Betül Kayışoğlu

A Ph.D. Thesis

AGU 2022

DEVELOPMENT OF MODELS AND SOLUTION METHODOLOGIES FOR TREE OF HUBS LOCATION AND ARC CAPACITATED HUB LOCATION PROBLEMS

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING & SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Betül KAYIŞOĞLU

January 2022

DEVELOPMENT OF MODELS AND SOLUTION METHODOLOGIES FOR TREE OF HUBS LOCATION AND ARC CAPACITATED HUB LOCATION PROBLEMS

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING & SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Betül KAYIŞOĞLU

January 2022

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Betül KAYIŞOĞLU

REGULATORY COMPLIANCE

Ph.D. thesis titled **“DEVELOPMENT OF MODELS AND SOLUTION METHODOLOGIES FOR TREE OF HUBS LOCATION AND ARC CAPACITATED HUB LOCATION PROBLEMS”** has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Betül KAYIŞOĞLU	Prof. Dr. İbrahim AKGÜN

Head of the Industrial Engineering Graduate Program

Prof. Dr. İbrahim AKGÜN

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled **"DEVELOPMENT OF MODELS AND SOLUTION METHODOLOGIES FOR TREE OF HUBS LOCATION AND ARC CAPACITATED HUB LOCATION PROBLEMS"** and prepared by Betül KAYIŞOĞLU has been accepted by the jury in the Industrial Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

27/01/2022

JURY:

Prof. Dr. İbrahim AKGÜN :……………………………

Prof. Dr. Bahar YETİŞ KARA :……………………………

Prof. Dr. Emel KIZILKAYA AYDOĞAN :……………………………

Assoc. Prof. Ayşegül ALTIN KAYHAN :……………………………

Assist. Prof. Dr. Selçuk GÖREN :……………………………

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / ……….. and numbered …………..……. .

….. /….. / 2022

Graduate School Dean

Prof. Dr. İrfan ALAN

ABSTRACT

DEVELOPMENT OF MODELS AND SOLUTION METHODOLOGIES FOR TREE OF HUBS LOCATION AND ARC CAPACITATED HUB LOCATION PROBLEMS

Betül KAYIŞOĞLU
Ph.D. in Industrial Engineering
Advisor: Prof. Dr. İbrahim AKGÜN

January 2022

In this dissertation, we study two different extensions to hub location problems, namely, *Multiple Allocation Tree of Hubs Location Problem (MATHLP)* that result from incorporating a tree topology requirement for the hub network and *Multiple Allocation Arc Capacitated Hub Location Problem (MACHLP)* that result from imposing capacities on the arcs. We consider both problems in a multiple allocation framework and try to minimize total flow cost by locating *p* hubs. Unlike most studies in the literature that use complete networks with costs satisfying the triangle inequality to formulate the problems, we define the problems on non-complete networks and develop a modeling approach that does not require any specific cost and network structure. Our proposed approach provides more flexibility in modeling several characteristics of real-life hub networks. We solve the proposed models using CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic. For MATHLP, we develop Benders decomposition-based heuristic algorithms and for MACHLP, we develop a heuristic algorithm based on simulated annealing. We conduct computational experiments using problem instances defined on non-complete networks with up to 500 and 400 nodes for MATHLP and MACHLP respectively. The results indicate that the proposed solution methodologies are especially effective in finding good feasible solutions for large instances.

Keywords: hub location problem, tree of hubs location problem, arc capacitated hub location problem, benders-type heuristics, simulated annealing

ÖZET

AĞAÇ YAPILI VE AYRIT KAPASİTELİ HUB YERLEŞİM PROBLEMLERİ İÇİN MODEL VE ÇÖZÜM METODOLOJİLERİNİN GELİŞTİRİLMESİ

Betül KAYIŞOĞLU
Endüstri Mühendisliği Anabilim Dalı Doktora
Tez Yöneticisi: Prof. Dr. İbrahim AKGÜN

Ocak 2022

Bu tezde, ana dağıtım üsleri (hub) arasında bir ağaç topolojisi gerektiren *Çok Atamalı Ağaç Yapılı Hub Yerleşim Problemi (AYHYP)* ve ayrıtlar üzerinden geçen akışlara üst limitler getiren *Çok Atamalı Ayrıt Kapasiteli Hub Yerleşim Problemi (AKHYP)* çalışılmıştır. Her iki problemde de çoklu atama stratejisi kullanılmıştır ve p adet hub yerleştirilerek toplam akış maliyeti en aza indirilmeye çalışılmıştır. Problemler için formülasyon geliştirilmesinde maliyetleri üçgen eşitsizliğini sağlayan tam serimlerin kullanıldığı literatürdeki birçok çalışmanın aksine, her iki problem tam olmayan serimler üzerinde tanımlanmış ve özel bir maliyet ile serim yapısı gerektirmeyen bir modelleme yaklaşımı geliştirilmiştir. Önerilen yaklaşım, gerçek hayattaki hub serimlerinin çeşitli özelliklerini modellemede daha fazla esneklik sağlamaktadır. Önerilen modeller, CPLEX tabanlı dal ve sınır algoritması ve NoRel sezgiseli ile birlikte Gurobi tabanlı dal ve sınır algoritması kullanılarak çözülmüştür. AYHYP için Benders ayrıştırma tabanlı sezgisel algoritmalar ve AKHYP için benzetimli tavlama metoduna dayalı bir sezgisel algoritma geliştirilmiştir. AYHYP ve AKHYP için sırasıyla 500 ve 400 düğüme kadar tam olmayan serimlerde tanımlanan problem örnekleri kullanılarak testler gerçekleştirilmiştir. Test sonuçları, önerilen çözüm metodolojilerinin özellikle büyük örnekler için iyi çözümler bulmada etkili olduğunu göstermektedir.

Anahtar kelimeler: Ana dağıtım üssü yer seçimi problemi, ağaç yapılı ana dağıtım üssü yer seçimi problemi, ayrıt kapasiteli ana dağıtım üssü yer seçimi problem, Benders ayrıştırma tabanlı sezgiseller, benzetimli tavlama

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. İbrahim Akgün for all his help, guidance and patience during my whole Ph.D. education. This dissertation began and was shaped with his invaluable ideas. Besides his guidance on conceptual, methodological and editorial aspects of this dissertation, I would also like to express my appreciation and thanks to him for he has been a mentor for me. What I learned from him about teaching and academic studies will always help me on the way to become a good academician. I feel lucky and privileged to have such a ‘*rigorous*’ advisor.

Besides my advisor, I would like to thank the rest of my dissertation monitoring committee: Assist. Prof. Selçuk Gören and Prof. Bahar Yetiş Kara for their insightful comments and support. Assist. Prof. Selçuk Gören also supported me by giving precious Ph.D. courses and taught many issues related to this dissertation within those courses. I would like to thank him for strengthening my background for being able to carry out this dissertation. I also feel very privileged that Prof. Bahar Yetiş Kara, an expert in ‘hub location’ is a member of my dissertation committee. Last but not least, I would like to thank Assist. Prof. Pınar Zarif Tapkan who was a member of my dissertation monitoring committee in the first two years and always encouraged my studies.

I would also like to express my appreciation to the rest my dissertation committee: Prof. Emel Kızılkaya Aydoğan and Assoc. Prof. Ayşegül Altın Kayhan for devoting their valuable time to read and review this dissertation. Their suggestions and comments are of great value to the quality of this dissertation.

I am also indebted to my family. My parents, Mehmet and Ayşe Nalbantoğlu always supported me during my whole education life. It is great to feel that they are always proud of me. My husband Esat Kayışoğlu always motivated me and believed in me that I can succeed. My daughters Nevin and İpek always brought joy to my life and helped me to get rid of my stress during this period. I also like to thank all the members of my ‘Nalbantoğlu’ and ‘Kayışoğlu’ families.

Finally, I am deeply grateful to my departmental colleagues, Şeyma Bekli, İsmet Söylemez, Abdülkerim Benli for being always there for me with both academic and spiritual support. Moreover, graduate study would not have been bearable without my colleagues Tuğba Değirmenci, Özlem Kevseroğlu, Nehir Gümüşlü, Gül Gündüz, Fulya

Keser Uslucan, Yasemin Doğan and Mithat Gökhan Atahan who always motivated me during this period.

I acknowledge that this dissertation was supported by the Scientific and Technological Research Council of Turkey **(TÜBİTAK Grant No: 114M363)** and the Research Fund of the Abdullah Gül University **(Grant No: FDK-2018- 123)**.

TABLE OF CONTENTS

1. INTRODUCTION 1
1.1 MULTIPLE ALLOCATION TREE OF HUBS LOCATION PROBLEM (MATHLP) 3
1.2 MULTIPLE ALLOCATION ARC CAPACITATED HUB LOCATION PROBLEM (MACHLP) 6
2. MULTIPLE ALLOCATION TREE OF HUBS LOCATION PROBLEM 9
2.1. LITERATURE REVIEW 10
2.1.1. Single Allocation Tree of Hubs Location Models based on the Classical Approach 12
2.1.2. Multiple Allocation Tree of Hubs Location Models based on the Classical Approach 15
2.2 COMPARISON OF THE HUB NETWORKS FOR DIFFERENT MODELING APPROACHES 17
2.3 PROBLEM DEFINITION AND MATHEMATICAL FORMULATION 24
2.4 PROPOSED SOLUTION METHODOLOGY 30
2.4.1 Benders Decomposition for MATHLP 32
2.4.2 Acceleration Strategies for the BD Algorithm 34
2.4.3 Benders-Type Heuristic Approach 37
2.5 COMPUTATIONAL EXPERIMENTS 40
2.5.1 Computational experiments for MATHLP using CPLEX and Gurobi with NoRel heuristic 41
2.5.2 Computational Experiments with the Benders-type Heuristics 42
2.6 CONCLUSION 45
3. MULTIPLE ALLOCATION ARC CAPACITATED HUB LOCATION PROBLEM 47
3.1 LITERATURE REVIEW 48
3.1.1 Arc Capacitated Hub Location Models based on Classical Approach 50
3.2 COMPARISON OF THE HUB NETWORKS FOR DIFFERENT MODELING APPROACHES 53
3.3 PROBLEM DEFINITION AND MATHEMATICAL FORMULATION 59
3.4 PROPOSED SOLUTION METHODOLOGY 61
3.4.1 The Overall SA Procedure 62
3.5 COMPUTATIONAL EXPERIMENTS 64
3.5.1 Computational Experiments for MACHLP using CPLEX and Gurobi with NoRel Heuristic 65
3.5.2 Computational Experiments with the Proposed SA-based Solution Methodology 69
3.6 CONCLUSION 75
4. CONCLUDING REMARKS AND FUTURE DIRECTIONS 77
4.1 CONCLUSIONS 77
4.2 SOCIETAL IMPACT AND CONTRIBUTION TO GLOBAL SUSTAINABILITY 79
4.3 FUTURE PROSPECTS 80

LIST OF FIGURES

Figure 1.1. Curitiba's transportation system 5
Figure 1.2 A public transportation system with different types of transportation modes and changing arc capacities changing according to the transportation mode used 7
Figure 2.1. Non-complete transportation network consisting of 30 cities of Turkey. 18
Figure 2.2. Tree structures obtained using a complete network whose distances do not satisfy the triangle inequality. (a), (b), and (c) represent the original complete network, the resulting hub network using the proposed model, and the resulting the hub network using the classical model. 20
Figure 2.3 A real-world example showing that the triangle inequality does not hold. ... 21
Figure 2.4. Optimal hub network obtained with the proposed model ($\boldsymbol{H}$=30, $\boldsymbol{p}$=3)....... 22
Figure 2.5. Optimal hub network obtained with the classical model ($\boldsymbol{H}$=30, $\boldsymbol{p}$=3)........ 22
Figure 2.6. Optimal hub network obtained with the proposed model ($\boldsymbol{H}$=20, $\boldsymbol{p}$=8)....... 23
Figure 2.7. Optimal hub network obtained with the classical model ($\boldsymbol{H}$=20, $\boldsymbol{p}$=8)........ 24
Figure 2.8. Three-layer MN $\boldsymbol{G0}$ constructed from RealN $\boldsymbol{G}$ 27
Figure 2.9. Illustration of the tree spanning all nodes in the hub layer (a) and the resulting tree structure connecting the hubs (b). 30
Figure 3.1 A non-complete network RealN with different arc capacities. 54
Figure 3.2 Non-complete transportation network consisting of 8 cities of Eagean Region of Turkey. 55
Figure 3.3 Hub Level Network (HLN) obtained using MACHLP and CAMARC when hub arc capacities are 40. 57
Figure 3.4 Flow routes of commodity 7 originating from node 7 using MACHLP and CAMARC when all arc capacities are 120 58
Figure 3.5. UB values achieved with the proposed SA based solution methodology as time passes for the test instances on PMED300. 70

LIST OF TABLES

Table 2.1 Sets, indices, parameters, and decision variables of the proposed model.......24
Table 2.2 Test results for MATHLP using CPLEX and Gurobi with NoRel heuristic. .42
Table 2.3 Test results for the instances using BDHEUR (T=24h) 44
Table 3.1 Objective function values achieved and hub nodes located with the proposed Model MACHLP and Model CAM_ArcCap when hub arc capacities are imposed. 56
Table 3.2 Objective function values achieved and hub nodes located with the proposed Model MACHLP and Model CAM_ArcCap when hub and access arc capacities are imposed...57
Table 3.3 Test results for MACHLP using CPLEX and Gurobi with NoRel heuristic with hub arc capacities. 67
Table 3.4 Test results for MACHLP using CPLEX and Gurobi with NoRel heuristic with hub and access arc capacities...68
Table 3.5 Test results for MACHLP using the proposed SA-based heuristics with hub arc capacities (T=24h)..72
Table 3.6 Test results for MACHLP using the proposed SA-based heuristics with hub and access arc capacities (T=24h) 73

LIST OF ABBREVIATIONS

OD	: Origin-destination
HLP	: Hub location problem
THLP	: Tree of hubs location problems
MATHLP	: Multiple Allocation Tree of Hubs Location Problem
MACHLP	: Multiple Allocation Arc Capacitated Hub Location Problem
RealN	: Real-world network
MN	: Modeled network
HN	: Hub network
HLN	: Hub-level network
AN	: Access network
BD	: Benders decomposition
SA	: Simulated Annealing
MIP	: Mixed Integer Programming
MP	: Master problem at iteration h of BD algorithm
SP	: Subproblem at iteration h of BD algorithm
Model SPD	: Linear problem obtained by fixing integer variables in MATHLP at iteration h
Model SP	: Subproblem obtained by taking the dual of SPD at iteration h
Model PRT	: Model to find a pareto-optimal solution
Model MPM	: Master problem with disaggregated cuts at iteration h
Model RelMP	: Relaxed master problem with a single cut
Model MCMP	: Relaxed master problem with multiple cuts
SPTree	: Rooted spanning tree formulation
BDHEUR1	: Benders-Type Heuristic 1 employing strong cut generation
BDHEUR2	: Benders-Type Heuristic 2 employing strong cut generation and cut disaggregation,

Chapter 1

INTRODUCTION

Hubs act as aggregation, distribution, switching, and sorting centers in telecommunication, transportation, and computer networks where commodities (e.g., data, packages, etc.) are sent between many origin-destination (OD) pairs. In these networks, instead of sending flows directly between each OD pair, the flows are sent through hub facilities in at most three movements: collection from the origin to a hub, transfer between hubs, and distribution from the last hub to the destination (transfer movement may be skipped for some OD pairs). The transfer of consolidated flow between hubs enables to capture the economies of scale. Moreover, advantages resulting from reducing setup costs, centralized commodity handling, and sorting operations are obtained.

A generic Hub Location Problem (HLP) is concerned with determining the locations of hubs, allocation of supply and demand points to hubs, and determining the routes between OD pairs such that total cost is minimized. The research on hub location addresses different types of problems, e.g., *p-hub median problem* locates *p* hubs and minimizes total transportation cost, *p-hub center problem* minimizes maximum transportation cost between OD pairs by locating *p* hubs, *hub location problem with fixed costs* assumes that the number of hubs to locate is not known a priori and focuses on minimizing the sum of hub fixed costs and total transportation cost, and *hub covering problem* maximizes the demand covered with a given number of hubs to locate. These problems are also categorized as single allocation and multiple allocation. In *single allocation* problems, all the incoming and outgoing traffic of each node is routed through a single hub. In *multiple allocation* problems, each origin (destination) node can be allocated to more than one hub to send (receive) flows. For a comprehensive review of the problems, see, for instance, Alumur and Kara [1], Campbell and O'Kelly [2], Farahani et al. [3], Contreras and O'Kelly [4], and Alumur [5]).

Akgün and Tansel [6] give a network terminology for HLPs that we will also use to further discussion. They specify five different different types of networks: (1) Real-world network (RealN): The physical network, e.g., road and rail networks, in which hub system will operate. (2) Modeled network (MN): The network used as an input in developing a model for the problem. MN is not necessarily the same as RealN but may be obtained from RealN through preprocessing. (3) Hub network (HN): The sub-network of MN that consists of the hub nodes, non-hub nodes, and the arcs on the service routes between OD pairs. (4) Hub-level network (HLN): The subnetwork of HN consisting of the hub nodes and the hub arcs connecting them. (5) Access network (AN): The sub-network of HN consisting of the hub nodes, non-hub nodes, and access arcs that connect non-hub origin and destination nodes to hub nodes.

The models developed for HLPs in the literature assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. If the real-world network RealN is not complete or complete but its distances do not satisfy the triangle inequality (e.g., bus fares) as is the case for most real-life networks, a preprocessing on RealN is required to construct a complete MN by an algorithm (e.g., Floyd [7]) that finds the shortest path lengths between all OD pairs in RealN. Thus, the resulting complete MN consists of the shortest path lengths in RealN and hence its distances satisfy the triangle inequality. That is, *an arc in a complete MN* (and hence in a hub network HN) *may actually correspond to a shortest path consisting of several arcs and not necessarily a single arc in RealN.* Most studies do not differentiate between RealN and MN and directly assume that a complete MN with arc distances satisfying the triangle inequality is given. Even though this approach has gained acceptance, this may cause several modeling and computational disadvantages. For example, when the triangle inequality is not satisfied, the models do not work correctly (e.g., Marin et al. [8]). Moreover, the shortest path may not be preferred or may not be the path with the least cost in some cases, e.g., communication networks.

Akgün and Tansel [6] discuss these issues in detail and propose a problem setting and modeling framework that allows (non-complete or complete) RealN with any cost structure to be directly used as MN. They present the modeling framework in the context of the *p*-hub median problem defined on non-complete networks (RealN) and show how to extend it to handle different hub location problems. The approach provides flexibility in modeling several characteristics of real-life hub networks, e.g., the interactions between location and routing decisions, arcs with different costs and capacities, different topology and service level requirements.

Considering the advantages resulting from the problem setting and modeling framework proposed by Akgün and Tansel [6], we study two different hub location problems, namely, *Multiple Allocation Tree of Hubs Location Problem (MATHLP)* and *Multiple Allocation Arc Capacitated Hub Location Problem (MACHLP),* built upon the problem setting proposed by Akgün and Tansel [6]. In MATHLP, a tree topology requirement is imposed in the hub level network. In MACHLP, capacities are imposed on the arcs of the network. We consider both problems in a multiple allocation framework and try to minimize total flow cost by locating *p* hubs.

1.1 Multiple Allocation Tree of Hubs Location Problem (MATHLP)

The *Multiple Allocation Tree of Hubs Location Problem* (MATHLP) imposes a tree topology requirement on the part of the network where the transfer between hubs is carried out, i.e., on the backbone or hub-level network, using multiple allocation strategy. Tree topology required for MATHLP on the HLN has been used or suggested for applications in railway transportation, telecommunication networks, electricity and water distribution networks, and pipeline transportation, where the connectivity between hubs is required but the setup costs for inter-hub links are significant. For example, a backbone network with tree topology is required in designing the high-speed train network in Spain with the stations being hubs [9]; in private data networks, metropolitan area networks, and community antenna television network systems in a hierarchical structure with concentrators (that aggregate and forward data packets) being hubs [10]; in gas pipeline networks with valve sets being hubs, which receive gas produced in wells through production pipes and transfer it to a station via gathering pipes [11]; in electricity power distribution networks with distribution substations being hubs [12]; in urban and public transport network with transfer points between cities and towns being hubs [13]. The aforementioned *tree of hubs location* studies consider single allocation framework and try to minimize the total fixed and flow cost. In this dissertation, we study *tree of hubs location problem* considering multiple allocation framework and minimize the total transportation cost.

As a motivating example for MATHLP, consider the problem of determining a public transportation network in a city where the nodes represent bus/rail stations and the arcs represent the roads/railways between bus/rail stations. The objective is to move the passengers from their origin stations to their destination stations such that the total transportation cost (e.g., travel time or operating cost) is minimized. There are different types of public transport networks, e.g., direct, trunk-and-feeder, radial, diameter, hybrid. [14]. In a trunk-and-feeder system, which is similar to the hub-and-spoke system, the demand on the feeder routes is served by small vehicles and combined on the trunk routes so that passengers from multiple feeder routes can all use a much larger trunk vehicle. The operating costs of large vehicles per passenger are lower than those of smaller vehicles. The trunk network, i.e., the back-bone network, has mostly a tree structure, e.g., a tram network and/or a road network with high-capacity buses operating with higher frequency along rapid transit corridors or paths. The stations on the backbone network that are incident to feeder lines are transfer points (hub nodes) where passengers change line. In a trunk-and-feeder system, a demand point is served by a single line and most passengers need to make two transfers, which is not desired by passengers. In this regard, a hybrid system where direct travel service between some OD pairs is made possible by allowing some bus lines to go beyond the trunk route as necessary. In such a case, these bus lines are required to pass through a transfer point on the backbone network for the passengers that need to change line. Moreover, more than one bus line may be planned to serve the same demand point (bus station) so that passengers can prefer the line closer to their destinations, i.e., multiple allocation. Most of the cities in Turkey operate such a hybrid public transport system. A tree-like backbone network is especially preferred in cities where the metro/tram system is still in its infancy. The metro/tram network is complemented by high-capacity buses operated on rapid transit corridors. In the context of public transportation, MATHLP allows us to determine the physical network including the tree-like trunk routes on which the flow of passengers is achieved with the minimum cost. The resulting network and passenger loads on the links may be used to determine bus lines and their frequencies in accordance with the public transport service planning process [15].

An example of MATHLP is the public transportation system in Curitiba, the capital and largest city in the Brazilian state of Paraná. Curitiba’s public transportation system is known worldwide as an example of a pragmatic, integrated, cost-effective, and efficient transportation system [16]. Figure 1.1 shows Curitiba’s public transportation network consisting of the lines (red) used by express buses and the lines (yellow, green, and orange) used by other buses. Red lines constitute the backbone network in a tree structure, i.e., the *hub-level network* is a tree. Other bus lines (yellow, green, and orange) are connected to the backbone network not at a single point, that is, a passenger may use more than one hub implying the multiple allocation case.

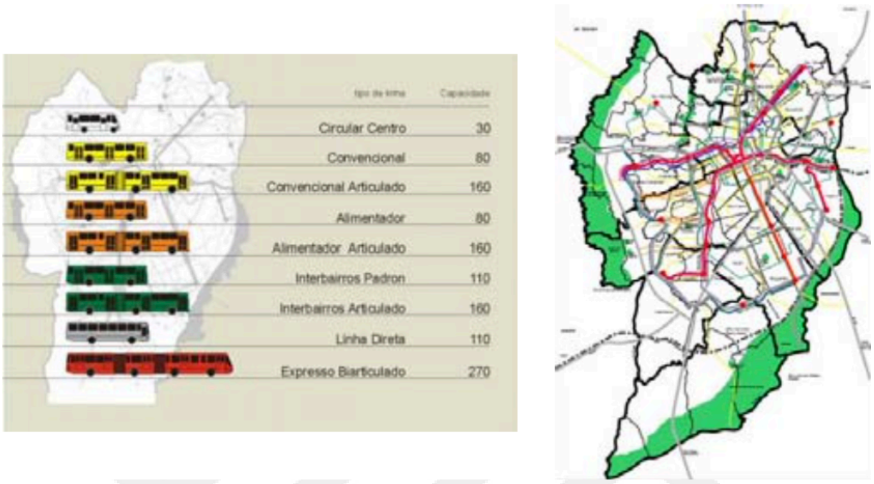

Figure 1.1. Curitiba's transportation system (*Adapted from Rehan and Mahmoud* [16])

We propose a new *mixed integer programming* (MIP) model for MATHLP built upon the problem setting adopted by Akgün and Tansel [6]. The proposed model is defined on a non-complete network but can also be used with complete networks. We solve the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic. MATHLP is difficult to solve using standard optimization software. This has led us to develop specialized solution methodologies. We develop Benders decomposition (BD)-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We conduct computational tests to assess the performance of the proposed heuristics using test instances on networks with up to 500 nodes. As the network size gets larger, the resulting optimality gaps get higher for the solutions found by CPLEX or Gurobi with NoRel heuristic. On the other hand, the proposed BD-based heuristics can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for those instances.

1.2 Multiple Allocation Arc Capacitated Hub Location Problem (MACHLP)

The *Multiple Allocation Arc Capacitated Hub Location Problem* (MACHLP) imposes an upper limit on the flow traversing the arcs of the network. Incorporation of capacity considerations when designing hub networks is an important extension to HLPs to delimit the allowable activities and operations. Capacity constraints on hub networks may be imposed both on nodes and arcs of the network. However, most studies, e.g., Campbell [17], Ernst and Krishnamoorthy [18], Ebery et al. [19], Boland et al. [20], Labbé et al. [21], Contreras et al. [22], Correia et al. [23], and Contreras et al. [24] consider capacity constraints only on incoming or outgoing flow at hub nodes. Contreras and O'Kelly [4] and Alumur et al. [25] state that, in the case of HLPs, the capacity constraints may arise not only at the hub facilities but also on the arcs of the network. There are several studies incorporating arc capacities in HLPs, e.g., Bryan [26], Sasaki and Fukushima [27], Rodríguez-Martín and Salazar-González [28] and Lin et al. [29]. The aforementioned studies assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. However, this assumption does not allow directly to model arc capacities existing in real transportation networks, as will be discussed in Section 3. In this regard, we deviate from the literature by adopting a modeling framework based on non-complete networks, i.e., real network, which allows us to model arc capacities directly.

Arc capacitated hub location problems may arise in telecommunication networks on which servers can be considered as hubs and fiberoptic cables can be considered as arcs. Fiberoptic cables can transit a limited amount of data in a certain time period. Water or natural gas distribution networks also have certain capacities on the pipes that limit the amount of water or natural gas transported in a given amount of time. Moreover, road, rail, and airway transportation networks may have arc capacities imposed by the number of available vehicles and the capacity of the infrastructure. Available vehicles on the hub level network may be over-size vehicles, trains, and airplanes while available vehicles on the access network may be small-size vehicles, trains, and airplanes. Bridges, subways, canals, and straits are examples of the infrastructure that puts a limit on transportation capacity. Just like the arc capacities on telecommunication, water, and gas distribution networks, arc capacities on transportation networks should have a temporal dimension, e.g., the number of drivers that can pass through a bridge in a day.

Public transportation networks may be arc capacitated as well. Kaveh et al. [30] presents a new multi-modal hub location problem for the design of an urban public transportation network imposing a limited capacity on both hubs and hub arcs. As a case study, they design an efficient public transportation network for the Qom city located in Iran. In each hub node, they establish one or both bus rapid transit (BRT) and metro stations and in each hub link, they establish one of the BRT or metro lines. Since the number of available BRT buses and metro trains change, the number of passengers travelling through the hub arcs depends on the transportation mode chosen for that hub arc. In other words, hub arcs have different capacities. Figure 1.2 illustrates one of the public transportation systems proposed by Kaveh et al. [30] for Qom city.

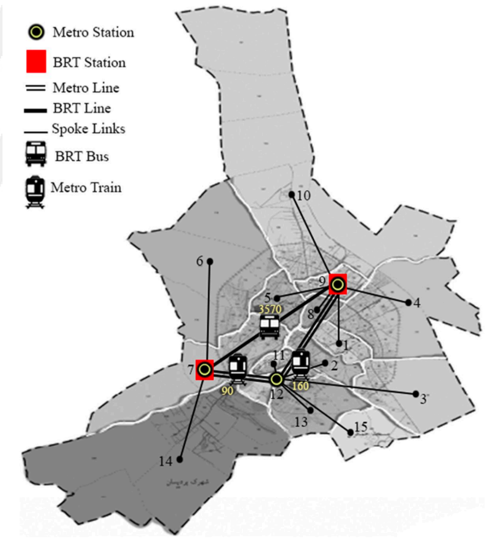

Figure 1.2 A public transportation system with different types of transportation modes and changing arc capacities changing according to the transportation mode used (*Adapted from Kaveh et al.* [30])

We propose a new MIP model for MACHLP built upon the problem setting adopted by Akgün and Tansel [6]. The proposed model is defined on a non-complete network but can also be used with complete networks. We solve the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic. We also develop a simulated annealing (SA)-based heuristic algorithm. We create test instances by defining capacities on different arcs, i.e., on only hub arcs, and on both hub and access arcs, and changing arc capacities. We conduct computational tests to assess the performance of the proposed heuristics using these test instances on networks with up to 400 nodes. The resulting optimality gaps are high for the solutions found by CPLEX and Gurobi with NoRel heuristic. However, the proposed SA heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for those instances.

In Chapter 2, we introduce the *multiple allocation tree of hubs location problem* (MATHLP) and present the related literature. Then, we compare hub networks that result from using different modeling approaches for MATHLP and show the advantages of the proposed approach. After presenting the details of the MIP model for MATHLP, we give the proposed BD-based heuristic approaches and the computational studies. An article mostly composed of Chapter 2 was published in the journal of Computers and Operations Research [31].

In Chapter 3, we introduce *multiple allocation arc capacitated hub location problem* (MACHLP) and give the related literature. We demonstrate on the necessity of using RealN as MN to be able to incorporate arc capacities into the HLPs. After defining a MIP model, we present a heuristic approach based on SA algorithm and give the computational studies.

Finally, in Chapter 4 we conclude the dissertation with some final remarks and future research directions.

Chapter 2

MULTIPLE ALLOCATION TREE OF HUBS LOCATION PROBLEM

In this chapter, we consider Multiple Allocation Tree of Hubs Location Problem (MATHLP) that imposes a tree topology requirement on the part of the network where the transfer between hubs is carried out, i.e., on the backbone or hub-level network, using multiple allocation strategy. The objective is to minimize the total transportation cost needed to transport the given flow between OD pairs by locating p hub nodes.

Most hub location problems are known to be NP-hard (e.g., Carello et al. [32]; Alumur and Kara [1]; Contreras and O'Kelly [4]). MATHLP is NP-hard as well. When the hub locations and the allocation of supply and demand points to hubs are fixed in MATHLP, the problem of finding a tree spanning the hub nodes is equivalent to the Optimum Communication Spanning Tree Problem, which is NP-hard (Johnson et al. [33]; Contreras et al. [9]).

In Section 2.1, we give the related literature for MATHLP. We propose a new MIP model for MATHLP that is built upon the problem setting adopted by Akgün and Tansel [6]. In Section 2.2, we show through examples that the proposed modeling approach may produce better solutions than the classical approach, which may result from the differences in the selected hubs, the flow routes between origin-destination points, and the assignment of non-hub nodes to hub nodes. In Section 2.3, we define the problem and present the details of the MIP model. The proposed model is defined on non-complete networks but can also be used with complete networks. We solve the model by the CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic. In Section 2.4, we develop BD-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We conduct computational tests to assess the performance of the proposed heuristics using instances defined on different networks with the number of nodes changing from 81 to

500. We present these computational studies in Section 2.5 and conclude the chapter in Section 2.6.

2.1. Literature Review

Tree of hubs location problems (THLP) require a tree topology on the part of the network where the transfer between hubs is carried out, i.e., on the backbone or hub-level network (HLN). Most studies in the literature impose a complete HLN topology but do not discuss whether this is a result of the nature of the application or the assumptions regarding the network or data structure. It occurs that there are three main assumptions in most HLP models (For a detailed discussion, see, e.g., Contreras and O'Kelly [4]; Akgün and Tansel [6]): (1) MN is a complete network with arc distances satisfying the triangle inequality, (2) transportation costs on all hub arcs are discounted by a constant factor independent of the actual amount of flow on the arcs, i.e., collection and distribution are more costly, and (3) all flows are routed via a set of hubs, i.e., no direct flows between non-hub nodes. A model with these three assumptions and a cost minimization objective and without any topological requirements produces a solution where the flows between an OD pair visit at most two hubs. In other words, a route between an OD pair in an HN consists of at most three arcs, namely, collection (access), transfer (hub), and distribution (access) arcs. Given non-zero flows between all OD pairs, the resulting HLN is a complete network, i.e., all hubs are fully interconnected by hub arcs.

Considering the complete HLN topology to be restrictive, several researchers have studied incomplete HLN topologies. Some of these studies (e.g., Alumur et al. [34]; Nickel et al. [35]; Mohri and Akbarzadeh [36]; Calık et al. [37]; Alumur and Kara [38]; Yoon and Current [39]; Alumur et al.[40]; Martins de Sa ́ et al. [41], [42]) require HLN only to be connected and do not impose any specific HLN topology. Some studies (e.g., Campbell et al. [43], [44]; Campbell [45]) do not even require HLN to be connected. The studies that impose a particular HLN topology other than a tree structure are Labb ́e and Yaman [46] and Yaman [47] that study a star HLN, Martins de Sa ́ et al. [48], [49] that address a line HLN, and Lee et al. [50] and Contreras et al. [51] that investigate a cycle HLN.

To our knowledge, all studies that require a tree HLN address the single allocation version of the problem. Contreras et al. [9] are the first to require a tree-like HLN topology. They propose a MIP formulation for THLP presenting several families of valid inequalities to strengthen the proposed formulation together with their exact separation procedures. The authors show the effectiveness of the proposed valid inequalities with a set of computational experiments. They are able to solve instances with up to 25 nodes optimally in reasonable computational time.

Contreras et al. [22] develop a new formulation for the single allocation THLP. Their aim is to develop a new formulation that yields tighter linear programming (LP) bounds than that in Contreras et al. [9]. They observe the quality of LP bounds produced by different formulations for classical single allocation hub location problems, and find out that the best LP bounds are usually obtained with four-index formulations proposed by Campbell [17] for the single allocation p-hub median problem. In this regard, Contreras et al. [22] propose a new formulation for the single allocation THLP based on the model proposed by Campbell [17] and an algorithm where the lower bounds (LB) are generated by the Lagrangean dual and the upper bounds (UB) are generated by a simple heuristic applied at each iteration of the subgradient optimization. They can find solutions with at most 10% deviation between lower and upper bounds for instances with at most 100 nodes.

Martins de Sa et al. [52] propose a Benders Decomposition (BD) approach to solve the model developed by Contreras et al. [22]. They develop a new cut selection scheme to improve the BD algorithm. The proposed BD Algorithm solves instances up to 100 nodes optimally. Sedehzadeh et al. [53] address a multi-objective and multi-modal problem with uncertain input data allowing hubs to have different capacity levels and to be connected with different transportation modes. They obtain Pareto-optimal solutions for instances with up to 100 nodes by using two different algorithms. Pessoa et al. [54] design a genetic algorithm for THLP. They test their algorithm using the instances with 25 nodes generated by Contreras et al. [9]. They can find better feasible solutions for instances not solved to optimality and optimal or near-optimal solutions for instances solved to optimality by Contreras et al. [9]. Blanco and Marin [55] offer two MIP models for upgrading nodes, which implies a decrease in the cost of traversing arcs connecting those upgraded nodes. They offer two MIP models, one based on the ideas presented in Contreras et al. [9] and one based on the disaggregation of the variables, and compare their computational performance using instances with up to 25 nodes. The proposed models in this study are hub location models with hub and/or arc fixed costs rather than p-hub median models.

We remark that the models that require a connected HLN may produce hub networks with a tree HLN depending on the data. For example, Martins de Sa et al. [42] obtain optimal solutions having a tree HLN for problem instances with sufficiently high fixed arc setup costs for a multiple allocation incomplete hub location problem with service time requirements.

Aforementioned formulations proposed for THLP assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. We call this approach as *the classical approach*. In section 2.1.1, we give the single allocation THLP models in the literature based on the classical approach. There is not a model for the multiple allocation version of THLP in the literature, *to represent the classical approach* for MATHLP, we extend the model of Ernst and Krishnamoorthy [56] developed for multiple allocation p-hub median problem by adding necessary constraints to achieve a tree HLN and present this formulation in Section 2.1.2.

2.1.1. Single Allocation Tree of Hubs Location Models based on the Classical Approach

Contreras et al. [9] are the first to require a tree-like HLN topology in HLPs. They propose a MIP formulation for THLP. They consider a complete network *G = (N, A)* whose set of nodes, $N = \{1, . . . , n\}$, represents the set of origins and destinations of a certain product that is routed through G via some hub nodes. w_{ij} denote the demand of product from i to j for each pair of nodes $i,j \in N$. The total amount of flow originating from node i is $O_i = \sum_j w_{ij}$ and the total amount of flow sent to node j is $D_j = \sum_i w_{ij}$. They denote the transportation cost of a unit of flow between i and j with c_{ij} and α represent the discount factor for hub-to-hub journeys. Any node of N can be chosen to become a hub, and there is a fixed number p that must be chosen to be hubs. For each pair $i,j \in N$, if i and j are non-hub nodes, the flow w_{ij} must go from i to j through one or more hubs. When i and j are hubs, the flow w_{ij} can go directly from i to j and it is also possible that the flow w_{ij} uses one or more intermediate hubs. Contreras et al. [9] require single allocation where every non-hub node i must be allocated to one single hub, so that all the incoming and outgoing traffic of that non-hub node is routed through this hub. Contreras et al. [9] aim to (1) locate p hubs, (2) link them so as to define a tree,

(3) allocate every non-hub node to one single hub node in such a way that the overall transportation cost is minimized.

In order to represent the routes between origins and destinations, they use the variables with three indices, x_{ikm} proposed by Ernst and Krishnamoorthy [57] for the single allocation *p*-hub median problem to represent the amount of flow with origin *i* traversing arc *(k,m)*. They also define binary variables z_{ik} taking on the value of 1 if node *i* is allocated to hub *k*, 0 otherwise and y_{km} taking on the value of 1 if arc *(k,m)* links two hubs, 0 otherwise. They suppose that there is a network with a node set N with n nodes. Each unit of product that traverses *(r,s)* incurs a cost $c_{rs} \geq 0$ whereas when both *r* and *s* are hubs a discount factor $0 \leq \alpha \leq 1$ is applied, and the per unit cost associated with arc *(r,s)* is αc_{rs}.

With these definitions, the model proposed by Contreras et al. [9] is given below:

$$Min \sum_{i \in N} \sum_{k \in N} (c_{ik} O_i + c_{ki} D_i) z_{ik} + \sum_{i \in N} \sum_{k \in N} \sum_{\substack{m \in N \\ m \neq k}} \alpha c_{km} x_{ikm} \qquad \textbf{(2.1)}$$

s.t.

$$\sum_{k \in N} z_{ik} = 1 \qquad i \in N \qquad \textbf{(2.2)}$$

$$\sum_{k \in N} z_{kk} = p \qquad \textbf{(2.3)}$$

$$z_{km} + y_{km} \leq z_{mm} \qquad k, m \in N \,; m > k \qquad \textbf{(2.4)}$$

$$z_{mk} + y_{km} \leq z_{kk} \qquad k, m \in N \,; m > k \qquad \textbf{(2.5)}$$

$$x_{ikm} + x_{imk} \leq O_i y_{km} \qquad i, k, m \in N \,; m > k \qquad \textbf{(2.6)}$$

$$O_i Z_{ik} + \sum_{\substack{m \in N \\ m \neq k}} x_{imk} \leq \sum_{\substack{m \in N \\ m \neq k}} x_{ikm} + \sum_{m \in N} w_{im} z_{mk} \qquad i, k \in N \,; i \neq k \qquad \textbf{(2.7)}$$

$$\sum_{k \in N} \sum_{m \in N} y_{km} = p - 1 \qquad \textbf{(2.8)}$$

$$x_{ikm} \geq 0 \qquad i, k, m \in N \qquad \textbf{(2.9)}$$

$$z_{km}, y_{km} \in \{0,1\} \qquad k, m \in N \qquad \textbf{(2.10)}$$

Objective function (2.1) minimizes the total transportation cost. Constraints (2.2) assign each non-hub node to a hub node and constraint (2.3) locates p hubs. Constraints (2.4) and (2.5) guarantee that non-hub nodes are allocated to open hubs and hub links are possible between open hubs. Constraints (2.6) ensure that the flow between hubs will only move through the tree structure. Constraints (2.7) are the flow balance constraints. Finally, constraints (2.8) define a tree structure within the hubs by choosing $p-1$ edges that are connected due to constraints (2.6) and (2.7). Constraints (2.9) and (2.10) define decision variables.

Contreras et al. [22] develop a new formulation for the single allocation THLP. Their problem is just same as the problem defined by Contreras et al. [9]. Their aim is to develop a new formulation for this problem that yields tighter LP bounds than that in Contreras et al. [9]. For that reason, they use the variables x_{ijkm} proposed by Campbell [17] for the single allocation p-hub median problem to represent the routes between origins and destinations. They define the binary variables x_{ijkm} taking on the value of 1 if the flow from i to j traverses arc *(k,m)* connecting hubs k and m, 0 otherwise.

The model proposed by Contreras et al. [22] with the same definitions used in the formulation of Contreras et al. [9] is given below:

$$Min \sum_{i\in N}\sum_{k\in N}(c_{ik}\,O_i + c_{ki}D_i)z_{ik} + \sum_{i\in N}\sum_{J\in N}\sum_{k}\sum_{\substack{m\in N\\ m\neq k}} \alpha\, w_{ij}c_{km}x_{ijkm} \qquad \textbf{(2.11)}$$

s.t.

$$\sum_{k\in N} z_{ik} = 1 \qquad i\in N \qquad \textbf{(2.12)}$$

$$\sum_{k\in N} z_{kk} = p \qquad \textbf{(2.13)}$$

$$\sum_{\substack{m\in N\\ m\neq k}} x_{ijkm} + z_{jk} - \sum_{\substack{m\in N\\ m\neq k}} x_{ijmk} - z_{ik} = 0 \qquad i,j,k\in N\,;\, i\neq j\ k\neq j \qquad \textbf{(2.14)}$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \qquad i,j,k\in N\,;\, m>k \qquad \textbf{(2.15)}$$

$$\sum_{k\in N}\sum_{\substack{m\in N\\ m>k}} y_{km} = p-1 \qquad i,k,m\in N\,;\, m>k \qquad \textbf{(2.16)}$$

$$x_{ijkm} \geq 0 \qquad i,j,k,m\in N\ ;\ k\neq m \qquad \textbf{(2.17)}$$

$z_{ik} \in \{0,1\}$ $\quad i, k \in N$ **(2.18)**

$y_{km} \in \{0,1\}$ $\quad k, m \in N; m > k$ **(2.19)**

Objective function (2.11) minimizes the total transportation cost. Constraints (2.12) assign each non-hub node to a hub node and constraint (2.13) locates p hubs. Constraints (2.14) are the flow balance constraints. Constraints (2.15) guarantee that the flow between hubs will only move through the tree structure. Finally, constraints (2.16) define a tree structure within the hubs by choosing $p-1$ edges that are connected due to constraints (2.15). Constraints (2.17), (2.18) and (2.19) define decision variables.

2.1.2. Multiple Allocation Tree of Hubs Location Models based on the Classical Approach

We extend the model of Ernst and Krishnamoorthy [56] developed for multiple allocation p-hub median problem by adding necessary constraints to achieve a tree HLN *to represent the classical approach* for MATHLP since there is no model for MATHLP in the literature.

Suppose that there is a complete network $G = (N, A)$ with the set of nodes $N = \{1, \ldots, n\}$. Node i generates a positive annual flow w_{ij} for at least one node $j \in N - \{i\}$. The total amount of flow originating from node i is $O_i = \sum_j w_{ij}$ and the total amount of flow sent to node j is $D_j = \sum_i w_{ij}$ where w_{ij} is the demand of product from i to j for each pair of nodes $i,j \in N$. Let c_{ij} denote the transportation cost of a unit of flow between i and j and α represent the discount factor for hub-to-hub journeys. Any node of N can be chosen to become a hub, and there is a fixed number p that must be chosen to be hubs. We require multiple allocation where every non-hub node i can be allocated to more than one hub. We aim to (1) locate p hubs, (2) link them so as to define a tree, (3) allocate every non-hub node to at least one hub node in such a way that the overall transportation cost is minimized.

Ernst and Krishnamoorthy [56] define X_{ilj} as the flow originating from $i \in N$ flowing from hub l to node j, Z_{ik} as the flow from node i to hub k, Y_{ikl} as the total amount of flow of commodity i that is routed between hubs k and l. H_k takes the value 1 if node k is a hub, 0 otherwise. In addition, we define yy_{km} that takes the value 1 when arc (k, m) links two hubs and 0 otherwise in order to define a tree-like hub level network. To create the tree structure on the hub network, we use the approach of

Contreras et al. [9] and Contreras et al. [22]. According to this approach, limiting the number of arcs connecting hub nodes to $p-1$ creates a tree structure on the hub network since the underlying network is a complete network.

The proposed model for MATHLP based on classical approach is given below:

Model CAM: The Model for the Classical Approach of MATHLP

$$\text{Min} \sum_i \left[\sum_k c_{ik} Z_{ik} + \sum_k \sum_l c_{kl} Y_{ikl} + \sum_l \sum_j \alpha\, c_{lj} X_{ilj} \right] \quad \textbf{(2.20)}$$

s.t.

$$\sum_k H_k = p \quad \textbf{(2.21)}$$

$$\sum_k Z_{ik} = O_i \qquad i \in N \quad \textbf{(2.22)}$$

$$\sum_l X_{ilj} = W_{ij} \qquad i,j \in N \quad \textbf{(2.23)}$$

$$\sum_l Y_{ikl} + \sum_j X_{ikj} - \sum_l Y_{ilk} - Z_{ik} = 0 \qquad i,k \in N \quad \textbf{(2.24)}$$

$$Z_{ik} \leq O_i H_k \qquad i,k \in N \quad \textbf{(2.25)}$$

$$\sum_i X_{ilj} \leq D_j H_l \qquad l,j \in N \quad \textbf{(2.26)}$$

$$yy_{kl} \leq H_k \qquad k,l \in N \quad \textbf{(2.27)}$$

$$yy_{kl} \leq H_l \qquad k,l \in N \quad \textbf{(2.28)}$$

$$\sum_k \sum_l yy_{kl} = p-1 \quad \textbf{(2.29)}$$

$$Y_{ilk} + Y_{ikl} \leq O_i yy_{kl} \qquad i,k,l \in N \quad \textbf{(2.30)}$$

$$H_k \leq \sum_l yy_{kl} + \sum_l yy_{lk} \qquad k \in N \quad \textbf{(2.31)}$$

$$H_k \in \{0,1\} \qquad k \in N \quad \textbf{(2.32)}$$

$$Y_{ikl}, X_{ilj}, Z_{ik} \geq 0 \qquad i,j,k,l \in N \quad \textbf{(2.33)}$$

$$yy_{kl} \in \{0,1\} \qquad k,l \in N \quad \textbf{(2.34)}$$

Objective function (2.20) together with constraints (2.21) − (2.26) and (2.32) − (2.33) constitute the formulation of Ernst and Krishnamoorthy [56]. Objective function (2.20) minimizes the total transportation cost. Constraint (2.21) locates p hubs. Constraints (2.22) and (2.23) satisfy the supply and demand requirements, respectively. Constraints (24) are the flow balance constraints. Constraints (2.25) and (2.26) ensure that flow incoming to and going from a hub node is possible only if that node is chosen as a hub. Constraints (2.32) and (2.33) define the decision variables.

Constraints (2.27) (2.31) and (2.34) ensure that the hubs are connected through a tree structure. Constraints (2.27) and (2.28) require that an arc (k, m) be chosen to form the tree structure between hubs if and only if both k and m are chosen as hubs. Constraint (2.29) limits the number of arcs connecting hub nodes to $p - 1$. Constraints (2.30) allow flows between hubs only in arcs selected as a part of the tree structure. Constraints (2.31) guarantee that a selected hub must be connected by an arc which is a part of the tree structure. Constraints (2.34) define the new binary variables.

In the next section, we will use *Model CAM* to represent the classical approach to be able to compare it with our proposed approach.

2.2 Comparison of the Hub Networks for Different Modeling Approaches

In this section, we investigate how the hub network topology and the total cost change depending on the modeling approach used under different assumptions. We compare two modeling approaches: (1) *The classical approach*: Modeled network MN is complete and its distances satisfy the triangle inequality. (2) *The proposed approach*: MN is the same as the real-world network RealN that may be complete or non-complete. For comparison purposes, we use two different types of networks: a *7-node complete network* whose distances do not satisfy the triangle inequality and a *30-node non-complete network*. The complete network is the network used by Marin et al. [8] to show that some hub location models do not work correctly when the triangle inequality is not satisfied. The distance matrix of this complete network used is given below;

$$(d_{ij}) = \begin{pmatrix} 0 & 1 & 100 & 100 & 100 & 100 & 100 \\ 1 & 0 & 1 & 100 & 100 & 100 & 100 \\ 100 & 1 & 0 & 1 & 100 & 100 & 100 \\ 100 & 100 & 1 & 0 & 1 & 100 & 100 \\ 100 & 100 & 100 & 1 & 0 & 1 & 100 \\ 100 & 100 & 100 & 100 & 1 & 0 & 1 \\ 100 & 100 & 100 & 100 & 100 & 1 & 0 \end{pmatrix}.$$

The non-complete network given in Figure 2.1 is the network that consists of 30 cities in Turkey as the nodes and the roads between neighboring cities as the arcs. The distances are direct distances between the neighboring cities. We assume that a discount factor of 0.7 is applied to the hub arc costs.

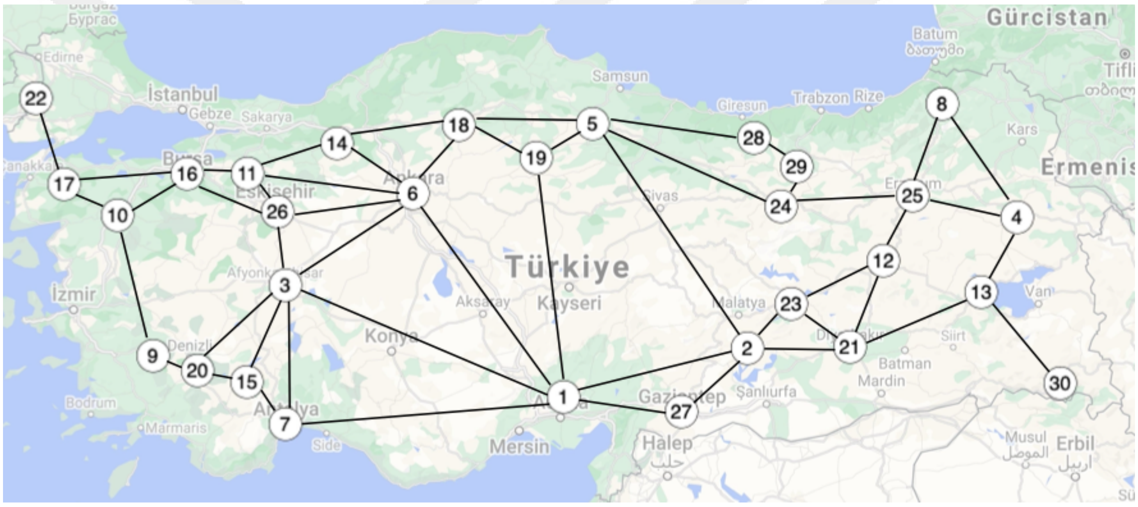

Figure 2.1. Non-complete transportation network consisting of 30 cities of Turkey.

To represent the classical approach, we use the Model CAM, the extention of the model of Ernst and Krishnamoorthy [56] developed for multiple allocation p-hub median problem with additional constraints to achieve a tree HLN. We remark that the classical model needs as MN a complete network whose distances satisfy the triangle inequality to work correctly. In this regard, we apply the Floyd's Algorithm [7] to the non-complete network to find all-pairs shortest path distances and construct a complete network in order to obtain a solution for the non-complete network using the classical model. *To represent the proposed approach*, we use our proposed model for MATHLP, whose details are given in Section 2.3. The proposed model can use any type of network, i.e., complete or non-complete, as MN. In this regard, the proposed model uses directly the non-complete network and complete network as MN.

Our first goal is to show that the classical model does not work correctly when a complete network whose distances do not satisfy the triangle inequality is used as MN. We solve both the proposed model and the classical model to optimality using the complete network given in Figure 2.2 (a) and setting p=3. Figure 2.2 (b) and Figure 2.2 (c) indicate hub networks for the proposed model and the classical model on the original network, respectively. Filled circles and empty circles represent hub nodes and non-hub nodes, respectively. Solid lines and dashed lines indicate hub arcs and access arcs, respectively. Figure 2.2 (b) and 2.2 (c) show that both models produce an HLN with a tree structure. However, the selected hubs, the tree structures, and the resulting objective function values are different. In Figure 2.2 (b), HLN consists of nodes 2 through 6 with nodes 2, 4, and 6 being the hub nodes and nodes 3 and 5 being the non-hub nodes. The tree in Figure 2.2 (b) consists of non-hub nodes 3 and 5 as intermediate nodes between hub nodes, which may be possible when the triangle inequality is not satisfied (e.g., Marin et al. [8]). We can think of nodes 3 and 5 as transshipment points in HLN. These nodes receive service from their adjacent hub nodes. Specifically, non-hub node 3 is assigned to hub nodes 2 and 4 while non-hub node 5 is assigned to hub nodes 4 and 6. Accordingly, dashed lines between hub nodes represent collection and distribution flows between non-hub nodes 3 and 5 and their adjacent hub nodes. Non-hub nodes 1 and 7 are assigned to a single hub. In Figure 2.2 (c), HLN consists of the nodes 2, 3, and 6 as the hub nodes. All non-hub nodes receive service from a single hub. The optimal objective function values for the proposed model and the classical model are 88 and 1266, respectively. This difference results from the fact that the proposed model is allowed to find and use routes with less cost than that of direct arcs between nodes to send flows while the classical model uses only direct arcs. For example, to send flow from node 2 to node 6, the classical model uses the direct arc (2,6) with a cost of 100 while the proposed model uses the path consisting of the arcs (2,3), (3,4), (4,5), and (5,6) with a cost of 4.

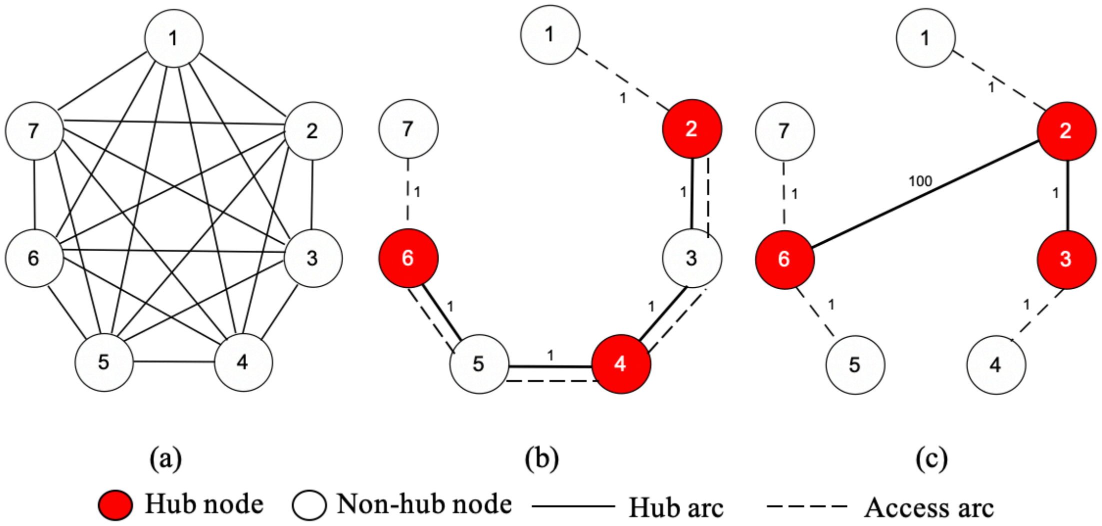

Figure 2.2. Tree structures obtained using a complete network whose distances do not satisfy the triangle inequality. (a), (b), and (c) represent the original complete network, the resulting hub network using the proposed model, and the resulting the hub network using the classical model.

The case in Figure 2.2 occurs because the cost matrix of the modeled network MN does not satisfy the triangle inequality. Such cost matrices may occur especially in cases where the cost is independent of the distance or when the costs or weights represent something else, e.g., travel time, bus fares, telecommunication costs. To give a real-world example, consider the map given in Figure 2.3 that represents a part of Marmara Region in Turkey. Suppose that a driver located at node 1 (Yalova) would like to go to Node 3 (Istanbul). The driver has two options: (a) going from 1 to 3 directly using the bridge or (b) going from 1 to 3 through node 2 (Izmit). Clearly, the direct distance following (a) is shorter than the indirect distance following (b). If the shortest path distance is used, (a) should be used. However, the cost of using (a) is almost twice the cost of using (b) because using the bridge is too costly. In this case, the lowest cost is not the direct path and hence the triangle inequality does not hold.

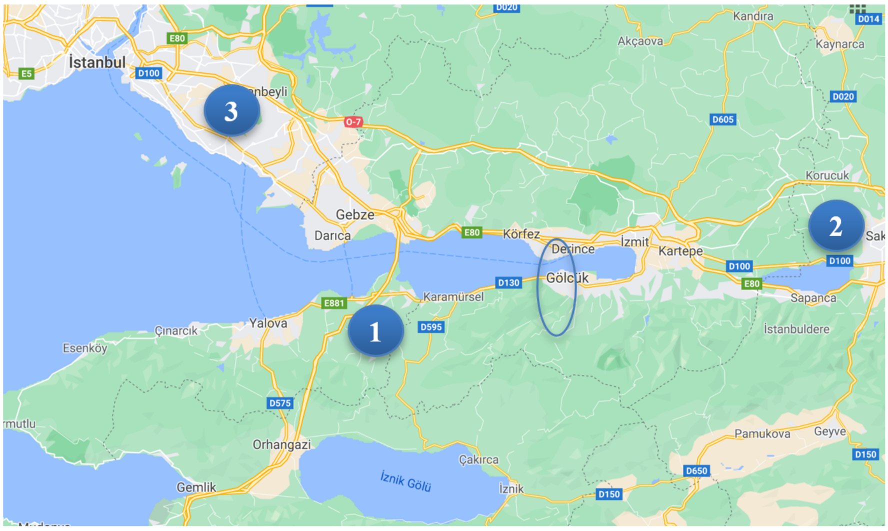

Figure 2.3 A real-world example showing that the triangle inequality does not hold.

Our second goal is to show that the proposed model may find a better solution than the classical model even when the triangle inequality is satisfied. We will consider *two cases*: (1) The proposed model and the classical model find different flow routes even though their optimal hub set and the tree structure are the same. (2) The proposed model and the classical model find different optimal hub sets and hence different hub networks.

For both cases, we solve the models to optimality using appropriate versions of the non-complete network in Figure 2.1 (non-complete version for the proposed model and complete version for the classical model). For *Case 1*, we allow all 30 nodes to be selected as hub nodes and set p=5. Figure 2.4 and Figure 2.5 indicate the resulting hub networks for the proposed model and the classical model, respectively. Nodes 3, 16, 19, 21, and 25 are selected as the hub nodes by both models. Hub network in Figure 2.4 directly gives the routing information on the real-world network RealN. For example, the route between hubs 3 and 21 is (3,1) - (1,2) - (1,21) with nodes 1 and 2 being transshipment points in HLN.

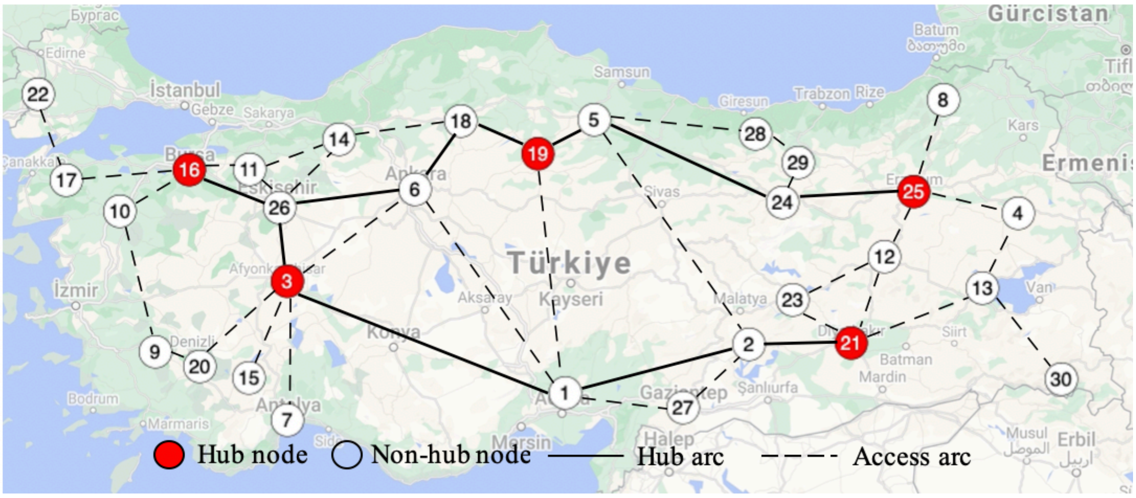

Figure 2.4. Optimal hub network obtained with the proposed model (*H*=30, *p*=3).

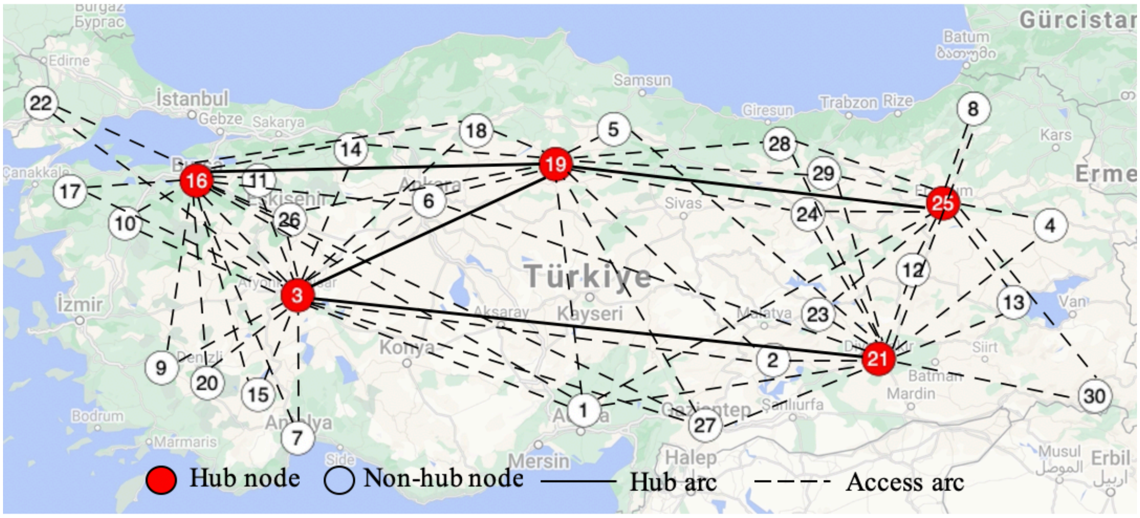

Figure 2.5. Optimal hub network obtained with the classical model (*H*=30, *p*=3).

In Figure 2.5, however, hub nodes 3 and 21 are connected by a single arc as expected and postprocessing is required to determine that the shortest-path arc (3,21) corresponds to the route consisting of the arcs (3,1)-(1,2)-(1,21) in RealN. Nevertheless, this should not be interpreted as that the routes between nodes are always the same in both models. Consider the routes between hub nodes 3, 16, and 19. In Figure 2.5, these nodes are connected to each other by direct arcs (16,19) and (19,3). Any flow sent from node 16 to node 3 follows the route consisting of the arcs (16,19) and (19,3). The direct arcs (16,19) and (19,3) in Figure 2.5 correspond to the routes (16,26)-(26,6)-(6,18)-(18,19) and (19,18)-(18,6)-(6,26)-(26,3) in RealN as shown in Figure 2.4, respectively. This means that flow sent from 16 to 3 covers the route (26,6)-(6,18)-(18,19) twice, one going from 16 to 19 and one going from 19 to 3. In Figure 2.4, however, flow from node 16 to node 3 follows the route (16, 26)-(26,3). That is, even though the tree structures of both models seem to be the same in RealN (after postprocessing the solution of the classical model), the resulting flows are different. This also affects the assignments of origin and destination nodes to the hubs. As a result, the optimal objective function values for the proposed model and the classical model are 14,279,772 and 14,297,370, respectively.

For *Case 2*, we allow 20 nodes indexed between 1-20 to be selected as hub nodes and set p=8. Figure 2.6 and Figure 2.7 indicate the resulting hub networks for the proposed model and the classical model, respectively. The resulting hub networks are different because different optimal hub sets (2-6, 8, 12, and 16 for the proposed model and 1, 2, 4, 5, 6, 12, 16, and 20 for the classical model) are selected. The optimal objective function values for the proposed model and the classical model are 13,762,410 and 13,795,031, respectively.

We remark that the results in Case 1 and Case 2 are valid when we address MATHLP. When we relax the tree-HLN requirement, both approaches find solutions that are the same in RealN as long as arc distances (costs) satisfy the triangle inequality. This is also true for the single allocation version.

To sum up, using the proposed approach in addressing MATHLP may allow to obtain better solutions that may result from different hub network topologies, assignments, flows, and costs. The approach may also provide more flexibility in modeling several real-life issues directly, e.g., arc capacities or disruptions.

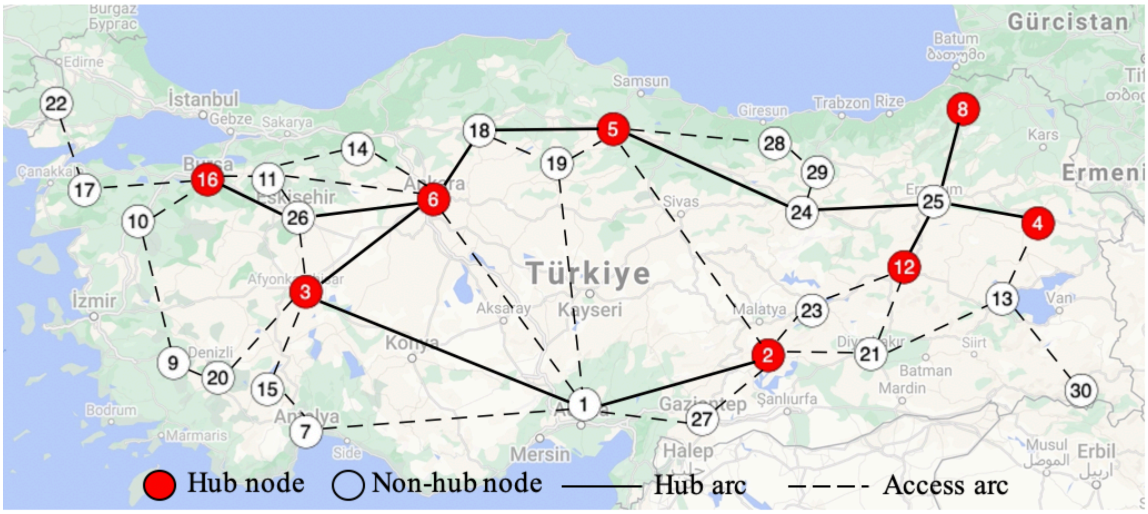

Figure 2.6. Optimal hub network obtained with the proposed model (*H*=20, *p*=8)

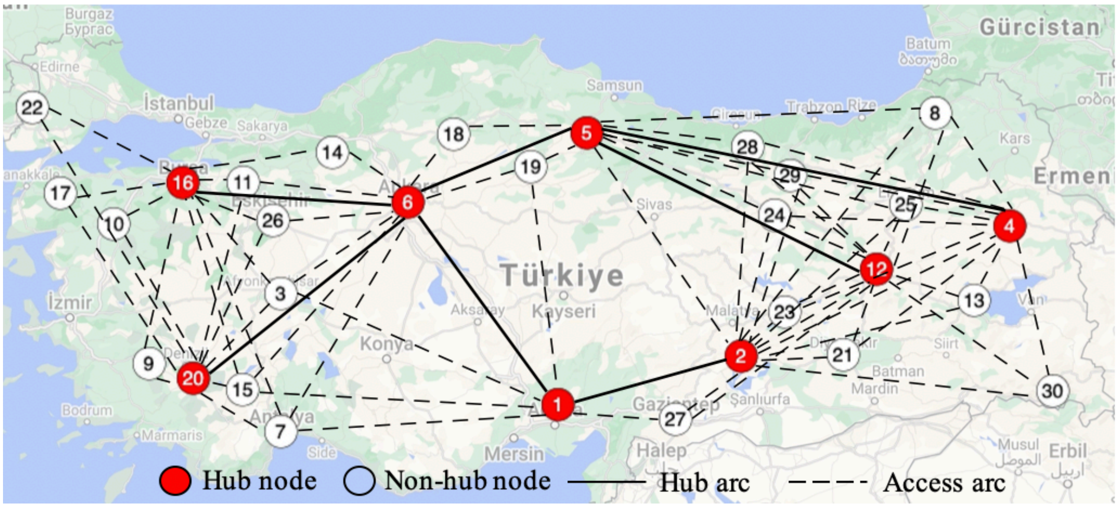

Figure 2.7. Optimal hub network obtained with the classical model (*H*=20, *p*=8)

2.3 Problem Definition and Mathematical Formulation

We define *Multiple Allocation Tree of Hubs Location Problem* (MATHLP) based on the modeling framework given by Akgün and Tansel (2018). Let $G = (N, E)$ be an undirected and connected network representing RealN with $N = \{1, \dots, n\}$ and E being the node set and edge set, respectively. N consists of supply/origin nodes S, demand/destination nodes D, and transshipment nodes T. The nodes in S generate a positive flow w_{ij} for at least one node $j \in D$. The same node can be the element of both S and D.

Table 2.1 Sets, indices, parameters, and decision variables of the proposed model.

	Sets, Indices, and Parameters
G=(N,E)	Undirected real-world network with node set N and edge set *E*, *N*=*S* ∪ *D* ∪ *T* where *S*, *D*, and *T* are the set of supply, demand, and transshipment nodes
$G^* = (N^*, E^*)$	Subnetwork of G that can be used for inter-hub transportation, E^* is the set of edges that can be used as hub arcs, N^* is the set of nodes that are incident to E^*
$G' = (N, A)$	Directed version of *G*=(*N*,*E*) obtained by replacing each edge $\{i, j\} \in E$ with a pair of directed arcs (i, j) and (j, i)
H	Set of nodes that can be hubs with $H \subseteq N^*$
$G_0 = (N_0, A_0)$	Three-layer modeled network, $G_0 = G_1 \cup G_2 \cup G_3$ with G_1, G_2, and G_3 representing the supply (first), hub (second), and distribution (third)

	layers of the network
$G_1 = (N_1, A_1)$	Supply layer network with $G_1 = G'$
$G_2 = (N_2, A_2)$	Hub layer network constructed from subnetwork of G' that corresponds to $G^* = (N^*, E^*)$
$G_3 = (N_3, A_3)$	Distribution layer network with $G_2 = G'$
$i,j \in N$	Nodes in the network G=(N,E)
$1i$	Nodes in the supply layer G_1
$2i$	Nodes in the hub layer G_2
$3i$	Nodes in the distribution layer G_3
$(1i,1j)$	Arcs in the supply layer
$(2i,2j)$	Arcs in the hub layer
$(3i,3j)$	Arcs in the distribution layer
A_{12}	Arcs connecting G_1 and G_2 with $A_{12} = \{(1i, 2i): i \in H\}$
A_{23}	Arcs connecting G_2 and G_3 with $A_{23} = \{(2i, 3i): i \in H\}$
$i,j \in N$	Nodes in the network G=(N,E)
k	Commodity type indicating the origin node of the flow, $k = i \in S$
l_{ij}	Length of arc (i, j)
χ_{ij}, α_{ij}, δ_{ij}	Cost of moving one unit of flow per unit length along arc (i,j) for supply, hub, and distribution layers, respectively
c_{ij}	Cost of arc (i, j) with $c_{ij} = l_{ij}\chi_{ij}$, $c_{ij} = l_{ij}\alpha_{ij}$, and $c_{ij} = l_{ij}\delta_{ij}$ for supply, hub, and distribution layers, respectively
w_{ij}	Flow to be sent from $i \in S$ to $j \in D$
W_i	Total supply of commodity i, $W_i = \sum_{j \in D} w_{ij}$
$b_{\beta k}$	Amount of supply/demand of commodity k at node $\beta \epsilon N_0$
F_β^{out} (F_β^{in})	Forward (inward) star of a node $\beta \in (N_1 \cup N_2 \cup N_3)$.
Decision Variables	
x_{ijk}	Amount of flow of commodity k in arc (i, j)
y_{2i}	1, if a hub is located at node $i \in H$ and 0, otherwise
s_{ij}	1, if arc (i, j) belongs to the tree structure in G_2 and 0, otherwise
t_{ij}	Flow variable used to construct the tree structure in G_2 and represents the amount of fictitious flow in arc (i, j)

Let $G^* = (N^*, E^*)$ *represent* the subnetwork of G that can be used for inter-hub transportation. E^* is the set of edges that can be used as *hub arcs* for some reason, e.g., they have high capacities or more appropriate for construction. N^* is the set of nodes that are incident to E^*. We define $H \subseteq N^*$ as the set of nodes that can be hubs.

Let l_{ij} denote the *length* of edge $\{i,j\}$ with $l_{ij} = l_{ji}$. The χ_{ij}, $\alpha_{ij,}$ and δ_{ij} are the cost of moving one unit of flow per unit length along the edge $\{i,j\}$ for *collection*, *transfer*, and *distribution*, respectively, with $\alpha_{ij} \leq \chi_{ij}$ and $\alpha_{ij} \leq \delta_{ij}$ to achieve economies of scale.

MATHLP aims to (1) select p *nodes* from hub set H, (2) determine the service routes between OD pairs that visit at least one hub node, (3) connect all hubs through a tree structure and require all flows to use this tree structure by using a multiple allocation strategy such that total transportation cost is minimized.

We formulate MATHLP using a three-layer network $G_0 = (N_0, A_0)$ as the modeled network MN where the first, second, and third layers represent the *collection/supply*, *transfer/hub*, and *distribution/demand* layers, respectively. To construct G_0, we use the directed version of $G = (N, E)$, $G' = (N, A)$, which is obtained by replacing each edge $\{i,j\} \in E$ with a pair of directed arcs (i,j) and (j,i) such that $l_{ij} = l_{ji}$.

The supply *layer* network $G_1 = (N_1, A_1)$ and the distribution layer network $G_3 = (N_3, A_3)$ are copies of $G' = (N, A)$ while the hub layer network $G_2 = (N_2, A_2)$ is the subnetwork of G' that corresponds to $G^* = (N^*, E^*)$ with $N_m = \{m1, m2, \dots, mn\}$ and $A_m = \{(mi, mj) : (i,j) \in A\}$ where $m = 1,2,3$. To exemplify, node 3 in RealN G is represented as 13, 23, and 33 in G_1, G_2, and G_3, respectively. G_1 and G_2 are connected by arcs of the form $A_{12} = \{(1\mathrm{i}, 2\mathrm{i}) : \mathrm{i} \in H\}$ while G_2 and G_3 are connected by arcs of the form $A_{23} = \{(2\mathrm{i}, 3\mathrm{i}) : \mathrm{i} \in H\}$. Thus, $N_0 = \bigcup_{m=1}^{3} N_m$ and $A_0 = \bigcup_{m=1}^{3} A_m \cup A_{12} \cup A_{23}$. Figure 2.8 shows a three-layer MN constructed using the structure of RealN G where $E^* = \{\{2,3\},\{2,4\},\{3,4\},\{4,5\}\}$, $N^* = \{2,3,4,5\}$, $H = \{3,4,5\}$, and $S = D = \{1,2,3,4,5\}$.

We *formulate* MATHLP as a multicommodity flow problem with side constraints in G_0. The flows w_{ij} with $i \in S$ and $j \in D$ are sent from $1i \in N_1$ to $3j \in N_3$ through G_0. We associate with each node $i \in S$ a different commodity. $W_i = \sum_{j \in D} w_{ij}$ is the total supply of commodity i at node $1i$. We use the parameter $b_{\beta k}$ to represent the amount of supply/demand of commodity k at node $\beta \epsilon N_0$. $b_{\beta\beta} = \sum_{j \epsilon D} w_{\beta j}$ for $\beta = 1i$ and $i \in S$, $b_{\beta k} = -w_{k\beta} = -w_{kj}$ for $\beta = 3j$ and $j \epsilon D$, and $b_{\beta k} = 0$ for all other nodes and $k \epsilon S$. F_{β}^{out} (F_{β}^{in}) is the forward (inward) star of a node $\beta \in (\mathrm{N}_1 \cup \mathrm{N}_2 \cup \mathrm{N}_3)$.

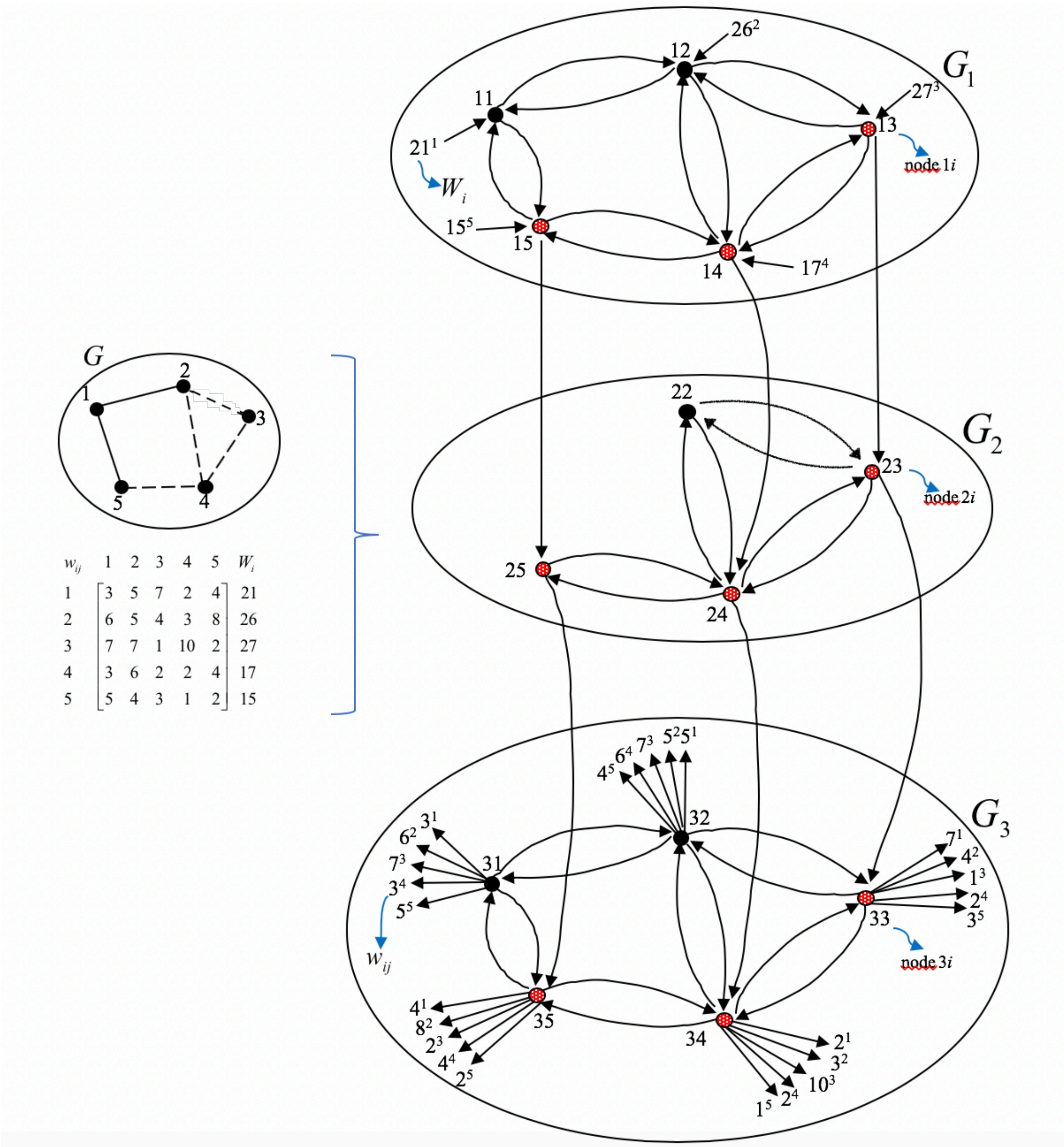

Figure 2.8. Three-layer MN G_0 constructed from RealN G (Adapted from Akgün and Tansel, [6]).

We also *associate* with each arc (i, j) a unit cost c_{ij}, where l_{ij} is the length of arc (i, j), as follows:

$$c_{ij} = \begin{cases} \chi_{ij} \times l_{ij} & for\ (1i, 1j), (i, j) \in A \\ \alpha_{ij} \times l_{ij} & for\ (2i, 2j), (i, j) \in A^* \\ \delta_{ij} \times l_{ij} & for\ (3i, 3j), (i, j) \in A \\ 0 & for\ (1i, 2i)\ \ or\ \ (2i, 3i), i \in H \end{cases}$$

We define the following decision variables: (1) x_{ijk} is the amount of flow of commodity $k \in S$ in arc (i, j), (2) y_{2i} is a binary variable that takes on the value of 1 when a hub is located at node $i \in H$ and 0 otherwise, (3) s_{ij} is a binary variable that takes on the value of 1 when arc (i, j) belongs to the tree structure in G_2 and 0 otherwise, and (4) t_{ij} is a flow variable used to construct the tree structure in G_2 and represents the amount of fictitious flow in arc (i, j).

We construct the tree structure in G_2 by using a rooted spanning tree formulation based on single-commodity flows t_{ij}. We define a node $\theta \in N_2$ as the root/supply node from which one unit of fictitious flow is sent to each other node $i \in (N_2 - \theta)$, i.e., a total of $|N_2 - 1|$ units of flow is sent from θ. The arcs with positive flows t_{ij} are selected as the arcs of the tree by the variables s_{ij}.

We present the sets, indices, parameters, and decision variables used in the formulation of the problem in Table 2.1. With these definitions, the proposed model, Model MATHLP, is given below:

Model MATHLP: Multiple Allocation Tree of Hubs Location Model

$$Z^* = \text{Min} \sum_{k \epsilon S} \sum_{(i,j) \epsilon A_0} c_{ij} x_{ijk} \quad \textbf{(2.35)}$$

s.t.

$$\sum_{j \in F_\beta^{out}} x_{\beta jk} - \sum_{j \in F_\beta^{in}} x_{j\beta k} = b_{\beta k} \qquad \beta \in (N_1 \cup N_2 \cup N_3), k \in S \quad \textbf{(2.36)}$$

$$\sum_{i \in H} y_{2i} = p \quad \textbf{(2.37)}$$

$$x_{(1i,2i)k} \leq W_k y_{2i} \qquad i \in H, k \in S \quad \textbf{(2.38)}$$

$$x_{(2i,3i)k} \leq W_k y_{2i} \qquad i \in H, k \in S \quad \textbf{(2.39)}$$

$$\sum_{j \in F_\theta^{out}} t_{\theta j} = |N_2 - 1| \quad \textbf{(2.40)}$$

$$\sum_{j \in F_\beta^{out}} t_{\beta j} - \sum_{j \in F_\beta^{in}} t_{j\beta} = -1 \qquad \beta \in (N_2 - \theta) \quad \textbf{(2.41)}$$

$$\beta \in (N_2 - \theta) \quad \textbf{(2.42)}$$

$$\sum_{j \in F_{\beta}^{in}} s_{j\beta} = 1$$

$$\sum_{j \in (F_{\theta}^{in} \cap N_2)} s_{j\theta} = 0 \qquad \textbf{(2.43)}$$

$$t_{ij} \leq |N_2 - 1| \, s_{ij} \qquad (i,j) \in A_2 \qquad \textbf{(2.44)}$$

$$x_{ijk} \leq W_k(s_{ij} + s_{ji}) \qquad (i,j) \in A_2, k \in S \qquad \textbf{(2.45)}$$

$$t_{ij} \geq 0\,, \; s_{ij} \in \{0,1\} \qquad (i,j) \in A_2 \qquad \textbf{(2.46)}$$

$$x_{ijk} \geq 0 \qquad (i,j) \in A_0, k \in S \qquad \textbf{(2.47)}$$

$$y_{2i} \in \{0,1\} \qquad i \in H \qquad \textbf{(2.48)}$$

The objective function (2.35) together with the constraints (2.36) − (2.39) constitute the formulation of Akgün and Tansel [6]. The objective function (2.35) minimizes the total transportation cost. Constraints (2.36) are the flow balance constraints for all the nodes in each layer of the network G_0 and commodities. Constraint (2.37) requires p hubs be selected. Constraints (2.38) and (2.39) ensure that the flow between layers is possible only through arcs $(1i, 2i)$ and $(2i, 3i)$ if a hub is located at node i, i.e, $y_{2i} = 1$.

Constraints (2.40)−(2.44) construct a spanning tree in G_0. Constraint (2.40) sends $|N_2 - 1|$ units of fictitious flow from root node θ. Constraints (2.41) are flow-balance constraints that ensure all nodes in $N_2 - \theta$ receive one unit of fictitious flow. Constraints (2.42) require that there be exactly one incoming arc to each node $i \in (N_2 - \theta)$. Constraint (2.43) ensures that there is no incoming arc to root node θ from the nodes in N_2. Constraints (2.44) require an arc with a positive fictitious flow to be selected as an arc of the spanning tree in the hub layer. Constraints (2.45) allow commodity flows only on the arcs of the spanning tree and hence the arcs with positive commodity flows connect all hubs in a tree structure. Figure 2.9(a) illustrates an example of a tree spanning all nodes in the hub layer constructed by positive fictitious flows. Accordingly, Figure 2.9(b) gives the resulting tree structure connecting the hubs. Nodes 1 and 6 are the non-hub nodes serving as transition nodes to connect the hubs through a tree. Constraints (2.46)−(2.48) define the decision variables.

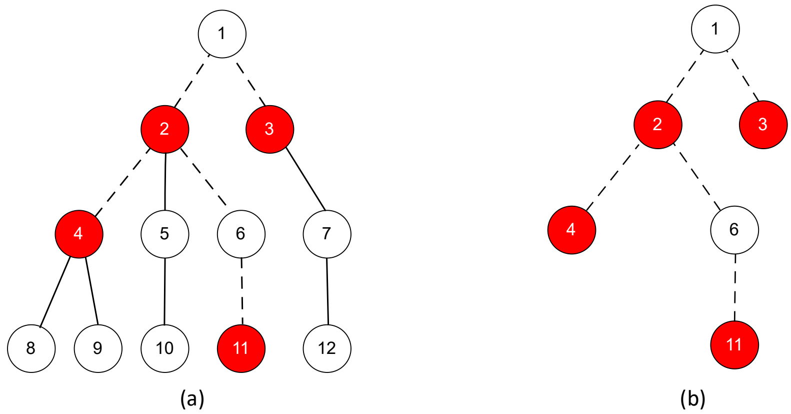

Figure 2.9. Illustration of the tree spanning all nodes in the hub layer (a) and the resulting tree structure connecting the hubs (b).

2.4 Proposed Solution Methodology

MATHLP is difficult to solve using standard optimization software. Computational studies indicate that CPLEX-based algorithm can find optimal or near-optimal solutions for problem instances defined on 81-node network with a run time of 24 h. However, for larger size networks, CPLEX either cannot find a solution or the resulting optimality gaps increase up to 36%. This has led us to develop a solution methodology based on Benders Decomposition (BD) [58], which has been used successfully in solving several variants of hub location problems including single allocation tree of hubs location problem (e.g., Camargo et al. [59], [60], [61], [62], [63], [64] Contreras et al. [65], [66], [67], Camargo and Miranda [68], Martins de Sá et al. [52], [48], [49], [42], Ghaffarinasab and Kara [69] , Mokhtar et al. [70], Taherkhani et al. [71]). However, in the application of BD algorithm even for small-size problems, the lower bound and the upper bound that we obtain from the algorithm were not close to each other. We have tried CPLEX's automatic BD and encountered with the same convergence problem. We have incorporated several acceleration strategies, namely, strong cut generation, cut disaggregation, and a combination of two strategies, in order to improve the convergence of the BD algorithm. However, these strategies have not been successful as well. For this reason, we have developed Benders-type heuristics. Of several heuristics, *the heuristic based on BD with strong cut generation and the heuristic based on BD with combined strong cut generation and cut disaggregation* have produced much better results than the other ones for all instances. In this regard, we present the development of these heuristics in this section. Computational studies indicate that the proposed heuristics are efficient and can find solutions for large-size problems with up to 500 nodes.

The BD approach partitions a difficult optimization problem into two simpler problems: an integer problem, named as the master problem (MP), and a linear problem, named as the subproblem (SP). The algorithm solves MP and SP iteratively and adds new constraints to the MP known as Benders cuts obtained from SP. The algorithm is terminated when the optimal objective function values of MP and SP are equal to each other.

One main challenge arising in the application of the BD algorithm is the need to solve difficult integer MPs in large size problems. As the number of iterations increases in this type of problems, the number of cuts added to the MP also increases, which makes the MP more difficult to solve and hence the convergence of the BD algorithm is too slow due to time and memory limitations. This has led the researchers to develop algorithms referred to as Benders-type heuristics. In most Benders-type heuristics, researchers use (meta)heuristic algorithms or relaxations to solve MPs. According to Boschetti and Maniezzo [72] BD algorithm provides a rich framework for developing heuristics since it uses dual information to reduce search space, verifies solution quality and obtains multiple starting points for local search. In most Benders-type heuristics, researchers relax the MP or use some kind of meta-heuristics for the MPs. Poojari and Beasley [73], Lai et al. [74], Lai et al. [75] solve the MP using genetic algorithm. Jiang et al. [76] use a similar approach, based on tabu search. Boschetti and Maniezzo [72] solve both MP and the subproblem using Lagrangean relaxation. Another approach in Benders type heuristics is to solve LP relaxation of the MP and to use round-off heuristics to find an integer solution. Pacqueau et al. [77] use the BD algorithm to solve the linear relaxation and then fix some of the variables to their upper/lower bounds. Optimality is not guaranteed with Benders-type heuristics. However, for computationally intractable problems, efficient Benders-type heuristics are able to reach near optimal solutions.

In the following, we give the BD algorithm, strong cut generation, cut disaggregation, and finally the Benders-type heuristics where a relaxed version of MP obtained is solved after removing some complicating constraints.

2.4.1 Benders Decomposition for MATHLP

As stated before, the BD algorithm decomposes the original problem into two simpler problems, namely, MP and SP, and solves MP and SP iteratively until their optimal objective function values are equal or a stopping criterion is reached. MP is a relaxed version of the original problem and involves the set of integer variables and associated constraints. SP is a linear program that is obtained by the dual problem formulated by fixing the values of integer variables in MATHLP.

Let $\boldsymbol{y}$ and $\boldsymbol{s}$ represent the vector of integer variables $y_{2i}, i \in H$, and $s_{ij}, (i,j) \in A_2$, respectively. Let SPD represent the linear problem obtained by fixing the values of integer variables $\boldsymbol{y}$ and $\boldsymbol{s}$ in MATHLP to $\boldsymbol{y^h}$ and $\boldsymbol{s}^h$, respectively, at iteration h. When $\boldsymbol{y}$ and $\boldsymbol{s}$ are fixed, the resulting SPD consists of (2.49)−(2.54) and is a linear routing problem that finds the routes between OD pairs because the hubs and hub arcs in the tree structure are fixed. We find SP by taking the dual of SPD with the dual variables e_{ik}, f_{ik}, g_{ik}, and tt_{ijk} defined for constraints (2.50) through (2.53), respectively. The resulting SP consists of (2.55)−(2.62).

Model SPD: Linear Problem at Iteration h

$$min \sum_{k\epsilon S} \sum_{(i,j)\epsilon A_0} c_{ij} x_{ijk} \qquad \textbf{(2.49)}$$

s.t.

$$\sum_{j \in F_\beta^{out}} x_{ijk} - \sum_{j \in F_\beta^{in}} x_{jik} = b_{ik} \qquad i \in (N_1 \cup N_2 \cup N_3), k \in S \qquad \textbf{(2.50)}$$

$$x_{(1i,2i)k} \le W_k y_i^h \qquad i \in H, k \in S \qquad \textbf{(2.51)}$$

$$x_{(2i,3i)k} \le W_k y_i^h \qquad i \in H, k \in S \qquad \textbf{(2.52)}$$

$$x_{ijk} \le W_k \ (s_{ij}^h + s_{ji}^h) \qquad (i,j) \in A_2 \qquad \textbf{(2.53)}$$

$$x_{ijk} \ge 0 \qquad (i,j) \in A, k \in S \qquad \textbf{(2.54)}$$

Model SP: Subproblem at Iteration h

$$Max \sum_{k\epsilon S} \sum_{i\epsilon N_0} b_{ik} e_{ik} \quad + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^h f_{ik} + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^h g_{ik} + \sum_{k\epsilon S} \sum_{a\epsilon A_2} W_k \big(s_{ij}^h + s_{ji}^h\big) tt_{ijk} \qquad \textbf{(2.55)}$$

s.t.

$$e_{ik} - e_{jk} \le l_{ij} \qquad (i,j) \in (A_1 \cup A_3), k \in S \qquad \textbf{(2.56)}$$

$$e_{ik} - e_{jk} + tt_{ijk} \leq l_{ij} \qquad (i,j) \in A_2, k \in S \tag{2.57}$$

$$e_{ik} - e_{jk} + f_{jk} \leq 0 \qquad (i,j) \in A_{12}, k \in S \tag{2.58}$$

$$e_{ik} - e_{jk} + g_{ik} \leq 0 \qquad (i,j) \in A_{23}, k \in S \tag{2.59}$$

$$f_{ik}, g_{ik} \leq 0 \qquad i \in H, k \in S \tag{2.60}$$

$$e_{ik} \quad free \qquad i \in (N_1 \cup N_2 \cup N_3), k \in S \tag{2.61}$$

$$tt_{ijk} \leq 0 \qquad (i,j) \in A_2, k \in S \tag{2.62}$$

We formulate MP by using the constraints associated with integer variables in MATHLP and adding Benders *optimality* cuts. A Benders optimality cut (2.63) can be derived from the objective function (2.55) of SP at iteration h. In (2.63), $e_{ik}^h, f_{ik}^h, g_{ik}^h$, and tt_{ijk}^h are the optimal values of the dual variables in SP at iteration h and η is the under-estimator for the total cost. The resulting MP consists of (2.64)−(2.73).

$$\eta \geq \sum_{k\epsilon S}\sum_{i\epsilon N_0} b_{ik}e_{ik}^h + \sum_{k\epsilon S}\sum_{i\epsilon N_0} W_k f_{ik}^h y_i + \sum_{k\epsilon S}\sum_{i\epsilon N_0} W_k g_{ik}^h y_i + \sum_{i\epsilon N_2}\sum_{j\epsilon N_2}\sum_{k\epsilon S} W_k * tt_{ijk}^h * (s_{ji} + s_{ji}) \tag{2.63}$$

Model MP: Master Problem at Iteration h

$$\text{Min } \eta \tag{2.64}$$

s.t.

$$\eta \geq \sum_{k\epsilon S}\sum_{i\epsilon N_0} b_{ik}e_{ik}^h + \sum_{k\epsilon S}\sum_{i\epsilon N_0} W_k f_{ik}^h y_i + \sum_{k\epsilon S}\sum_{i\epsilon N_0} W_k g_{ik}^h y_i + \sum_{i\epsilon N_2}\sum_{j\epsilon N_2}\sum_{k\epsilon S} W_k * tt_{ijk}^h * (s_{ji} + s_{ji}) \tag{2.65}$$

$$\sum_{i\in H} y_{2i} = p \tag{2.66}$$

$$\sum_{j\in F_\theta^{out}} t_{\theta j} = |N_2 - 1| \tag{2.67}$$

$$\sum_{j\in F_\beta^{out}} t_{\beta j} - \sum_{j\in F_\beta^{in}} t_{j\beta} = -1 \qquad \beta \in (N_2 - \theta) \tag{2.68}$$

$$\sum_{j \in F_\beta^{in}} s_{j\beta} = 1 \qquad \beta \in (N_2 - \theta) \tag{2.69}$$

$$\sum_{j \in (F_\theta^{in} \cap N_2)} s_{j\theta} = 0 \tag{2.70}$$

$$t_{ij} \leq |N_2 - 1|\, s_{ij} \qquad (i,j) \in A_2 \tag{2.71}$$

$$t_{ij} \geq 0\,,\ s_{ji} \in \{0,1\} \qquad (i,j) \in A_2 \tag{2.72}$$

$$y_{2i} \in \{0,1\} \qquad i \in H \tag{2.73}$$

MP determines the locations of p hubs and connects them in a tree structure through constraints (2.66) - (2.71) at each iteration. Given $\boldsymbol{y}^h$ and $\boldsymbol{s}^h$ at iteration h, SPD is mainly a network flow problem. Because supply and demand quantities are equal and there are no capacity constraints in SPD, it is always feasible. Moreover, the objective function value of SPD is bounded because transportation costs are non-negative and finite, which means that SP, the dual of SPD, is feasible and has a bounded objective function value. Thus, at each iteration of the Benders algorithm, we obtain a feasible solution. This is why we do not need to generate and add Benders *feasibility* cuts, we add only Benders optimality cuts to MP. Otherwise, we would have to add feasibility cuts as well.

MP is a relaxation of the original problem and hence its objective function value provides a lower bound to that of MATHLM. We improve this bound at each iteration by adding Benders optimality cut (45). By construction, the objective function value of SP provides an upper bound on the objective function value of MATHLM.

2.4.2 Acceleration Strategies for the BD Algorithm

The performance of the BD Algorithm is mainly determined by the number of iterations and the time required to complete each iteration (e.g., Rahmaniani et al. [78]). The number of iterations may be high if the improvement rate of the LB is low, which results from weak Benders cuts. This also adversely affects the solution time of MP and memory requirements because the higher the number of cuts added to MP, the more difficult it becomes to solve MP. In some cases, solving SP may take excessive time, too. All of these may cause poor convergence of the algorithm as we have experienced.

There are several strategies employed in the literature to accelerate the progress of the BD Algorithm. In this study, we employ *generating strong cuts and disaggregating the Benders cuts together*.

2.4.2.1 Generating Strong Cuts

Magnanti and Wong ([79] suggest an approach to generate *stronger* optimality cuts based on the determination of a *core point*. To formalize the approach, let C_a and C_b represent two different cuts generated using (2.63) from two different solutions $(\boldsymbol{e}^a, \boldsymbol{f}^a, \boldsymbol{g}^a, \boldsymbol{tt}^a)$ and $(\boldsymbol{e}^b, \boldsymbol{f}^b, \boldsymbol{g}^b, \boldsymbol{tt}^b)$, respectively. Then, C_a is stronger than or dominates C_b if the right-hand-side value of C_a is greater than or equal to that of C_b with a strict inequality for at least one point $(\boldsymbol{y}, \boldsymbol{s})$. That is, the cut that gives a better bound is a dominating cut. A cut is *pareto-optimal* if it is not dominated by any other cuts. Accordingly, the solution $(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{tt})$ is pareto-optimal if the cut defined by $(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{tt})$ is pareto optimal.

A pareto optimal solution $(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{tt})$ can be obtained by solving an optimization problem that finds a pareto-optimal point among alternative optimal solutions using a *core point*. A core point $(\boldsymbol{y}^0, \boldsymbol{s}^0)$ is a point in the relative interior of the convex hull of $\boldsymbol{y} \in Y$ and $\boldsymbol{s} \in \bar{S}$.

To define the optimization problem to solve, let $(\boldsymbol{y}^h, \boldsymbol{s}^h)$ be the optimal solution of MP and z^*_{SPh} be the optimal objective function value of SP at iteration h. The optimal solution $(\boldsymbol{e}^{\boldsymbol{0}}, \boldsymbol{f}^{\boldsymbol{0}}, \boldsymbol{g}^{\boldsymbol{0}}, \boldsymbol{tt}^{\boldsymbol{0}})$ to the optimization problem PRT comprised of (2.74)−(2.78) is a pareto-optimal solution. The Benders cut obtained from the objective function (2.74) is a pareto-optimal cut. This cut is closest to the chosen core point $(\boldsymbol{y}^0, \boldsymbol{s}^0)$.

Model PRT: Model to Find a Pareto-Optimal Solution

$$Max \sum_{k\epsilon S} \sum_{i\epsilon N_0} b_{ik} e_{ik} + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^0 f_{ik} + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^0 g_{ik} + \sum_{i\epsilon N_2} \sum_{j\epsilon N_2} \sum_{k\epsilon S} W_k * \left(s_{ij}^0 + s_{ji}^0\right) * tt_{ijk} \qquad \textbf{(2.74)}$$

$s.t.$

$$\sum_{k\epsilon S} \sum_{i\epsilon N_0} b_{ik} e_{ik} + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^h f_{ik} + \sum_{k\epsilon S} \sum_{i\epsilon N_0} W_k y_i^h g_{ik} + \qquad \textbf{(2.75)}$$

$$\sum_{i \epsilon N_2} \sum_{j \epsilon N_2} \sum_{k \epsilon S} W_k * \left(s_{ij}^h + s_{ji}^h\right) * tt_{ijk} = z_{SPh}^*$$

$f_{ik}, g_{ik} \leq 0$	$i \in H, k \in S$	**(2.76)**
e_{ik} *free*	$i \in (N_1 \cup N_2 \cup N_3), k \in S$	**(2.77)**
$tt_{ijk} \leq 0$	$(i,j) \in A_2, k \in S$	**(2.78)**

The approach is based on the fact when SP has multiple optimal solutions, different cuts with different strengths may be defined. PRT generates the strongest cut possible. A pareto-optimal cut may be added at each iteration or periodically considering the tradeoff between additional computational burden and the reduction in the number of iterations. In this study, we add pareto-optimal cuts in each iteration. In the Benders Algorithm with pareto-optimal cuts, PRT is solved after solving SP and the cut generated using a given core point is added to MP.

Finding a core point may be a challenge for some problems (e.g., Martins de Sa et al., 2013 [52]). Mercier et al. [80] state that using a core point that is not in the interior of the convex hull does not preclude finding a valid Benders cut. However, the further the core point is from the interior of the convex hull, the weaker the Benders cuts that are generated by this method. This demonstrates the importance of finding a good core point. Mercier et al. [80] state that values of binary variables close to 0 or 1 generate stronger cuts for different problem types. In this study, we conduct computational experiments by setting $\boldsymbol{y}$ and $\boldsymbol{s}$ to 0 and 1 to identify a good core point. Computational results indicate that setting $\boldsymbol{y} = \mathbf{1}$ and $\boldsymbol{s} = \mathbf{0}$ yields stronger cuts.

2.4.2.2 Disaggregating the Benders Cuts and Adding Multiple Cuts

Another strategy to improve the progress of BD Algorithm is to add multiple cuts instead of just one cut at each iteration. We can achieve this by disaggregating Benders cuts (2.63) as proposed by Birge and Louveaux [81]. Disaggregation is possible because SP can be decomposed into smaller problems based on commodity type k. This allows us to add $|S|$ cuts simultaneously. The resulting MP, **MPM**, is comprised of (2.66)−(2.73) and (2.79)−(2.80).

Model MPM: Master Problem With Disaggregated Cuts at Iteration h

$$Min \sum_{k \in S} \eta_k \qquad \textbf{(2.79)}$$

s.t.

In addition to Constraints (2.66) - (2.73)

$$\eta_k \geq \sum_{i \in N_0} b_{ik} e_{ik}^h + \sum_{i \in N_0} W_k f_{ik}^h y_i + \sum_{i \in N_0} W_k g_{ik}^h y_i + \sum_{i \in N_2} \sum_{j \in N_2} W_k * tt_{ijk}^h * (s_{ij} + s_{ji}) \qquad \forall\, k \in S \qquad \textbf{(2.80)}$$

In the Benders Algorithm with multiple cuts, MPM rather than MP is solved.

It may be possible to disaggregate (29) further, e.g., based on commodity type k and node i. However, the number of cuts in this case is $3n^2$ and solving MPM even for small-size problems becomes computationally very expensive. This is why we prefer to disaggregate based on k.

2.4.2.3 Combining Strong Cut Generation and Multiple Cuts

We use the strategies explained in 2.4.2.1 and 2.4.2.2 simultaneously. In doing that, we first find the pareto-optimal cut and then disaggregate this pareto-optimal cut into multiple cuts. This requires solving PRT to find a pareto-optimal solution after solving SP and then disaggregating the cut as given in MPM.

2.4.3 Benders-Type Heuristic Approach

Computational experiments with the BD-based algorithms with or without acceleration strategies indicate that they are not promising to be used to solve large-scale problem instances. We observe that the progress of the algorithms is limited due to difficulty in solving the master problems. This leads us to develop two Benders-Type heuristics that facilitate the solution of the master problems and keeping the rest of the algorithms essentially the same.

Our approach is based on obtaining $(\boldsymbol{y}^h, \boldsymbol{s}^h)$ at iteration h in two steps: (1) determining the hubs to locate by solving a relaxed master problem and (2) finding the tree structure connecting the located hubs by solving a rooted spanning tree formulation. We adopt this approach because we observe that the constraints (2.67)−(2.71) that ensure the tree structure in the master problems increase the solution time of the master problems significantly or make them almost impossible to solve for large-size problems.

The master problems that need to be solved in the heuristic algorithms can be stated by eliminating the variables $\boldsymbol{s}^h$ and associated terms from the formulations. We define two relaxed master problems, **RelMP** to be used while adding a single cut and **MCMP** to be used while adding multiple cuts.

Model RelMP: Relaxed Master Problem with a Single Cut

$$\text{Min } \eta \tag{2.81}$$

s.t.

$$\eta \geq \sum_{k\in S}\sum_{i\in N_0} b_{ik}e_{ik}^h + \sum_{k\in S}\sum_{i\in N_0} W_k f_{ik}^h y_i + \sum_{k\in S}\sum_{i\in N_0} W_k g_{ik}^h y_i \tag{2.82}$$

$$\sum_{i\in H} y_{2i} = p \tag{2.83}$$

$$y_{2i} \in \{0,1\} \qquad \forall\, i \in H \tag{2.84}$$

Model MCMP: Relaxed Master Problem with Multiple Cuts

$$Min \sum_{k\in S} \eta_k \tag{2.85}$$

s.t.

In addition to (2.83)-(2.84)

$$\eta_k \geq \sum_{i\in N_0} b_{ik}e_{ik}^h + \sum_{i\in N_0} W_k f_{ik}^h y_i + \sum_{i\in N_0} W_k g_{ik}^h y_i \qquad \forall\, k \in S \tag{2.86}$$

RelMP and MCMP solved at iteration h determine the values of $\boldsymbol{y}^h$ where p of the variables are 1 and the remaining are zero. To determine the tree structure among the given hub locations i with $y_{2i} = 1$, we solve a rooted spanning tree formulation, **SPTree,** based on single commodity flows t_{ij}. We define one of the hub locations, say node θ, as the root/supply node from which one unit of fictitious flow is sent to other hub locations, i.e., a total of $p-1$ units of flow is sent from θ. After solving SPTree, for the arcs with $t_{ij} > 0$, we set $s_{ij} = 1$. The solution $(\boldsymbol{y}^h, \boldsymbol{s}^h)$ is then fed into SP.

Model SPTree: Rooted Spanning Tree Formulation Restricted to Hub Locations

$$min\ Z^* = \sum_{k\in S}\sum_{(i,j)\in A_0} c_{ij}t_{ij} \tag{2.87}$$

s.t.

$$\sum_{j\in F_\theta^{out}} t_{\theta j} = p - 1 \tag{2.88}$$

$$\sum_{j\in F_\beta^{out}} t_{\beta j} - \sum_{j\in F_\beta^{in}} t_{j\beta} = -1 \qquad \beta \in (H-\theta) \tag{2.89}$$

$$\sum_{j\in F_\beta^{out}} t_{\beta j} - \sum_{j\in F_\beta^{in}} t_{j\beta} = 0 \qquad \beta \in (N_2-H) \tag{2.90}$$

$$t_{ij} \geq 0 \qquad (i,j)\in A_2 \tag{2.91}$$

Given the relaxed master problems (RelMP and MCMP) and SPTree, we define two Benders-Type Heuristic algorithms, **BDHEUR1** and **BDHEUR2**, that are mainly different in the acceleration strategies employed and the master problem solved. In BDHEUR1, we only *generate strong cuts*. In the application of the algorithm, we first determine a pareto-optimal cut as explained in Section 2.4.2.2 and then add it to RelMP. In BDHEUR2, we use two strategies together, *generating strong cuts* and *disaggregating Benders cuts*. In the application of the algorithm, we determine a pareto-optimal cut, disaggregate this cut into multiple cuts as explained in Section 2.4.2.3, and add them to MCMP. We outline the steps of BDHEUR1 below. The steps of BDHEUR2 are the same as BDHEUR1 except that we solve MCMP instead of RelMP, i.e., replace RelMP with MCMP. In the application of the algorithms, we solve SP and SPTree to optimality and RelMP/MCMP until an optimality gap of 10% is achieved. Moreover, for the test problems on PMED400 and PMED500, we set a time limit of 1 h and 5 h, respectively, for the Model PRT that finds a pareto-optimal cut because we observe that strong cuts are obtained within that time limit.

In the algorithm, UB and z_{SP}^* represent the upper bound and the optimal objective function value for SP, respectively.

Algorithm BDHEUR1: Benders-Type Heuristic 1 Employing Strong Cut Generation.

Step 1: (Initialization)
Set $\boldsymbol{y}^h$ and $\boldsymbol{s}^h$ to an initial feasible integer solution.
Set time limit.
Set *UB* = +∞
Set $h = 0$
Find a core point $(\boldsymbol{y}^0, \boldsymbol{s}^0)$ for strong cut generation
Step 2: Solve SP (2.55)−(2.62)
Step 3: Set UB=min(UB, z_{SP}^*)
Step 4: Solve PRT (2.74)−(2.78) to choose a strong cut
Step 5: Add cut(s) to RelMP (2.81)−(2.84)

Step 6: Solve RelMP (2.81)−(2.84) to get y
Step 7: Solve SPTree (2.87)−(2.91) to get $\boldsymbol{s}$
Step 8: Set *h*=*h*+1
Step 9: Set $\boldsymbol{y}^h = \boldsymbol{y}$ and $\boldsymbol{s}^{\boldsymbol{h}} = \boldsymbol{s}$
Step 11: If elapsed time > time limit, stop. Otherwise, go to Step 2

2.5 Computational Experiments

We conduct computational experiments to test the performance of the proposed model and solution methodologies. Specifically, we observe the performance of MATHLP using CPLEX, Gurobi, Gurobi with NoRel Heuristic, and LocalSolver and Benders-type heuristics. We prefer the aforementioned solvers and algorithms because they are known to be effective for difficult MIP problems. In the application of Gurobi with NoRel heuristic, which may be useful for models where the root relaxation is quite expensive, the NoRel heuristic first tries to find a high-quality feasible solution in the allocated time and then Gurobi implements the branch and cut algorithm with the feasible solution found by the NoRel heuristic [82]. Localsolver is an innovative optimization solver combining exact and heuristic techniques and finds high-quality solutions for large-scale optimization problems [83]. However, LocalSolver cannot find feasible solutions even for small-size instances. Gurobi with NoRel heuristic performs much better than Gurobi for all instances in finding feasible solutions. Because of this, we only present the results obtained by CPLEX and Gurobi with NoRel Heuristic against the results obtained by Benders-type heuristics.

We define test problems on TR81, PMED200, PMED300, PMED400, and PMED500 networks. TR81 is defined by Akgün and Tansel [6] and the non-complete transportation network of Turkey including all 81 cities of Turkey. The edges on TR81 are defined only between adjacent cities. The length of the edges on TR81 are assumed to be the direct distances from the high-way transportation network of Turkey. PMED200 through PMED500 are the non-complete networks used for the p-median problem instances (e.g., Beasley [84]) with the numbers indicating the number of nodes. Different test problems are created on the networks by changing $|H|$ and p where $E^* = E, N^* = N$ and $H \subseteq N^*$. For all test problems, w_{ij} is uniformly distributed with the interval (10,30). For all arcs, χ_{ij} and δ_{ij} are taken as 1 whereas α_{ij} is taken as 0.7. In all problems, $S = D = N$.

We code the models and the algorithms using GAMS and conduct the experiments on a PC with 3.6 GHz Intel Core i7-7700CPU processor and 32 GB of RAM for TR81, PMED200, PMED300 instances and on a server with Intel® Xeon® CPU E5-2683 V4 @ 2.1 GHz 64 core processor and 256 GB of RAM for PMED400 and PMED500 instances due to memory requirements. The runtime for the solvers and the algorithms is set to 24 h (86,400 secs). In using Gurobi with NoRel heuristic, we assign 12 h for the NoRel heuristic and 12 h for the branch and cut algorithm because we obtain high-quality feasible solutions with this setting.

In the tables, we present (1) the runtime (T) in CPU secs, the lower bound (LB), the objective function value of the best integer solution at the end of runtime (BP) obtained by CPLEX or Gurobi with NoRel heuristic, and the relative optimality gap (Gap%) between LB and BP for MATHLP and (2) the best integer solution achieved either from CPLEX or Gurobi with NoRel heuristic (BP^*); the number of iterations (# of iters), LB, and UB achieved by the heuristic algorithms.

2.5.1 Computational experiments for MATHLP using CPLEX and Gurobi with NoRel heuristic

Table 2.2 gives the results obtained solving MATHLP by CPLEX and Gurobi with NoRel heuristic. In solving MATHLP using CPLEX, we use two different parameter settings that differ only in the value of *mipemhasis* parameter. The mipemphasis parameter value tells CPLEX what the balance between finding better feasible solutions and proving optimality should be in solving a model. We use two values of mipemphasis parameter, namely, “balance feasibility and optimality” and “feasibility” because our focus is to find better feasible solutions. With regard to finding BP values, no setting dominates the other one. In this regard, we present the results of the setting under which better BP value is obtained for each instance in Table 2.2.

In Table 2.2, **bold** and *italic* values indicate the same or better **BP** and *LB* values for each instance. **Gurobi with NoRel heuristic** (CPLEX) mostly finds better **BP** (*LB*) values than CPLEX (Gurobi with NoRel heuristic); however, it cannot find a solution for Problem 30. The last column indicates the best BP values, BP^*. Table 2.2 shows that as the problem size increases, the optimality gaps increases considerably indicating that it becomes more difficult to solve large-scale problems.

Table 2.2 Test results for MATHLP using CPLEX and Gurobi with NoRel heuristic.

						CPLEX			Gurobi with NoRel Heuristic			
Pr. Id	Network	|N|	|H|	p	T (secs)	LB	BP	Gap (%)	LB	BP	Gap (%)	BP*
1	TR81	81	30	3	84840	*113261890*	**113261890**	0.0	113120288	**113261890**	0.0	113261890
2		81	30	5	86400	100804000	**103530000**	2.7	*102044941*	**103530000**	1.4	103530000
3		81	50	3	86400	*107115000*	**112944470**	5.4	105087303	**112944470**	7.0	112944470
4		81	50	5	86400	*96404800*	103014691	6.9	95881444	**102883188**	6.8	102883188
5		81	50	8	86400	*91322400*	97216421	6.5	91015050	**96098414**	5.3	96098414
6		81	50	10	86400	*89545200*	93852241	4.8	88953013	**93793657**	5.2	93793657
7		81	60	3	86400	*102576000*	**112944470**	10.1	100724739	**112944470**	10.8	112944470
8		81	60	5	86400	*95209200*	103426356	8.6	94446443	**102940709**	8.3	102940709
9		81	60	8	86400	89687000	96185436	7.2	*90368120*	**96098414**	6.0	96098414
10		81	60	10	86400	*89205300*	94098044	5.5	88302192	**93793657**	5.9	93793657
11		81	81	3	86400	*99773300*	**113067656**	13.3	98948604	**113067656**	12.5	113067656
12		81	81	5	86400	*93627400*	102909588	9.9	93297541	**102883188**	9.3	102883188
13		81	81	8	86400	*89710800*	98401597	9.7	89309497	**96184127**	7.1	96184127
14		81	81	10	86400	86910600	94461872	8.7	*87454873*	**93710445**	6.7	93710445
15	PMED200	200	200	3	86400	69962197	83921015	20.0	*70242011*	**82844187**	15.2	82844187
16		200	200	5	86400	*67681170*	81743444	20.8	67065627	**77314463**	13.2	77314463
17		200	200	8	86400	*65226836*	74061217	13.5	64502322	**72475694**	11.0	72475694
18		200	200	10	86400	*63947068*	72858377	13.9	63309299	**71176399**	11.0	71176399
19	PMED300	300	300	3	86400	100542251	152379779	34.0	*102105607*	**129512902**	21.2	129512902
20		300	300	5	86400	93990900	148400000	36.3	*99812107*	**126641742**	21.2	126641742
21		300	300	8	86400	*97915345*	118263997	17.2	97631794	**117762104**	17.1	117762104
22		300	300	10	86400	96342738	**111688878**	13.7	*97046000*	112496864	13.7	111688878
23	PMED400	400	400	3	86400	*129405098*	2521891782	94.9	126492232	**188065072**	32.0	188065072
24		400	400	5	86400	*126975841*	**159334212**	20.3	125777210	217002369	42.0	159334212
25		400	400	8	86400	*125823824*	191544185	34.3	124213039	**185954774**	33.0	185954774
26		400	400	10	86400	*125305399*	**152776681**	18.0	123575853	164174581	24.0	152776681
27	PMED500	500	500	3	86400	*172189807*	3954137863	95.7	169677900	**288398427**	41.2	288398427
28		500	500	5	86400	*172544439*	3920051544	95.6	163650171	**267833441**	38.8	267833441
29		500	500	8	86400	*171012235*	3952102853	95.7	159240544	**238740589**	33.3	238740589
30		500	500	10	86400	*169673118*	**4094682459**	95.9	160733186	no solution	-	4094682459

2.5.2 Computational Experiments with the Benders-type Heuristics

We initiate the algorithms with an initial solution $(\boldsymbol{y}, \boldsymbol{s})$ where $\boldsymbol{y}$ is found by setting its first p elements to 1 and the remaining to 0 and $\boldsymbol{s}$ is found by solving SPTree. We use a core point with $\boldsymbol{y}^0 = 1$ and $\boldsymbol{s}^0 = 0$ since computational results indicate that this core point yields stronger cuts.

Table 2.3 presents the results. In the table, we give GapHeur (%) defined as 100×(UB - BP^*)/BP^* in order to compare the solutions of the heuristics to BP*, the best solution found by either CPLEX or Gurobi with NoRel heuristic. A *positive (negative)* value indicates that the UB achieved by the heuristic algorithm is *worse (better)* than BP*. Instances with **bold UB values** are the ones for which a heuristic can either find the same solution or a better solution than CPLEX or Gurobi with NoRel heuristic.

Italic (normal or ***bold)*** UB values are used to show the heuristic that produces a better UB than the other heuristic. BDHEUR1 produces better UB values for 8 problems (4, 6, 9, 10, 11, 24-26) while BDHEUR2 produces better UB values for 19 problems (2, 5, 8, 12-23, 27-30). For the remaining problems, they find the same solutions. UB values found by BDHEUR1 (BDHEUR2) are on the average 0.39% (5.2%) better than those of BDHEUR2 (BDHEUR1). Considering these results, we can conclude that BDHEUR2 performs better than BDHEUR1. In this regard, we will continue our analysis with BDHEUR2.

For TR81 instances, GapHeur values change from 0% to 3% with an average of 1.6%. The heuristic can find a solution equivalent to BP* for three instances (problems 1, 3, and 7) out of 14 instances. For the remaining instances for which BP* values are better, GapHeur values change from 0.1% to 3%.

For PMED200 instances, GapHeur values range from −0.3% to 1.8% with an average of 0.6%. The heuristic can find a better solution for one instance (problem 15) and the same solution for one instance (problem 16) out of 4 instances. For PMED300 instances, GapHeur values change from −2% to 3.9% with an average of 1.5%. The heuristic can find a better solution for one instance (problem 21) out of 4 instances. For PMED400 instances, GapHeur values change from −17% to −0.2% with an average of -7.5%. For PMED500 instances, GapHeur values change from −94.8% to −7.3% with an average of −31.8%. The heuristic can find a better solution for all instances.

The results show that, as the network size gets larger, CPLEX or Gurobi with NoRel heuristic find a solution with high optimality gaps. On the other hand, the heuristics can find solutions either close to or better than those found by CPLEX or Gurobi with NoRel heuristic, i.e., Benders-type heuristic algorithms are effective in finding good solutions.

Table 2.3 Test results for the instances using BDHEUR (T=24h)

Pr. Id.	Network	|N|	|H|	p	BP*	BDHEUR1 # of Iters	BDHEUR1 UB	BDHEUR1 GapHeur (%)	BDHEUR2 # of Iters	BDHEUR2 UB	BDHEUR2 GapHeur (%)
1	TR81	81	30	3	113261890	4287	**113261890**	**0.0**	854	**113261890**	**0.0**
2	TR81	81	30	5	103530000	2670	105345943	1.8	520	*105027039*	1.4
3	TR81	81	50	3	112944470	3018	**112944479**	**0.0**	368	**112944479**	**0.0**
4	TR81	81	50	5	102883188	1415	*105242582*	2.3	514	*105297298*	2.3
5	TR81	81	50	8	96098414	1229	98468128	2.5	636	*98340058*	2.3
6	TR81	81	50	10	93793657	1175	*95893977*	2.2	1326	96563871	3.0
7	TR81	81	60	3	112944470	2539	**112944479**	**0.0**	284	**112944479**	**0.0**
8	TR81	81	60	5	102940709	1422	105242582	2.2	446	*104985323*	2.0
9	TR81	81	60	8	96098414	815	*98911891*	2.9	574	98970843	3.0
10	TR81	81	60	10	93793657	709	*95947980*	2.3	1048	96141575	2.5
11	TR81	81	81	3	113067656	1878	***112944479***	***-0.1***	245	113203327	0.1
12	TR81	81	81	5	102883188	659	105692069	2.7	115	*104534769*	1.6
13	TR81	81	81	8	96184127	555	98815936	2.7	547	*98015933*	1.9
14	TR81	81	81	10	93710445	499	96494930	3.0	1114	*96046631*	2.5
							Max	3.0		Max	3.0
							Min	-0.1		Min	0.0
							Average	1.8		Average	1.6
15	PMED200	200	200	3	82844187	335	86751752	4.7	199	***82579184***	**-0.3**
16	PMED200	200	200	5	77314463	180	84192026	8.9	232	***77297928***	**-0.6**
17	PMED200	200	200	8	72475694	132	79995083	10.4	200	*73794169*	1.8
18	PMED200	200	200	10	71176399	135	78755379	10.6	165	*71904326*	1.0
							Max	10.6		Max	1.8
							Min	4.7		Min	-0.3
							Average	8.7		Average	0.6
19	PMED300	300	300	3	129512902	30	146804453	13.4	35	*134514203*	3.9
20	PMED300	300	300	5	126641742	34	137386549	8.5	36	*130470514*	3.0
21	PMED300	300	300	8	117762104	34	132241629	12.3	40	***115380696***	**-2.0**
22	PMED300	300	300	10	111688878	34	128240795	14.8	42	*112950607*	1.1
							Max	14.8		Max	3.9
							Min	8.5		Min	-2.0
							Average	12.2		Average	1.5
23	PMED400	400	400	3	188065072	23	177360845	**-5.7**	23	***165976456***	**-11.7**
24	PMED400	400	400	5	159334212	23	***156768212***	**-1.6**	23	157830551	**-0.9**
25	PMED400	400	400	8	185954774	23	***154057167***	**-17.2**	23	154424692	**-17.0**
26	PMED400	400	400	10	152776681	23	***151006447***	**-1.2**	23	152523259	**-0.2**
							Max	**-1.2**		Max	**-0.2**
							Min	**-17.2**		Min	**-17.0**
							Average	**-6.4**		Average	**-7.5**
27	PMED500	500	500	3	288398427	5	260076970	**-9.8**	5	***242867355***	**-15.8**
28	PMED500	500	500	5	267833441	5	249145118	**-7.0**	5	***242398659***	**-9.5**
29	PMED500	500	500	8	238740589	5	234735216	**-1.7**	5	***221405699***	**-7.3**
30	PMED500	500	500	10	4094682459	5	231288249	**-94.4**	5	***214659947***	**-94.8**
							Max	-1.7		Max	-7.3
							Min	-94.4		Min	-94.8
							Average	-28.2		Average	-31.8

2.6 Conclusion

In this chapter, we present the *Multiple Allocation Tree of Hubs Location Problem* where the hub-level network is required to have a tree topology and transportation cost of sending flows between OD pairs is minimized. Most studies in the literature assume a complete network with costs satisfying the triangle inequality to formulate the problem. If the underlying real-life network is not complete or complete but its distances do not satisfy the triangle inequality, a preprocessing on the underlying network is implemented to construct a complete network whose costs satisfy the triangle inequality.

Unlike the previous studies, we have defined the problem on non-complete networks and developed a modeling approach that does not require any specific cost and network structure. The modeling approach allows us to use the structure of the real physical network directly in the formulation of the problem. We have shown that the proposed modeling approach may produce better solutions than a modeling approach that uses a complete network structure whose costs satisfy the triangle inequality, which may result from the differences in the selection of the hubs, the flow routes between hubs, and the assignments of non-hub nodes to hub nodes. The proposed approach may also provide more flexibility in modeling several characteristics real-life hub networks, e.g., the interactions between location and routing decisions, arcs with different costs and capacities, different topology and service level requirements.

In the study, we have solved the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic and developed BD-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We have conducted computational experiments using networks with up to 500 nodes. As the network size gets larger, the resulting optimality gaps are high for the solutions found by CPLEX or Gurobi with NoRel heuristic. On the other hand, the heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for all instances, i.e., Benders-type heuristic algorithms are effective in finding good solutions.

In the future, we may incorporate other acceleration strategies not considered in this study, e.g., reduction of the model size and selection of good initial cuts, to improve the progress of exact Benders algorithms or Benders-type heuristics. We may develop hybrid algorithms utilizing metaheuristics and Benders decomposition to improve the effectiveness of the heuristic algorithms. A problem specific branch-and-bound algorithm may be developed as well.

Chapter 3

MULTIPLE ALLOCATION ARC CAPACITATED HUB LOCATION PROBLEM

In this chapter, we consider *Multiple Allocation Arc Capacitated Hub Location Problem* (MACHLP) that imposes an upper limit on the flow traversing some of the arcs in the network. The objective of the problem is to minimize the total transportation cost needed to transport the given flow between OD pairs via hub nodes with locating *p* hubs and satisfying the capacity constraints defined on the arcs.

Most hub location problems are known to be NP-hard (e.g., Carello et al. [32]; Alumur and Kara [1]; Contreras and O'Kelly [4]). MACHLP is NP-hard as well since MACHLP can be transformed into the uncapacitated multiple allocation hub location problem known to be NP-hard [85].

In Section 3.1, we give the related literature for MACHLP. We propose a new MIP model for MACHLP that is built upon the problem setting adopted by Akgün and Tansel [6]. We use the 3-layered framework introduced before. In Chapter 3.2 we discuss the advantages of using the proposed modeling approach. In Section 3.3, we define the problem and present the details of the MIP model. The proposed model is defined on non-complete networks but can also be used with complete networks. We solve the model by the CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic. In Section 3.4, we develop a heuristic approach based on Simulated Annealing (SA) algorithm. In Section 3.5, we present the computational tests conducted to assess the performance of the proposed heuristic using instances defined on different networks with the number of nodes changing from 81 to 400. In Section 3.6, we conclude the chapter.

3.1 Literature Review

Capacity constraints on the hub networks may be imposed both on the nodes and the arcs of the network. However, most studies incorporate capacity constraints only on the nodes. In these studies, an upper limit is imposed on the incoming or outgoing flow at the hub nodes. A motivating example for hub capacitated HLP is the postal delivery application. In postal delivery, the volume of mail that hubs can sort is limited by time constraints resulting in capacity restriction on the hub nodes. Hub capacitated versions of HLPs with single allocation strategy are studied by Campbell [17], Ernst and Krishnamoorthy [18], Labbé et al. [21], Correia et al. [23], Contreras et al. [24], [25]. Capacitated versions of HLPs with multiple allocation strategy are studied by Campbell [17], Ebery et al. [19], Boland et al. [20], and Marín [86].

Carello et al. [32], Yaman and Carello [87], and Yaman [47] study the single allocation hub capacitated HLPs with modular link capacities. They consider fixed costs of installing hubs and fixed costs of installing the required capacity on each edge to route the traffic. They install a number of links with fixed capacity on the edges and determine the capacity required to route the traffic on an edge. Their aim is to design the hub network by minimizing the sum of hub costs and link costs. These problems appear to be a design rather than an allocation problem. In this regard, these problems are different from the problem that we address. Our problem MACHLP deviates from these studies by minimizing the total transportation cost by satisfying the capacity constraints defined apriori on the arcs.

Contreras and O'Kelly [4] and Alumur et al. [25] state that the capacity constraints may arise not only at the hub facilities but also at the arcs of the network, and arc capacities are important in some settings of HLPs in which amount of flow traversing the arcs has an upper limit. Bryan [26] is the first to introduce a model in which capacities are associated with the hub arcs rather than with the hub nodes. He states that large amounts of flow traveling across the same hub arc can create practical problems and, moreover, physical constraints can place a limit on the amount of flow that can be handled on any single hub arc. Bryan [26] modifies the model (FLOWLOC) developed by Bryan and O'Kelly [88] to include capacitated hubs arcs by adding the required constraints. The FLOWLOC model is a multiple allocation HLP model with the objective of minimizing the total flow cost. It allows unit costs to vary with the amount flow traversing the hub arcs instead of assuming a fixed discount for the hub arcs. For this purpose, the model includes a piecewise linearization of a nonlinear cost function in which costs increase at a decreasing rate as flows increase. With the computational experiments carried out, Bryan and O'Kelly [88] conclude that only a few of the hub arcs amass large amounts of flow when they use the FLOWLOC model. For that reason, Bryan [26] proposes that a capacitated network may be required to prevent congestion problems on the heavily traveled hub arcs and modifies the FLOWLOC model by incorporating the hub arc capacities. In the computational experiments, Bryan [26] uses a data set based on airline passenger travel with 100 nodes. He uses fixed hub locations to be able to examine the effects of capacity constraints and also compares alternative sets of hub locations while maintaining reasonable computation times. In order to examine the effect on network design as the capacity changes, several different capacity levels are examined. Hub arc flows from the uncapacitated version of the model are used as a guide in determining the upper and lower bounds for the capacity levels. As expected, total network cost per unit flow increases as the capacity level decreases and heavily traveled hub arcs are avoided with the hub arc capacities.

Sasaki and Fukushima [27] present a new formulation of a one-stop capacitated hub-and-spoke model that involves arc capacity constraints as well as hub capacity constraints. They state that arc capacity may represent the number of available aircrafts for the airline company on that arc. For the computational experiments, they use the CAB data set that contains the data of 25 US cities with the highest traffic of airline passengers in 1970. Rodríguez-Martín and Salazar-González [28] consider capacities both on the arcs and hubs. They propose two exact solution methods. One of the methods is a branch-and-cut algorithm based on a two-level nested decomposition scheme that performs better than the other method based on standard Benders decomposition. They evaluate and compare these algorithms on instances with up to 25 commodities and 10 potential hubs. For large instances, they develop a hybrid heuristic based on solving a sequence of linear programs. They conduct computational experiments on instances with 25 commodities and a number of hubs ranging from 30 to 50 to test the performance of their proposed heuristic. They conclude that they are able to obtain near-optimal solutions. Lin et al. [29] study capacitated p-hub median problem in which they consider both node and arc capacity constraints. They make an application to a Chinese air cargo network with the number of nodes equivalent to 40.

The aforementioned formulations proposed for arc capacitated hub location problems assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. We call this approach as *the classical approach*. In section 3.1.1, we give arc capacitated HLP models in the literature based on the classical approach.

3.1.1 Arc Capacitated Hub Location Models based on Classical Approach

Bryan [26] is the first to introduce a model in which capacities are associated with the hub arcs. Sasaki and Fukushima [27], Rodríguez-Martín and Salazar-González [28] and Lin et al. [29] also incorporate arc capacities in HLPs, but the problem given by Bryan [26] is closer to the problem that we address. For that reason, we give the model proposed by Bryan [26] in this section. We also make some modifications to this model to make it address exactly our problem. We call this modified model CAMARC and use it in Section 3.2 to be able to compare our *proposed approach* with the *classical approach*.

Bryan [26] considers a complete network $G = (N, A)$ whose set of nodes, $N = \{1, . . . , n\}$, represents the set of origins and destinations nodes. Indices i,j,k,m refer to locations while q refers to different levels of costs corresponding to different flow volumes. Bryan [26] allows unit costs to vary with the amount flow traversing the hub arcs instead of assuming a fixed discount for the hub arcs. For this purpose, the model includes a piecewise linearization of a nonlinear cost function in which costs increase at a decreasing rate as flows increase. For this purpose, a_q and FC_q represent interhub discount factor (the slopes of the piece-wise lines) and fixed cost (the intercepts of the piece-wise lines) respectively. Bryan [26] denote w_{ij} as the demand of product from i to j for each pair of nodes $i,j \in N$ and c_{ij} as the transportation cost of a unit of flow between i and j. Bryan [26] defines X_{ijkm} as the proportion of flow from i to j that is routed via hubs k and m, respectively, R_{ikm} as the total flow originating from node i on the hub arc (k, m). Z_k takes the value 1 if node k is a hub, 0 otherwise and Y_{qkm} takes the value 1 if the flow on interhub link (k,m) will be charged FC_q, 0 otherwise.

With these definitions, the model proposed by Bryan [26] is given below:

$$\text{Min} \sum_i \sum_j \sum_k \sum_m w_{ij} \left(c_{ik} + c_{jm}\right) X_{ijkm} + \sum_q \sum_k \sum_m c_{km} \left(a_q R_{ikm} + FC_q\, Y_{qkm}\right) \quad \textbf{(3.1)}$$

s.t.

$$\sum_k Z_k = \text{p} \quad \textbf{(3.2)}$$

$$\sum_k \sum_m X_{ijkm} = 1 \qquad \forall\, i,j \quad \textbf{(3.3)}$$

$$\sum_m X_{ijkm} - Z_k \leq 0 \qquad \forall\, i,j,k \quad \textbf{(3.4)}$$

$$\sum_k X_{ijkm} - Z_m \leq 0 \qquad \forall\, i,j,m \quad \textbf{(3.5)}$$

$$\sum_q R_{qkm} = \sum_i \sum_j w_{ij}\, X_{ijkm} \qquad \forall\, k,m,k \neq m \quad \textbf{(3.6)}$$

$$R_{qkm} - Y_{qkm} \sum_i \sum_j w_{ij} \leq 0 \qquad \forall\, k,m,k \neq m \quad \textbf{(3.7)}$$

$$X_{kmkm} \geq Z_k + Z_m - 1 \qquad \forall\, k,m \quad \textbf{(3.8)}$$

$$\sum_q Y_{qkm} - X_{kmkm} = 0 \qquad \forall\, k,m,k \neq m \quad \textbf{(3.9)}$$

$$\sum_i R_{qkm} \leq CAP \qquad \forall\, k,m \quad \textbf{(3.10)}$$

$$X_{ijkm} \geq 0 \qquad \forall\, i,j,k,m \quad \textbf{(3.11)}$$

$$R_{qkm} \geq 0 \qquad \forall\, q,k,m \quad \textbf{(3.12)}$$

$$\text{Z}_\text{k} \in \{0,1\} \qquad \forall\, k \quad \textbf{(3.13)}$$

$$Y_{qkm} \in \{0,1\} \qquad \forall\, q,k,m \quad \textbf{(3.14)}$$

The objective function (3.1) minimizes the total transportation cost. Constraint (3.2) locates p hubs and Constraints (3.3) ensure that every pair of nodes *(i,j)* is allocated to a path via hub nodes k and m. Constraints (3.4) and (3.5) guarantee that the flow will not be routed via hubs k and m unless k and m are actually hubs. Constraints (3.6) calculate the total amount of flow traveling across each hub arc. When total transportation cost is calculated, the amount of flows on each hub arc is multiplied by the corresponding slope (a_q) and the fixed cost (FC_q) is then added. Constraints (3.7) ensure that the correct fixed cost is associated with its corresponding hub arc discount. Constraints (3.8) require two hubs to utilize their own hub arc when interacting with each other, while Constraints (3.9) say that exactly one Y_{qkm} must be equal one if both k and m are hubs. Constraints (3.8) and (3.9) force all hub arcs to be open and used. Constraints (3.10) state that the amount of flow traveling across a hub arc must be less than or equal to the capacity of the arc. Lastly, Constraints (3.11)−(3.14) define decision variables.

To obtain CAMARC, three main changes are made in the model of Bryan [26]. Specifically, (1) unit costs varying with the amount of flow traversing the hub arcs is replaced with fixed discount factor (α) for the hub arcs, (2) access arc capacities are incorporated in addition to hub arc capacities, and (3) complete hub-level network requirement is relaxed. Since we use a fixed discount factor, we do not use the decision variable Y_{qkm}. In CAMARC, three decision variables are used; (1) X_{ijkm}, (2) Z_k, and (3) R_{ikm}.

The model for the classical approach of MACHLP with the same definitions used in the formulation of Bryan [26] is given below:

Model CAMARC: The Model for the Classical Approach of MACHLP

$$\text{Min} \sum_i \sum_j \sum_k \sum_m w_{ij} \left(c_{ik} + c_{jm}\right) X_{ijkm} + \sum_i \sum_k \sum_m \alpha c_{km} R_{ikm} \quad \textbf{(3.15)}$$

s.t.

$$\sum_k Z_k = p \quad \textbf{(3.16)}$$

$$\sum_k \sum_m X_{ijkm} = 1 \qquad \forall\, i, j \quad \textbf{(3.17)}$$

$$\sum_m X_{ijkm} - Z_k \leq 0 \qquad \forall\, i, j, k \quad \textbf{(3.18)}$$

$$\sum_k X_{ijkm} - Z_m \leq 0 \qquad \forall\, i, j, m \quad \textbf{(3.19)}$$

$$\sum_i R_{ikm} = \sum_i \sum_j w_{ij} X_{ijkm} \qquad \forall\, k, m, k \neq m \qquad \textbf{(3.20)}$$

$$\sum_i R_{ikm} \leq cap_{km} \qquad \forall\, k, m \qquad \textbf{(3.21)}$$

$$\sum_j \sum_m w_{ij} X_{ijkm} \leq cap_{ik} \qquad \forall\, i, k \qquad \textbf{(3.22)}$$

$$\sum_i \sum_k w_{ij} X_{ijkm} \leq cap_{ik} \qquad \forall\, j, m \qquad \textbf{(3.23)}$$

$$X_{ijkm} \geq 0 \qquad \forall\, i, j, k, m \qquad \textbf{(3.24)}$$

$$R_{ikm} \geq 0 \qquad \forall\, i, k, m \qquad \textbf{(3.25)}$$

$$Z_k \in \{0,1\} \qquad \forall\, k \qquad \textbf{(3.26)}$$

The objective function (3.15) minimizes the total transportation cost. Constraint (3.16) locates p hubs and Constraints (3.17) ensure that every pair of nodes *(i,j)* is allocated to a path via hub nodes k and m. Constraints (3.18) and (3.19) guarantee that the flow will not be routed via hubs k and m unless k and m are actually hubs. Constraints (3.20) calculate the total amount of flow traveling across each hub arc. Constraints (3.21) state that the amount of flow traveling across a hub arc must be less than or equal to the capacity of the arc. Constraints (3.22) and (3.23) impose an upper limit on the flow on collection and distribution arcs respectively. Lastly, Constraints (3.24)−(3.26) define decision variables.

3.2 Comparison of the Hub Networks for Different Modeling Approaches

In this section, we investigate how the total cost, hub locations, and flow route between OD pairs change depending on the modeling approach used under different assumptions. We compare two modeling approaches: (1) *The classical approach*: Modeled network MN is complete and its distances satisfy the triangle inequality. (2) *The proposed approach*: MN is the same as the real-world network RealN that may be complete or non-complete. Before proceeding further we would like to clarify *the classical approach* more. To our knowledge, all studies on arc capacitated hub location problems assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. However, arc capacities on a non-complete RealN may not be easily incorporated when a complete MN is used because an arc in a complete MN may actually correspond to a shortest path consisting of several arcs with different capacities and not necessarily a single arc in RealN.

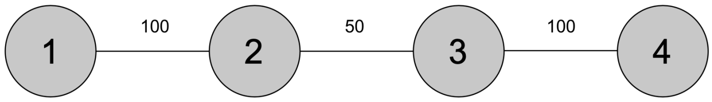

Figure 3.1 A non-complete network RealN with different arc capacities.

As an example, consider the non-complete network in Figure 3.1 where the numbers on the arcs represent the arc capacities. When a complete MN is created from this RealN by an algorithm (e.g., Floyd [7]), the arc (1,4) in MN actually corresponds to the shortest path consisting of arcs (1,2), (2,3), and (3,4) in RealN and hence the amount of flow that can be sent from node 1 to node 4 cannot exceed 50. In this case, to incorporate arc capacities in RealN into MN, one may be tempted to assign a capacity of 50 to arc (1,4) in MN. Similarly, a capacity of 50 may be assigned to arcs (1,3), (2,3), (2,4), (3,2), (3,1), (4,1), and (4,2) in MN. However, this assignment of arc capacities is not correct because this approach will allow 400 units to be sent between nodes 2 and 3 in MN, while the actual capacity is just 50 in RealN. In this regard, it is not easy to correctly represent the arc capacities in RealN with the current approach. Path-based formulation may be developed; however, finding all paths is computationally too expensive. For this reason, we adopt an approach where RealN is used as a part of MN, which allows us to easily incorporate arc capacities.

For comparison purposes, we use an 8-node non-complete network that consists of 8 cities in the Eagean region of Turkey as the nodes and the roads between neighboring cities as the arcs given in Figure 3.2. We assume that a discount factor of 0.7 is applied to the hub arc costs. We create different instances using different capacities for the arcs. We assume that all the arcs have the same capacity. We also assume that the shortest path arcs created for the complete network have the same capacity. In all instances, p=3 and w_{ij}'s are drawn from a uniform distribution with the interval $[10,30]$.

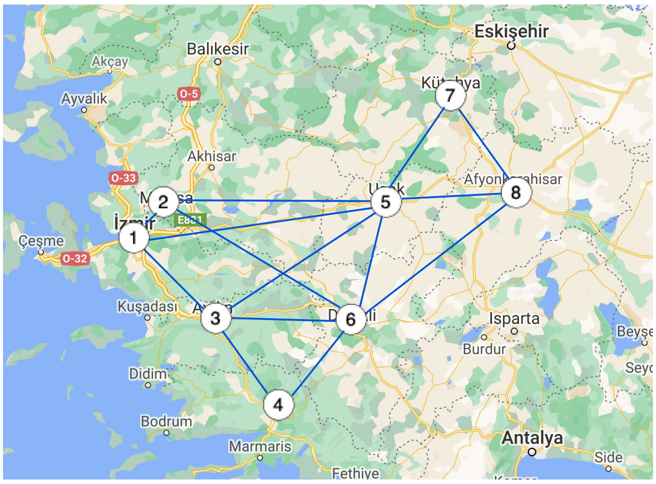

Figure 3.2 Non-complete transportation network consisting of 8 cities of Eagean Region of Turkey.

We use CAMARC to *represent the classical approach.* We remark that the classical model needs as MN a complete network whose distances satisfy the triangle inequality to work correctly. In this regard, we apply the Floyd's Algorithm [7] to the non-complete network to find all-pairs shortest path distances and construct a complete network to obtain a solution for the non-complete network using the classical model. *To represent the proposed approach*, we use our proposed model for MACHLP, whose details are given in Section 3.3. The proposed model can use any type of network, i.e., complete or non-complete, as MN. In this regard, the proposed model uses directly the non-complete network and complete network as MN. We consider two cases: (1) Case 1: All hub arcs are assumed to have the same capacity, (2) Case 2: All hub and access arcs are assumed to have the same capacity.

In Case 1, we impose arc capacities only on hub arcs. We determine the capacities after finding the optimal flow values on hub arcs when there are no arc capacities. Table 3.1 shows the objective function values and hub nodes obtained by MACHLP and CAMARC for different capacity values. Please note that we assign the same capacity to all arcs in each instance in Table 3.1. When there are no arc capacities and arc capacities are not restrictive, i.e., 70, the same optimal objective function values and hub sets are obtained by MACHLP and CAMARC. When arc capacities are restrictive, i.e., changing from 60 to 10, the optimal objective function values of MACHLP are better than those of CAMARC even though the same hub set is found by both models.

Table 3.1 Objective function values achieved and hub nodes located with the proposed Model MACHLP and Model CAM_ArcCap when hub arc capacities are imposed.

	MACHLP		CAMARC	
Capacity of each arc	Objective Function Value	Hub Nodes	Objective Function Value	Hub Nodes
No Capacity	230138	2,3,5	230138	2,3,5
70	231584	1,4,5	231584	1,4,5
60	231780	1,4,5	232008	1,4,5
50	232607	1,4,5	234042	1,4,5
40	234506	1,4,5	236762	1,4,5
30	237524	1,4,5	240822	1,4,5
20	242655	1,5,6	244771	1,5,6
10	246637	1,5,6	247847	1,5,6

When we examine the solution of both models, we observe that the difference in the objective function values results from the fact that CAMARC can use only direct paths between hubs while MACHLP can use alternative paths connecting hubs. As an example, consider the HLNs in Figure 3.3 obtained by both models when the hub arc capacities are set to 40. When hub arc capacities are 40, nodes 1,4, and 5 are chosen as hubs with both of the models. CAMARC use the shortest path distances to connect the hubs and create the HLN as given in Figure 3.3 (b). On the other hand, MACHLP can also use alternative paths connecting hubs, e.g., arcs (1,2) and (2,5) to connect the hubs 1 and 5 as given in Figure 3.3 (a). In this case, when the capacities of arcs (1,5) and (5,1) are not enough for the flow between hubs 1 and 5, MACHLP also use the capacities of the arcs (1,2) and (2,5) to reach from hub 1 to hub 5 or use the capacities of the arcs (5,2) and (2,1) to reach from hub 5 to hub 1.

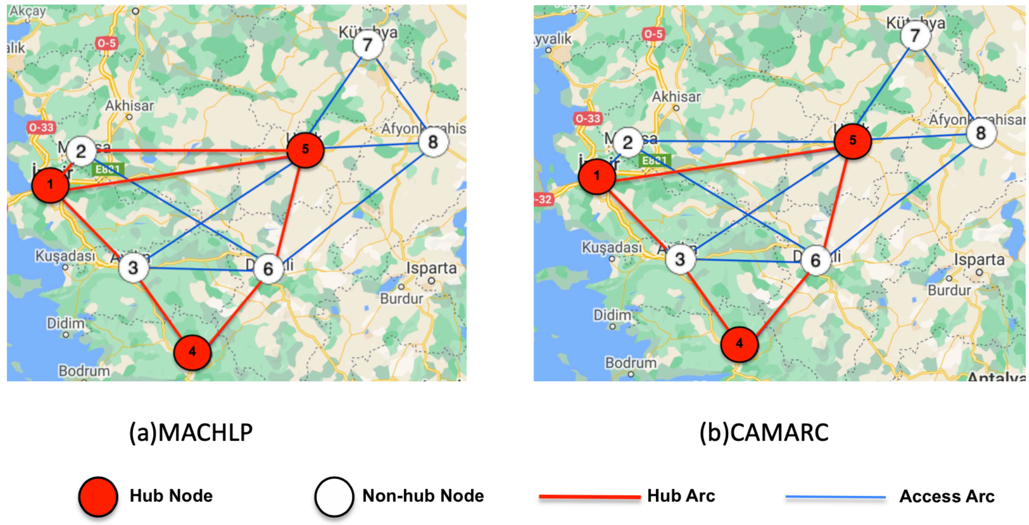

Figure 3.3 Hub Level Network (HLN) obtained using MACHLP and CAMARC when hub arc capacities are 40.

In case 2, we impose arc capacities on both hub arcs and access arcs and solve to optimality assigning capacities changing from 60 to 160 to all arcs. We assign the capacity values considering the solutions obtained when there are no constraints. Table 3.2 shows the objective function values and hub nodes found by MACHLP and CAMARC.

Table 3.2 Objective function values achieved and hub nodes located with the proposed Model MACHLP and Model CAM_ArcCap when hub and access arc capacities are imposed.

	MACHLP		CAMARC	
Capacity of each arc	Objective Function Value	Hub Nodes	Objective Function Value	Hub Nodes
No Capacity	224024	2,3,5	224024	2,3,5
160	224024	2,3,5	224024	2,3,5
140	224024	2,3,5	224884	2,3,5
120	230862	2,3,5	233278	2,3,5
100	237834	1,5,6	245537	2,3,5
80	247603	2,3,8	259089	3,5,6
60	Infeasible	NA	300790	3,5,6

When there are no arc capacities or when arc capacities are not restrictive, i.e., 160, both approaches give the same objective function value and hub set. When arc capacities change from 100 to 140, the optimal objective function values of MACHLP are better than those of CAMARC even though the same hub sets are found by both models. When arc capacities are 80 and 100, the optimal objective function values and hub sets obtained by both models are different. The objective function values of MACHLP are better than those of CAMARC. When the arc capacity is 60, MACHLP cannot find a feasible solution while CAMARC finds a solution. When we examine the solutions of both models, we observe that the differences result from the facts that CAMARC can use only direct paths while MACHLP can use alternative paths as in Case1 and CAMARC uses the capacities of the arcs more than once. As an example, consider the flows of commodity 7 originating from node 7 in Figure 3.4 obtained by both models when all the arc capacities are set to 120. Red arcs indicate the flow routes and numbers on the red arcs show the amount of flow. The total amount of flow originating from node 7 is 140. Since arc capacities are 120, MACHLP sends 120 units of flow from node 7 to hub 5 directly and sends the remaining 20 units of flow from node 7 to hub 5 on an alternative path (7,8) and (8,5) as shown in Figure 3.4 (a). On the other hand, CAMARC sends 120 units of flow from node 7 to hub 5 and sends the remaining 20 units of flow from node 7 to hub 2 directly. The direct path (7,2) in MN actually corresponds to the shortest path consisting of the arcs (7,5) and (5,2). Since the capacity of (7,5) is already used, CAMARC uses its capacity again while sending flows on (7,2).

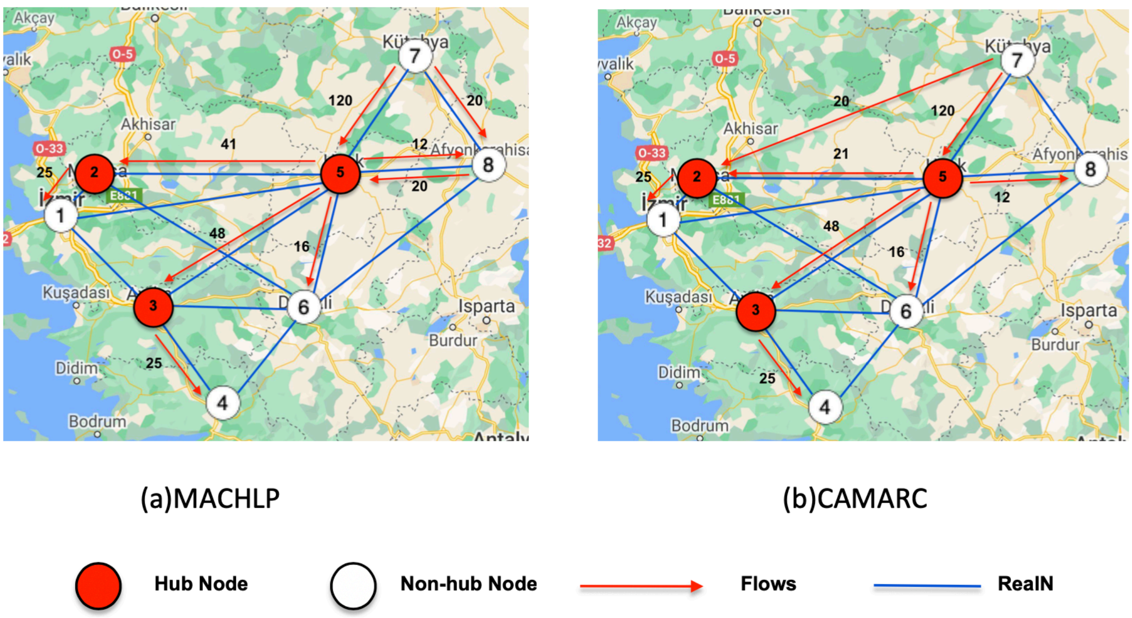

Figure 3.4 Flow routes of commodity 7 originating from node 7 using MACHLP and CAMARC when all arc capacities are 120

The examples in this section indicate that incorporating arc capacities in RealN is not possible with the classical approach even when all arcs are assigned the same capacity unless RealN is a complete network with arc distances satisfying the triangle inequality. On the other hand, the proposed approach allows incorporating capacities easily. When capacities are restrictive, the proposed approach may find better solutions. Moreover, in some cases, the classical approach may not satisfy the arc capacity constraints.

3.3 Problem Definition and Mathematical Formulation

We define *Multiple Allocation Arc Capacitated Hub Location Problem* (MACHLP) based on the modeling framework given by Akgün and Tansel [6] who propose a problem setting and modeling framework that allows non-complete or complete RealN with any cost structure to be used as MN for the multiple allocation p-hub median problem. The proposed problem setting and modeling framework are discussed in Section 2. However, for the purposes of easy reference and integrality of the section, we repeat the relevant parts here as well.

We consider an undirected and connected network $G = (N, E)$ (RealN) with node set $N = \{1, \dots, n\}$ and edge set E. Node set N has subsets as $S \subseteq N$ and $D \subseteq N$ that are supply (demand) nodes and demand (destination) nodes respectively. The same node can be the element of both S and D. A node $i \in S$ generates flows $w_{ij} > 0$ for some $j \in D$. We define $G^* = (N^*, E^*)$ as the subnetwork of G where inter-hub transportation is possible. E^* stands for the set of edges that can be used as hub arcs and N^* stands for the set of nodes that are incident to E^*. $H \subseteq N^*$ is the set of nodes that are appropriate to be chosen as hubs.

We define l_{ij} as the length of edge $\{i, j\}$ with $l_{ij} = l_{ji}$. The χ_{ij}, $\alpha_{ij,}$ and δ_{ij} stand for the cost of moving one unit of flow per unit length along the edge $\{i, j\}$ for collection, transfer, and distribution, respectively. To achieve economies of scale from inter-hub transportations, the cost of moving one unit of flow per unit length along an inter-hub edge is defined less than the cost of moving one unit of flow per unit length along collection and distribution edges as $\alpha_{ij} \leq \chi_{ij}$ and $\alpha_{ij} \leq \delta_{ij}$.

We formulate the problems using a three-layer network in which first, second and third layers represent collection/supply, transfer/hub and distribution/demand layers, respectively as Akgün and Tansel [6]. We construct the modeled network MN as $G_0 = (N_0, A_0)$ using the directed version of $G = (N, E)$, $G' = (N, A)$. We create the supply layer network $G_1 = (N_1, A_1)$ and the distribution layer network $G_3 = (N_3, A_3)$ by exactly copying $G' = (N, A)$. The hub layer network $G_2 = (N_2, A_2)$ is the subnetwork of G' that corresponds to $G^* = (N^*, E^*)$ with $N_m = \{m1, m2, \dots, mn\}$ and $A_m = \{(mi, mj): (i, j) \in A\}$ where $m = 1,2,3$ representing the layers of the network. We assume that in MACHLP, capacity constraints may not arise on all arcs of the network. For that reason, we define $A^{\text{cap}} \subseteq A_1 \cup A_2 \cup A_3$ as the set of arcs with capacity limits.

MACHLP aims to (1) select p nodes from hub set H, (2) determine the service routes between OD pairs that visit at least one hub node by satisfying the capacity requirement on the amount of allowable flow on the arcs A (collection arcs A_1, inter-hub arcs A_2 and distribution arcs A_3) such that total transportation cost is minimized with using a multiple allocation strategy.

We define w_{ij}'s as the flows with $i \in S$ and $j \in D$ sent from $1i \in N_1$ to $3j \in N_3$ through G_0. W_i is the total supply of commodity i at node $1i$ defined as $W_i = \sum_{j \in D} w_{ij}$. The $b_{\beta k}$ is the amount of supply/demand of commodity k at node $\beta \epsilon N_0$. F_β^{out} $\left(F_\beta^{in}\right)$ is the forward (inward) star of a node $\beta \in (N_1 \cup N_2 \cup N_3)$.

The unit cost of each arc c_{ij} is defined as;

$$c_{ij} = \begin{cases} \chi_{ij} \times l_{ij} & for\ (1i, 1j), (i, j) \in A \\ \alpha_{ij} \times l_{ij} & for\ (2i, 2j), (i, j) \in A^* \\ \delta_{ij} \times l_{ij} & for\ (3i, 3j), (i, j) \in A \\ 0 & for\ (1i, 2i)\ or\ (2i, 3i), i \in H \end{cases}$$

To incorporate arc capacities, we define cap_{ij} as the capacity on the amount of allowable flow on arc $(i, j) \in A^{cap}$.

The decision variables used in the formulation are (1) x_{ijk} that presents the amount of flow of commodity $k \in S$ in arc (i, j) and (2) y_{2i} that takes on the value of 1 when a hub is located at node $i \in H$ and 0 otherwise.

With these definitions, the proposed model for MACHLP is given below:

Model MACHLP: Model Multiple Allocation Arc Capacitated Hub Location Model

$$Z^* = \text{Min} \sum_{k \epsilon S} \sum_{(i,j) \epsilon A_0} c_{ij} x_{ijk} \tag{3.27}$$

$s.t.$ **(3.28)**

$$\sum_{j\in F_\beta^{out}} x_{\beta jk} - \sum_{j\in F_\beta^{in}} x_{j\beta k} = b_{\beta k} \qquad \beta \in (N_1 \cup N_2 \cup N_3), k \in S \qquad \textbf{(3.29)}$$

$$\sum_{i\in H} y_{2i} = p \qquad \textbf{(3.30)}$$

$$x_{(1i,2i)k} \le W_k y_{2i} \qquad i \in H, k \in S \qquad \textbf{(3.31)}$$

$$x_{(2i,3i)k} \le W_k y_{2i} \qquad i \in H, k \in S \qquad \textbf{(3.32)}$$

$$\sum_{k\in S} X_{ak} \le cap_a \qquad \forall\, a \in A^{cap} \subseteq A_1 \cup A_2 \cup A_3 \qquad \textbf{(3.33)}$$

$$x_{ijk} \ge 0 \qquad (i,j) \in A_0, k \in S \qquad \textbf{(3.34)}$$

$$y_{2i} \in \{0,1\} \qquad i \in H \qquad \textbf{(3.35)}$$

Objective function (17) together with constraints (18)-(23) constitute the formulation of multiple allocation p-hub median problem developed by Akgün and Tansel [6]. We add constraints (69) to the formulation of Akgün and Tansel [6] to develop the formulation of MACHLP. Constraints (69) state that the amount of flow traveling on an arc cannot exceed the capacity of that arc cap_a.

3.4 Proposed Solution Methodology

MACHLP is an NP-hard problem and is difficult to solve using standard optimization software. Computational studies indicate that CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic cannot find optimal solutions for problem instances defined on 81-node network with a run time of 24 h. Because we may have problems on larger networks in real-life applications, we propose a solution methodology based on Simulated Annealing (SA) for solving large-size problem instances.

SA algorithm was developed in 1953 by Metropolis et al. [89]. SA is an effective metaheuristic in solving combinational optimization problems and is commonly used to solve HLPs. Ernst and Krishnamoorthy [90] propose a solution methodology based on SA algorithm for single assignment p-hub median problem. For small-size problems, they propose a BB algorithm in which they derive an upper bound using their SA algorithm. They use their proposed algorithm for instances with 50 nodes. For the instances of larger problems with 100 and 200 nodes, they are able to obtain solutions using the SA algorithm. These instances are the ones for which they can not obtain any solution with exact methods. Ernst and Krishnamoorthy [57] develop heuristic algorithms based on SA algorithm and random descent (RDH) for capacitated single assignment p-hub median problem. They solve instances with up to 200 nodes. They compare the performance of the SA algorithm and that of RDH and show that SA generally performs slightly better than RDH for large problems. Abdinnour-Helm [91] implements a SA-based heuristic approach for the single assignment p-hub median problem. He solves instances with 80 nodes. Chen [92] proposes a hybrid heuristic based on the SA algorithm and tabu list for single allocation hub location problem. He conducts computational experiments using instances with up to 200 nodes. He can obtain optimal solutions for all small-scaled problems and for large-scaled problems obtain the best-known solutions with smaller CPU time. Jabalameli et al. [93] address the single allocation maximal covering hub location problem and propose an efficient SA-based heuristic to solve it. Computational results for the instances with up to 50 nodes prove the efficiency of the proposed heuristic both in terms of solution quality and CPU time. Ghaffarinasab et al. [94] present SA-based heuristic approaches for the competitive single and multiple allocation HLPs. They conduct computational experiments for the instances with 25 and 81 nodes. For all the instances, they can obtain optimal solutions. However, for the larger instances where the optimal solutions are not known, they cannot prove the optimality of the solutions obtained by the SA.

3.4.1 The Overall SA Procedure

The SA algorithm starts with an initial solution and initial temperature *T0*. The algorithm proceeds by moving from the current solution to a neighboring solution. In order to initiate the algorithm, we determine an initial solution by selecting the first p elements of the hub set H as hub nodes and the remaining nodes as non-hub nodes. This initial solution generation is a quick way to produce a feasible solution. This initial solution is set as the 'besthubset' at the beginning of the algorithm. To obtain the objective function value of the solution, we run MACHLP by fixing the values of decision variables corresponding to the selected hubs. This objective function value is set as the 'bestbound'. We use an operator called 'Swap_One_Hub' for generating neighboring solutions as in Ghaffarinasab et al. [94]. This operator randomly selects a hub node and a non-hub node in the current solution. The selected hub node becomes a non-hub and the selected non-hub node becomes a hub in the new, neighboring solution. In order to find the objective function value of the neighboring solution, we again use MACHLP by fixing the values of location variables corresponding to hub nodes. If the new solution is infeasible (the infeasibility may result from arc capacities), another solution is obtained by using the 'Swap_One_Hub' operator. If the objective function of the new solution is better than the previous solution, the current solution is replaced by the new solution. We compare the objective function value of the new solution with 'bestbound' and if it is better, the 'besthubset' and 'bestbound' are updated. If the objective function of the new solution is worse than the previous solution, we do not reject that solution directly. The algorithm calculates a probability as $e^{-\Delta E/T}$ where ΔE is the difference of objective function values between the current solution and the new solution and T is the current temperature. We generate a number between 0 and 1 randomly. If the generated number is less than $e^{-\frac{\Delta E}{T}}$ we accept the new solution and update the temperature as $T = \delta T$ where δ is the coefficient that controls the cooling schedule. In other words, with the probability of $e^{-\frac{\Delta E}{T}}$, the algorithm accepts the new solution. As the algorithm proceeds, the temperature is reduced slowly. As the temperature is reduced, the probability of accepting the worse solutions decreases. In the early iterations of the algorithm, there is a high probability to accept a worse solution in order not to get stuck in local optimum solution. In the final stages, there is less probability to accept a bad solution so that the algorithm converges to a good solution. The critical point in the algorithm is not to reduce the temperature too fast so that you can get stuck in local optimum solution or too slow so that you can not reach good quality solutions. The algorithm continues similarly until the stopping criterion is reached. In this study, we use TimeLimit as the stopping criterion. However, a final temperature or an iteration count may be used as well.

Algorithm SA-Heuristic:

Step 1: (Initialization)

Generate a random initial solution for the location of hubs.

Set an initial temperature *T0*.

Set TimeLimit.

Set the location of the hubs as 'besthubset'

Set $h = 0$

Step 2: Run MACHLP (3.27)−(3.35) with the values of location decision variables fixed. Set this objective function value as 'bestbound'.

Step 3: Use Swap_One_Hub operator: First, select a hub node and a non-hub node in the current solution and assign the selected hub node as non-hub node and the selected non-hub node as a hub node in the new solution.

Step 4: Run MACHLP (3.27)−(3.35) with the values of location decision variables fixed**.**

If MACHLP is infeasible go to Step 3, otherwise;

Compare this objective function value with the objective function value of the previous solution.

If it is better, accept the new solution.

If the objective function of the new solution is better than 'bestbound', update 'bestbound', 'besthubset', and the temperature as $T=\delta T$, set $h=h+1$, go to Step3, otherwise; update the temperature as $T=\delta T$, set $h=h+1$, go to Step3 without updating 'bestbound' and 'besthubset'

If it is worse, keep the 'bestbound' and 'besthubset'. Calculate $e^{-\Delta E/T}$ and generate a number between 0 and 1 randomly.

If the generated number < $e^{-\Delta E/T}$ accept the new solution, update the temperature as $T=\delta T$, set $h=h+1$, go to Step3, otherwise; reject the new solution, keep the existing solution, update the temperature as $T=\delta T$, set $h=h+1$, go to Step3.

Step 5: Repeat Steps 3-4 until the '*Timelimit'* is reached.

3.5 Computational Experiments

We conduct computational experiments to test the performance of the proposed models and solution methodologies. We test the performance of MACHLP using CPLEX, Gurobi, and Gurobi with NoRel Heuristic that are known to be efficient solvers for the MIP problems. When Gurobi is used with NoRel Heuristic, NoRel heuristic first tries to find a high-quality feasible solution in the allocated time and then Gurobi implements the branch and cut algorithm with the feasible solution found by the NoRel heuristic. Gurobi with NoRel Heuristic can be useful on models where root relaxation is quite expensive since since it searches for high-quality feasible solutions before solving the root relaxation [82]. We observe that Gurobi with NoRel Heuristic finds much better feasible solutions than Gurobi for all instances. For that reason, we do not present the results obtained by Gurobi and only present the results obtained by CPLEX and Gurobi with NoRel Heuristic.

We define test problems on TR81, PMED100, PMED200, PMED300, and PMED400 networks. TR81 defined by Akgün and Tansel [6] is the non-complete transportation network of Turkey including all 81 cities of Turkey. The edges on TR81 are defined only between adjacent cities. The length of the edges on TR81 are assumed to be the direct distances from the high-way transportation network of Turkey. PMED100 through PMED400 are the non-complete networks used for the p-median problem instances (e.g., Beasley [84]) with the numbers indicating the number of nodes. Different test problems are created on the networks by changing p and *cap(i,j)* where $E^* = E, N^* = N$ and $H \subseteq N^*$. For all the test problems, w_{ij} is uniformly distributed with the interval (10,30). For all arcs, χ_{ij} and δ_{ij} are taken as 1 whereas α_{ij} is taken as 0.7. In all problems, $S = D = N$.

To determine A^{cap}, the set of arcs with limited capacities, we use the flow values on the arcs in the best solution obtained by running CPLEX to solve MACHLP without capacity constraints for 24 h. We sort the arcs in each layer (collection, hub, distribution) considering the flow values from the highest to the lowest and set the first 20% of the arcs on the sorted list as A^{cap}. The flow values of the arcs in A^{cap} are considered as an upper limit on the amount of flow traversing the corresponding arcs, which we define as *maxarccap.* We set the arc capacities of the corresponding arcs, i.e., cap_{ij}, 20%, 50%, and 80% of *maxarccap* to obtain different problem instances.

We code the models and the algorithms using GAMS and conduct the experiments on a PC with 3.6 GHz Intel Core i7-7700CPU processor and 32 GB of RAM for TR81, PMED100, PMED200, PMED300 instances and on a server with Intel® Xeon® CPU E5-2683 V4 @ 2.1 GHz 64 core processor and 256 GB of RAM for PMED400 instances due to memory requirements.

3.5.1 Computational Experiments for MACHLP using CPLEX and Gurobi with NoRel Heuristic

We conduct computational experiments for two different cases. In the first case, we impose arc capacities only on the hub arcs. In the second case, we impose arc capacities both on the hub arcs and access arcs. In both cases, the capacitated arcs are determined as described above. Table 3.3 and Table 3.4 give the results obtained by solving MACHLP using CPLEX and Gurobi with NoRel heuristic for these cases, respectively. The runtime of all instances is 24 h. When we use Gurobi with NoRel heuristic, we assign 12 h for the NoRel heuristic. After 12 h, Gurobi starts the branch and cut algorithm with the feasible solution found by NoRel heuristic and uses the remaining 12 h for branch and cut. Different time limits are also tried for NoRel heuristic; however, we can obtain high-quality solutions with this time setting.

In the tables, we present the runtime (T) in CPU secs, the lower bound (LB), the objective function value of the best integer solution at the end of runtime (BP) obtained by CPLEX or Gurobi with NoRel heuristic, and the relative optimality gap (*Gap%*) between LB and BP, and the best integer solution achieved either by CPLEX or Gurobi with NoRel heuristic (BP^*). In the tables, **bold** and *italic* values indicate the same or better **BP** and *LB* values for each instance.

Table 3.3 shows that Gurobi with NoRel heuristic mostly finds better BP values than CPLEX for TR81, PMED100, and PMED200 instances whereas CPLEX mostly finds better BP values than Gurobi with NoRel heuristic for PMED300 and PMED400 instances. The average gap for the small-size instances (TR81 and PMED100) is ranging from 3.7% to 7.5% whereas the average gap for the large-size instances (PMED200, PMED300 and PMED400) is ranging from 14.4% to 23.9%. As the network size gets larger, the number of capacitated arcs increases and arc capacities become more restrictive, resulting in higher optimality gaps.

Table 3.4 shows that Gurobi with NoRel heuristic mostly finds better BP values than CPLEX for all instances except PMED300 instances. The average gap for the small-size instances (TR81 and PMED100) is ranging from 7.2% to 13.2% whereas the average gap for the large-size instances (PMED200, PMED300 and PMED400) is ranging from 22.5% to 69.1%. The results show that, as the network size gets larger and as the number of arcs with capacities increases, i.e., when we impose capacities on both access and hub arcs rather than only on hub arcs, optimality gaps increase.

Table 3.3 Test results for MACHLP using CPLEX and Gurobi with NoRel heuristic with hub arc capacities.

Pr. Id	Network	|N|	p	Decrease in maxarccap (%)	T (secs)	CPLEX LB	CPLEX UB	CPLEX Gap (%)	Gurobi with NoRel Heuristics LB	Gurobi with NoRel Heuristics UB	Gurobi with NoRel Heuristics Gap (%)	BP*
1	TR81	81	3	20	86400	*102945951*	**112890530**	9.7	98772886	**112890530**	12.5	112890530
2		81	3	50	86400	*102811383*	**113092521**	10.0	98923941	**113092521**	12.5	113092521
3		81	3	80	86400	*102848053*	**113420833**	10.3	98846960	113899711	13.2	113420833
4		81	5	20	86400	*96071305*	101824824	5.7	93081273	**101630907**	8.4	101630907
5		81	5	50	86400	*96123674*	102076752	5.8	93438095	**101838411**	8.2	101838411
6		81	5	80	86400	*96452760*	102943176	8.0	93598954	**102286195**	8.4	102286195
7		81	8	20	86400	*91001266*	93859641	3.1	89535609	**93859641**	4.6	93859641
8		81	8	50	86400	*91264410*	**94832303**	3.8	89744040	94896754	5.4	94832303
9		81	8	80	86400	*92114134*	96205262	4.3	90662508	**96065406**	5.6	96065406
10		81	10	20	86400	*89256910*	**90996970**	1.9	87732680	**90996970**	3.5	90996970
11		81	10	50	86400	*89234894*	91929070	2.9	87875082	**91612434**	4.0	91612434
12		81	10	80	86400	*89763880*	93066574	3.6	88583710	**92807614**	4.5	92807614
							Average	5.4		Average	7.5	
13	PMED100	100	3	20	86400	*31782543*	**32930289**	3.5	30005567	**32930289**	8.8	32930289
14		100	3	50	86400	*31858910*	**32939139**	3.3	30155918	**32939139**	8.4	32939139
15		100	3	80	86400	*32004919*	**32969502**	2.9	30080649	**32969502**	8.7	32969502
16		100	5	20	86400	*28909200*	**30622886**	5.6	28486360	**30622886**	6.9	30622886
17		100	5	50	86400	*28922934*	30724600	5.9	28695781	**30706921**	6.5	30706921
18		100	5	80	86400	*28962741*	30837964	6.1	28777851	**30800113**	6.5	30800113
19		100	8	20	86400	*27747579*	**28412960**	2.3	27376398	**28412960**	3.6	28412960
20		100	8	50	86400	*27776810*	**28638604**	3.0	27328632	**28638604**	4.5	28638604
21		100	8	80	86400	*27855993*	**28891436**	3.6	27456902	**28891436**	4.9	28891436
22		100	10	20	86400	*27087666*	**27715709**	2.3	26737125	**27715709**	3.5	27715709
23		100	10	50	86400	*27119451*	28007152	3.2	26730025	**27964398**	4.4	27964398
24		100	10	80	86400	*27243841*	28272424	3.6	26897481	**28227890**	4.7	28227890
							Average	3.7		Average	5.9	
25	PMED200	200	3	20	86400	68757637	**88409107**	22.2	66748355	**88409107**	24.5	88409107
26		200	3	50	86400	68554486	87770593	21.9	*68828454*	**82761876**	16.8	82761876
27		200	3	80	86400	*68774183*	**89375366**	23.1	68554153	**89375366**	23.3	89375366
28		200	5	20	86400	*66116031*	85524862	22.7	66076587	**76824557**	13.9	76824557
29		200	5	50	86400	*66235691*	81788733	19.3	65965218	**77140939**	14.4	77140939
30		200	5	80	86400	*66184537*	85942385	23.0	66183518	**77175834**	14.2	77175834
31		200	8	20	86400	*64087284*	77909694	17.7	63766287	**71409434**	10.7	71409434
32		200	8	50	86400	*64066832*	88458636	27.6	63865576	**71397604**	10.5	71397604
33		200	8	80	86400	63566448	73265967	13.2	*63887723*	**72347091**	11.6	72347091
34		200	10	20	86400	*62581616*	72591859	22.2	61531446	**69552738**	11.5	69552738
35		200	10	50	86400	*62827030*	72583918	19.4	62698139	**70202498**	10.6	70202498
36		200	10	80	86400	*62940237*	70732885	11.0	62734262	**70451517**	10.9	70451517
							Average	20.2		Average	14.4	
37	PMED300	300	3	20	86400	100039256	**126904538**	21.2	*101162865*	130958127	22.7	126904538
38		300	3	50	86400	99795546	**126904538**	21.4	*101309492*	127442389	20.5	126904538
39		300	3	80	86400	99942282	**126904538**	21.3	*101140916*	129265963	21.8	126904538
40		300	5	20	86400	*98368346*	119755392	17.9	98332583	**119295140**	17.6	119295140
41		300	5	50	86400	98397675	119755392	17.8	*98583762*	**119288148**	17.4	119288148
42		300	5	80	86400	96961998	**119783476**	19.1	*98625828*	122740251	19.6	119783476

43		300	8	20	86400	95213553	**112850544**	15.6	*96909585*	113591891	14.6	112850544
44		300	8	50	86400	95700525	**112850544**	15.2	*96829462*	113434988	14.6	112850544
45		300	8	80	86400	95166765	**112591330**	15.5	*97035546*	114366263	15.2	112591330
46		300	10	20	86400	92057452	112126767	17.9	*95702271*	**109773739**	12.8	109773739
47		300	10	50	86400	94287150	**110088775**	14.4	*95440976*	110331974	13.5	110088775
48		300	10	80	86400	94414605	**110350594**	14.4	*95778008*	112492561	14.8	110350594
							Average	17.6		Average	17.0	
49	PMED400	400	3	20	86400	128655781	**172247869**	25.3	*130632203*	178185447	26.6	172247869
50	PMED400	400	3	50	86400	129217406	172247869	25.0	*131114396*	**162536528**	19.3	162536528
51	PMED400	400	3	80	86400	128778160	**171645514**	25.0	*131135000*	188758266	30.5	171645514
52	PMED400	400	5	20	86400	126078000	**159291000**	20.9	*128089009*	165348673	22.5	159291000
53	PMED400	400	5	50	86400	126068781	**158639802**	20.5	*127495090*	166888998	23.6	158639802
54	PMED400	400	5	80	86400	*127277784*	**159891522**	20.4	126513881	168055183	24.7	159891522
55	PMED400	400	8	20	86400	*124825000*	**150321000**	17.2	124041193	164813376	24.7	150321000
56	PMED400	400	8	50	86400	*125458768*	**151895521**	17.4	123534699	157033380	21.3	151895521
57	PMED400	400	8	80	86400	*124936045*	**149557957**	16.5	123155435	157608234	21.8	149557957
58	PMED400	400	10	20	86400	123019000	**147057000**	16.4	*124039776*	173500823	28.5	147057000
59	PMED400	400	10	50	86400	*123428000*	**148308000**	16.8	122244814	149314586	18.1	148308000
60	PMED400	400	10	80	86400	*124373734*	**149717000**	16.9	122096730	163732832	25.4	149717000
							Average	19.8		Average	23.9	

Table 3.4 Test results for MACHLP using CPLEX and Gurobi with NoRel heuristic with hub and access arc capacities.

Pr. Id	Network	|N|	p	Decrease in maxarccap (%)	T (secs)	CPLEX			Gurobi with NoRel Heuristics			BP*
						LB	UB	Gap (%)	LB	UB	Gap (%)	
61	TR81	81	3	20	86400	*103526953*	**114914918**	11.0	99655975	115281313	13.5	114914918
62	TR81	81	3	50	86400	*104416477*	117071754	12.1	99734529	**115694288**	13.7	115694288
63	TR81	81	3	80	86400	*105220416*	118046785	12.2	100596696	**117011966**	14.0	117011966
64	TR81	81	5	20	86400	*96071305*	**101824824**	5.7	93506453	102965309	9.1	101824824
65	TR81	81	5	50	86400	*96466020*	105117294	8.2	93339294	**104498908**	10.6	104498908
66	TR81	81	5	80	86400	*98523400*	**106022593**	7.1	94928043	**106022593**	10.4	106022593
67	TR81	81	8	20	86400	*91210840*	95748062	4.7	89748708	**94742442**	5.2	94742442
68	TR81	81	8	50	86400	*91849206*	97417496	5.7	89971897	**96915551**	7.1	96915551
69	TR81	81	8	80	86400	*93841975*	99094584	7.0	92086213	**98836098**	6.8	98836098
70	TR81	81	10	20	86400	*89024967*	92040568	3.3	87820889	**91981590**	4.5	91981590
71	TR81	81	10	50	86400	*89666623*	94601963	5.2	88172666	**93892359**	6.0	93892359
72	TR81	81	10	80	86400	*91816969*	**95753742**	4.1	90308654	**95753742**	5.6	95753742
								7.2			8.8	
73	PMED100	100	3	20	86400	*30778970*	34287262	10.2	29973516	**34157981**	12.3	34157981
74	PMED100	100	3	50	86400	*30247451*	36543053	17.2	29905880	**36071269**	17.0	36071269
75	PMED100	100	3	80	86400	*32350494*	**37884305**	14.6	30858409	**37884305**	18.5	37884305
76	PMED100	100	5	20	86400	*28924711*	31556850	8.3	28734286	**31556347**	8.9	31556347
77	PMED100	100	5	50	86400	*29114625*	33609757	13.4	28430677	**33301518**	14.6	33301518
78	PMED100	100	5	80	86400	*30989826*	**35878431**	13.6	29872326	**35878431**	16.7	35878431
79	PMED100	100	8	20	86400	*27868285*	29394313	5.2	27325774	**29253295**	6.5	29253295
80	PMED100	100	8	50	86400	*27619903*	30862630	10.5	27594765	**30844544**	10.5	30844544
81	PMED100	100	8	80	86400	*29172254*	**32802869**	11.1	28645243	32988664	13.1	32802869
82	PMED100	100	10	20	86400	*27242125*	**28335970**	3.9	25145322	**28335970**	11.2	28335970
83	PMED100	100	10	50	86400	26877733	**29744519**	9.6	*28807415*	33622974	14.3	29744519

84		100	10	80	86400	28275715	**32014511**	11.7	*31006734*	36669754	15.4	32014511
								10.7			13.2	
85		200	3	20	86400	*68798258*	92679549	25.8	68790221	**87950587**	21.7	87950587
86		200	3	50	86400	68249315	113305564	39.8	*68660539*	**96173074**	28.6	96173074
87		200	3	80	86400	*70062038*	107833727	35.0	69779371	**94955419**	26.5	94955419
88		200	5	20	86400	*66584867*	95067609	30.0	66193354	**85427037**	22.5	85427037
89		200	5	50	86400	*66901631*	94079140	28.9	66427391	**85992946**	22.7	85992946
90	PMED200	200	5	80	86400	*68534169*	92759870	26.1	67828920	**92082116**	26.3	92082116
91		200	8	20	86400	*65240463*	**76882178**	15.1	63893749	77472013	17.5	76882178
92		200	8	50	86400	*64581472*	88040609	26.7	64365814	**84224024**	23.5	84224024
93		200	8	80	86400	*67389398*	91566807	26.4	66574756	**88885932**	25.1	88885932
94		200	10	20	86400	*63171148*	78523829	19.6	62745677	**73763549**	14.9	73763549
95		200	10	50	86400	*63716795*	95711583	33.4	63318400	**79379925**	20.2	95711583
96		200	10	80	86400	*66401974*	94781986	29.9	65560105	**82581387**	20.6	82581387
								28.0			22.5	
97		300	3	20	86400	99066880	**140643156**	29.6	*100551924*	158715823	36.6	140643156
98		300	3	50	86400	*100029467*	**141991514**	29.6	98958009	181219781	45.3	141991514
99		300	3	80	86400	*102853759*	**154420261**	33.4	102438975	162566454	36.9	154420261
100		300	5	20	86400	97481250	**150142080**	35.1	*97900651*	150416020	34.9	150142080
101		300	5	50	86400	*98308505*	**142235824**	30.9	97137470	147335327	34.0	142235824
102	PMED300	300	5	80	86400	*100726631*	**141216710**	28.7	98824717	158495066	37.6	141216710
103		300	8	20	86400	94528300	**124436000**	24.0	*94741260*	149052657	36.4	124436000
104		300	8	50	86400	*95894711*	**129374170**	25.9	94863801	147125287	35.5	129374170
105		300	8	80	86400	*98493520*	**132672564**	25.8	96920872	156144691	37.9	132672564
106		300	10	20	86400	93227083	**116758410**	20.2	*94461289*	141942302	33.5	116758410
107		300	10	50	86400	94138066	**125892543**	25.2	*94182626*	126429640	25.5	125892543
108		300	10	80	86400	*96985961*	**126327965**	23.2	94911748	133525381	28.9	126327965
								27.6			35.2	
109		400	3	20	86400	127706003	3608530345	96.5	*129101086*	**243230077**	47.2	243230077
110		400	3	50	86400	126122232	2637792689	95.2	*130190004*	**285158797**	54.3	285158797
111		400	3	80	86400	126122232	2637792689	95.2	*132504697*	**236120100**	43.8	236120100
112		400	5	20	86400	*126805640*	**172253818**	26.4	125980900	202268343	37.7	172253818
113		400	5	50	86400	*128192803*	**206850078**	38.0	126630071	218983603	42.1	206850078
114	PMED400	400	5	80	86400	*131554972*	2569061646	94.9	127401700	**250749067**	52	250749067
115		400	8	20	86400	*124446872*	200542266	37.9	123257052	**193667054**	36.3	193667054
116		400	8	50	86400	*126735799*	**180416522**	29.8	124027978	216698126	42.7	180416522
117		400	8	80	86400	*131433916*	4021263142	96.7	126332881	**210938589**	40.1	210938589
118		400	10	20	86400	*123852655*	**166750263**	25.7	122092256	189330359	35.8	166750263
119		400	10	50	86400	*122692282*	4248884252	97.1	122421613	**196846174**	37.8	196846174
120		400	10	80	86400	*129019523*	3959976011	96.7	122798697	**218859960**	43.8	218859960
								69.1			42.8	

3.5.2 Computational Experiments with the Proposed SA-based Solution Methodology

We find the initial solution and the corresponding objective function value as described in Section 3.4. We set the initial temperature *T0* to 10000000 after observing the quality of solutions with a set of trial tests. *T0* affects the probability $e^{-\Delta E/T}$, the probability of accepting worse solutions. ΔE is the difference of objective function values between the current solution and the new solution and this difference is mostly in the values of millions. If T is a small number, the probability $e^{-\Delta E/T}$ becomes very small. However, in the early iterations of the algorithm, we would like it to accept the worse solutions not to get stuck in local optimum solution. For that reason, instead of small values of $T0$ we use a large value 10000000 that gives us good quality solutions. For cooling schedule δ, we use two different values, 0.99 and 0.95. We set TimeLimit to 24 h with a focus on high-quality solutions rather than shorter solution times considering the strategic nature of MACHLP. However, we observe that the best solutions are found in about 12 h for most of the instances. Figure 3.5 shows how the UB values change as time passes for the PMED300 test instances defined with arc capacities both on hub and access arcs.

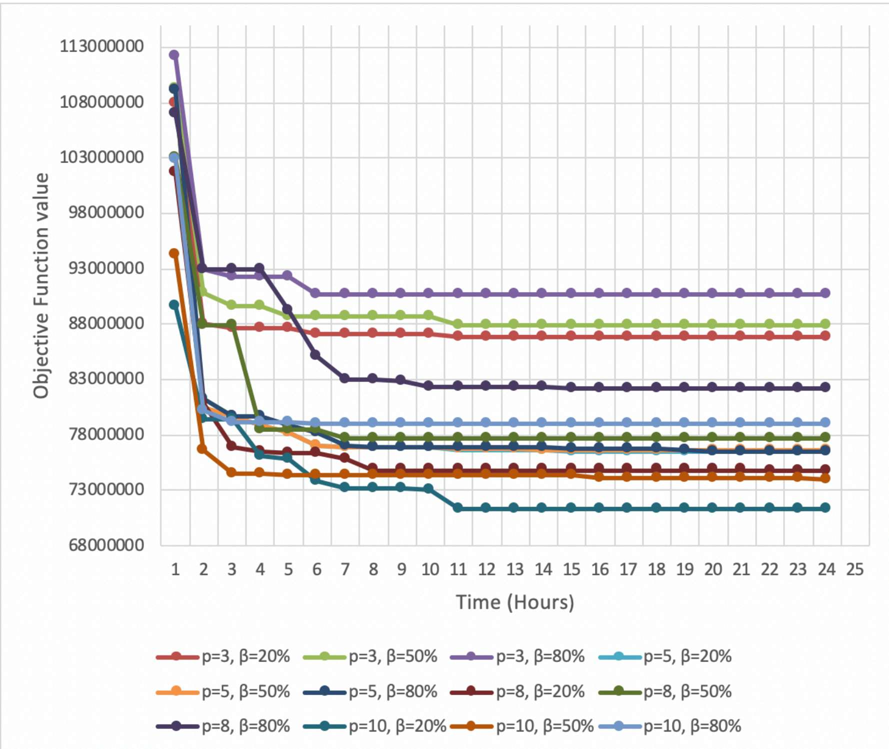

Figure 3.5. UB values achieved with the proposed SA based solution methodology as time passes for the test instances on PMED300.

Table 3.5 (3.6) presents the results of the proposed SA-based heuristics for different test instances of MACHLP with only hub arc capacities (with both hub and access arc capacities). In the tables, we give GapHeur (%) defined as 100×(UB - BP^*)/BP^* in order to compare the heuristic solution UB to BP*, the best solution found by either CPLEX or Gurobi with NoRel heuristic. A *positive (negative)* value indicates that the UB achieved by the heuristic algorithm is *worse (better)* than BP*. Instances with **bold UB values** are the ones for which the heuristic can either find the same solution or a better solution than CPLEX or Gurobi with NoRel heuristic.

The results indicate that average *GapHeur* values with the cooling factor δ=0.95. for PMED100 and PMED300 instances when only hub arc capacities are imposed (Table 3.5). For all other test instances in Table 3.5 and Table 3.6, average *GapHeur* values with δ=0.95 are better than those with δ=0.99. In this regard, we will continue our analysis with the parameter setting δ=0.95.

Table 3.5 shows that, the proposed heuristic finds the same solution as CPLEX or Gurobi with NoRel heuristic for 13 instances out of 24 TR81 and PMED100 instances. However for PMED200, PMED300 and PMED400 instances, the proposed heuristic finds a solution equivalent to or better than BP^* for 31 instances out of 36. For PMED400 instances, average GapHeur is the lowest with a value of −1.9% which shows that as the network size gets larger, the heuristic can find solutions better than those found by CPLEX or Gurobi with NoRel heuristic.

Table 3.6 indicates that, the proposed heuristic finds the same solution as CPLEX or Gurobi with NoRel heuristic for 17 instances out of 24 TR81 and PMED100 instances. However for PMED200, PMED300 and PMED400 instances, the proposed heuristic finds a solution better than BP^* for all the 36 instances except two of them. For PMED200 instances, GapHeur values range from −1.3% to −22.7% with an average of −8.4%. The heuristic can find a better solution for all the PMED200 instances. For PMED200 instances, GapHeur values range from –0.2% to −13% with an average of −4.7%. The heuristic can find a better solution for all the PMED300 instances like PMED200 instances. For PMED400 instances, GapHeur values change from 7.6% to −30.3% with an average of −10.8%. The heuristic cannot find a better solution for two instances (Pr. Id 116 and 11) out of 12 PMED400 instances. As network size increases, the proposed heuristic is able to find much better solutions than the solutions achieved with CPLEX or Gurobi with NoRel heuristic.

The results show that, as the network size gets larger and as the number of arcs with capacities increases, i.e., when we impose capacities on both access and hub arcs rather than only on hub arcs, CPLEX or Gurobi with NoRel heuristic find a solution with high optimality gaps. On the other hand, the proposed heuristic can find solutions either close to or better than those found by CPLEX or Gurobi with NoRel heuristic, which shows the efficiency of the proposed heuristic to find high-quality solutions.

Table 3.5 Test results for MACHLP using the proposed SA-based heuristics with hub arc capacities (T=24h)

Pr. Id	Network	|N|	p	Decrease in maxarccap (%)	BP*	UB when δ=0.99	GapHeur (%)	UB when δ=0.95	GapHeur (%)
1	TR81	81	3	20	112890530	112988100	0.1	112988100	0.1
2		81	3	50	113092521	113217000	0.1	113217000	0.1
3		81	3	80	113420833	**113472500**	**0.0**	**113472500**	**0.0**
4		81	5	20	101630907	**101591900**	**0.0**	**101591900**	**0.0**
5		81	5	50	101838411	102345000	0.5	102345000	0.5
6		81	5	80	102286195	**102286200**	**0.0**	**102286200**	**0.0**
7		81	8	20	93859641	98188900	4.6	**93859640**	**0.0**
8		81	8	50	94832303	98488100	3.9	95034380	0.2
9		81	8	80	96065406	99434450	3.5	**96065410**	**0.0**
10		81	10	20	90996970	91095720	0.1	91495800	0.5
11		81	10	50	91612434	94760650	3.4	94760650	3.4
12		81	10	80	92807614	92866590	0.1	92866590	0.1
						Max	4.6	Max	3.4
						Min	0.0	Min	0.0
						Average	1.4	Average	0.4
13	PMED100	100	3	20	32930289	**32930289**	**0.0**	**32930289**	**0.0**
14		100	3	50	32939139	**32939139**	**0.0**	**32939139**	**0.0**
15		100	3	80	32969502	**32969502**	**0.0**	**32969502**	**0.0**
16		100	5	20	30622886	32106400	4.8	**30622886**	**0.0**
17		100	5	50	30706921	32141520	4.7	**30706921**	**0.0**
18		100	5	80	30800113	32188456	4.5	**30800113**	**0.0**
19		100	8	20	28412960	**28412960**	**0.0**	**28412960**	**0.0**
20		100	8	50	28638604	**28638604**	**0.0**	31653670	10.5
21		100	8	80	28891436	29635738	2.6	31771380	10.0
22		100	10	20	27715709	27775585	0.2	**27715709**	**0.0**
23		100	10	50	27964398	28007542	0.2	27988478	0.1
24		100	10	80	28227890	29801188	5.6	30115991	6.7
						Max	5.6	Max	10.5
						Min	0.0	Min	0.0
						Average	1.9	Average	2.3
25	PMED200	200	3	20	88409107	**82761876**	**-6.4**	**82761876**	**-6.4**
26		200	3	50	82761876	**82761876**	**0.0**	**82761876**	**0.0**
27		200	3	80	89375366	**82761876**	**-7.4**	**82761876**	**-7.4**
28		200	5	20	76824557	**76514167**	**-0.4**	**76514167**	**-0.4**
29		200	5	50	77140939	**76533753**	**-0.8**	**76533753**	**-0.8**
30		200	5	80	77175834	**76565576**	**-0.8**	**76565576**	**-0.8**
31		200	8	20	71409434	**71334036**	**-0.1**	**71334036**	**-0.1**
32		200	8	50	71397604	**71371852**	**0.0**	**71371852**	**0.0**
33		200	8	80	72347091	**71437881**	**-1.3**	**71437881**	**-1.3**
34		200	10	20	69552738	69639444	0.1	**69525782**	**0.0**
35		200	10	50	70202498	**69902498**	**-0.4**	**69626947**	**-0.8**
36		200	10	80	70451517	**69792291**	**-0.9**	**69792291**	**-0.9**
						Max	0.1	Max	0.0
						Min	-7.4	Min	-7.4
						Average	-1.5	Average	-1.6

37		300	3	20	126904538	**126904538**	**0.0**	**126904538**	**0.0**
38		300	3	50	126904538	**126904538**	**0.0**	128159829	1.0
39		300	3	80	126904538	**126904538**	**0.0**	**126904538**	**0.0**
40		300	5	20	119295140	**118513751**	**-0.7**	119374121	0.1
41		300	5	50	119288148	**118513751**	**-0.6**	119374121	0.1
42	PMED300	300	5	80	119783476	**119092250**	**-0.6**	**118866634**	**-0.8**
43		300	8	20	112850544	**112281042**	**-0.5**	**111644121**	**-1.1**
44		300	8	50	112850544	**112288763**	**-0.5**	**111641119**	**-1.1**
45		300	8	80	112591330	**112490053**	**-0.1**	112694443	0.1
46		300	10	20	109773739	**108745773**	**-0.9**	**108622588**	**-1.0**
47		300	10	50	110088775	**108423976**	**-1.5**	**108655305**	**-1.3**
48		300	10	80	110350594	**108617127**	**-1.6**	**108862157**	**-1.3**
						Max	0.0	Max	1.0
						Min	-1.6	Min	-1.3
						Average	-0.6	Average	-0.4
49		400	3	20	172247869	**163765348**	**-4.9**	**163765348**	**-4.9**
50		400	3	50	162536528	163765348	0.8	163765348	0.8
51		400	3	80	171645514	**163793379**	**-4.6**	**163793379**	**-4.6**
52		400	5	20	159291000	**155155800**	**-2.6**	**155116513**	**-2.6**
53		400	5	50	158639802	**155667145**	**-1.9**	**155212330**	**-2.2**
54	PMED400	400	5	80	159891522	167775982	4.9	**155520422**	**-2.7**
55		400	8	20	150321000	158720000	5.6	**148986483**	**-0.9**
56		400	8	50	151895521	**148477092**	**-2.3**	**149147770**	**-1.8**
57		400	8	80	149557957	**148846650**	**-0.5**	**149568864**	**0.0**
58		400	10	20	147057000	**145888433**	**-0.8**	145888433	-0.8
59		400	10	50	148308000	**146288037**	**-1.4**	146120442	-1.5
60		400	10	80	149717000	**146712600**	**-2.0**	146692757	-2.0
						Max	5.6	Max	0.8
						Min	-4.9	Min	-4.9
						Average	-0.8	Average	-1.9

Table 3.6 Test results for MACHLP using the proposed SA-based heuristics with hub and access arc capacities (T=24h)

Pr. Id	Network	|N|	p	Decrease in maxarccap (%)	BP*	UB when δ=0.99	GapHeur (%)	UB when δ=0.95	GapHeur (%)
61		81	3	20	114914918	115762300	0.7	**114914900**	**0.0**
62		81	3	50	115694288	116593200	0.8	116593200	0.8
63		81	3	80	117011966	120498000	3.0	117495200	0.4
64		81	5	20	101824824	102881000	1.0	102881000	1.0
65		81	5	50	104498908	105050700	0.5	105077000	0.6
66		81	5	80	106022593	106772400	0.7	106772400	0.7
67		81	8	20	94742442	**94664910**	**-0.1**	**94664910**	**-0.1**
68		81	8	50	96915551	100403700	3.6	**96772750**	**-0.1**
69		81	8	80	98836098	102673900	3.9	**98836100**	**0.0**
70		81	10	20	91981590	92189660	0.2	**91957650**	**0.0**
71		81	10	50	93892359	**93884420**	**0.0**	**93824300**	**-0.1**
72	TR81	81	10	80	95753742	100115000	4.6	**95753740**	**0.0**
						Max	4.6	Max	1.0

						Min	-0.1	Min	-0.1
						Average	1.6	Average	0.3
73		100	3	20	34157981	**34157981**	**0.0**	**34157981**	**0.0**
74		100	3	50	36071269	**36071269**	**0.0**	**36071269**	**0.0**
75		100	3	80	37884305	**37884305**	**0.0**	**37884305**	**0.0**
76		100	5	20	31556347	**34383958**	**9.0**	**31556347**	**0.0**
77		100	5	50	33301518	**33301518**	**0.0**	**33301518**	**0.0**
78	PMED100	100	5	80	35878431	**35718658**	**-0.4**	37794415	5.3
79		100	8	20	29253295	32327704	10.5	32327704	10.5
80		100	8	50	30844544	**30664118**	**-0.6**	**30664118**	**-0.6**
81		100	8	80	32802869	32988664	0.6	**32802869**	**0.0**
82		100	10	20	28335970	**28326930**	**0.0**	**28324510**	**0.0**
83		100	10	50	29744519	**29542474**	**-0.7**	**29583921**	**-0.5**
84		100	10	80	32014511	**31545144**	**-1.5**	**31476793**	**-1.7**
						Max	10.5	Max	10.5
						Min	-1.5	Min	-1.7
						Average	1.4	Average	1.1
85		200	3	20	87950587	**86811528**	**-1.3**	**86811528**	**-1.3**
86		200	3	50	96173074	**87964453**	**-8.5**	**87964453**	**-8.5**
87		200	3	80	94955419	**90769904**	**-4.4**	**90769904**	**-4.4**
88		200	5	20	85427037	**76515205**	**-10.4**	**76515204**	**-10.4**
89		200	5	50	85992946	**76521679**	**-11.0**	**76521679**	**-11.0**
90	PMED200	200	5	80	92082116	**76537222**	**-16.9**	**76537222**	**-16.9**
91		200	8	20	76882178	86195094	12.1	**74734122**	**-2.8**
92		200	8	50	84224024	87963257	4.4	**77765212**	**-7.7**
93		200	8	80	88885932	**82388929**	**-7.3**	**82153312**	**-7.6**
94		200	10	20	73763549	**71975139**	**-2.4**	**71301223**	**-3.3**
95		200	10	50	95711583	**74219984**	**-22.5**	**73955861**	**-22.7**
96		200	10	80	82581387	**78896072**	**-4.5**	**78997931**	**-4.3**
						Max	12.1	Max	-1.3
						Min	-22.5	Min	-22.7
						Average	-6.1	Average	-8.4
97		300	3	20	140643156	**137893377**	**-2.0**	**137893377**	**-2.0**
98		300	3	50	141991514	**141366577**	**-0.4**	**141758889**	**-0.2**
99		300	3	80	154420261	**150996461**	**-2.2**	**148634916**	**-3.7**
100		300	5	20	150142080	**130814258**	**-12.9**	**130548934**	**-13.0**
101		300	5	50	142235824	**135594227**	**-4.7**	**132836398**	**-6.6**
102	PMED300	300	5	80	141216710	**138067286**	**-2.2**	**136909483**	**-3.1**
103		300	8	20	124436000	**124426170**	**0.0**	**121656760**	**-2.2**
104		300	8	50	129374170	**124028838**	**-4.1**	**123265563**	**-4.7**
105		300	8	80	132672564	140704046	6.1	**126142428**	**-4.9**
106		300	10	20	116758410	119615011	2.4	**112221735**	**-3.9**
107		300	10	50	125892543	**122997368**	**-2.3**	**115067938**	**-8.6**
108		300	10	80	126327965	**121032083**	**-4.2**	**121828034**	**-3.6**
						Max	6.1	Max	-0.2
						Min	-12.9	Min	-13.0
						Average	-2.2	Average	-4.7
109		400	3	20%	243230077	**186316093**	**-23.4**	**186316093**	**-23.4**
110		400	3	50%	285158797	**198884325**	**-30.3**	**198884325**	**-30.3**
111	PMED400	400	3	80%	236120100	**198804444**	**-15.8**	**198891600**	**-15.8**
112		400	5	20%	172253818	**171575551**	**-0.4**	**168785289**	**-2.0**
113		400	5	50%	206850078	**181032300**	**-12.5**	**176149487**	**-14.8**
114		400	5	80%	250749067	**185380026**	**-26.1**	**183609478**	**-26.8**

115	400	8	20%	193667054	**173492484**	**-10.4**	**157445600**	**-18.7**
116	400	8	50%	180416522	194093410	7.6	194093410	7.6
117	400	8	80%	210938589	**205193956**	**-2.7**	**205193956**	**-2.7**
118	400	10	20%	166750263	**162259100**	**-2.7**	**162259100**	**-2.7**
119	400	10	50%	196846174	199449959	1.3	199449959	1.3
120	400	10	80%	218859960	**216223226**	**-1.2**	**216223226**	**-1.2**
					Max	7.6	Max	7.6
					Min	-30.3	Min	-30.3
					Average	-9.7	Average	-10.8

3.6 Conclusion

In this chapter, we present *Multiple Allocation Hub Arc Capacitated p-hub Median Problem* (MACHLP) in which we impose an upper limit on the flow traversing the arcs. MACHLP minimizes transportation cost of sending flows between OD pairs by locating *p* hubs and satisfying the capacity requirements. Capacity constraints on the hub networks may be imposed both on the nodes and arcs of the network. However, most studies incorporate capacity constraints on the nodes and few studies address arc capacities with some restrictive assumptions. However, arc capacities are important in some settings of HLPs, e.g., bridges, subways, canals and straits are examples of the infrastructure that create capacities on the arcs of a tranportation network.

Studies addressing arc capacities assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. If the underlying real-life network is not complete or complete but its distances do not satisfy the triangle inequality, a preprocessing on the underlying network is implemented to construct a complete network whose costs satisfy the triangle inequality. However, arc capacities on a non-complete RealN may not be easily incorporated when a complete MN is used because an arc in a complete MN may actually correspond to a shortest path consisting of several arcs with different capacities and not necessarily a single arc in RealN. We develope a modeling approach that does not require any specific cost and network structure and that uses RealN to be used as MN. The proposed approach allows us to incorporate capacities on any arc of the network with our modelin approach.

In the this chapter, we solve the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic and develop a SA-based heuristic algorithm. We conduct computational experiments using networks with up to 400 nodes. We create test instances by defining capacities on different arcs, i.e., on only hub arcs, and both hub and access arcs and changing arc capacities. As the network size gets larger, the number of capacitated arcs increases and arc capacities become more restrictive resulting in high optimality gaps for the solutions found by CPLEX or Gurobi with NoRel heuristic. However, the heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for those instances.

In the study, we only incorporate arc capacities; however we may easily incoporate hub capacities as well. The SA-based heuristic algorithm is effective in finding good solutions especially for large-size problems but other heuristic algorithms based on different metaheuristics, e.g., tabu search and genetic algorithm, may also be developed. Exact solution algorithms such as a problem specific branch-and-bound algorithm may be developed as well.

Chapter 4

CONCLUDING REMARKS AND FUTURE DIRECTIONS

4.1 Conclusions

In this dissertation, we study two different hub location problems, namely, *Multiple Allocation Tree of Hubs Location Problem (MATHLP)* that result from incorporating a tree topology requirement for the hub network and *Multiple Allocation Arc Capacitated Hub Location Problem (MACHLP)* that result from imposing capacities on the arcs.

The models developed for HLPs in the literature assume that the underlying network is a complete network with arc distances (costs) satisfying the triangle inequality. If the real-world network is not complete or complete but its distances do not satisfy the triangle inequality (e.g., bus fares) as is the case for most real-life networks, a preprocessing is required to construct a complete network by an algorithm (e.g., Floyd [7]) that finds the shortest path lengths between all OD pairs in RealN. Even though this approach has gained acceptance, this may cause several modeling and computational disadvantages. For example, the interactions between location and routing decisions, arcs with different costs and capacities, different topology and service level requirements may not be modeled. Considering these issues, Akgün and Tansel [6] propose a problem setting and modeling framework that allows (non-complete or complete) real-world network with any cost structure to be directly used. For that reason, we study MATHLP and MACHLP built upon the problem setting adopted by Akgün and Tansel.

To our knowledge, all studies about tree of hubs location problem address the single allocation version of the problem whereas we study the multiple allocation version. Considering multiple allocation is essential in some cases of THLP, e.g., public transportation networks. We also deviate from the literature by using a new modeling framework that allows real-world network with any cost structure to be used. We show through examples that the proposed modeling approach may produce better solutions than the classical approach, which may result from the differences in the selected hubs, the flow routes between origin-destination points, and the assignment of non-hub nodes to hub nodes. We solve the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic and develop BD-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We conduct computational experiments using networks with up to 500 nodes. As the network size gets larger, the resulting optimality gaps are high for the solutions found by CPLEX or Gurobi with NoRel heuristic. On the other hand, the heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for all instances.

Capacity constraints on the hub networks may be imposed both on the nodes and arcs of the network. However, most studies in the literature incorporate capacity constraints on the nodes. Few studies address arc capacities assuming that the underlying network is a complete network with arc distances (costs) satisfying the triangle inequality. However, arc capacities on a non-complete real-world network may not be easily incorporated when a complete network is created from a real-world network. With the modeling approach we use, we can easily incorporate capacities on any of the arcs of the underlying network. We show through examples that using this proposed modeling approach is critical in the presence of arc capacities because the classical approach does not guarantee optimal even feasible solutions. We solve the proposed model by CPLEX-based algorithm and Gurobi-based algorithm with NoRel heuristic and develop an SA-based heuristic algorithm. We conduct computational experiments using networks with up to 400 nodes. We create test instances by defining capacities on different arcs and changing the amount of the capacity on the arcs. As the network size gets larger, as the number of arcs with capacities and the amount of capacity on the arcs increase the resulting optimality gaps are high for the solutions found by CPLEX or Gurobi with NoRel heuristic. On the other hand, the heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with NoRel heuristic for all instances.

4.2 Societal Impact and Contribution to Global Sustainability

United Nations Department of Economic and Social Affairs [95] provide Sustainable Development Goals (SDGs) that are an urgent call for taking action by all countries in a global partnership. Among these SDGs there are two goals, namely, ‘building resilient infrastructure, promote inclusive and sustainable industrilization and foster innovation’ and ‘making cities and human settlements inclusive, resilient and sustainable’. According to these two goals, it is utmost importance that urban and public transportation systems, gas, water and electricity distribution systems, and telecommunication network systems are smart, resilient and sustainable.

The application areas of the problems MATHLP and MACHLP that we study in this dissertation range from the optimization of fiber internet backbone to the exact configuration of the physical road network of the transportation networks of the cargo companies, from the improvement of computer or wireless communication networks to the establishment of smart electricity, water or gas distribution networks in the most efficient way, from efficient airway and railway transportation systems to smart public transportation systems with different transportation modes. MATHLP and MACHLP are directly applicable to a wide range of systems that serve to achieve the goals of building resilient infrastructure, sustainable transportation systems, sustainable cities and human settlements.

We propose a new modelling approach for the problems MATHLP and MACHLP that allows us to use the structure of the real physical network directly in the formulation of the problems. This approach provides more flexibility in modeling several characteristics of real-life hub networks. The developed models will find more application areas because they better represent real life problems. Moreover, one main challenge arising in real-life applications is the problem size. Mostly it is not possible to solve them with the standard optimization softwares. However, we are able to solve large-size problems that arise in real life with our proposed solution methodologies.

4.3 Future Prospects

We consider both problems MATHLP and MACHLP in a multiple allocation framework as p-hub median problems. In the future, we may adapt our approach to different types of hub location problems, e.g., p-hub center problem, hub location problem with fixed costs or hub covering problem as well. We may also adapt our approach to the single allocation framework. We assume that all the data is already known. However, the amount of flow generated between demand points has stochastic nature. For future research, we can incorporate stochasticity into our problems. The inclusion of stochasticity in the problems may lead to more robust solutions.

For MATHLP, in the future, we may incorporate other acceleration strategies not considered in this study, e.g., reduction of the model size and selection of good initial cuts, to improve the progress of exact Benders algorithms or Benders-type heuristics. We may develop hybrid algorithms utilizing metaheuristics and Benders decomposition to improve the effectiveness of the heuristic algorithms. A problem specific branch-and-bound algorithm may be developed as well.

For MACHLP, we only incorporate arc capacities but in the future we may also incorporate hub capacities besides arc capacities in the problem. We may also study MACHLP as multimodal hub location problem in which different transportation modes are used to design the hub networks. The proposed SA-based heuristic algorithm is effective in finding good solutions for MACHLP but other heuristic algorithms may also be developed. Exact solution algorithms as a problem specific branch-and-bound algorithm may be developed as well.

BIBLIOGRAPHY

[1] S. Alumur and B. Y. Kara, “Network hub location problems: The state of the art,” *Eur. J. Oper. Res.*, vol. 190, no. 1, pp. 1–21, 2008, doi: 10.1016/j.ejor.2007.06.008.

[2] J. F. Campbell and M. E. O’Kelly, “Twenty-five years of hub location research,” *Transp. Sci.*, vol. 46, no. 2, pp. 153–169, 2012, doi: 10.1287/trsc.1120.0410.

[3] R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh, “Hub location problems: A review of models, classification, solution techniques, and applications,” *Comput. Ind. Eng.*, vol. 64, no. 4, pp. 1096–1109, 2013, doi: 10.1016/j.cie.2013.01.012.

[4] I. Contreras and M. O’Kelly, “Location Science,” in *Location Science*, 2015.

[5] E. V. Marianov and Z. Drezner, “Contributions to Location Analysis,” 2019.

[6] İ. Akgün and B. Tansel, “P-Hub Median Problem for Non-Complete Networks,” *Comput. Oper. Res.*, vol. 95, pp. 56–72, 2018, doi: 10.1016/j.cor.2018.02.014.

[7] R. W. Floyd, “Algorithm 97: Shortest path,” *Commun. ACM*, vol. 5, no. 6, p. 345, 1962, doi: 10.1145/367766.368168.

[8] A. Marín, L. Cánovas, and M. Landete, “New formulations for the uncapacitated multiple allocation hub location problem,” *Eur. J. Oper. Res.*, vol. 172, no. 1, pp. 274–292, 2006, doi: 10.1016/j.ejor.2004.09.047.

[9] I. Contreras, E. Fernández, and A. Marín, “The Tree of Hubs Location Problem,” *Eur. J. Oper. Res.*, vol. 202, no. 2, pp. 390–400, 2010, doi: 10.1016/j.ejor.2009.05.044.

[10] J. G. Klincewicz, “Hub location in backbone/tributary network design: A review,” *Locat. Sci.*, vol. 6, no. 1–4, pp. 307–335, 1998, doi: 10.1016/S0966-8349(98)00042-4.

[11] J. Zhou, J. Peng, G. Liang, and T. Deng, “Layout optimization of tree-tree gas pipeline network,” *J. Pet. Sci. Eng.*, vol. 173, no. October 2018, pp. 666–680, 2019, doi: 10.1016/j.petrol.2018.10.067.

[12] J. Sabattin, G. Fuertes, M. Alfaro, L. Quezada, and M. Vargas, “Optimization of large electric power distribution using a parallel genetic algorithm with dandelion

strategy," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 26, no. 5, pp. 2648–2660, 2018, doi: 10.3906/elk-1801-261.

[13] W. Zhong, Z. Juan, F. Zong, and H. Su, "Hierarchical hub location model and hybrid algorithm for integration of urban and rural public transport," *Int. J. Distrib. Sens. Networks*, vol. 14, no. 4, 2018, doi: 10.1177/1550147718773263.

[14] V. R. Vuchic, *Urban Transit Systems and Technology*. John Wiley & Sons Inc, Hoboken, New Jersey., 2007.

[15] A. Ceder and A. Ceder, *Public Transit Planning and Operation*. 2007.

[16] G. M. Rehan and H. S. Mahmoud, "The integration between transportation solutions, economic development and community development as an approach for sustainability - A case study of Curitiba, Brazil," *World Acad. Sci. Eng. Technol.*, vol. 73, no. 1, pp. 705–711, 2011.

[17] J. F. Campbell, "Integer programming formulations of discrete hub location problems," *Eur. J. Oper. Res.*, vol. 72, no. 2, pp. 387–405, 1994, doi: 10.1016/0377-2217(94)90318-2.

[18] A. T. Ernst and M. Krishnamoorthy, "Solution algorithms for the capacitated single allocation hub location problem," *Ann. Oper. Res.*, vol. 86, pp. 141–159, 1999, doi: 10.1023/a:1018994432663.

[19] J. Ebery, M. Krishnamoorthy, A. Ernst, and N. Boland, "Capacitated multiple allocation hub location problem: Formulations and algorithms," *Eur. J. Oper. Res.*, vol. 120, no. 3, pp. 614–631, 2000, doi: 10.1016/S0377-2217(98)00395-6.

[20] N. Boland, M. Krishnamoorthy, A. T. Ernst, and J. Ebery, "Preprocessing and cutting for multiple allocation hub location problems," *Eur. J. Oper. Res.*, vol. 155, no. 3, pp. 638–653, 2004, doi: 10.1016/S0377-2217(03)00072-9.

[21] M. Labbé, H. Yaman, and E. Gourdin, *A branch and cut algorithm for hub location problems with single assignment*, vol. 102, no. 2. 2005.

[22] I. Contreras, E. Fernández, and A. Marín, "Tight bounds from a path based formulation for the tree of hub location problem," *Comput. Oper. Res.*, vol. 36, no. 12, pp. 3117–3127, 2009, doi: 10.1016/j.cor.2008.12.009.

[23] I. Correia, S. Nickel, and F. Saldanha-da-Gama, "The capacitated single-

allocation hub location problem revisited: A note on a classical formulation," *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 92–96, 2010, doi: 10.1016/j.ejor.2010.04.015.

[24] I. Contreras, J. A. Díaz, and E. Fernández, "Branch and price for large-scale capacitated hub location problems with single assignment," *INFORMS J. Comput.*, vol. 23, no. 1, pp. 41–55, 2011, doi: 10.1287/ijoc.1100.0391.

[25] S. A. Alumur, J. F. Campbell, I. Contreras, B. Y. Kara, V. Marianov, and M. E. O'Kelly, "Perspectives on modeling hub location problems," *Eur. J. Oper. Res.*, vol. 291, no. 1, pp. 1–17, 2021, doi: 10.1016/j.ejor.2020.09.039.

[26] D. Bryan, "Extensions to the hub location problem: Formulations and numerical examples," *Geogr. Anal.*, vol. 30, no. 4, pp. 315–330, 1998, doi: 10.1111/j.1538-4632.1998.tb00405.x.

[27] M. Sasaki and M. Fukushima, "On the hub-and-spoke model with arc capacity conatraints," *J. Oper. Res. Soc. Japan*, vol. 46, no. 4, pp. 409–428, 2003, doi: 10.15807/jorsj.46.409.

[28] I. Rodríguez-Martín and J. J. Salazar-González, "Solving a capacitated hub location problem," *Eur. J. Oper. Res.*, vol. 184, no. 2, pp. 468–479, 2008, doi: 10.1016/j.ejor.2006.11.026.

[29] C. C. Lin, J. Y. Lin, and Y. C. Chen, "The capacitated p-hub median problem with integral constraints: An application to a Chinese air cargo network," *Appl. Math. Model.*, vol. 36, no. 6, pp. 2777–2787, Jun. 2012, doi: 10.1016/J.APM.2011.09.063.

[30] F. Kaveh, R. Tavakkoli-Moghaddam, C. Triki, Y. Rahimi, and A. Jamili, *A new bi-objective model of the urban public transportation hub network design under uncertainty*, vol. 296, no. 1–2. Springer US, 2021.

[31] B. Kayışoğlu and İ. Akgün, "Multiple allocation tree of hubs location problem for non-complete networks," *Comput. Oper. Res.*, vol. 136, no. July, 2021, doi: 10.1016/j.cor.2021.105478.

[32] G. Carello, F. Della Croce, M. Ghirardi, and R. Tadei, "Solving the hub location problem in telecommunication network design: A local search approach," *Networks*, vol. 44, no. 2, pp. 94–105, 2004, doi: 10.1002/net.20020.

[33] D. S. Johnson, J. K. Lenstra, and A. H. G. R. Kan, "The complexity of the network design problem," *Networks*, vol. 8, no. 4, pp. 279–285, 1978, doi: 10.1002/net.3230080402.

[34] S. A. Alumur, B. Y. Kara, and O. E. Karasan, "The design of single allocation incomplete hub networks," *Transp. Res. Part B Methodol.*, vol. 43, no. 10, pp. 936–951, Dec. 2009, doi: 10.1016/J.TRB.2009.04.004.

[35] S. Nickel, A. Schöbel, and T. Sonneborn, "Hub Location Problems in Urban Traffic Networks," pp. 95–107, 2001, doi: 10.1007/978-1-4757-3357-0_6.

[36] S. S. Mohri, H. Karimi, A. A. Kordani, and M. Nasrollahi, "Airline hub-and-spoke network design based on airport capacity envelope curve: A practical view," *Comput. Ind. Eng.*, vol. 125, no. July, pp. 375–393, 2018, doi: 10.1016/j.cie.2018.09.010.

[37] H. Calik, S. A. Alumur, B. Y. Kara, and O. E. Karasan, "A tabu-search based heuristic for the hub covering problem over incomplete hub networks," *Comput. Oper. Res.*, vol. 36, no. 12, pp. 3088–3096, Dec. 2009, doi: 10.1016/J.COR.2008.11.023.

[38] S. Alumur and B. Y. Kara, "A hub covering network design problem for cargo applications in Turkey," *J. Oper. Res. Soc.*, vol. 60, no. 10, pp. 1349–1359, 2009, doi: 10.1057/jors.2008.92.

[39] M. G. Yoon and J. Current, "The hub location and network design problem with fixed and variable arc costs: Formulation and dual-based solution heuristic," *J. Oper. Res. Soc.*, vol. 59, no. 1, pp. 80–89, 2008, doi: 10.1057/palgrave.jors.2602307.

[40] S. A. Alumur, B. Y. Kara, and O. E. Karasan, "Multimodal hub location and hub network design," *Omega*, vol. 40, no. 6, pp. 927–939, 2012, doi: 10.1016/j.omega.2012.02.005.

[41] E. Martins de Sá, R. Morabito, and R. S. de Camargo, "Benders decomposition applied to a robust multiple allocation incomplete hub location problem," *Comput. Oper. Res.*, vol. 89, pp. 31–50, 2018, doi: 10.1016/j.cor.2017.08.001.

[42] E. Martins de Sá, R. Morabito, and R. S. de Camargo, "Efficient Benders decomposition algorithms for the robust multiple allocation incomplete hub

location problem with service time requirements," *Expert Syst. Appl.*, vol. 93, pp. 50–61, 2018, doi: 10.1016/j.eswa.2017.10.005.

[43] J. F. Campbell, A. T. Ernst, and M. Krishnamoorthy, "Hub arc location problems: Part I - Introduction and results," *Manage. Sci.*, vol. 51, no. 10, pp. 1540–1555, 2005, doi: 10.1287/mnsc.1050.0406.

[44] J. F. Campbell, A. T. Ernst, and M. Krishnamoorthy, "Hub Arc location problems: Part II - Formulations and optimal algorithms," *Manage. Sci.*, vol. 51, no. 10, pp. 1556–1571, 2005, doi: 10.1287/mnsc.1050.0407.

[45] J. F. Campbell, "Designing hub networks with connected and isolated hubs," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 1–10, 2010, doi: 10.1109/HICSS.2010.138.

[46] M. Labbé and H. Yaman, "Solving the hub location problem in a star-star network," *Networks*, vol. 51, no. 1, pp. 19–33, 2008, doi: 10.1002/net.20193.

[47] H. Yaman, "Star p-hub median problem with modular arc capacities," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 3009–3019, Sep. 2008, doi: 10.1016/J.COR.2007.01.014.

[48] E. Martins De Sá, I. Contreras, and J. F. Cordeau, "Exact and heuristic algorithms for the design of hub networks with multiple lines," *Eur. J. Oper. Res.*, vol. 246, no. 1, pp. 186–198, 2015, doi: 10.1016/j.ejor.2015.04.017.

[49] E. Martins de Sá, I. Contreras, J.-F. Cordeau, R. S. de Camargo, and G. De Miranda, "The Hub Line Location Problem," vol. 8, no. 2, pp. 142–168, 2015.

[50] C. ho Lee, H. bong Ro, and D. wan Tcha, "Topological design of a two-level network with ring-star configuration," *Comput. Oper. Res.*, vol. 20, no. 6, pp. 625–637, Aug. 1993, doi: 10.1016/0305-0548(93)90117-2.

[51] I. Contreras, M. Tanash, and N. Vidyarthi, "Exact and heuristic approaches for the cycle hub location problem," *Ann. Oper. Res.*, vol. 258, no. 2, pp. 655–677, 2017, doi: 10.1007/s10479-015-2091-2.

[52] E. M. De Sá, R. S. De Camargo, and G. De Miranda, "An improved Benders decomposition algorithm for the tree of hubs location problem," *Eur. J. Oper. Res.*, vol. 226, no. 2, pp. 185–202, 2013, doi: 10.1016/j.ejor.2012.10.051.

[53] S. Sedehzadeh, R. Tavakkoli-Moghaddam, A. Baboli, and M. Mohammadi, “Optimization of a multi-modal tree hub location network with transportation energy consumption: A fuzzy approach,” *J. Intell. Fuzzy Syst.*, vol. 30, no. 1, pp. 43–60, 2016, doi: 10.3233/IFS-151709.

[54] L. S. Pessoa, A. C. Santos, and M. G. C. Resende, “A biased random-key genetic algorithm for the tree of hubs location problem,” *Optim. Lett.*, vol. 11, no. 7, pp. 1371–1384, 2017, doi: 10.1007/s11590-016-1082-9.

[55] V. Blanco and A. Marín, “Upgrading nodes in tree-shaped hub location,” *Comput. Oper. Res.*, vol. 102, pp. 75–90, 2019, doi: 10.1016/j.cor.2018.10.007.

[56] A. T. Ernst and M. Krishnamoorthy, “Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem,” *Eur. J. Oper. Res.*, vol. 104, no. 1, pp. 100–112, 1998, doi: 10.1016/S0377-2217(96)00340-2.

[57] A. T. Ernst and M. Krishnamoorthy, “Efficient algorithms for the uncapacitated single allocations-Hub median problem,” *Rech. - Transp. - Secur.*, vol. 62, no. 3, pp. 139–154, 1999.

[58] J. F. Benders, “Partitioning procedures for solving mixed-variables programming problems,” *Numer. Math.*, vol. 4, no. 1, pp. 238–252, 1962, doi: 10.1007/BF01386316.

[59] R. S. de Camargo, G. Miranda, and H. P. Luna, “Benders decomposition for the uncapacitated multiple allocation hub location problem,” *Comput. Oper. Res.*, vol. 35, no. 4, pp. 1047–1064, Apr. 2008, doi: 10.1016/J.COR.2006.07.002.

[60] R. S. De Camargo, G. De Miranda, and H. P. L. Luna, “Benders decomposition for hub location problems with economies of scale,” *Transp. Sci.*, vol. 43, no. 1, pp. 86–97, 2008, doi: 10.1287/trsc.1080.0233.

[61] R. S. De Camargo, G. De Miranda, and H. P. L. Luna, “Benders decomposition for hub location problems with economies of scale,” *Transp. Sci.*, vol. 43, no. 1, pp. 86–97, 2009, doi: 10.1287/trsc.1080.0233.

[62] R. S. De Camargo, G. De Miranda, and R. P. M. Ferreira, “A hybrid Outer-Approximation/Benders Decomposition algorithm for the single allocation hub location problem under congestion,” *Oper. Res. Lett.*, vol. 39, no. 5, pp. 329–337, 2011, doi: 10.1016/j.orl.2011.06.015.

[63] R. S. de Camargo, G. de Miranda, M. E. O'Kelly, and J. F. Campbell, "Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints," *Appl. Math. Model.*, vol. 51, pp. 274–301, 2017, doi: 10.1016/j.apm.2017.06.035.

[64] R. S. de Camargo, G. de Miranda, and A. Løkketangen, "A new formulation and an exact approach for the many-to-many hub location-routing problem," *Appl. Math. Model.*, vol. 37, no. 12–13, pp. 7465–7480, 2013, doi: 10.1016/j.apm.2013.02.035.

[65] I. Contreras, J. F. Cordeau, and G. Laporte, "Stochastic uncapacitated hub location," *Eur. J. Oper. Res.*, vol. 212, no. 3, pp. 518–528, Aug. 2011, doi: 10.1016/J.EJOR.2011.02.018.

[66] I. Contreras, J. F. Cordeau, and G. Laporte, "Benders decomposition for large-scale uncapacitated hub location," *Oper. Res.*, vol. 59, no. 6, pp. 1477–1490, 2011, doi: 10.1287/opre.1110.0965.

[67] I. Contreras, J. F. Cordeau, and G. Laporte, "Exact solution of large-scale hub location problems with multiple capacity levels," *Transp. Sci.*, vol. 46, no. 4, pp. 439–459, 2012, doi: 10.1287/trsc.1110.0398.

[68] R. S. De Camargo and G. Miranda, "Single allocation hub location problem under congestion: Network owner and user perspectives," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3385–3391, 2012, doi: 10.1016/j.eswa.2011.09.026.

[69] N. Ghaffarinasab and B. Y. Kara, "Benders Decomposition Algorithms for Two Variants of the Single Allocation Hub Location Problem," *Networks Spat. Econ.*, vol. 19, no. 1, pp. 83–108, 2019, doi: 10.1007/s11067-018-9424-z.

[70] H. Mokhtar, M. Krishnamoorthy, and A. T. Ernst, "The 2-allocation p-hub median problem and a modified Benders decomposition method for solving hub location problems," *Comput. Oper. Res.*, vol. 104, pp. 375–393, 2019, doi: 10.1016/j.cor.2018.09.006.

[71] G. Taherkhani, S. A. Alumur, and M. Hosseini, "Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes," *Transp. Sci.*, vol. 54, no. 6, pp. 1446–1470, 2020, doi: 10.1287/trsc.2020.1003.

[72] M. Boschetti and V. Maniezzo, "Benders decomposition, Lagrangean relaxation and metaheuristic design," *J. Heuristics*, vol. 15, no. 3, pp. 283–312, 2009, doi: 10.1007/s10732-007-9064-9.

[73] C. A. Poojari and J. E. Beasley, "Improving benders decomposition using a genetic algorithm," *Eur. J. Oper. Res.*, vol. 199, no. 1, pp. 89–97, 2009, doi: 10.1016/j.ejor.2008.10.033.

[74] M. C. Lai, H. S. Sohn, T. L. Tseng, and C. Chiang, "A hybrid algorithm for capacitated plant location problem," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8599–8605, 2010, doi: 10.1016/j.eswa.2010.06.104.

[75] M. C. Lai, H. S. Sohn, T. L. Tseng, and D. L. Bricker, "A hybrid Benders/genetic algorithm for vehicle routing and scheduling problem," *Int. J. Ind. Eng. Theory Appl. Pract.*, vol. 19, no. 1, pp. 33–46, 2012.

[76] W. Jiang, L. Tang, and S. Xue, "A hybrid algorithm of tabu search and benders decomposition for multi-product production distribution network design," *Proc. 2009 IEEE Int. Conf. Autom. Logist. ICAL 2009*, no. 70728001, pp. 79–84, 2009, doi: 10.1109/ICAL.2009.5262971.

[77] R. Pacqueau, F. Soumis, and L.-N. Hoang, "A Fast and Accurate Algorithm for Stochastic Integer Programming, Applied to Stochastic Shift Scheduling," *TPublication G-2012-29,Groupe d'etudes recherch ́e en Anal. Decis. (GERAD), Univ. Montr. Montr. QC, Canada.*, 2012, [Online]. Available: http://www.gerad.ca/fichiers/cahiers/G-2007-21.pdf.

[78] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The Benders decomposition algorithm: A literature review," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 801–817, Jun. 2017, doi: 10.1016/J.EJOR.2016.12.005.

[79] T. L. Magnanti and R. T. Wong, "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria.," *Oper. Res.*, vol. 29, no. 3, pp. 464–484, 1981, doi: 10.1287/opre.29.3.464.

[80] A. Mercier, J. F. Cordeau, and F. Soumis, "A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem," *Comput. Oper. Res.*, vol. 32, no. 6, pp. 1451–1476, 2005, doi: 10.1016/j.cor.2003.11.013.

[81] J. R. Birge and F. V. Louveaux, “A multicut algorithm for two-stage stochastic linear programs,” *Eur. J. Oper. Res.*, vol. 34, no. 3, pp. 384–392, Mar. 1988, doi: 10.1016/0377-2217(88)90159-2.

[82] “NoRelHeurWork.” https://www.gurobi.com/documentation/9.1/refman/norelheurwork.html (accessed Oct. 20, 2021).

[83] “LocalSolver: Overview.” https://www.localsolver.com/product.html (accessed Oct. 20, 2021).

[84] J. E. B Easley, “Or-library: Distributing test problems by electronic mail,” *J. Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069–1072, 1990, doi: 10.1057/jors.1990.166.

[85] H. W. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn, “Adapting polyhedral properties from facility to hub location problems,” *Discret. Appl. Math.*, vol. 145, no. 1 SPEC. ISS., pp. 104–116, 2004, doi: 10.1016/j.dam.2003.09.011.

[86] A. Marín, “Formulating and solving splittable capacitated multiple allocation hub location problems,” *Comput. Oper. Res.*, vol. 32, no. 12, pp. 3093–3109, Dec. 2005, doi: 10.1016/J.COR.2004.04.008.

[87] H. Yaman and G. Carello, “Solving the hub location problem with modular link capacities,” *Comput. Oper. Res.*, vol. 32, no. 12, pp. 3227–3245, 2005, doi: 10.1016/j.cor.2004.05.009.

[88] M. E. O’Kelly and D. L. Bryan, “Hub location with flow economies of scale,” *Transp. Res. Part B Methodol.*, vol. 32B, no. 8, pp. 605–616, 1998, doi: 10.1016/S0191-2615(98)00021-6.

[89] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *J. Chem. Phys.*, vol. 21, 1953, doi: 10.1016/j.ejor.2004.07.022.

[90] A. T. Ernst and M. Krishnamoorthy, “Efficient algorithms for the uncapacitated single allocation p-hub median problem,” *Locat. Sci.*, vol. 4, no. 3, pp. 139–154, Oct. 1996, doi: 10.1016/S0966-8349(96)00011-3.

[91] S. Abdinnour-Helm, “Using simulated annealing to solve the p-Hub median problem,” *Int. J. Phys. Distrib. Logist. Manag.*, vol. 31, no. 3, pp. 203–220, 2001,

doi: 10.1108/09600030110389532.

[92] J. F. Chen, “A hybrid heuristic for the uncapacitated single allocation hub location problem,” *Omega*, vol. 35, no. 2, pp. 211–220, Apr. 2007, doi: 10.1016/J.OMEGA.2005.05.004.

[93] M. S. Jabalameli, F. Barzinpour, A. Saboury, and N. G. Nasab, “A simulated annealing-based heuristic for the single allocation maximal covering hub location problem,” *Int. J. Metaheuristics*, vol. 2, no. 1, p. 15, 2012, doi: 10.1504/ijmheur.2012.048213.

[94] N. Ghaffarinasab, A. Motallebzadeh, Y. Jabarzadeh, and B. Y. Kara, “Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems,” *Comput. Oper. Res.*, vol. 90, pp. 173–192, 2018, doi: 10.1016/j.cor.2017.09.022.

[95] “THE 17 GOALS | Sustainable Development.” https://sdgs.un.org/goals (accessed Jan. 05, 2022).

CURRICULUM VITAE

2001 – 2006	B.Sc., Industrial Engineering, Middle East Technical University, Ankara, TURKEY
2006 – 2011	Oracle Inventory and Production Planning Module Specialist, IT Department, Yataş, Kayseri, TURKEY
2013 – 2014	M.Sc., Industrial Engineering, Erciyes University, Kayseri, TURKEY
2013 – present	Research Assistant, Industrial Engineering, Abdullah Gül University, Kayseri, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) B. Kayışoğlu and İ. Akgün, “Multiple allocation tree of hubs location problem for non-complete networks,” Comput. Oper. Res., vol. 136, no. July, 2021, doi: 10.1016/j.cor.2021.105478.

C1) Kıdır S., Işılak R., Doğan Z., Kayışoğlu B., Akgün İ., " Trim Loss Problem in Cardvoard Production", 39th Operations Research and Industrial Engineering (YAEM) Congress, Başkent University, ANKARA, TURKEY, 12-14 June 2019.

C2) Kapar Y., İnce F., Şahin N., Uysal S.Z., Kayışoğlu B., Akgün İ., " Scheduling Problem in Textile Industry to Minimize Two-Dimensional Trim Loss and Machine Setting Time", 39th Operations Research and Industrial Engineering (YAEM) Congress, Başkent University, ANKARA, TURKEY, 12-14 June 2019.

C3) Ünal B., Demirci M., Kayışoğlu B., Akgün İ., "Distribution Center Location Problem", 39th Operations Research and Industrial Engineering (YAEM) Congress, Başkent University, ANKARA, TURKEY, 12-14 June 2019.

C4) Kayışoğlu B., Akgün İ., "A New Mathematical Model for Multiple Allocation Tree-of- Hubs Location Problem", 29th European Conference on Operational Research (EURO 2018), VALENCIA, SPAIN, 8-11 July 2018.

C5) Kayışoğlu B., Akgün İ., " A New Mathematical Model for Single Allocation Tree-of- Hubs Location Problem ", 38th Operations Research and Industrial Engineering (YAEM) Congress, Anadolu University, ESKİŞEHİR, TURKEY, 26-29 June 2018.

C6) Kayışoğlu B., Akgün İ., " Multi-Objective Facility Location of Service Points in Case of a Disaster", 35th Operations Research and Industrial Engineering (YAEM) Congress, Middle East Technical University, ANKARA, TURKEY, 09-11 September 2015.

C7) Kayışoğlu B., Özbakır L., " Classification of Lung Cancer with Cost Sensitive Data Mining Methodologies ", 34th Operations Research and Industrial Engineering (YAEM) Congress, Uludağ University, BURSA, TURKEY, 25-27 June 2014.