTION_M(out1[s],#outer1,Hub[1])

for i =1 **to** n

begin

If (i ≠ Hub[1] or Hub[2])

and (i ≠ Asgn(Hub[1],b) for all b=1,…, #assigned_nodes_H[1])

and (i $\neq$ Asgn(Hub[2],b) for all b=1,…, #assigned_nodes_H[2])

then increment #outer2 and out2[#outer]←i

end

CONNECTION_FUNCTION_M(out2[s],#outer2,Hub[2])

for i =1 **to** #assigned_nodes_H[1]**;** j =1 **to** #assigned_nodes_H[2]

begin

If $T_{Asgn(Hub[2],j),Hub[2]} + \alpha * T_{Hub[2],Hub[1]} + T_{Hub[1],Asgn(Hub[1],i)} > \beta$

begin

for t = 1 **to** #hubs

begin

If $T_{Asgn(Hub[1],i),Hub[t]} + T_{Hub[t],Asgn(Hub[2],j)} <= \beta$

begin

If (t $\neq$ 1) **and**

(i $\neq$ Asgn(Hub[t],b) for all b=1,…, #assigned_nodes_H[t])

then increment #assigned_nodes_H[t] and

Asgn(Hub[t], #assigned_nodes_H[t])←Asgn(Hub[1],i)

If (t $\neq$ 2) **and**

(j $\neq$ Asgn(Hub[t],b) for all b=1,…, #assigned_nodes_H[t])

then increment #assigned_nodes_H[t] and

Asgn(Hub[t], #assigned_nodes_H[t])←Asgn(Hub[2],j)

end

end

else

If ($\alpha * T_{Asgn(Hub[1],i),Hub[2]} + T_{Hub[2],Asgn(Hub[2],j)} <= \beta$)

begin

increment #hubs and Hub[#hubs]←Asgn(Hub[1],i)

increment #links and **activate** Asgn(Hub[1],i)-Hub[1]) link

increment #links and **activate** Asgn(Hub[1],i)-Hub[2]) link

end

else

begin

increment #hubs and Hub[#hubs]←Asgn(Hub[1],i)

increment #hubs and Hub[#hubs]→Asgn(Hub[2],j)

increment #links and **activate** Asgn(Hub[1],i)-Hub[1] link

increment #links and **activate** Asgn(Hub[2],j)-Hub[2] link

increment #links and **activate** Asgn(Hub[1],i)- Asgn(Hub[2],j) link

end

end

end

for i=1 **to** #hubs**;** j=1 **to** #hubs; k=1 **to** #hubs

begin

If (Hub[i]-Hub[j] link is **activated**) **and** (Hub[i]-Hub[k] link is **activated**) **and** (Hub[k]-Hub[j] link is **activated**)

and ($\alpha * T_{Hub[i],Hub[k]} + \alpha * T_{Hub[k],Hub[j]} > \beta$)

then increment #links and **activate** Hub[i]-Hub[j] link

end

end

procedure *CONNECTION_FUNCTION_M(out[s],#outer,Hub[a])*

begin

for i =1 **to** #assigned_nodes_Hub[a]**;** s = 1 **to** #outer

begin

If $T_{out[s],Hub[a]} + T_{Hub[a],Asgn(Hub[a],i)} <= \beta$,

then increment #assigned_nodes_Hub[a] and Asgn(Hub[a], #assigned_nodes_Hub[a])←out[s]

else

for t = 1 **to** #hubs

begin

If $T_{out[s],Hub[t]} + \alpha * T_{Hub[t],Hub[a]} + T_{Hub[a],Asgn(Hub[a],i)} <= \beta$

then increment #assigned_nodes_Hub[t] and

Asgn(Hub[t], #assigned_nodes_Hub[t])←out[s]

end

else

for t = 1 **to** #hubs

begin

If $T_{out[s],Hub[t]} + T_{Hub[t],Asgn(Hub[a],i)} <= \beta$

begin

increment #assigned_nodes_Hub[t] and

Asgn(Hub[t], #assigned_nodes_Hub[t])←out[s]

increment #assigned_nodes_Hub[t] and

Asgn(Hub[t], #assigned_nodes_Hub[t])←Asgn(Hub[1],i)

end

end

else

If $\alpha * T_{out[s],Hub[a]} + T_{Hub[a],Asgn(Hub[a],i)} <= \beta$

begin

increment #hubs and Hub[#hubs]←out[s]

increment #links and **activate** out[s]-Hub[a] link

end

else

If $T_{out[s],Asgn(Hub[a],i)} <= \beta$

begin

increment #hubs and Hub[#hubs]←out[s]

increment #assigned_nodes_Hub[#hubs] and

Asgn(Hub[#hubs], #assigned_nodes_Hub[#hubs])←Asgn(Hub[1],i)

increment #links=#links+1 and **activate** out[s]-Hub[a] link

end

else

begin

increment #hubs and Hub[#hubs]←out[s]

increment #hubs and Hub[#hubs]← Asgn(Hub[1],i)

increment #links and **activate** out[s]-Hub[a] link

increment #links and **activate** Asgn(Hub[a],i)-Hub[a] link

increment #links and **activate** out[s]-Asgn(Hub[a],i) link

end

end

end