MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

PROBLEM REDUCTION ALGORITHMS FOR MEDIAN TYPE HUB LOCATION PROBLEMS

CİHAT ÖZTÜRK

Ph.D. THESIS

Department of Industrial Engineering

Thesis Supervisor

Prof. Dr. Serol BULKAN

Thesis CO- Supervisor

Prof. Dr. Gülfem TUZKAYA

ISTANBUL, 2022

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

PROBLEM REDUCTION ALGORITHMS FOR MEDIAN TYPE HUB LOCATION PROBLEMS

CİHAT ÖZTÜRK
(624414702)

Ph.D. THESIS

Department of Industrial Engineering

Thesis Supervisor

Prof. Dr. Serol BULKAN

Thesis CO- Supervisor

Prof. Dr. Gülfem TUZKAYA

ISTANBUL, 2022

ACKNOWLEDGMENT

Although just my name appears on the cover page of my thesis, there are many people who worked behind the scenes to make this thesis achievable. As a result, I'd want to express my gratitude to every one of them for their unending support, assistance, and understanding.

First and foremost, I want to thank my supervisor, Prof. Dr. Serol Bulkan, for his unending encouragement, inspiration, and useful guidance. Whenever I felt stuck, demotivated, or anxious, his mentoring was there to assist, and I always left his office happier, more motivated, and optimistic. I would also want to thank him for always sharing his vast knowledge and expertise with me in order for me to be a competent academic and researcher.

Second, I'd want to express my heartfelt gratitude to my co-supervisor, Prof. Dr. Gülfem Tuzkaya, for her great support, encouragement, and patience. Through our fascinating discussion, I learned a lot from her, and her criticism helped me a lot. Furthermore, she always appreciated my views and perspectives while being true and understanding, for which I am eternally thankful. I am also grateful to my friends and colleagues; I am confident that their togetherness and support made my Ph.D. journey a lot simpler and smoother.

If there is one person who deserves full credit for this thesis, it is without a doubt my lovely wife, Betül Öztürk. She was always there for me throughout my Ph.D., even when times were hard. She sacrificed her own interests, aspirations, and career for me, demonstrating amazing compassion and patience. I would not have been able to accomplish without her love, companionship, and understanding.

Since 30 June 2016, when my lovely son Yusuf Ali Öztürk was born, I have had additional moral support; he has been increasing my energy and hope with his presence, encouraging me in coping with the tough periods throughout my Ph.D.

Last but not least, I am immensely thankful to my lovely family, particularly my father, Adem Öztürk, and my mother, Ayşe Öztürk. They have always encouraged me to continue my education, to which they have committed their whole lives, from the beginning of my life. They gave up everything, even their own comfort and wants, so that

I may succeed. Their assistance, mentoring, understanding, patience, and compassion have always been unending.

June 2022 **Cihat Öztürk**

TABLE OF CONTENTS

ÖZET

ORTANCA TİP HUB YERLEŞİM PROBLEMLERİ İÇİN PROBLEM AZALTMA ALGORİTMALARI

Günümüzün küreselleşen dünyasında, lojistik ve iletişim alanlarındaki son gelişmeler modern vatandaşlar ve organizasyonlar için birçok fırsatı da beraberinde getirmektedir; ancak bu gelişmeler aynı zamanda, karar vericiler tarafından ele alınması gereken çok sayıda zorlu problemi de ortaya çıkarmaktadır. Bu zorlukların birçoğu, optimizasyon problemleri olarak temsil edilmeye uygundur. Ulaştırma ve iletişim ağ sistemlerinde düğüm tahsisi için ağ maliyeti minimizasyonu çerçevesi buna bir örnektir. Ulaştırma ve iletişim ağlarında, hizmet kalitesi ihtiyaçları ile ilgili belirli özellikleri karşılayan bir ağ topolojisi oluşturan konsept ve grup haberleşmede katman ağlar ise diğer örneklerdir. Bu tip problemlerden bir tanesi olan hub konum problemleri de genellikle stratejik seviyeli ağ optimizasyonu problemlerinden bir tanesidir ve ölçek ekonomisi tabanlı ağ yapısını dikkate alır.

Bu tezde de konum problemlerinin varyantlarından biri olan hub konum problemleri ele alınır. Hub konum problemleri son yıllarda araştırmacıların en fazla ilgi gösterdikleri konuların başında gelmektedir. Ölçek ekonomisi konsepti ve maliyetleri düşürme stratejileri göz önünde bulundurulduğunda özellikle lojistik ve haberleşme alanlarında uygulanabilirliği yüksek bir problem tipidir.

Hub konum problemleri temel olarak üç farklı kategoride değerlendirilebilir. Bunlar p-hub ortanca, p-hub merkez ve hub kapsama problemleridir. Bu tezde odaklanılan problem tipi p-hub ortanca problemleridir. Fakat p-hub ortanca problemlerinin varsayımlarından biri tüm hub konumlarının birbirine bağlı olmasıdır. Fakat gerçek hayat problemlerinde bu durum gerçekçi değildir. Bu yüzden, tamamlanmamış hub konum problemlerine de ayrıca odaklanılmaktadır. Tamamlanmamış hub konum problemlerinde hub ağındaki tüm merkez konumların birbirlerine bağlı olma varsayımı gevşetilmektedir.

İlk olarak, küçük boyutlu CAB, TR ve AP veri setleri p-hub ortanca problemleri (hem tamamlanmış hem de tamamlanmamış ağlar için) çerçevesinde derinlemesine analiz edilir. Bu analizler neticesinde problem bazlı olarak optimal hub konumlarına dair özellikler çıkartılmıştır. Bu özellikler çerçevesinde CBCA, HCBCA, BCBCA ve HBCBCA olmak üzere dört farklı çözüm metodolojisi sunulmuştur. Bu metodolojiler temel olarak düğümler arası akış, merkezilik ve uzaklık metriklerini kullanan sezgisel metotlardır. Bu metotların verimliliğinin kanıtlanmasının ardından mevcut meta-sezgisel algoritmalar çerçevesinde çözüm metotları geliştirilir. İlk olarak tavlama benzetimi algoritması, genetik algoritma ve genelleştirilmiş değişken komşu arama algoritması dikkate alınan probleme entegre edilerek çözümler alınır. Daha sonra problem boyutu küçültme stratejisi ile azaltılmış ve genelleştirilmiş değişken komşu arama metodu geliştirilir. Bu metodun klasik meta sezgisel metotlara göre verimliliği küçük, orta ve büyük boyutlu problemler için karşılaştırılır. Elde edilen sonuçlar çerçevesinde R-GVNS algoritması özellikle büyük boyutlu problemlerin çözümü açısından etkili bir yaklaşım olarak görünmektedir. Sonuçların anlamlı olup olmadığı parametrik olmayan istatistiksel testlerden biri olan Wilcoxon İşaretli Sıralar Testi ile karşılaştırılır. Bu bağlamda R-GVNS algoritmasının birçok senaryoda klasik meta sezgisel metotlardan hem çözüm süresi hem de çözüm kalitesi açısından açık ara iyi sonuçlar verdiği saptanmıştır.

ABSTRACT

PROBLEM REDUCTION ALGORITHMS FOR MEDIAN TYPE HUB LOCATION PROBLEMS

In today's globalized world, the recent developments in logistics and communication bring many opportunities for modern citizens and organizations.; however, these developments also raise a number of challenging problems that need to be addressed by decision makers. Several of these difficulties are amenable to representation as optimization methods. An example is the network cost minimization framework for node allocation in transport and communication network systems. In transport and communication networks, concept and overlay multi-cast networks are other examples, which create a network topology that meets certain characteristics related to service quality needs. Hub location problems, one of these types of problems, are generally strategic level network optimization problems and considered economies of scale-based network structure.

In this thesis, hub location problems, one of the variants of location problems, are discussed. Hub location problems have been one of the topics that researchers have shown the most interest in recent years. Considering the economy of scale concept and cost reduction strategies, it is a highly applicable problem type, especially in the fields of logistics and telecommunication.

Hub location problems can basically be evaluated in three different categories. These are p-hub median, p-hub center and hub covering problems. The problem type focused in this thesis is p-hub median problems. But one of the assumptions of p-hub median problems is that all hub locations are connected to each other. But in real life problems, this situation is not realistic. Thus, incomplete-hub location problems are also focused on. In incomplete hub location problems, the assumption that all locations in the hub network are interconnected is relaxed.

First, small-sized CAB, TR and AP datasets are analyzed in depth within the framework of both complete and incomplete p-hub median problems. As a result of these analyzes, the characteristics of the optimal hub locations are extracted specifically for the problem. Based on these features, four different solution methodologies are presented, namely CBCA, HCBCA, BCBCA and HBCBCA. These methodologies are basically heuristic methods that use internode flow, centrality, and distance metrics. After proving the efficiency of these methods, solution methods are developed within the framework of existing meta-heuristic algorithms. First, simulated annealing, genetic algorithm and generalized variable neighborhood methods are integrated into the problem and solutions are obtained. Then, the reduced generalized variable neighborhood method is developed with the problem size reduction strategy. The efficiency of this method compared to classical meta-heuristic methods is compared for small, medium and large sized problems. Within the framework of the results obtained, the R-GVNS algorithm seems to be an effective approach, especially in terms of solving large-sized problems. Whether the results are significant or not is compared with the Wilcoxon Signed Rank Test, which is one of the non-parametric statistical tests. In this context, it has been determined that the R-GVNS algorithm gives better results than classical metaheuristic methods in terms of both solution time and solution quality in many scenarios.

CLAIM FOR ORIGINALITY

The following are some of the valuable contributions that this thesis research provides to the literatures on hub location problems (HLP) and meta-heuristic solution methodologies:

- Hub location problems are problem types that are in the NP-Hard class and are very difficult to solve. Especially for complex network structures, different approaches are needed. For this, four different solution methods have been developed as Centrality Based Clustering Approach (CBCA), Hybrid Centrality Based Clustering Approach (HCBCA), Bounded Centrality Based Clustering Approach (BCBCA) and Hybrid Bounded Centrality Based Clustering Approach (HBCBCA).
- As far as the author knows, there are no meta-heuristic solution approaches developed for incomplete p-hub median problems in the literature. In this context, simulated annealing (SA), Genetic Algorithm (GA) and General Neighborhood Search (GVNS) algorithms are integrated into incomplete p-hub median problems.
- For the developed GVNS algorithm, novel different local search operators based on clones and betweenness centrality have been proposed and their efficiency has been proven.
- A new meta-heuristic approach, named R-GVNS, was introduced by integrating the problem reduction technique into the GVNS algorithm. This approach aims to get quality results in a short time by reducing the number of decision variables. The R-GVNS algorithm was compared in terms of performance with other proposed GA, SA and VNS meta-heuristics.
- The performance of all developed algorithms was also analyzed with the Wilcoxon Signed Rank Test, one of the non-parametric statistical tests, and comparisons between the developed algorithms were presented.
- Upper bound values are provided for incomplete p-hub median problems (especially large size problems).

SYMBOLS

$\boldsymbol{\alpha}$	:	economies of scale coefficient
χ	:	collection flow coefficient
$\boldsymbol{\delta}$	:	distribution flow coefficient
$\boldsymbol{B}$	:	betweenness centrality
$\boldsymbol{C}$	:	closeness centrality
$\boldsymbol{C_{ij}}$	:	transportation cost between node i and node j
$\boldsymbol{D_i}$	:	total demand for node i
$\boldsymbol{d_{ij}}$	:	distance between node i and node j
$\boldsymbol{f_{ij}^k}$	:	flow between node i to node j on hub k
$\boldsymbol{N}$	:	node set in a network
$\boldsymbol{O_i}$	:	total order for node i
$\boldsymbol{p}$	:	number of hubs
$\boldsymbol{P_r}$	:	node r_{th} feature for problem reduciton
$\boldsymbol{q}$	:	number of hub connection links
$\boldsymbol{r_k}$	:	rank position for node k
$\boldsymbol{T_n}$	:	temperature level in period n
$\boldsymbol{x_{ij}}$	:	decision variable for assignments
$\boldsymbol{w_{ij}}$	:	flow between from node i to node j

ABBREVIATIONS

AP	:	Australian Post
BCBCA	:	Bounded Centrality Based Clustering Algorithm
CAB	:	Civil Aeronautics Board
CBCA	:	Centrality Based Clustering Algorithm
CMAp-HMP	:	Capacitated Multiple Allocation p-Hub Median Problems
CSAp-HMP	:	Capacitated Single Allocation p-Hub Median Problems
GA	:	Genetic Algorithm
GVNS	:	Generalized Variable Neighborhood Search
HBCBCA	:	Hybrid Bounded Centrality Based Clustering Algorithm
HCBCA	:	Hybrid Centrality Based Clustering Algorithm
HLP	:	Hub Location Problems
MAHLP	:	Multiple Allocation Hub Location Problems
p-HMP	:	p-Hub Median Problems
R-GVNS	:	Reduced Generalized Variable Neighborhood Search
SA	:	Simulated Annealing
SAHLP	:	Single Allocation Hub Location Problems
TR	:	Turkish
UMAp-HMP	:	Uncapacitated Multiple Allocation p-Hub Median Problems
URAND	:	Uniform Random
USAp-HMP	:	Uncapacitated Single Allocation p-Hub Median Problems
USAp-IHMP	:	Uncapacitated Single Allocation Incomplete p-Hub Median Problems
USIp-HMP	:	Uncapacitated Single Allocation p-Hub Median Problem
VNS	:	Variable Neighborhood Search

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

About any business and governmental organization which we can consider of has addressed the issue of facility location at some period in its lifespan. Therefore, location decisions for manufacturing, assembly, and service plants, as well as warehouses, must be determined by industrial companies. For instance, retail outlets should focus on location decisions for their stores. There are facility location selection decisions for branch office locations in cargo transportation. The government, on the other hand, should manage many decision processes regarding public services including as schools, hospitals, fire stations, ambulance bases, vehicle inspection stations and landfills. In this context, all non-profit or profit oriented organizations face many problems in facility location decisions. Thus, a firm's capacity to effectively manufacture and market its products or to provide high-quality services in the service sector depends on the position of the firms' facilities in relation to other facilities and their customers.

Location models in analytical perspective, including a series of questions. How many facilities should be installed? What could the location of each facility be? What capacity should each facility have? How can the need for the facilities' resources be allocated? The responses of all questions are highly dependent on the context in which the location problem is being solved as well as the priorities driving the problem. In some situations, such as emergency site location optimization problems, we would want to location the facilities as close to the requested sites (densely populated locations) as possible. In an opposite case such as armory site location selection, we would like to be in a seismically safe locations and as far away from density population centers as possible.

Hub locations are special facility types that have an important place in many sectors such as logistics, telecommunication, and air transportation. These type facilities are performed switching, sorting, connecting, and consolidation/break-bulk functions for traffic between supply and demand nodes. Instead of establishing a direct connection between all origin-destination pairs, providing flows through hub facilities yield great advantage in terms of both cost and efficiency of the transportation. Because of the number of connections in the distribution network is decreasing and economies of scale are benefited due to large amounts of transportation. On a distribution network, flows that sent from a supply node to any demand node must visit at least one hub location. Transportation is not performed by establishing a direct connection to the flows for each node. Instead, flows from various nodes are collected at selected hub locations and distributed from these locations.

Hub location problems involve critical planning decisions that facilitate in strategic improvement in terms of environmental, societal, political, and, most significantly, economic aspects. Therefore, decisions in hub location problems are hard to overcome. In this regard, the optimal design of hub distribution networks, which are becoming increasingly complex on a global scale, is critical in the many aspects. Hub location problems generally involve two decision processes. The first is the optimal location of the hubs, and the other is the optimal assignment of non-hub nodes to optimal hub locations. The main purpose of such problems is to distribute the flows on the network to the nodes at the least cost or in time. For these reasons, hub location problems can be defined as a network design process.

Flows originating from the same supply location and distributed to different demand locations or flows that are distributed to the same demand locations from different supply locations are subjected to load consolidation at hub centers. These load combinations can be directly between hub-to-hub, as well as node-to-hub, and hub-to-node directions (Alumur and Kara, 2008).

In traditional hub location problems, generally there is a network structure in which origin-destination node pairs are considered. In these type problems, the hub locations determining factors can be listed as the coefficient economy of scale, amount of flows between nodes, centrality, costs, and distances between node pairs. These factors are discussed in the literature from both deterministic and stochastic aspects.

We classify different hub location problem types in Figure 1 to provide a framework for the variations of the models in terms of objective functions. The presented problems contain different characteristics in terms of cost and service level objectives and constraints. The presented models with different features are useful for reflecting real life problems more effectively. A classification enables researchers to recognize and compare their own work in the literature, focus fully on research areas and find appropriate solution methods. In addition, the classification and structuring of existing literature illuminates the direction and challenges of future research and determines the motivation for future outlook in the literature.

In this context, hub location models are basically examined in terms of two features as discrete and continuous approaches. The discrete models discussed in this thesis can be examined in three different frameworks. These have minmax, maxmin and minsum type objective functions. The main purpose of early studies on hub location problems is the minimization of variable and fixed costs. Variable costs are usually of transportation, distribution, or vehicle related origin. Fixed costs, on the other hand, consist of classical logistics costs such as opening a hub location, establishing a connection between hubs. Studies that consider both variable and fixed costs are also included in the early hub location literature.

As seen in Figure 1.1, in hub location problems, the flow is delivered to the demand point after a three-stage process. The flow from the supply region first arrives at the hub location to which this location is connected. This stage is called collection part. In the transfer phase, consolidated flows are moved between two hub regions (only one hub is visited if origin-destination nodes are connected to the same hub location). Flows coming to the destination hub are forwarded to the destination point connected to this hub. This last stage refers to the distribution process. Thanks to the framework in Figure 1, costs are reduced by taking advantage of economies of scale. In such problems, hub locations indicate facilities such as airport zones, postal distribution centers, bus terminals. Flows are used to express the loads (passengers, products) carried by vehicles such as trucks, buses, planes between supply regions and demand centers. In addition, there are networks based on the hub distribution concept in the telecommunications and energy sectors.

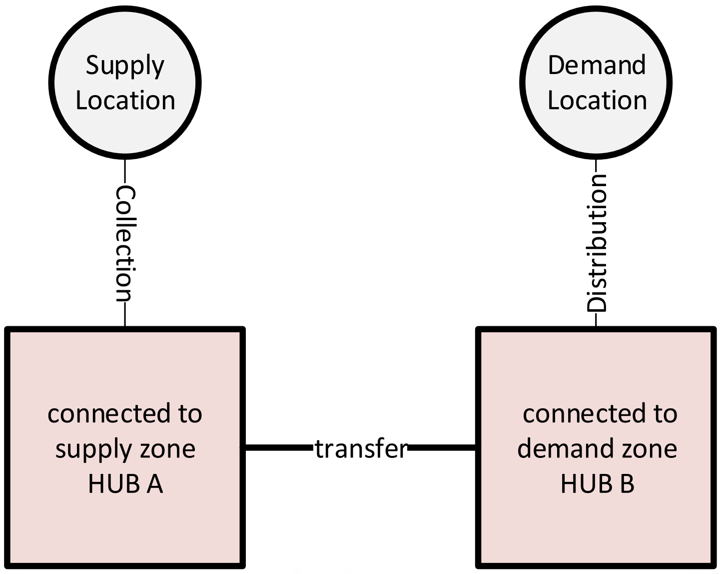


Figure 1.1. Origin-to-Destination Transportation

One of the largest costs in setting up a communication or transport network is the transportation costs between nodes. Typically, such network architectures accommodate multiple hub locations, eliminating the need to connect all nodes. This type of problem consists of two sub-problems. The first one is the location problem and the other one is the allocation problem. While examining which nodes are the most suitable hub locations in the location problem, the allocation problem focuses on the connection relationships in the network structure with access arcs. In traditional hub location problems, each node region has to be assigned to one or more hub nodes. No connection or flow between nodes is allowed in such problems (except for special types of problems). That is, in hub location network design problems, the flow between origin destination pairs can be accomplished by visiting at least one hub location.

1.1.Objectives and Contributions of the Thesis

This thesis examines p-hub median problems, which is one of the popular hub location problem variants. However, it also relaxed the assumption that all hub locations are topologically interconnected. In addition, capacity constraints are neglected for both edge and hub locations. As far as the author knows, there is no solution approach on the partial hub network model in the literature. Therefore, the proposed solution approaches provide computational based contributions to the literature. In this context, firstly, p-hub median (complete and incomplete) problems are analyzed in depth and four different problem reduction-based algorithms are proposed. These algorithms present an approach to reduce the number of decision variables in the problem by considering the centrality, distance, and flow features. In addition, for this type of problems, for the first time, four different meta-heuristic solution methods implemented for incomplete p-hub median problems. These algorithms are SA, GA, GVNS and R-GVNS solution methods, respectively. SA and GA algorithms are presented in order to compare the performances of the GVNS and R-GVNS methods we have developed. Effective novel local search operators are defined in the GVNS method. The R-GVNS approach, on the other hand, is a method that aims to provide quality results by reducing the problem size over node features based on the GVNS algorithm. It offers a two-stage solution methodology for this. At the first stage, various graph-based and statistical metrics are presented for nodes. Based on these metrics, candidate hub sets are constructed, and solutions are searched from this set with a certain probability. In the second stage, features and metrics defined for connections between hubs within the framework of selected hub location sets. At this stage, the most probable hub connections are made over the determined metrics and the local search is apply this network. These approaches are evaluated comparatively, and inferences presented for results. The proposed R-GVNS method a new methodology different from conventional meta-heuristic approaches.

1.2.Organization of the Thesis

The organization of the remaining parts of the thesis is as follows. In the second chapter, detailed literature research on hub location problems is presented. This section also shows the classification of hub location problems and some model examples for different types of problems. In the third chapter, the model focused on in the thesis and the data used are included. In addition, in-depth analyzes are presented on the small-scale test data on the focused problem. Based on these analyses, four different solution approaches are suggested. In the fourth chapter, four different algorithms for the solution of the model presented in the third chapter are explained. The fifth chapter consists of the solutions of the proposed algorithms and the statistical evaluations made on these solutions. In the sixth chapter, which is the last chapter, there are results and future directions.

2. LITERATURE REVIEW

Hub centers are special type of facilities that have an important place in many sectors such as logistics, telecommunication, and air transportation. This type of facility performs switching, sorting, connecting, and consolidation/break-bulk functions for traffic between supply and demand nodes. Instead of establishing a direct connection between all origin-destination pairs, providing flows through hub centers yield great advantage in terms of both cost and efficiency of the transportation. Because of the number of connections in the distribution network is decreasing and economies of scale are benefited due to large amounts of transportation. On a distribution network, flows that sent from a supply node to any demand node must visit at least one hub location. Rather than locating each origin-destination pair directly, flows originating from various nodes are collected in a hub and flows allocated to a certain node are transmitted from that whole hub.

Hub location problems involve critical planning decisions that facilitate in strategic improvement in terms of environmental, social, political, and, most significantly, economic aspects. Therefore, decisions in hub location problems are hard to overcome. In this regard, the optimal design of hub distribution networks, which are becoming increasingly complex on a global scale, is critical in the many aspects. The hub location problem is related to the location of hubs and the allocation of non-hub nodes to hubs. The objective of the problems is to find the best hub locations and distribute the corresponding hubs to the nodes with the minimum cost or maximum service level. These two decisions may have an impact on each other and are widely examined in the literature by the researchers. Thus, hub location problems can be considered as a network design problem.

Flows originating from the same supply location and distributed to different demand locations or flows that are distributed to the same demand locations from different supply locations are subjected to load consolidation at hub centers. These load combinations can be directly between hub-to-hub, as well as node-to-hub, and hub-to-node directions (S. Alumur & Kara, 2008). In Figure 2.1, an example network with 15 nodes and 3 (represented blues square) of the 15 nodes are selected as hub locations. Connections in the network explain the hub network structure.

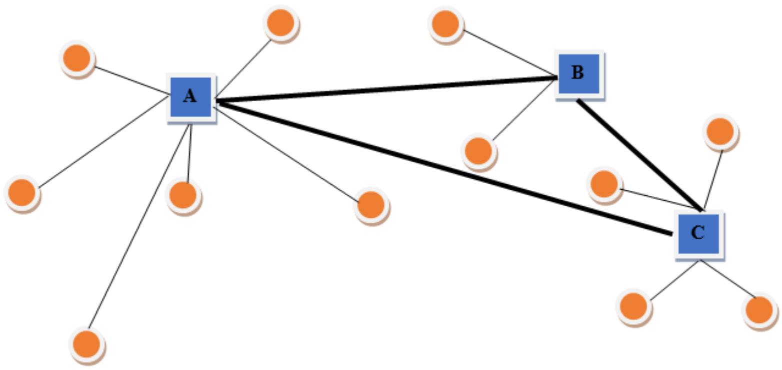


Figure 2.1. Complete Hub Network Representation

In traditional hub location problems, generally there is a network structure in which origin-destination node pairs are considered. In these type problems, the hub locations determining factors can be listed as the coefficient economy of scale, amount of flows between nodes, centrality, costs, and distances between node pairs. These factors are discussed in the literature for all aspects.

Although first reference papers dealing with hub location problems were presented by O'Kelly (1986;1987) in a mathematical perspective, ideas on hub location problems were introduced earlier. Hakimi (1964) is presented one of the leading studies on the origin of hub location problems. Toh and Higgins (1985), on the other hand, discussed hub network location models in air transportation. However, O'Kelly's (1986, 1987) studies can be considered as a milestone in considering core hub location models. Then, the first hub location model based on airline passenger transportation networks was presented by O'Kelly, (1987). It can be said that there has been a significant increase in studies dealing with hub location problems after the mid-1980s.

For early hub location models, O'Kelly (1987, 1992) presented models based on the quadratic formulation. These models were later developed by Campbell (1994, 1996) and defined with different mathematical formulations. Aykin (1994) proposed a capacitated variant of the hub location problems with fixed costs and hubs with restricted capacity. Aykin (1995) has investigated the similar problem with fixed costs and identified fixed number of hubs. Another milestone study based on hub location problems presented by (Ernst & Krishnamoorthy (1999). In this study, they deal with the capacity-constrained single-allocation models. Their two proposed formulations are improved versions of the previously developed mixed integer programming models.

However, after the early hub location models (especially in the mid-1990s), the number of articles published by researchers in this area has increased significantly. For this reason, the application areas and modeling approaches of hub location problems are changing year by year and many kinds of hub location problems have been derived. Thus, we classify different hub location problem types in Figure 2.2 to provide a framework for the variations of the models in terms of objective functions. The presented problems contain different characteristics in terms of cost and service level objectives and constraints. The presented models with different features are useful for reflecting real life problems more effectively. A classification enables researchers to recognize and compare their own work in the literature, focus fully on research areas and find appropriate solution methods. In addition, the classification and structuring of existing literature illuminates the direction and challenges of future research and determines the motivation for future outlook in the literature.

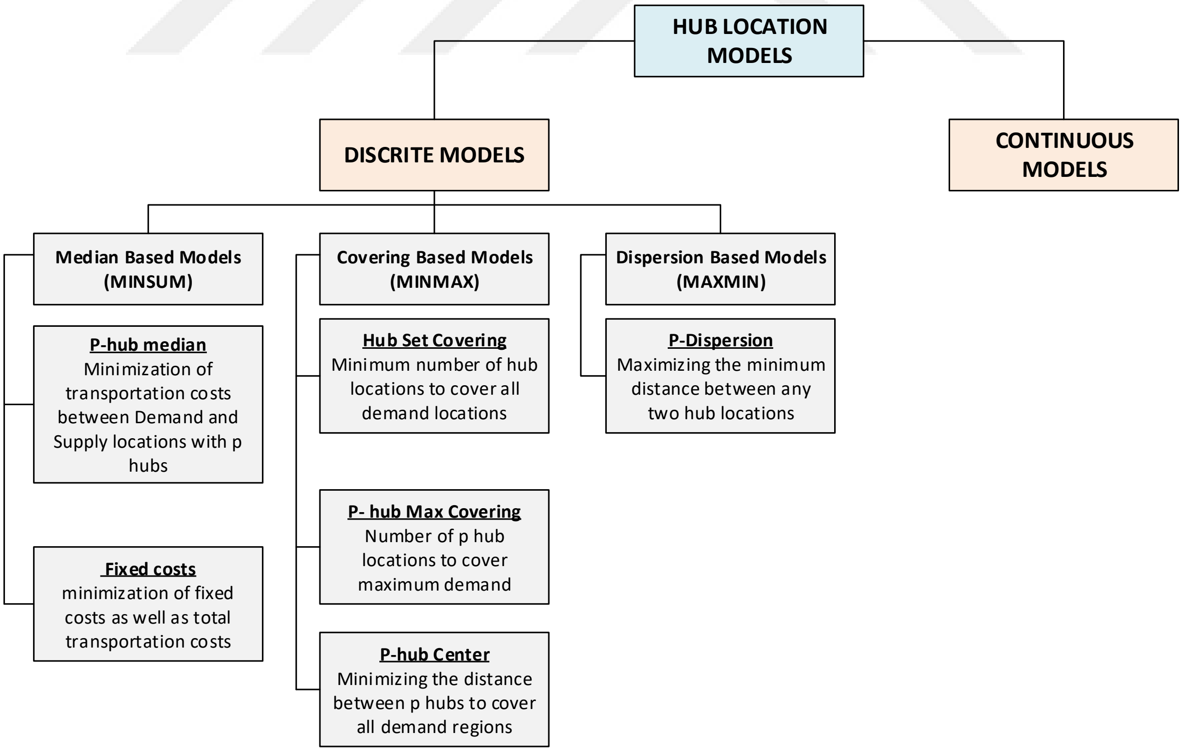


Figure 2.2. Hub Location Models Classification

In this context, hub location models are basically examined in terms of two features as discrete and continuous approaches. The discrete models discussed in this work can be examined in three different frameworks. These have minmax, max-min and min-sum type formulations. The objective functions most considered in the early hub location literature are based on minimization of collection, distribution, hub opening, arc setting and other specified costs. While the aim of these problems is to minimize the total transportation costs, there are also studies that take into account the fixed costs.

Hub location problems are built on a collection of recurring patterns that may be described as the problem's assumptions or logical constraints. Identifying each pattern will make it easier to understand the problem's complexity. It will also produce a guide to describe the issues and categorize them. Real-world applications are being updated with new and changing specifications, such as managerial and operational decisions. Thus, real-world applications engage with the conceptual perspective of the problem. As a nutshell, the emphasis of this research is on classification of the aspects that describes the structure of the hub location problems and to present a literature review on different hub location problem variants.

The literature studies on hub locations that have been conducted to far have addressed previous research from a variety of perspectives. Alumur & Kara (2008) presented a detailed review of the three most known hub location models as p-hub median, hub covering and p-hub center problems. Campbell & O'Kelly (2012), on the other hand, discussed the origins of hub location models and the current status and future of studies in this area, rather than classical literature review of the hub location problems. Farahani et al. (2013) presented a classification of hub location problems through mathematical models. In current studies, hub location problems are basically classified based on metrics such as objective functions, model constraints, economies of scale, modeling approaches, and topological network differences (Conteras, 2020; Conteras and O'Kelly, 2019). Besides these studies, the research provided by Alumur et al. (2021) reports on the present and future development of hub location models.

In the literature research part of this thesis, we classify the hub location literature under six main topics. These can be listed as objective function, network topology, modeling framework, solution approaches, extensions of hub location models and application areas.

Classifications such as mathematical formulation techniques (linear, non-linear, quadratic etc.), solution domain (discrete, continuous etc.) can also be added to the classification criteria. However, in each section, such criteria are mentioned in the content of the article. The general framework of the classification concept discussed in our study is shown in the Figure 2.3. The most important topic that distinguishes our study from other literature review approaches is the recent extensions of hub location problems. In particular, we focused more on this section because of the crucial role of models for realistic hub location problems.

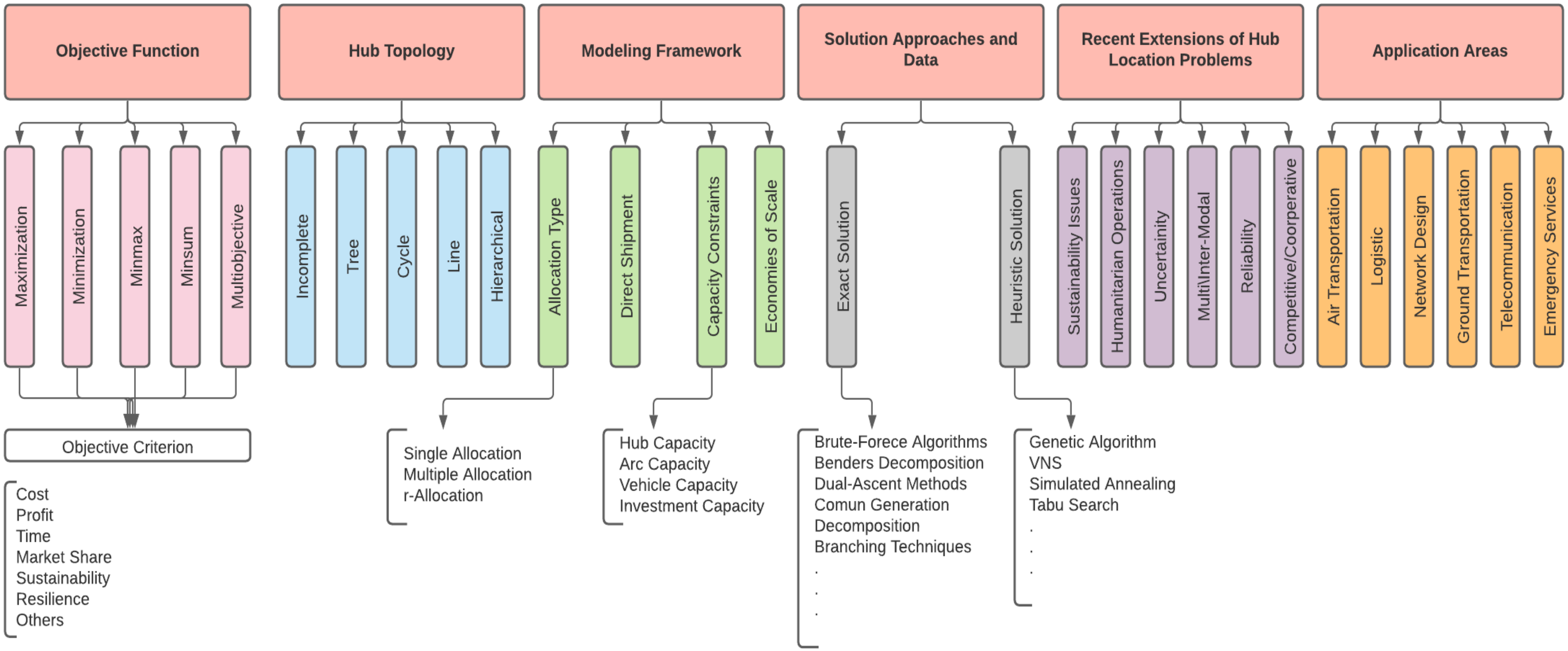


Figure 2.3. Hub Location Problems Based on Different Aspects

2.1.Paper Selection Process

This section provides information on the literature review's procedure. The methodological approach consists of three main parts as search phase on relevant articles, categorization phase, and source statistics.

2.1.1. Search Phase

We consider peer-reviewed papers published mostly between 2010 and 2021, but the publication year is not limited to this period. Papers were searched using Thomson Reuters Web of Knowledge and Scopus databases. Keywords like intermodal hub network design, hub location, hub and spoke network design, p-hub median, p-hub center, and hub covering were used. We also focus on articles by the best-known authors in the literature on hub location problems. The network of authors and their relations, who contributed the most to this field and created a hub location community, was presented in Figure 2.4. The publications of authors in this field generally consist of the most cited and contributing studies in the related literature. In addition, Mendeley (www.mendeley.com) is selected as the reference manager for paper collecting and sorting. The collection and sorting of references are made much easier and customizable by using the software application.

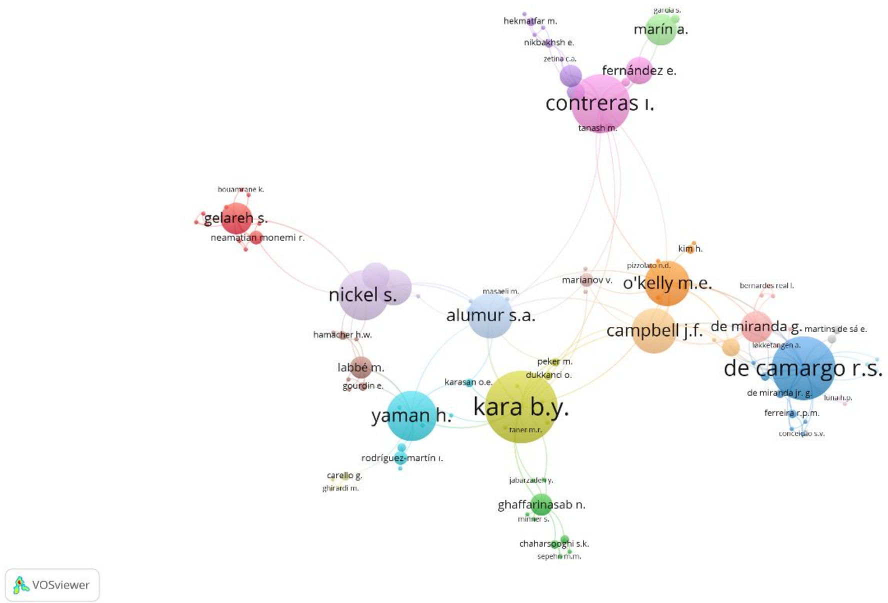


Figure 2.4. Authors and collaborations contributing to the hub location literature

2.1.2. Related Papers and Categorization

The basis developed by previous literature reviews was used to maintain consistency. The study presented by (Alumur & Kara, 2008) is a good starting point for this. Many authors have begun to deal with studies in this field based on emergent concepts and technologies. Especially with the development of unmanned aerial vehicles, autonomous vehicles and additive manufacturing systems, the concept of hub location evolves to a different point. Trends in this area are analyzed next sections in the study. Journals in which studies on hub location problems are concentrated are presented in the Figure 2.5. Computers and Operations Research, European Journal of Operation Research, Computers and Industrial Engineering, Transportation Research Part E and Transportation Research Part B seem to be the journals that publish the most articles in this field. In addition, Annals of Operations Research and Applied Mathematical Modeling are also coming as successor journals. It is noteworthy that most of the studies in this field are published in top journals focused on operations research.

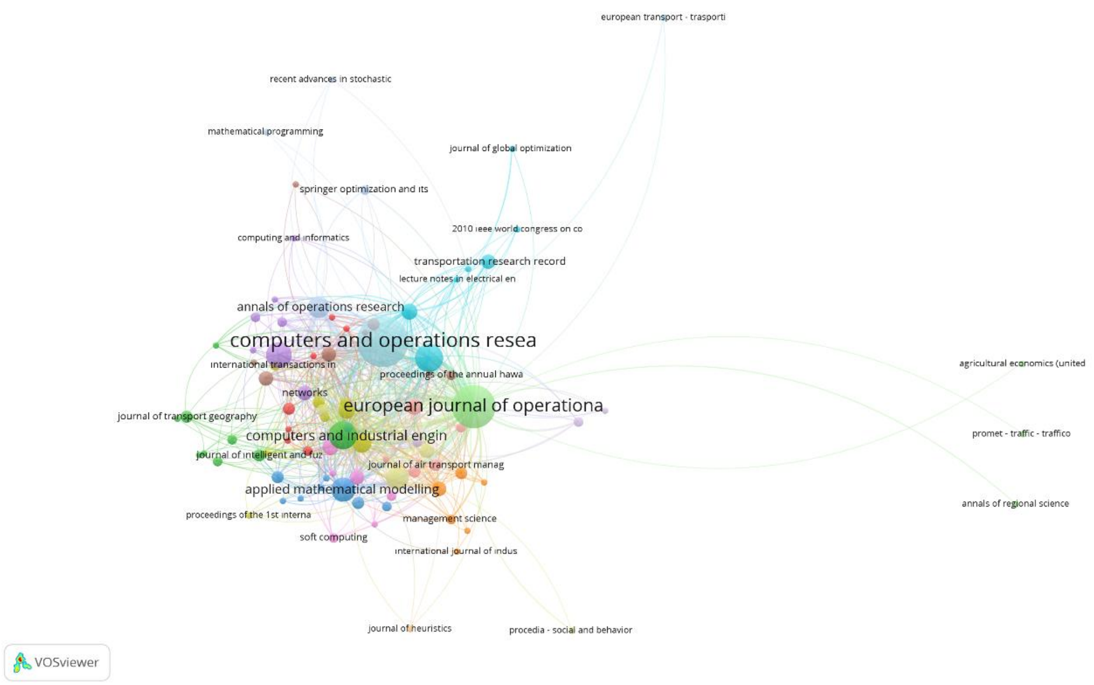


Figure 2.5. Major journal network on hub location literature

With the help of the proposed categorization, researchers can identify and compare their own work in the literature, as well as researchers know exactly what field of study they are working in and the best solution approaches to use. HLP can take on a more complex classification structure by describing numerous variants of the subject as different objectives, constraints, or solution methods. Existing literature can be classified and organized to reveal future research directions and issues and the rationale behind such efforts.

2.1.3. Source Statistics

In the hub location literature, we found approximately 475 relevant studies by searching the databases (Scopus, Web of Science, etc.) presented in the section 1.1.1. We reduced the number of studies to 214 based on the criteria we considered. Many studies have been presented on the hub location problem so far, and many valuable contributions have been made in this area. According to Figure 2.6, the quantity of papers has risen over time. In the early publications emphasize model-building and solution approaches from the early 2000s onwards, then optimization and completion of models later, and eventually novel problem variations as well as exact solution methodologies in more recent years. Given the wide range of possible expansions, it's remarkable that there's been such a large gap between academic discourse and real-life problems.

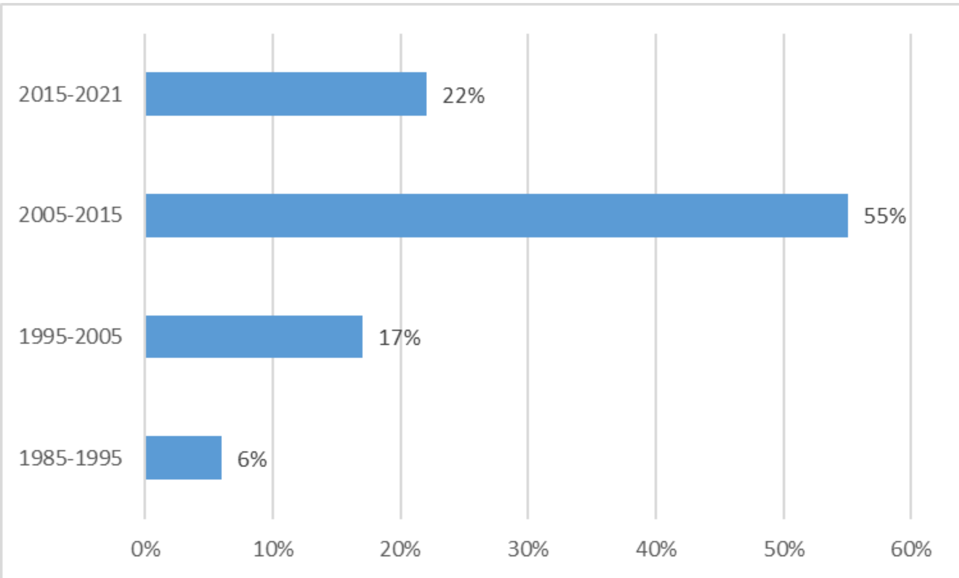


Figure 2.6. Article publication rates according to 10-years intervals based on hub location literature

The origin locations of the considered studies are examined. Iran was found to be the country with the most publications in this field with 23%. Iran is followed by China

(19%), USA (12%) and Turkey (10%). Since China and the USA have strong economies and high trade volumes, their position in this ranking is quite normal. However, the fact that two countries such as Turkey and Iran have more publications in this field is an issue worth examining. The fact that these two countries are in a position to connect Asia and Europe may be one of the important factors. In addition, development and population potential should not be ignored.

2.2.Objective Function

In hub location problems, objective functions are mainly based on two most known criteria as cost and service level. In cost-oriented objective functions, cost or distance (directly related to costs) minimization is considered. However, other measures on an economic basis objective such as profit, and market sharing have also been used by researchers in location models in recent years. The emergence of many actors in sectors such as cargo, transportation and telecommunication increase the competition in the market. Therefore, competitiveness is one of the focal points of hub location problems for more profitable investments in terms of market sharing. Thus, the considered objective types in the hub location-based studies conducted in recent years have also revealed original approaches. However, due to the main theme of hub location problems, we consider the following classification:

- *Hub Location Models:* The number of hubs to identify is unknown in advance, but a fixed installation cost for each hub is assumed in Hub Location Problems. The aim is to reduce the total hub installation fixed costs and transportation costs across the hub network.
- *p-Hub Median Models:* In median-type p-hub location problems, the number of hubs to be installed is predetermined. Considering p hub locations, minimization of transportation costs constitutes the objective function. In addition, there are studies that considered the fixed hub installation costs.
- *Hub Covering Models:* Hub covering problems aim to maximize of the service level, in other words, the minimization of the service time, and the objective function is the fixed costs minimization of the hubs to be installed. The transportation times between each origin destination pairs are limited to certain times or distances. However, the coverage criteria considered in such problems

may differ. These criteria can be considered in three main aspects; The first one is to determine a certain threshold value for the distance or transportation time between each pair of nodes based on a path (if we consider i and j as a hub pair and k and m as hub locations on (i,j) destination, then (i,k,m,j) is a path for (i,j) origin destination pairs). Second, define a threshold value (based on time or distance) for each of the arcs between each pair of nodes. Finally, a bound is defined for each node-to-hub or hub-to-node arcs. In hub covering problems, one of these three criteria is considered while minimizing the fixed hub installation costs. The p-hub maximal coverage models, which is one of the extensions of the covering type hub location problems, are examined the optimal locations for hubs in order to maximize demands within a set coverage distance with a fixed number of hubs.

- *p-Hub Center Models:* Similar to hub covering problems, p-hub center problems also consider the distance or transportation time between two pairs of nodes. It is expressed the minimization objective function of the maximum distance between two origin-destination nodes defined as i and j (minmax type problems). The maximum distance or time considered in p-hub center problems can be define in three different perspectives. The first of these, the maximum time or distance value considered and among all origin-destination pairs the maximum one is minimized. Secondly, arcs over the entire network are taken into account, (hub and access arcs) and finally, maximum values are minimized for access arcs.
- *p-Hub Dispersion Models:* The objective of the p-hub dispersion models is to locate p hub facilities on a network in such a way that the minimal separation distance between any origin-destination pair is maximized. Generally, this type models applies to locations that constitute a risk to each other, as well as retail or service franchise networks.

Multi-objective problems can also be handled by considering the above-mentioned problem types with different objective functions within the same model (Musavi & Bozorgi-Amiri, 2017). There are different and more sophisticated variants of objective functions have also been discussed in the literature. Especially in recent papers, many concepts such as equality, competitiveness, robustness, sustainability, congestion and reliability based objective functions frequently considered. For instance, Mahmoodjanloo et al. (2020)

proposed a competitive based multimodal hub location and pricing model. In this study, they investigate the design of transportation network for an entrant company with flexible customer demand, where an existing transportation company operates its main distribution network. The objective function is aimed at maximizing the profits of the new entrants to the market. Mohammadi et al. (2016) presented a reliable and consistent supply chain by utilizing a hub location network that is less vulnerable to disturbances in the hubs. To reduce the entire sum of nominal and predicted failure costs, the authors suggested a single-objective mixed-integer programming (MIP) model. Bütün et al. (2021) proposed a capacitated directed cycle hub location and routing problem under congestion. Unlike other most known models, the objective function of this study includes hub congestion costs as well as distribution, collection and fixed costs. Wang et al. (2020) focuses on hub location models from the perspective of multiple commodities distributions under demand and cost uncertainties in capacitated and uncapacitated hub location problems. They use a penalty cost for commodity demand that is not met in the objective function. Similarly, the other model based on robustness, Zetina et al. (2017) considered both uncertain demand and transportation costs and hub location models. Golestani et al. (2021) discussed hub location models in environmental perspectives. They proposed bi-objective green hub location model for multiple perishable products with various storage temperatures. The first objective in the model consists of minimizing the sum of transportation, distribution, carbon emission and cooling costs. The second aim is based on maximizing product quality. Extended versions of hub location models are also discussed in detail in the following sections.

2.3.Fixed Costs

The opening cost of hubs is not considered in p-hub median problems. The single assignment hub location problem with fixed costs was initially put up by O'Kelly (1992), and it was afterwards improved extensively. The objectives are to reduce operational costs is used to identify the number of hubs to be developed as the model incorporates the cost of constructing hubs into account. Furthermore, hub capacity characteristics might be incorporated into the model construction. Aykin (1994) provided capacitated variations of the problem, which permitted direct connections in circumstances when capacities prevented information flow via hubs. Campbell (1994) was able to develop formulations for all versions of the problem using linear model. Nickel et al. (2001) created a model that incorporated fixed costs for both hub and spoke fixed connections costs.

Yaman (2005) proposed modular capacity and investigated polyhedral problem outcomes. To describe the fixed cost of utilizing an edge, Yaman & Carello (2005) suggest a stepwise cost function. Abdinnour-Helm & Venkataramanan (1998) illustrate previous HLP solution methods with fixed costs. They demonstrated the Branch and Bound (B&B) technique as well as the genetic algorithm. Ernst & Krishnamoorthy (1999) investigated simulated annealing fixed cost variants of hub location problems, which offers upper bound and random descent, in order to find the upper bound and random descent heuristic. Ebery et al. (2000) presented shortest path heuristic methods in their study. Mayer & Wagner (2002) presents a B&B methodology and they find better lower bounds for tighter formulations as well as improved computing performance of the algorithm. Topcuoglu et al. (2005) and Chen (2007) two other studies that focused on fixed cost based HLPs were proposed genetic algorithm solution approaches. The heuristic presented by Chen (2007) outperformed the genetic algorithm developed by Topcuoglu et al. (2005) in terms of both quality of the solutions and solution time.

Labbé & Yaman (2004) provided several valid inequalities on the hub location problems with fixed costs that were facet-defining in form. Marín et al. (2006) redefined their previous formulation by removing the requirement that satisfy the triangular inequality of cost (Marín, 2005b). A branch-and-cut model was implemented by Labbé et al. (2005), who introduced the concept of hub capacity in terms of the amount of traffic that flows through it. Rodríguez-Martín & Salazar-González (2008) demonstrated that Benders decomposition algorithm becomes ineffective when compared to Double Benders decomposition algorithm or their branch and cut algorithms based on decomposition methods. Another finding of this article is that the suggested algorithm is capable of addressing capacitated HLP types that are not solvable by commercial MIP solvers.

Path-dependent variables were included into the model by Contreras et al. (2009a; 2009b) and they lead to tight lower bounds through a Lagrangean relaxation. Lagrangean relaxation and column generation methods, along with a branch-and-price algorithm, could successfully solve cases up to 200 nodes.

According to (Correia et al., 2010), the optimum capacity level for each hub is established based on a set of capacity levels. Based on their concept, the hubs should be of a certain size. The authors note that the capacitated single allocation HLP, as described in the literature, is ignoring certain critical information, which is necessary to get an optimum solution, and which is therefore infeasible. They redesigned the problem, which resulted in the adoption of a new set of constraints. Contreras et al. (2012) focus on determining of hub capacity levels in their model. In addition to splitable and non-splitable commodities, a new definition of demand is given as well.

Environmental issues were examined by O'Kelly (2012) in relation to the hub location problems, with the fuel consumption and aircraft cost being the primary objectives. It was suggested by Taherkhani & Alumur (2019) to use r-allocation modeling with the objectives of profit maximization while considering both traditional operational costs and routing costs.

2.4. Network Topology

In hub location problems, it is generally assumed that there is a direct connection between all hub pairs. However, this is not possible in real life problems. At the same time, it is possible to provide almost the same service quality level of complete hub networks structures in terms of cost and service time with an incomplete network designed in efficient manner. Thus, incomplete hub networks have been attract researcher’s attention in recent years. Table 2.1 represents current hub location literature according to the network topology and problem types.

Table 2.1. Hub Network Topology and Problem Types

	Incomplete	Star	Tree	Line	Hierarchical	Cycle
Nickel et al. (2001)	UMAHLP					
Lin and Chen (2004)					UMAHLP	
Yaman et al. (2007)	USAp-HCP					
Thomadsen and Larsen (2007)					UMAHLP	
Yaman (2008)		CSAp-HMP				
Labbé and Yaman (2008)		USAHLP				
Martin and Gonzalez (2008)	CMAHLP					
Gelareh (2008)	UMAHLP					
Yaman (2009)					USAp-HMP	
Alumur et al. (2009)	USAp-HMP,HCOP, HCP					
Calık et al. (2009)	USAHCOP					
Contreras et al. (2009b)			USAp-HLP			
Lin (2010)					UMAHLP	
Campbell (2010)	UMAHALP					
Contreras et al. (2010b)			USAp-HMP			
Alumur et al. (2012b)						
Alumur et al. (2012a)					USAHp-COP	
Davari and Zarandi (2012)					USAp-HMP	
Yaman and Elloumi (2012)					USAp-HMP	
Yaman and Elloumi (2012)					USAp-HCP	
Martins de Sá et al. (2013)			USAHLP			
Lüer-Villagra and Marianov (2013)	UMAHLP					
Davari et al. (2013)	USAHCOP					
Contreras et al. (2013)						USAHLP
Moghaddam and Sedehzadeh (2014)			CSAHLP			
Karimi and Setak (2014)	UMAHLP					
Contreras et al. (2015)						USAp-HLP
Fontes and Goncalves (2015)					UMAHLP	
Martins de Sá et al. (2015a)				UMAHLP		
Martins de Sá et al. (2015b)				UMAHLP		
O'Kelly et al. (2015a)						
Campbell et al. (2015)						
Alibeyg et al. (2016)						
Contreras et al. (2016)						USAp-HLP
Camargo et al. (2017)						
Rabbani et al. (2017a)					USAHLP	
Dukkanci and Kara (2017)					USAHp-COP	

	Incomplete	Star	Tree	Line	Hierarchical	Cycle
Alibeyg et al. (2017)						
Real et al. (2018)					USAHLP	
Akgün and Tansel (2018)	UMAp-HMP					
Martins de Sá et al. (2018b)	UMAHLP					
Martins de Sá et al. (2018a)	UMAHLP					
Chen et al. (2018)		USAp-HCP				
Blanco and Marín (2019)			UMAp-HLP			
Dai et al. (2019)	UMAHLP					

In this context, Alumur et al., (2009) proposed models for incomplete p-hub median, p-hub center and hub covering problems. In this study, the number of hub connections are bounded by a constant q value. In other words, the number of hub links is a fixed with a constant q value and the formulation provided that all hubs have at least one hub connection. Gelareh and Nickel (2011) also focus on the incomplete hub location problems, and they relaxed direct node-to-node transportation assumption. Also, they consider fixed and variable hub and link setup costs. In this study, while the exact solutions are obtained with primal benders decomposition approach up to 50 nodes, greedy heuristic is used for larger size instances.

In the study presented by Kelly (2015), there is a three-index incomplete hub location model. Also, in this study, the effect of fixed hub installation costs on hub positions is analyzed. Similarly, Camargo et al. (2017) focused on incomplete hub location problems with hop constraint features. Unlike Alumur et al. (2009) study, the number of connections between hubs is not known in their proposed models, and these connections are exogenous to the minimum total cost value. Sá et al. (2018a, 2018b) presented two different studies on incomplete hub location problems. In the first study, there is uncertainty on demand and hub installation costs. In addition, direct connections between node pairs are not allowed. As the solution method, they proposed the benders decomposition methods and a constructive heuristic. In another study, there is uncertainty over service time requirements. In this study, two special benders decomposition solution methods are developed. Campbell et al. (2005) proposed a new concept based on arc positioning for hub location problems. Rather than determining hub locations, this study focuses on locating connections between hubs. The results obtained are compared with the classical four special hub location models in detail. Dai et al. (2019) proposed a heuristic solution based on cost minimization for incomplete hub location problems. The heuristic, called HUBBI, adopts the hub positioning approach according to the node pair quality between two nodes.

Contreras et al. (2010) focused on tree hub location problems. In Tree hub location problems, p hub locations are located on a non-directed network, and hub sites have at most two at least one hub connections. In the study in which various valid inequalities were proposed and solutions were obtained from AP and CAB data sets with Xpress solver. In another tree hub location study, de Sá et al. (2013) proposed three variant of benders decomposition method. In this study, a new benders cut selection scheme is proposed. They developed a solution approach based on cost information by choosing a pareto optimality cut. The presented method provides optimal solutions to problems of up to 100 nodes. Table 2.2 represents some milestone studies about hub location problems based on problems specifications and solution methods. Also Figure 2.7 shows some different hub network topologies.

Table 2.2. Hub Location Studies based on Different Network Topologies

	Direct Connection	Topology	Allocation Strategy	Capacity Constraint	Solution Methodology
Alumur et al. (2009)	X	Incomplete	S	X	cplex solver
Gelareh and Nickel (2011)	✓	Incomplete	M	X	Benders Decomposition /Greedy Heuristic
de Sá et al. (2018a)	X	Incomplete			Benders Decomposition /Constructive Heuristic
de Sá et al. (2018b)	X	Incomplete			Benders Decomposition
Kelly et al. (2015)		Incomplete			
Conteras et al. (2017)	✓	Incomplete	M		
Conteras et al. (2010)	X	Tree	S	X	Xpress Solver
de Sá et al. (2013)	X	Tree	S	X	Benders Decomposition
Dai et al. (2019)	✓	Incomplete/Cycle	M	X	HUBBI Heuristic

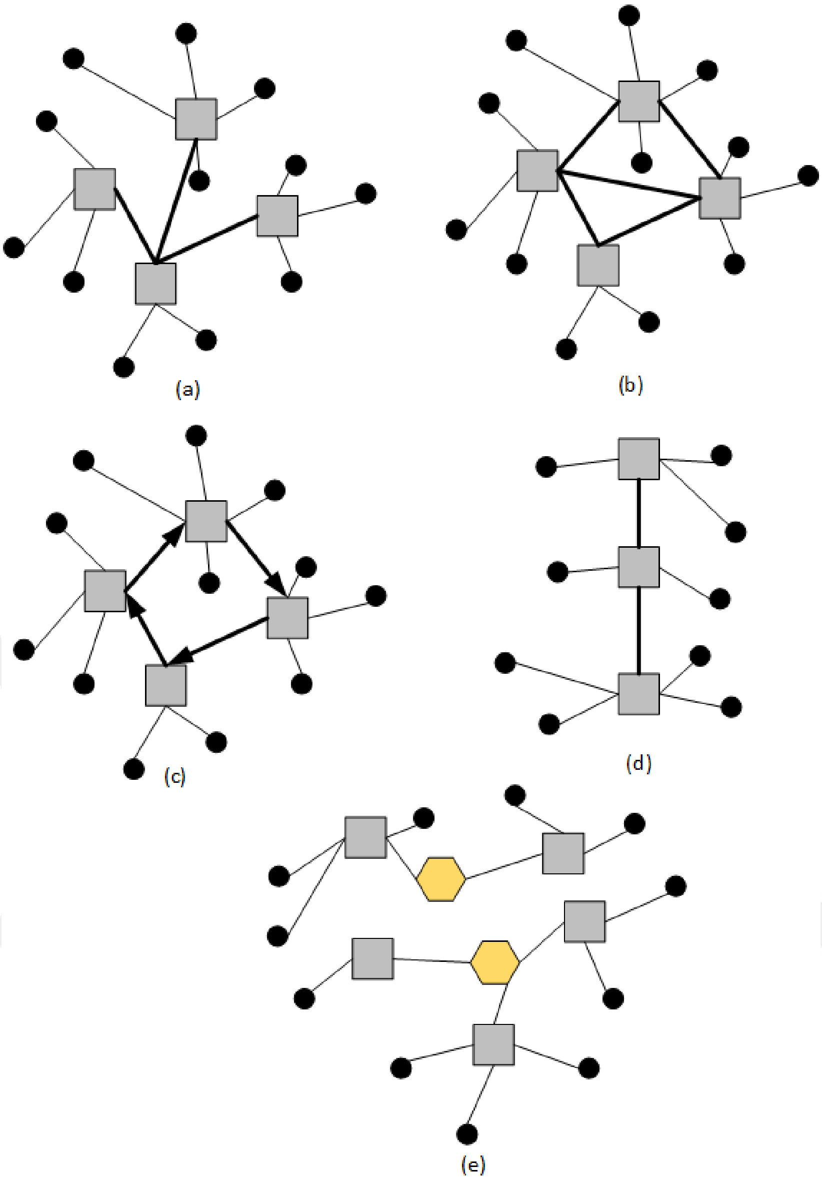


Figure 2.7. Different Network Topologies Representation

2.5.Modeling Framework of Hub Location Problems

In this section, we present the milestone formulations for hub location problems and the modeling framework of these formulations. Some presuppositions are taken into consideration in modeling hub location problems. These assumptions consist of hypothetical situations such as origin/destination flow only through hub locations, strategies of node allocations to hub locations, capacity constraints, the economies of scale implementations, fixed hub link setup costs and satisfying distances triangular inequality.

2.6.Direct Shipment Between Two Non-Hub Node

Hub location can be considered as a center where products are collected from different nodes, consolidated, and redistributed. In the hub network concept, it is aimed to provide flows collectively and thus to reduce costs in the economies of scale perspective. In this context, product flows are provided through hub locations. However, in recent studies, this assumption is relaxed. The reason for this approach is that in some scenarios (especially if two nodes are very close to each other, the cost of node-hub-node transportation is lower than the cost of node-to-node transportation) direct shipment is allowed (Dai et al., 2019; de Camargo et al., 2017; E. de Sá et al., 2018b; Taherkhani & Alumur, 2019).

2.7. Allocation Strategies

On a hub network, nodes can be assigned to hub locations with different strategies. Single allocation and multiple allocation strategies are the best known of these and in many studies, one of these assignment forms is chosen. Additionally, the r-allocation strategy has been used extensively in recent years. Although single allocation strategy requires each non-hub node to assign only one hub to send flows to the corresponding destination node, multiple allocation hub location versions allow a non-hub node to allocate to each available hub locations. Three variants of the allocation techniques may indeed be cumbersome for real-world hub location problem applications. A hub network in which all non-hub nodes are assigned to all potential hubs becomes economically infeasible. Single allocation, on the other hand, it unnecessarily increases transportation costs (since vehicle routes getting longer). Thus, Yaman (2011) presented a new formulation on r-allocation hub network structure. The proposed formulation concept limited the number of non-hub nodes assigned to hubs to such a predefined threshold value (r). Furthermore, some current research investigated to r-allocation modeling to propose solution methodologies based on heuristics. Figure 2.8 represents the percentage of studies based on allocation strategies. Also, Table 2.4 indicates some milestone studies based on allocation strategies and problems types.

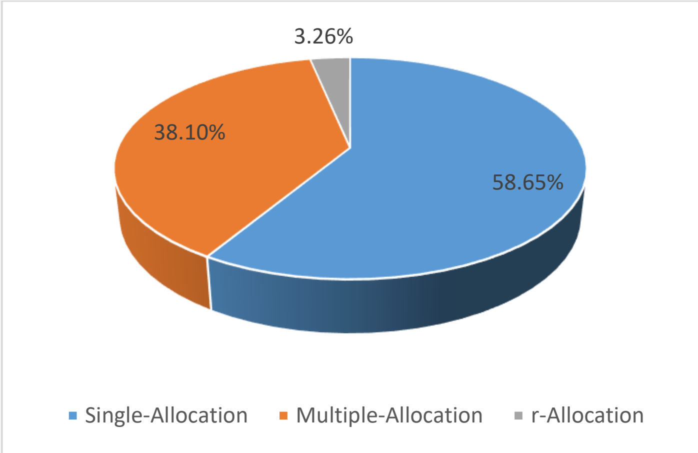


Figure 2.8. Article publication rate according to the allocation strategies

2.7.1. Single Allocation Strategies

In hub location problems, if each non-hub node is assigned to only one hub location, a single allocation network structure is obtained. In this network structure, let i and j be origin and destination nodes and if there is a flow between these nodes the f_{ij} parameter defines the amount of flow between i and j. We use the x_{ik} binary decision variable to define the node type and non-hub nodes allocated to hub node k. If node i is assigned to hub location k, then $x_{ik} = 1$. If $i = k$ equality is provided, then k can be defined a hub location from the equation $x_{kk} = 1$. In the single allocation strategy, at least one hub location in the flow route between the origin node i and the destination node j is fixed (if i and j are allocated to different hub locations, at least two hub locations are fixed). In the quadratic programming model originally presented by O'Kelly (1987), the single allocation strategy is used, and capacity constraints are not considered for hubs and links (arcs and hub connections). In the model that focuses on minimization of total transportation costs (min-sum criterion), and number of fixed p hub locations are considered (number of hub locations defined exogenously). In addition, all hub locations are completely connected to each other and there is no fixed hub installation cost. A coefficient α is used in hub-to-hub transportation. α coefficient, which takes a value between 0 and 1, represents the reduction of transportation costs based on economies of scale. This is fact that node-to-hub or hub-to-node transportation costs are smaller than hub-to-hub transportation costs. In this context, the model is as follows:

$$\min \sum_i \sum_j \sum_k f_{ij} x_{ik} c_{ik} + \alpha \sum_i \sum_j \sum_k \sum_m f_{ij} x_{ik} x_{jm} c_{km} + \sum_i \sum_j \sum_m f_{ij} x_{jm} c_{jm} \qquad (2.1)$$

s.t.

$$x_{ij} \leq x_{jj} \quad \forall i,j \in N \qquad (2.2)$$

$$\sum_j x_{ij} = 1 \quad \forall i \in N \qquad (2.3)$$

$$\sum_j x_{jj} = p \qquad (2.4)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in N \qquad (2.5)$$

Eq. (2.1) minimizes the total transportation costs of node-to-hub, hub-to-node and hub-to-hub. The first part of the objective function refers to the flow costs from nodes to hub locations. The second part of the function, on the other hand, defines the sum of transportation costs between hubs, considering the economies of scale. The last part of the objective function is the costs of distribution from hub locations to the nodes. Eq. (2.2) indicates that node j must be a hub for the node i to be assigned to the node j hub. Eq. (2.3), on the other hand, allows each node to be allocated to only one hub location. Eq. (2.4) means that the total number of hubs must be p. Finally, Eq. (2.5) defines the binary variables. However, there is an extra quadratic term in the second part of the objective function (hub-to-hub flows). This situation makes the problem difficult. There are linearized model suggestions in the related literature.

Skorin-Kapov et al. (1996) proposed path-based formulation for single allocation uncapacitated hub location problems (USAPHLP). In this model, let the origin node be $i \in N$ and the destination node be $j \in N$. Also, the flow between i and j be through the hub locations k and m. In this case, the decision variable y_{ijkm} is equal to 1 and otherwise it is equal to 0. The USAPHLP model is as follows:

$$\min \sum_{i,j,k,m \in N} f_{ij} c_{ijkm} y_{ijkm} \tag{2.6}$$

s.t.

$constraints\ (2.2 - 2.3, 2.5)$

$$\sum_{k \in N} y_{ijkm} \le x_{kk} \qquad \forall\, i,j,m \in N \tag{2.7}$$

$$\sum_{m \in N} y_{ijkm} \le x_{mm} \qquad \forall\, i,j,k \in N \tag{2.8}$$

$$\sum_{k, \in N} y_{ijkm} = x_{ik} \qquad \forall\, i,j,m \in N \tag{2.9}$$

$$\sum_{m \in N} y_{ijkm} = x_{jm} \qquad \forall\, i,j,k \in N \tag{2.10}$$

$$y_{ijkm} \ge 0 \qquad i,j,k,m \in N \tag{2.11}$$

2.7.2 Multiple Allocation Strategies

In hub location problems, a node can be associated with more than one hub region. That is, the flow from or entering a node can be provided through different hub locations. These types of problems are called multiple-allocation hub location problems and have been studied by many researchers in the literature. Considering that origin destination nodes can be connected to more than one hub region in multiple-allocation hub location problems, path-based models with fewer variables can be presented than a single assignment hub location problem.

Table 2.3. Some Milestone Studies Based on Allocation Strategies and Problem Types

	SAHLP	MAHLP	SApHMP	MApHMP	p-HCP	p-HCP
O'Kelly (1986;1987)			X			
Aykin (1990)			X			
Klincewicz (1991, 1992)			X			
J. F. Campbell (1992)				X		
M. E. O'Kelly (1992)	X					
J. F. Campbell (1994)	X	X	X	X	X	X
Aykin (1994)	X					
Skorin-Kapov et al. (1996)				X		
Ernst & Krishnamoorthy (1998b)			X			
O'Kelly et al. (1996)			X			
Klincewicz (1996)		X				
J. F. Campbell (1996)			X			
Sohn & Park (1998)			X			
Sasaki et al. (1999a)				X		
Ernst and Krishnamoorthy (1999)	X					
Ebery et al. 2000a)		X				
Ernst and Krishnamoorthy (2000)					X	
Kara & Tansel (2000)					X	
Nickel et al. (2001)		X				
Kara & Tansel (2001)						X
Ebery (2001)			X			
Kara & Tansel (2003)						X
Wagner (2004)						X
Boland et al. (2004)				X		
Hamacher et al. (2004)		X				
Labbé & Yaman (2004)	X					
Yaman & Carello (2005)		X				
Yaman (2005)		X				
Marín (2005)		X				
Labbé et al. (2005)		X				
Kimms (2006)				X		
Yaman et al. (2007)					X	
Campbell et al. (2007)					X	
Wagner (2008b)						X
Costa et al. (2008)	X					
Ernst et al. (2009)					X	
Gavriliouk (2009)					X	
O'Kelly (2009)			X			
Alumur & Kara (2009)						X
Correia et al. (2010)	X					
Yaman (2011)			X			
Puerto et al. (2011)			X			
O'Kelly (2012)		X				
Yaman and Elloumi (2012)					X	

	SAHLP	MAHLP	SApHMP	MApHMP	p-HCP	p-HCP
Hwang & Lee (2012)						X
Campbell (2013a)			X			
Campbell (2013b)				X		
Liang (2013)					X	
Lowe and Sim, (2013)						X
Peker et al. (2015)			X			
Peker and Kara (2015)						X
Meier et al. (2016)			X			
Ernst et al. (2018)						X
Taherkhani and Alumur (2019)		X				
Corberán et al. (2019)				X		
Dukkancı et al. (2019)						X
Čvokić et al. (2020)	X					
Brimberg et al. (2020)					X	
Atta and Sen (2020)		X				
Atta and Sen (2021)	X					
Monemi et al. (2021)		X				
Momayezi et al. (2021)	X					
Öztürk et al. (2021)			X			
Total	10	13	16	8	9	11

2.8. Capacity Constraints

In many studies on hub location problems, capacity limitations are another consideration. The capacity constraint in hub location problems can be examined in three basic frameworks. The first of these is the capacity at hub locations and is the most common capacity constraint. The second is the capacity of the hub network or access links, and in the network design each arc has a flow capacity (depending on its type). Finally, vehicle capacities can be considered in relation to arc capacities. The number of vehicles or vehicle capacities that carry out transport on a particular arc can be defined as constraints. Apart from all these, the flows on the hub network are splittable or unpartable. Especially in multiple allocation strategies in hub location problems, if a product is not splitable, transportation is provided over a single route between origin-destination pairs. This has an impact on costs. This has a serious impact on costs because one of the main purposes of the multi-allocation hub location problems is cost minimization over product splittability.

Early studies for multi-allocation and capacity constrained hub location problems were presented by Campbell (1994), Ebery et al. (2000), Boland et al. (2004). The common feature of these studies is that the input-output flow to the hub node has a certain limit value. In other words, hub nodes have a capacity constraint. In another model proposed by Bryan et al. (1998), the capacity constraints are applied on hub lines instead of hub vertices.

Marine (2005) formulated a model in which goods are splittable, while Rodriguez-Martin and Salazar-González (2008) provided a concept in which splittable goods can be supplied to their destinations by alternative pathways. Apart from these, there are studies in which non-hub nodes and access arcs on the network are designed over capacity constraints.

There are also studies that considering the capacity constraint in single allocated hub location problems. Early hub location studies with single assignment and capacitated version integrated studies, Ernst and Krishnamoorthy (1999), Contreras et al. (2011d), Contreras et al. (2009a), Correia et al. (2010), Campbell (1994b), and Labbé et al. (2005) as well. All of these single allocation and capacity restricted hub position designs take the maximum inbound and outbound traffic over the hub node into account. In the studies proposed by Aykin (1994, 1995), a capacity constraint is also considered on the direct transportation link between origin-destination pairs, as well as the amount of flow to and from the hub node. There are capacitated HLPs with modular connection capabilities in various research. Capacity restrictions on incoming and outgoing traffic at hubs were examined in these studies (Carello et al., 2004; Yaman and Carello (2005); Yaman (2008)).

Most of the above models, based on capacity constraints, assume that node and link capacities are determined exogenously. In other words, potential hub locations and connection capacities between hubs are predetermined and used in the model. Considering that capacities have decisive effects on hub locations and connections between hubs, it can be seen that innovative models are needed. Some researchers state that hub and hub link capacity levels should be part of the decision process. Correia et al. (2010) determined hub capacities as a decision variable in their model. In this context, they developed a capacity-constrained hub location model based on the single allocation strategy. Elhedhli and Wu (2010) are introduced a capacitated model in which hub capacity is also a decision variable. Contreras et al. (2012) are presented several models with multiple assignments in which the amount of capacity installed at the hubs is part of the decision process, for both splittable and non-splittable commodity cases. Similarly, Elhedhli and Wu (2010) defined hub capacity as a decision variable in their study. Contreras et al. (2012), on the other hand, developed several models based on multiple-allocation strategies, in which they include hub capacity levels in the decision process for both divisible and non-divisible products.

2.9. Economies of Scale

Economies of scale are a crucial concept for any company in any industry since they reflect the cost reductions and gain competitive advantage that larger businesses have over smaller ones. There are several reasons why economies of scale lead to reduce per-unit costs. Initially, manpower specialization and more integrated technology increase output quantities. Additionally, reduced per-unit expenses might result from supplier bulk orders, greater promotional purchases, or a lower cost of capital. Furthermore, spreading internal function expenses across a greater number of units produced and sold contributes to cost reduction.

Similar economies of scale approach can be applied to transportation models. Economies of scale relate to an average cost curve that slopes downward as the size of the transportation commodity amount. Because of the availability of economies of scale, as the size of the transportation amount increases, the average or unit cost decreases. In terms of transportation amount, economy of scale is one of the most important issues in hub location problems. The financial benefit for hub and spoke networks is provided by economies of scale (mostly owing to the aggregation of flows on interhub arcs). Presently, studies of hub location in operations research and transportation do not fully reflect the scale economies that occur as a result of flow aggregation.

Approaches in which the flow amounts between hub locations are handled within the framework of discounted cost can provide appropriate results when related over rapid transportation modes. However, in realistic approaches, economies of scale discount factor should be flow dependent. Fixed discount costs can lead to the fact that flows on the network are routed only between hub connections. For this reason, it can also be applied flow-dependently on the node-to-hub or hub-to-node arcs of the discount factor. For this reason, existing approaches employ rather simplified economies of scale. It is also quite common to use the same economies of scale coefficient, even though the flow amounts are different on each hub-to-hub connection. Therefore, hub-to-hub transportation costs that are calculated independently of flow may be calculated incorrectly. In addition, this situation is also effective on node assignments and hub selection decisions. Drawing attention to this situation, the researchers drew attention to the use of such flow-independent fixed discount factors and presented formulations for flow-dependent discount factor applications. In O'Kelly & Bryan (1998), the first hub location model that specifically accounted for economies of scale by enabling discount factors on hub arcs to be a function of flows was developed. To calculate the transportation cost in each hub link, this formulation, known as FLOWLOC, employs a non - linear objective functions in which costs grow at a declining rate as flows rise. The cost is believed to be smaller than the linear cost associated with a fixed discount factor for any quantity of flow. To provide a linear integer programming model for the subject, this function is modeled by a piece-wise linear linear function. Authors present several FLOWLOC formulation modifications that remove the assumption of complete connectivity between hubs by employing a minimal predefined threshold to activation a hub to hub line and include a flow amount dependent cost function for both the hub and access arcs. The FLOWLOC model may be simplified to a conventional problems, as demonstrated by Klincewicz (2002), once the hub locations are determined. Different nonlinear flow cost models are presented by Horner & O'Kelly (2001) depending upon arcs performance metrics that are often utilized in urban transportation design. In both hub and access arcs, flow-dependent costs are modeled using this equation. For the purpose of designing inter - modal transportation systems for freight lines, Racunica & Wynter (2005) investigate an adaptation of HLPs. Their approach solely models flow-dependent reduced costs on the transport and redistribution legs, using a different kind of non-linear concave function. This formula, in contrast to the FLOWLOC model, is dependent on such an efficient thresholds that takes into account the fact that reduced flow costs should be more than the linear cost up to a limit and less expensive after that. In almost all the network's connections, flow-dependent reduced costs are modeled using a new method by Kimms (2006) that is based on fixed-charge cost functions that are frequently employed in other hub system design problems. The offered function consists of fixed setup and variable costs. Fixed installation costs are not flow-dependent, while variable costs are defined as flow-dependent. In this article, two models are presented, capacity-constrained and non-capacity-constrained. In addition, the capacity constrained model has been extended in a multimodal framework according to different transportation models. In order to simulate flow-dependent costs on both hub and access arcs, Meier (2017) takes into account a stepwise function. In addition, in another study where piecewise-linear costs were considered, economy of scale application was discussed on p-hub median problems (Lüer-Villagra et al., 2019). Also, most vehicle routing problems simulate the shipping costs using calculations of this kind (Laporte, 2009).

2.10. Solution Approaches and Data

Hub location problems are difficult to solve (especially for large-scale problems) due to the complex nature of the network design process. Many exact solution methodologies have been presented in the literature for solving hub location problems. However, due to the difficulty of hub location problems, heuristic approaches also have an important place in the literature. In this section, exact and heuristic solution methods are discussed and several important studies about hub location problems complexity are conducted as well.

2.10.1. Complexity

With a few notable exceptions, it is well known that most hub location problems are NP-hard. However, there are many extended versions of hub location problems, and only a small number of complexity analysis studies have been conducted on these variants so far. (Sohn & Park, 2000), for instance, presented a proof that the uncapacitated p-hub median problem with single allocation problems (UpHMPSA) can be solved in polynomial time when the number of nodes is equal to 2 ($p = 2$). The UpHMPSA variant, on the other hand, is included in the N-hard class in cases when the number of hubs is three or more ($p \geq 3$). Additionally, Campbell et al. (2007) provided a variety of complexity analyses as well as integer programming formulations for both the uncapacitated and capacitated instances, respectively. They presented complexity results some exceptional uncapacitated instances that are polynomially solvable, such as when $\alpha = 0,\ p = 2$, and when the hub network is a tree or route.

It has been shown that the allocation sub-problem of multi-allocation uncapacitated HLPs in the complete structure is simplified to the shortest route problem when the hub locations are known in advance, and that type problems can be solved in polynomial time Ernst & Krishnamoorthy (1998a). Contreras & Fernández (2014) also showed that multiple-allocation generalized hub location problems are NP-hard.

According to Kara & Tansel (2000), the uncapacitated single allocation p-hub center location problem is NP-complete since it is a reduction of the dominating set problem, which is NP-complete. This problem is also NP-hard, as shown by Ernst et al. (2009) for the multiple assignment variant. They further demonstrate that the single allocation subproblem with respect to a particular set of hubs is already NP-hard, while the multiple allocation subproblem is not. Ernst et al. (2002) demonstrates that the uncapacitated p-hub center single allocation problem is NP-complete by reducing it to the independent transversal problem. Whenever $\alpha = 0$, the uncapacitated p-hub center single allocation problem can be solved in $O(np)$ time by finding the optimal solution. Also, they proved that the optimum solution to the single allocation 2-hub center problem can be found in $O(n^4 \log n)$ time Campbell et al. (2007). According to Liang (2013), the star p-hub center problem is substantially NP-hard. In addition, Kara & Tansel (2003) show that hub set-covering problems with single allocations are NP-hard.

Contreras et al. (2010) showed that tree hub location problems are NP-hard. Therefore, tree hub location problems are much more difficult to solve than the classical models that assume a complete backbone network. Also, in the case of cycle-star topologies, linking the hub nodes with a cycle is analogous to solving the Hamiltonian cycle problem, which is known to be NP-hard Contreras et al. (2017). On the other hand, Gavriliouk (2009) presented a complexity analysis for hub location problems over the aggregation paradox.

2.10.2. Exact Methods

Exact solution methods are preferred when attempting to obtain optimality. Exact solution methods, on the other hand, require a greater sacrifice of time and resources, as well as a more expensive computing cost. Preprocessing methods are advantageous because they focus on decreasing the size of the problems rather than on increasing the complexity of it. By initialization, Boland et al., (2004), Ebery et al. (2000b) and Skorin-Kapov et al., (1996) were able to modify tighter models. In the following years, researchers presented various studies to obtain more stringent models.

Brute force algorithms are useful for providing the best solutions for hub location problems by presenting all potential options, such as the assignment of non-hub nodes or the quantity of flows transferred between node pairs. Brute force-based solution methodologies were proposed by Aykin (1995), Ernst & Krishnamoorthy (1998a, 1998b), Abdinnour-Helm & Venkataramanan (1998), Klincewicz (2002), J. F. Campbell et al. (2003), C.-C. Lin & Chen (2004), J. F. Campbell et al. (2005b), C.-C. Lin (2010), Sasaki et al. (2014), Mahmutogullari & Kara (2016). However, the computational cost of such algorithms is quite high, and they are insufficient especially in solving large-scale problems.

Linear programming-based solution methods are widely used in the solution of hub location problems in the literature. These algorithms can be listed as Lagrange relaxation, benders decomposition, dual ascent techniques, column-row generation, and branching techniques. These approaches have been used in conjunction with heuristic methods in certain research. Cánovas et al. (2007) are presented one of the studies that uses the dual ascent methodology. A heuristic approach based on a dual-ascent methodology is intended to solve the dual problem of a four-indexed formulation. This heuristic, which is strengthened by many subprocesses and does not need an additional linear problem processor, is the fundamental tool contained in a precise branch-and-bound design. A two-phase method is presented by Meyer et al. (2009), whereby the first phase computes a collection of possible optimum hub combinations utilizing a branch and bound algorithm that is based on the shortest route principle. In allocation phase, employing a reduced-sized formulation, is then performed, with the optimum solution being returned. They are obtained good upper bound for the branch and bound problem and developed an ant colony optimization method that was used to produce a heuristic solution for the single allocation p-hub center problems. Other important studies are presented based on linear programming solution methodology are Y. Lee et al. (1996), Klincewicz (1996), Sung & Jin, (2001), Mayer & Wagner (2002).

Branching methods, such as branch and bound, branch and cut, branch and price, and so on, are the most often employed in hub location optimization problems. By splitting the original problem and obtaining smaller parts of it, branch and bound technique searches the feasible area of the problem. Puerto et al. (2013) are presented a novel formulation for the single-allocation ordered median hub location problems, as well as a branch-and-bound and cut-based method for solving this model efficiently. A hybrid optimization technique is suggested by Stanojević et al. (2015). For addressing the capacitated single allocation hub placement problem, the approach consists of an evolutionary algorithm and a branch-and-bound method. Tanash et al. (2017) developed a branch-and-bound method that employs a Lagrangean relaxation to obtain lower and upper bounds at enumeration tree nodes. For benchmark instances with up to 75 nodes, numerical results are provided. Other studies that offer solutions with the branch and bound methodology have contributed to the literature by Aykin (1994, 1995), Ernst & Krishnamoorthy (1996, 1998b, 1999), Klincewicz et al. (1998), Ebery et al. (2000a), Nickel et al. (2001), Ebery (2001), Ernst et al. (2009), Berman & Wang (2010), Ishfaq & Sox (2011), Puerto et al. (2016), An et al. (2015), Alibeyg et al. (2018).

The branch and cut methodology provide additional cutting planes by tightening the LP relaxations and improves the results achieved. With this technique, larger dimensional problems can be solved. Labbé et al., (2005) are studied the polyhedral characteristics of hub location problems and used the findings to develop a branch and cut method. García et al. (2012) are introduced a novel formulation as well as a branch-and-cut method for the multiple allocation p-hub median problems. They solve more complex instances than have previously been solved in the literature. The suggested method performs particularly well for high values of p. Rodríguez-Martín et al. (2014) proposed a mixed integer programming model for hub location-routing problems, which is strengthened by valid inequalities. They establish separation procedures for these inequalities and develop a branch-and-cut algorithm, which they verify on CAB and AP cases from the literature. The findings indicate that the formulation is valid, and that the branch-and-cut method can handle cases with up to 50 nodes. Catanzaro et al. (2011) are concentrated on partitioning hub location routing problems, and they developed an integer programming approach for addressing the problem exactly and exploring potential valid inequalities to enhance it. The other studies based on branch and cut methodology can be listed as

Yaman et al. (2007), Rodríguez-Martín et al. (2014; 2008), Labbé & Yaman (2008), Contreras et al. (2010, 2017), García et al. (2012), Yıldız & Karaşan (2015), Zetina et al. (2017), Rothenbächer et al. (2016), Boccia et al. (2018), Pearce & Forbes (2018), Quadros et al. (2018), Yıldız et al. (2021).

As can be seen, the lagrangian relaxation technique is widely utilized in the literature on hub location problems. Lagrangian relaxation is a partitioning method that offers tight bounds to decrease the problem's computational cost. The constraints that make it difficult to find a solution are eliminated from the formulation and incorporated into the objective function with dual variables using lagrangian relaxation. A combination of hyper-heuristics and relax-and-cut solution methods is presented by Danach et al. (2019), which involves relations multiple basic heuristics that are regulated and guided by a process of learning. It provides a framework for using the dual information obtained by the lagrangian relaxation approach to drive localized searching for combination method feasible solutions, hence minimizing the quantity of time consumed. Alkaabneh et al. (2019) offer a lagrangian method for obtaining tight upper and lower bounds, which they claim will provide tight upper and lower bounds. The lagrangian decomposition takes use of the problem's structure and divides it into subproblems that are convex and concave in form. Furthermore, they introduce certain valid inequalities to the lagrangian heuristic in order to speed up the rate at which it converges. Karimi & Setak (2014) provide some lower bounds for incomplete hub location-routing problems utilizing a lagrangian relaxation method and valid inequalities. Computational studies are being used to measure the performance of lower bounding implementations and valid inequalities. For some other examples for lagrangian relaxation method Aykin & Brown (1992), Y. Lee et al. (1996), Pirkul & Schilling (1998), Elhedhli & Hu (2005), Yaman (2008), Contreras, Díaz, et al. (2009), Contreras, Fernández, et al. (2009), Gelareh et al. (2010), Elhedhli & Wu (2010), Ishfaq & Sox (2011), Lin et al. (2012), He et al. (2015), Rostami et al. (2016), Neamatian Monemi et al. (2017), Dukkanci & Kara (2017), Tiwari et al. (2021), Tiwari et al. (2021)can be examined.

An additional partitioning method is the benders decomposition algorithm, which repeatedly executes the solution process of the problems derived from the original problem. Until the optimum solution is found, the master problem and sub-problems generated from the original problem are solved repeatedly, and the solutions achieved are associated to the corresponding problems. Taherkhani et al. (2020) are provided a strong deterministic formulation of the problem, and a benders reformulation is proposed to best solution large-size instances of the problem. To decompose the benders subproblem, a novel two-phase approach is established, and two effective separation procedures are derived to enhance the benders optimality cuts. By piecewise-linearizing non-linear cost components, Najy & Diabat (2020) have addressed the multiple allocation hub network design problem as a mixed-integer linear program. Since the proposed methodology is difficult to solve with standard commercial solvers, a customized benders decomposition method is developed to tackle the problem. (Mokhtar et al., 2019b) are proposed a mathematical formulation for the 2-allocation p-hub median problems, as well as a modified benders decomposition technique developed for solving the problem. This is accomplished by converting the relevant subproblems into minimum cost network flow problems. The proposed technique to deal with large sized of instances effectively. Apart from these, many studies have been presented in the hub location literature that consider the benders decomposition method (Bernardes Real et al., 2018; Contreras et al., 2011a, 2012; de Camargo, de Miranda, et al., 2009a; de Camargo et al., 2008, 2011, 2013, 2017; de Camargo, Miranda Jr., et al., 2009; de Camargo & Miranda, 2012; de Miranda Junior et al., 2011; E. M. de Sá et al., 2013b; Gelareh & Pisinger, 2011; Ghaffarinasab & Kara, 2019; Hult et al., 2014; Martins de Sá et al., 2018a, 2018b; Meraklı & Yaman, 2017; Oliveira et al., 2022; Pearce & Forbes, 2018; Rahmati et al., 2021; Rostami et al., 2018; Wu et al., 2021).

2.10.3. Heuristic, Metaheuristic and Math-heuristic Methods

Heuristic and meta-heuristic solution approaches are widely used methodologies in optimization problems and aim to obtain the best among feasible solutions. Heuristic and meta-heuristic algorithms do not guarantee the best solution. However, the superiority they provide in terms of providing fast and effective solutions makes these approaches preferable. In hub location problems, although small-sized problems can be solved effectively with exact solution methods, large-sized problems are more suitable to be solved with heuristic methods because they are Np-hard. However, while novel mathematical solutions approaches (e.g., benders decomposition and branch and pricing methods) can tackle large-scale and challenging hub location problems, extremely effective heuristic and meta-heuristic algorithms have been created to overcome computing costs. Although the computation capabilities of computers have increased today, complex processes eliminate this advantage and the problems that arise require the efficiency of heuristic methods. The hub location literature contains numerous heuristic approaches. In the previous hub location literature, the problems used heuristic methods to find the best feasible solution, particularly when such network scale is wide or the modelling is complicated. Moreover, the easy implementation of a heuristic algorithm is the main reason why it is often used in hub location literature.

In this section, the developed heuristic and metaheuristic methods are evaluated in different classes. Firstly, a classification is presented according to single and multiple allocation strategies. Then, the proposed approaches that considering fixed costs are analyzed chronologically.

Many heuristic and meta-heuristic methodologies were developed between 1987 and 2010 to solve the quadratic hub location model developed by O'Kelly (1987) and then the integer programming formulation proposed by Campbell (1994). However, the VNS-based method developed by Ilić et al. (2010) remains the best meta-heuristic method developed for pure hub location problems with uncapacitated, single allocation and complete network structure. In this context, after the study published by Ilić et al. (2010), many of the proposed heuristic methods are generally developed for problems and models that have been extended under titles such as uncertainty, reliability, congestion, transportation mode, network topology and so on.

Heuristic and meta-heuristic approaches have been developed for hub location problems based on a single allocation strategy approximately last three decades. First two heuristic methods have been proposed by O'Kelly (1987) for the single allocation-hub median problems. Both proposed heuristic approaches are based on possible selections of p-hub locations with enumeration methodology. The HEUR1 method assigns demand nodes to the closest hub location, and the HEUR2 heuristic selects the best of the objective function values of the first and second closest hub locations. Proposed heuristic methods were applied on the CAB data set. Another heuristic method on single allocation p-hub median problems was developed by Klincewicz (1991). The heuristic method, which is based on improvement through local search, is based on the single and double exchange procedure. Computational results show that the proposed heuristic method is superior to HEUR1 and HEUR2 methods presented by O’Kelly (1987). Another heuristic method proposed by Klincewicz (1992) is based on the tabu search algorithm and the greedy randomized search procedure (GRASP). In the proposed solution approaches, non-hub nodes are allocated to the closest hub location. In this study, CAB data and 52-nodes sets are considered. In addition, the number of hubs up to 10 problems are tested. Another heuristic method in which the obtained computational results were compared with HEUR1-HEUR2 (proposed by (O’kelly, 1987)) and the approach presented by Klincewicz (1992) was developed by Skorin-Kapov & Skorin-Kapov (1994). Although the computational results obtained are superior, the proposed method has a higher computational cost. The proposed method is similarly based on the tabu search algorithm, but it offers a different approach to the allocation of nodes to hub locations compared to the HEUR1 and HEUR2 methods. Then, O’Kelly et al. (1995) linearizes the quadratic objective function in single allocation p-hub median problems and provides lower bounds. They presented that the obtained results of the tabu search algorithm proposed by Skorin-Kapov & Skorin-Kapov (1994) give an average gap of 3.3% for small-sized problems (10-15 nodes) and an average of 5.9% for medium-sized problems (20-25 nodes).

On the other hand, Campbell (1996) is developed new heuristic methods for single allocation problems based on the fact that optimal solutions of multiple-allocation hub location problems provide lower bound for single allocation problems. The MAXFLO and ALLFLO heuristics proposed by Campbell, (1996) provide solutions for single allocation hub location problems based on multi-allocation hub location problems. While the presented approach different the other studies from node assignments strategies perspective, location selection decisions are similar to other algorithms.

Ernst & Krishnamoorthy (1996) proposed simulated annealing algorithm for hub location problems. In addition, they presented the obtained solutions in comparison with the tabu search algorithm proposed by Skorin-Kapov & Skorin-Kapov (1994), based on solution quality and computational cost. As a second approach, they integrated with the simulated annealing algorithm solutions into the LP-based branch-and-bound method as upper bound. They tested both solution algorithms with CAB and AP data sets. However, they could only provide solutions for problems up to 50 nodes. Also, in the study presented by Abdinnour-Helm (1998), genetic algorithm and tabu search algorithm are hybridized and used in solving hub location problems without capacity constraints. The algorithm that reached the best results found in the literature so far has been compared with the results presented by Skorin-Kapov et al. (1996). In another study, Abdinnour-Helm (2001) introduce simulated annealing algorithm to solve single allocation p-hub median problems. They compared their solutions with the tabu search algorithm (Skorin-Kapov & Skorin-Kapov, 1994), MAXFLO (Campbell, 1996), ALLFLO (Campbell, 1996) and SA algorithms (Ernst & Krishnamoorthy, 1996). In the study where similar results were obtained with the tabu search algorithm and no comparison was made regarding the solution times. Smith et al. (1996) presented the Hopfield neural network method for single allocation p-hub median problems. For this, they used the quadratic integer programming formulation, which has fewer variables and constraints and recommended by O'Kelly (1987). The obtained solutions are compared with the simulated annealing algorithm proposed by Ernst & Krishnamoorthy (1996). They showed that the Hopfield neural network method provides similar solutions with the simulated annealing algorithm.

Pirkul & Schilling (1998) proposed a heuristic based on Lagrangian relaxation for p-hub median location problems. The proposed approach reaches quality solutions in reasonable time limits. The proposed method begins with a previously proposed tight linear programming formulation and uses sub-gradient optimization on the Lagrangian relaxation of the model. The algorithm that is run on eighty-four test problems and it gives gap an average 0.048%.

Topcuoglu et al. (2005) focused on the genetic algorithm solution method in their studies where capacity constraint was not considered. In addition, they comparatively discuss the genetic-tabu hybrid algorithm and simulated annealing approaches proposed by Abdinnour-Helm (1998) and Ernst & Krishnamoorthy (1996) for validation phase of the study. However, the simulated annealing algorithm has been modified and used. In another study where tabu search and simulated annealing methods were hybridized, single allocation hub location problems were focused (Chen, 2007). The obtained solutions were compared with the genetic and simulated annealing algorithms proposed by Topcuoglu et al. (2005). It is claimed that the solution time of the proposed hybrid approach is shorter and the solution quality is better when the scale economy value (α) and problem size increase.

Kratica et al. (2007) presented two genetic algorithm-based solution approaches for single allocation p-Hub median problems without capacity. In this work, new encoding types are integrated with the appropriate objective functions. Both proposed approaches maintain the feasibility of solutions using specific representations and modified genetic operators. The second GA approach obtains all previously known optimum solutions and the best-known solutions in large scale samples (max 200 nodes). To compare the obtained solutions, the results of algorithms such as simulated annealing Ernst & Krishnamoorthy (1996) and tabu search (Klincewicz, 1992), which were previously presented in the literature, were made.

Ilić et al. (2010) presented a general variable neighborhood search approach to the uncapacitated single allocation p-hub median problems. This study uses three different neighborhood structures to calculate the current total flow in a hub network, and the updated solution calculations are done efficiently. In addition to the classical sequential based variable neighbor search algorithm, a nested based algorithm is also proposed in the study. The proposed algorithms effectively solve problems of up to 1000 nodes within satisfactory time limits.

Qin & Gao (2017) adopted the genetic algorithm approach to solve uncertain programming models in hub location models. Zhalechian et al. (2017) proposed fuzzy programming model to solve multi objective problem with uncertain parameters including capacity level, demands, transportation costs and times. Also, they developed an evolutionary based meta-heuristic to solve proposed model. Shahabi & Unnikrishnan (2014) developed a robust optimization model for uncapacitated hub network design problems for both single and multiple allocation strategies in uncertainty demand perspective.

He et al. (2015) proposed a hybrid heuristic method that combined branch & bound, Lagrangian relaxation, and linear programming relaxation methods and they used the hybrid MIP heuristic method to solve intermodal hub location problems.

Contreras et al. (2017) developed GRASP based solution methodology for cycle hub location problems. The obtained solutions are considered as initial upper bounds for exact solution methods as branch and bound method.

2.11. Data Sets

There are three most known data sets used in hub location models in the literature. These are CAB, AP and TR datasets and are frequently used in comparative analysis and measuring the performance of proposed solution methods. General information about these datasets is as follows:

CAB: The CAB data set, which was first used by (M. E. O'Kelly, 1986), refers to information received from the Civil Aeronautics Board of the United States of America. The CAB instance has nominally symmetric demand, and the distance matrices between each pair of nodes are calculated for each pair. The instances are available in five different sizes: 10, 15, 20, and 25 nodes. The distribution and collection factors and are both equal to one.

AP: Data from the Australian Post service was used to create the AP set, which was developed by Ernst & Krishnamoorthy (1996). This network includes an asymmetric demand matrix, coordinates for each node to calculate the Euclidean distance between every pair of nodes, and nominal hub fixed costs, all of which are included in the cost of the network. The distribution and collection factors and are equal to 3 and 2 respectively, whereas the discount factor is equal to 0.75 in all cases in AP data sets.

TR: The TR81 data set contains information about freight transportation between 81 cities in Turkey (Tan & Kara, 2007).

Networks of different sizes between 10 and 200 nodes are derived from the above benchmark datasets. URAND and PlanetLab are two data sets that have been utilized to solve bigger and more challenging problems in recent years, despite the fact that they are less frequently reported in the literature. The URAND data set, presented by Meyer et al. (2009) is composed of random instances of up to 400 nodes. Ilic et al. (2010) produced the instances with 1000 nodes. Flow data was created randomly with nodes at randomly generated locations from 0 to 100,000. Information on internet broadband connections is obtained via PlanetLab instances (Ilić et al., 2010). In these networks, all of the parameters (distribution, collection and economies of scale factors) are equal to 1, and the distance matrix does not satisfy the triangle inequality. The figure represents the usage rates of the most known instances in the literature.

2.12. Recent Extended Versions Hub Location Problems

In recent years, changing lifestyles and business processes have evolved in the application areas of hub location and hub network design problems under the influenced of emerging technologies. Frade & Ribeiro (2015), proposed a new location model based on a bike sharing system. In this model, operational and strategic decisions are combined such as bicycle movements (optional), number of bicycles and bicycle stations decisions. In the study, the limited budget is considered while the amount of covered demand is maximized. In the study,which focused on the problem of locating bike sharing stations in an urban area, the maximum coverage model associated with demand, and it expanded with budget and service level constraints. In another study, which also focuses on the locations of bike sharing stations, bicycle inventory level is considered (J.-R. Lin et al., 2013). The authors added the inventory decisions to the hub location problem. In the study, other considered design parameters are the number and location of bicycle stations, the user choices between the origin-destination points and the bicycle lanes between the stations. To set the design decisions in the problem, they focused costs and service levels as classical hub location models. The heuristic solution developed for the proposed system design problem is based on two basic elements. The first element is cycling stations and a greedy heuristic to identify paths between these stations. The secondary element is a heuristic method based on the purpose of cost minimization.

Drones have also been used extensively in transportation, military, and especially the last mile delivery systems in recent years. Drone hubs are considered to maximize service levels and reduce costs (oversized trucks are both costly and long transportation times). In addition, the location of drone hubs is one of the most important optimization issues of logistics systems, and the dynamics of drone hubs are different from traditional hub location models. Thus, a novel hub design problem introduces the literature (Pan et al., 2020).

After the 2010s, as social media has become popular and marketing activities are concentrated on these platforms, hub location problems may also involve social media networks. Applications in this area may also have different design features than conventional hub location models. Primarily, the problem in the digital environment can be much larger and there may be more than one layer at the design stage. Similarly, the rapid spread of wireless and modem networks, incredible advances in cellular communication technologies create new road maps with regard to problems in these research areas.

2.13. Social and Environmental Responsibility

As with other OR problems, there are studies that considered the sustainability metrics in hub location problems. Especially in recent years, awareness is raised about the importance of the concept of sustainability with climate changes, epidemics, irregularities, and human abuse (Dukkanci et al. 2019a; Dukkanci et al. 2019b). For this reason, environmental and social gains come to the fore in hub location models. Dukkanci et al. (2019) has reconstructed hub location problems by expanding them with sustainability dimensions. The environmental framework is presented with location problems in terms of minimization of emission amounts.

2.14. Humanitarian Operations from the Hub Location Problem Perspectives

Humanitarian logistic is one of the most important issues in location problems. In a disaster scenario, the flow, storage and control of aid and food supplies must be planned to protect of the vulnerable people. Thus, aid zones, meeting areas, distribution networks and evacuation zones need to be properly designed. Hub location problems for humanitarian operations may have different objectives and within this framework, network design can be at a strategic or tactical level. Locations of temporary health centers and product distribution centers and selection of shelter locations are some of them (Dönmez et al., 2021). Generally, tactical level location problems are evaluated based on post-disaster actions, while strategic level humanitarian hub locations are included in pre-disaster planning. Apart from this, location models for humanitarian operation activities may be oriented towards not only disaster scenarios, but also organizing aid activities for people in need. However, interestingly, hub location problems are a relatively new issue in the context of humanitarian logistics.

Eskandari-Khanghahi et al. (2018) proposed a multi-objective mathematical model to sustainable supply chain in blood bank context. Their model is based on probabilistic environment, and they considered blood collection facilities, distribution centers, and hospitals in facility location concept. In addition, a simulated annealing algorithm is developed for large scale problems. Similarly, Sha & Huang (2012) proposed a planning model for blood supply based on p-hub median location model. The proposed study is a multi-period location-allocation model and a heuristic algorithm based on Lagrangian relaxation model is developed.

Liu et al. (2019) proposed a bi-objective optimization model for determining medical service locations to maximize the number of survivors and to minimize operational costs in a disaster scenario. They developed an iterative ε-constraint method to solve developed model. In other studies, on the location of health centers or emergency services, Zarrinpoor et al., 2017) has developed a location-allocation model for scanning a health service network. In this model, there is the interrupted risk of health services. They used Benders decomposition-based algorithm to solve developed model. On the other hand, For activities that might be conducted in a disaster case, Ghasemi et al. (2019) suggested an uncertain multi-objective multi-commodity multi-period multi-vehicle location-allocation mixed-integer mathematical programming formulation. In the proposed model, location-allocation of facilities costs and relief shortage minimization are considered. In the study, which focuses on a real case in Tehran, three methods such as modified multiple-objective particle swarm optimization (MMOPSO), Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) and epsilon constraint method are applied to solution of the model.

Shelters are safe facilities that protect a population from possible damaging effects of a disaster. Thus, shelter site decisions are one of the most popular problems for emergency location problems. A model developed by Bayram & Yaman (2018) that simultaneously addresses the shelter location and traffic assignment decisions in an effective evacuation planning, and also it takes into consideration uncertainty situations such as time, place and scale of the disaster. Dalal & Üster (2018) also presented an emergency response network design that considers uncertainty and integrates relief and evacuation decisions. The model focuses on minimizing fixed costs from the perspective of supply center and shelter locations. Kınay et al. (2018) proposed a maximum probabilistic programming model including two types of probabilistic constraints. One of the constraints is the shelter utilization rate while the other is related to the capacity of the shelters. For the application area of the proposed study, only one district is initially considered, and in the second stage, the entire city is taken into account. Ozbay et al. (2019) also presented a study dealing with the determination of shelter locations and the allocation of the population affected by the disaster to these locations. In the study, a three-stage stochastic mixed integer programming model is proposed. In addition, capacity constraint (for shelters) is another issue that is taken into consideration. In the study, a heuristic solution is developed to solve large scale problems.

Karsu et al. (2021) focused on a network design problem to meet the clean water demand in the camps of refugee people. In the bi-objective hub location designed problem, the first objective is the minimization of costs, while the second objective is maximization of access to water. In the solution phase of the problem, they proposed two metaheuristics algorithm such as based on NSGA-II and MOEA.

Balcik & Beamon (2008) presented a study on facility decisions in post-disaster response. In the maximal covering-based model they developed, the optimization of the locations and the number of distribution centers in a humanitarian network is the focus of the study to meet the needs of disaster victims. In addition, the amount of aid material to be stored in each location is determined in the study. Namely, inventory decisions are also integrated into the maximal covering problem.

2.15. Multi-Modal and Inter-Modal Hub Location Models

The transportation of goods through an intermodal container or truck involving several or more modes or distributors from shipper to vendor is referred to as intermodal transportation. Multimodal transportation, also known as consolidated transportation, includes the use of two or three modes or shipment service providers, rather than through a single contract with a single carrier that is technically responsible for the whole operation. This carrier might not have any of the services required to complete the shipment's journey. Multimodal and intermodal transportation processes are also considered in hub location problems as vehicle routing problems. In this context, recent studies are focused on the hub network design problems that involve transportation in various modes.

In the extension of HLP models, there are several modes of transportation. The main reason for this is that service providers have various types of service based on the needs of their customers, such as same-day shipment, next-day distribution, normal delivery, and so on. According to the analysis provided by SteadieSeifi et al. (2014), HLP is the major tool used to make strategic decisions on multimodal and intermodal transportation services. The idea of multimodal and intermodal transport is intuitively linked to the use of various transportation modes to minimize cost or maximize profits. This means that each transportation mode will have its own cost structure, network accessibility, and other characteristics based on the service type (Mahmoodjanloo et al., 2020). Traditional HLP models do not have various modes of transport; however, there are some varieties that do.

Most important assumptions may need to be relaxed in multimodal and intermodal transport hub network design problems. In basic hub location problems, there are two types of locations, hub and node. However, in a real distribution networks may have several layers. Thus, multi-modal hub location models are directly related to networks in a hierarchical structure. Transport mode may vary and costs may differ in each layer of the network these type networks. In this context, while the transportation between the central hub locations is provided by the airline, the vehicle sizes in the other access nodes can be hierarchically listed as trucks, vans and delivery cars in decreasing proportion of vehicle size. In basic hub location problems, there are two types of locations such as hubs and nodes. However, in a real distribution network, the number of layers may be higher than the classical hub networks. The types of service providers between these layers are often different, and several cost types may arise from an economy of scale perspective (Alumur et al., 2012a; 2012b; Real et al., 2018; Yaman, 2009).

It is important in the synchronization between different transportation modes in multimodal and intermodal transportation. For instance, situations such as connecting flights in airline transportation, synchronization of different vehicle types in last-mile delivery operations come to the fore in multi-modal hub location models. The main purpose of providing synchronization is to increase the service level by reducing waiting times. Especially in air transport, it is the most important factor to consider together with the costs.

One of the reference studies on intermodal hub network design is based on formulations that based on fixed number of intermodal hub locations developed by Arnold et al. (2004). In another study, Groothedde et al. (2005a) deal with the multimodal hub location model from the perspective of cooperation. Truck and inland barges transport modes for the distribution of fast-moving consumer goods are considered in the proposed hub network design. They proposed a heuristic approach that works iteratively to solve the developed model. The obtained solutions emphasized that barge transportation is suitable for the stable part of the demand, however, the necessity of truck transportation for short-term demand variations. In another study, Racunica & Wynter (2005) proposed a hub location problem based optimization model in order to increase the share of freight rails in intermodal transportation. The proposed model includes a generalized location problem that allows nonlinear and concave cost functions.

Limbourg & Jourquin (2010) discussed multi-modal hub location model in a road-rail transportation network. In this context, they have developed a heuristic approach that consider the road network in the first step of the algorithm and then takes into account the cost reductions by adding railway connections to the road network. The developed method works iteratively until it cannot get a better solution.

Ishfaq & Sox (2011) proposed a model based on the multimodal and multi-allocation p-hub median approach. The model considers transportation costs, different mode connection costs and fixed location costs under the service requirements within the framework of modal dynamics. They used the tabu search algorithm to solve the model especially in large-scale problems (100 nodes). In the other study, Ishfaq & Sox (2012) proposed a mixed integer nonlinear programming model that considered several variants of the cost and travel times for each transportation modes.

Meng & Wang (2011) discussed the equilibrium approach between modes in hub network design. In this respect, they proposed a model that includes a balance behavior in the transportation process (carriers, operation managers, intermodal operators, etc.), in which many stakeholders are involved.

In another study, Alumur et al. (2012a) considered the multi-modal transportation scenario in incomplete hub network design problem. In this study, where the assumption the interconnection of all hub locations to be relaxed, and it takes into account the service level as well as the installation and transportation costs in the model. In another work of Alumur et al. (2012b), two different shipment modes are considered as ground and air transportation within the hub covering formulation. The model has cost-based objective function while guarantee delivery times are also integrated into the model.

More recently, He et al. (2015) developed MIP heuristic to solve intermodal hub location problems. The heuristic based on the hybridization of the LP relaxation, Lagrangian relaxation, and branch-and-bound methodologies. Meraklı & Yaman (2017) also considered intermodal hub location problems and they incorporate demand uncertainty based on the robust optimization perspective. They used Benders decomposition algorithms to solve the developed models. Another multimodal mixed integer linear programming model is presented by Karimi et al. (2018), the researchers proposed different cost mechanisms that including separate fixed costs and variable costs for each mode.

Mokhtar et al. (2019) extended classical hub location models by intermodal transportation framework on a sparse network structure. In this paper, they considered three different transportation modes and two hub variants, and the model has an incomplete hub network structure. The other study presented a mixed integer linear programming (MILP) model capacitated version of multimodal and multi-commodity hub location problems. The new developed model is relaxed non-direct transportation assumption between nodes in traditional hub location models (Osorio-Mora et al., 2020).

2.16. Uncertainty in Hub Location Models

One of the most important topics in decision-making problems is uncertain data, and the ecosystem of hub location problems have many uncertainties such as demand quantity, congestion, and disruption. In addition, hub network design problems complicate decision models at the strategic level. Thus, the effects of costs arising from uncertainties at this level are also greater the other levels as tactical and operational. In this context, it is necessary to deal with the uncertainty in three basic cost items such as demand, transportation, and installation costs in hub location problems. For instance, there are many factors that affect fixed hub installation costs, such as land prices, raw material prices, and taxation. Although fixed installation costs can be predicted to a certain level, the slightest change in the strategic level can result in serious financial losses. Thus, it is essential that uncertainty is included in the model and considered in hub network design problems.

Thus, the uncertainty has been an important topic in the current hub location problem literature. In this context, some studies considered uncertain flow, cost, time, and demand and discussed the hub location problems with different optimization methods (heuristic and exact). In recent years, stochastic and robust optimization approaches have been used frequently in the solution of hub location models to overcome uncertainties. In one of the first studies in this direction, M/D/c queuing systems were modeled in the hub location problem and Poisson arrivals was considered. Tabu search algorithm has been proposed for the developed model Marianov & Serra (2003). In most of the studies on transportation and location, the uncertainty is on shipment and traffic congestion Bollapragada et al. (2005). However, there are also studies that deal with different factors such as demand and costs within the framework of uncertainty. Thus, the researchers addressed the hub location problems by considering the randomness feature. Yang (2009) is focused on hub location models in airline transportation, which assume that demand is subject to seasonal changes. Contreras et al. (2011b) addressed stochastic uncapacitated hub location problems where uncertainty is related to demands and transportation costs. In the model proposed by Alumur et al. (2012), they focus on two main sources of uncertainty. These are hub fixed setup costs and shipment flow between nodes. The proposed models are solved with single and multiple allocation versions.

In recent studies, a simulated annealing and imperialist competitive metaheuristic algorithms based on chance constrained and fuzzy programming have been proposed by Mohammadi et al. (2014). Kazemian & Aref (2017) are discussed the capacitated hub location models with the uncertainty framework and presented the robust mixed integer programming model. Kaveh et al. (2021) is developed the particle swarm metaheuristic algorithm to solve the bi-objective fuzzy capacitated hub network design model. Li et al.

(2020) focus on cost minimization for single and multiple allocation hub location models in their studies where they discuss robust binary models.

2.17. Reliability

Reliability is a crucial factor while designing the network infrastructure Gavish & Neuman (1992). It refers to the capability of the network which performs a successful operation in the case of disruption. In a scenario which the hubs of a network experience a disaster and cannot serve, a hub breakdown strategy must be established Rostami et al. (2018). Because random or intended failures are often faced in real life operation of a system, therefore, maintaining operations is a vital concern (Colbourn, 1999; Kansal et al., 1995).

Kim & O'Kelly (2009) focused on many newly defined hub location problems, namely, reliable p-hub location problems which are (1) p-hub maximum reliability, (2) mandatory dispersion models. p-Hub model reliability is considered in terms of single-assignment (MRSA) and multiple-assignment (MRMA) variants. The value losses can occur during the logistic activities. To prevent events such as robbery, Hamidi et al. (2014) studied on preventive reliable hub location problem and proposed a new problem type named preventive reliable hub location problem for a safe hub network structure. For a reliable model that the properties can be kept safe during transportation, three new objects are implemented to the network: fake hub, fake allocation, fake flow. A mixed integer linear programming model is established with cost minimization objective and reliability constraints. The model is tested by using LINGOv11 with TR dataset with 5 and 15 nodes. Also, Monte Carlo simulation is used to validate the results. The optimality gap is found to be at most 8%.

An et al. (2015) focused on reliable single and multi-allocation hub-and-spoke network design models. To solve reliable hub-and-spoke network design problems nonlinear mixed integer formulations are structured. The objective function is to minimize the total cost which includes the operating cost of normal situation and disruption situation. Mohammadi et al. (2016) concentrated on reliable bi-objective single allocation p-hub center-median problem. A bi-objective single allocation p-hub center-median problem is tackled while considering all the uncertainty in flows, costs, times and hub operations.

This problem is modeled as a bi-objective mixed-integer non-linear programming. The objective function is to (1) minimize the total cost, and (2) minimize the travel time of the longest path in the network. A fuzzy-queuing approach is employed to model the uncertainties in the network. Tran et al. (2017) studied unreliable uncapacitated p-hub median problem. A mixed integer nonlinear programming model is constructed. The objective of the model is to minimize expected demand weighted travel cost and a penalty if all hubs fail. The model is linearized by using a novel network flow structure called a probability lattice. A tabu search algorithm is developed to with the purpose of finding optimal to near optimal solutions for large problems.

For the case in which the hubs of a network experience a disaster and cannot serve, a hub breakdown strategy must be established. For this reason, single allocation hub location problem which includes the reallocation of sources to a backup hub in case the hub breaks down are tackled by Rostami et al. (2018). The model of the proposed problem is established in non-linear structure with two-stage formulation. Mixed integer quadratically constrained quadratic program is used in this paper to maintain a valuable problem for their solution methodology. To solve the problem, the researchers developed a two-stage decomposition, where in the first stage the breakdown scenario is solved, and in the second scenario reacts to hub breakdown is evaluated. Experimental tests display that the presented approach leads to an essential improvement in the performance. Torkestani et al. (2018) focused on the single and multiple allocation hierarchical multimode transportation hub location problem (HMMTHLP) in the case of dynamic network disruption. A novel mixed-integer mathematical programming formulation is established for hierarchical multi-modes transportation hub location problem (HMMTHLP). Model formulation is divided into two for different scenarios. In the first scenario, optimization procedure is handled without any disruption. The second scenario solves the disruption case.

Mohammadi et al. (2019) studied bi-objective capacitated single-allocation reliable p-hub location problem (BOCSRpHLP). The single allocation p-hub location problems and the impact of uncertainty on delivery are tackled. The model works on both the complete and partial disruptions on the hubs and partial disruption on links. The purpose of this problem is modeled as a new mixed-integer non-linear programming model with two objectives.

First is to minimize the total cost including transportation cost and expected failure cost, and the second is to minimize the maximum transportation time between each pair of O-D nodes. An efficient approximation approach to provide a lower bound for the optimal Pareto-frontier. In addition, a new hybrid meta-heuristic algorithm based on self-adaptive non-dominated sorting genetic algorithm II (SNSGA-II) and variable neighborhood search (VNS) algorithm is proposed.

Rahimi et al. (2019) handled multi-objective p-hub median protection model with backup hubs under a single-assignment policy. The problem is modeled as a multi-objective mixed integer linear formulation which aims to (1) minimize the total transportation cost of a p-hub median protection model, (2) maximize the flow between each O-D pair and (3) minimize the total transportation time. This study employs Robust Possibilistic Programming. A fuzzy multi-objective decision making-based approach is developed. This approach is to solve small-sized problems optimally. The proposed model and solution approach are tested on real transportation data of Iranian Road Transportation Sector.

Shen et al. (2021) tackled reliable hub location model. In this study, a tractable mixed-integer linear program reformulation is proposed. The objective of the model is to minimize the cost function which consists of associated costs and the penalty for unserved demands. The model produces a primary path and a backup path for every O-D pair in case a disruption occurs. For the solution of the problem tackled, two effective heuristics are developed. The disruptions are forecasted based on the data of Storm Prediction Center of the National Oceanic and Atmospheric Administration (NOAA). The proposed method is tested on CAB dataset with 25 nodes. It was indicated that the proposed model barely escalates the network cost in the case of a disruption but improves the service level.

2.18. Competitive and Cooperative Hub Location Models

To the best of the authors knowledge, the first study on competitive hub location models was proposed by Marianov et al. (1999). In this study, they purpose to determine the hub locations to maximize of the flows, and tabu search algorithm developed for solving the proposed competitive hub location models. In addition, the solution methodology applied for different network structures in the study. On the other hand, in another study where two competitive carriers' contractors competed for hub locations, the bi-level programming formulations were developed for continuous hub location problems Sasaki et al. (1999b). Sasaki & Fukushima (2001) presented another study that integrates the leader approach with the follower in hub location problems. The models are based on a Stackelberg games, and the customer's behavior was considered by using a logit function. To solve the proposed models, they used the sequential quadratic programming methodology. In addition, Sasaki (2005) developed an enumeration algorithm and a greedy heuristic solution methodology for the discrete version of this problem. Wagner (2008a) on the other hand, revealed the defects of the solution method proposed by Marianov et al. (1999) in the criticism article. In this study, redundant variables have been eliminated and solution bounds have been tightened. In addition, Wagner's (2008a) proposed solution approach guarantees optimality, although it works more slowly than the Tabu Search heuristic Marianov et al. (1999). Eiselt & Marianov (2009) presented a study based on the competitive hub location problems. In the study the objective is maximizing the market share of a new entrant in the market. The output of the study is showed that the share of the new market participant can reach up to 70% in all network flows for the 25-node AP data set.

Lin & Lee (2010) presented an integrated constrained game theoretical model for time definite less-than-truckload freight services in an oligopolistic market. One of the results of the study is that the geographical central location preferences of the participants are evident. Asgari et al. (2013) analyzes the competition and cooperation strategies between three stakeholders, such as two central container hub ports and shipping companies. In the study, game theoretical network design models are presented over three different scenarios. The scenarios considered are full cooperation between hub ports, full competition between hub ports, and cooperation between hub ports and carrier companies. The data of the two most important ports of the Asian region considered for the solution of the models. In addition, the interval branch and bound solution method has been developed for the solution of the proposed models. In another study, an approach is presented that measures competitiveness on hub locations based on airlines. In this study, which aims to minimize the travel distance between hub airports, also different geographical markets are taken into consideration (Redondi et al., 2011). The other study Lüer-Villagra & Marianov (2013) focused on the hub location and pricing problems in the competitive environment. They examined the study proposed by Marianov et al. (1999) that considered market share based on profit maximization. In addition, they considered the price decision variable in the model and to solve the model they proposed a genetic algorithm.

In the one of the recent studies by Sasaki et al. (2014), companies decide on hub arc positions instead of hub location. Čvokić 2020 and Čvokić et al. (2021) is also proposed a leader-follower hub location problem in a market where prices have fixed markups. The number of hubs to be installed is not constant and actors want to maximize profits rather than market share maximization in the proposed problem. The demands are split according to the logit model in this study. Mahmutogullari & Kara, (2016) discussed a duopoly formulation in a Stackelberg approach, where two company sequentially select hub locations. Their objective is to maximize total flow on the network. The problem is modelled as a bilevel HLP and solved using implicit enumeration of the leader's problem. Ghaffarinasab et al. (2018) discussed the competitive HLPs both single and multiple allocation strategies and they define the model based on duopoly market. They proposed efficient simulated annealing algorithm to solve the two-type node assignment version of the HLPs. At the end of the study, they compare the obtained results with Mahmutogullari & Kara (2016) solutions. Tiwari et al., (2021), on the other hand, is examined the competitive hub location problem of an airline company that considers designing hub network to maximize its market share. They provided two type formulation for competitive hub location problems as mixed integer second order conic program reformulation and Kelley's cutting plane method within Lagrangian relaxation. They are considered both single and multiple allocation strategies for each model.

2.19. Application Areas

Because of its diverse and broad application areas, hub location problems have been a major study subject during the last three decades. Communication and transportation have been the primary application areas for identifying the basics of the hub location problems. Figure 1 shows that over half of the studies found in the literature are idealized general models not designed for a particular application area that we have designated as a hub network for this kind of research. Whereas other studies are used to represent the specific characteristics of the application with related constraints and objectives. Figure 2.9 shows percentage values of hub locations problem application areas.

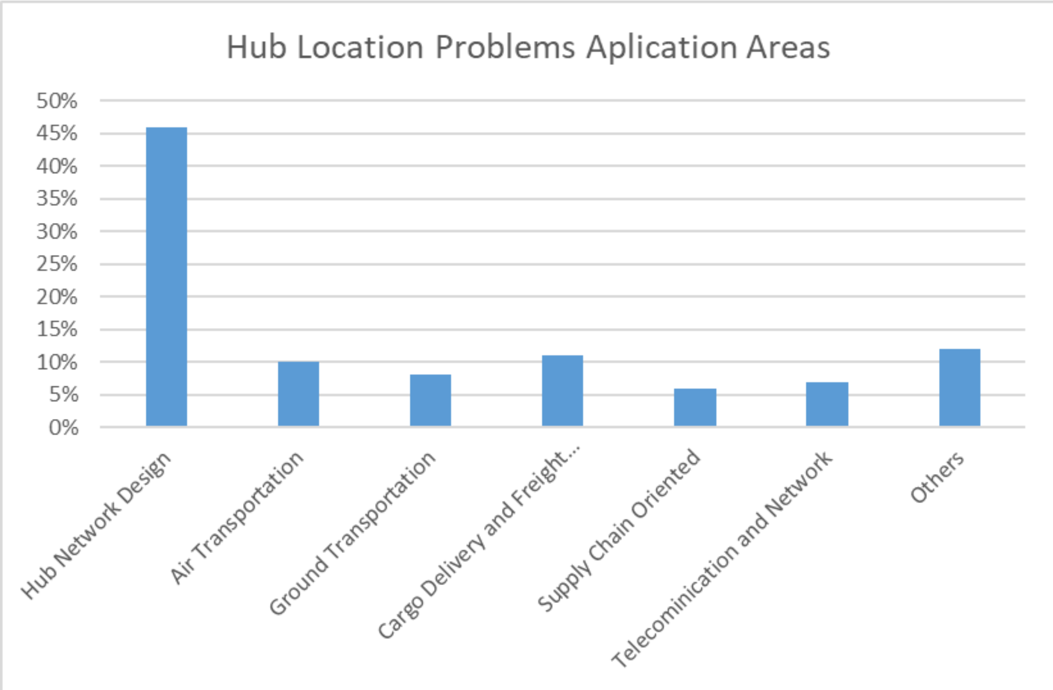


Figure 2.9. Real-life applications covered by hub location problems

Hub location problems may arise in the flow of freight and passengers when there are three main and most known transportation modes: ground (roads, rails, pipelines, and so on), water (marine, liner shipping), and air transportation. Furthermore, distribution networks, logistic and supply chain networks, and transportation are all well-suited for the application of the hub location problems due to their unique characteristics and requirement as economies of scale. Models and formulations related with air transportation are extensively discussed in the literature, with particular emphasis on the connection between cost and flow distributions, direct flights as opposed to flights with single or many stops, and topological variations. For passenger transportation, O'Kelly & Bryan, (1998) conducted research on the economies of scale in order to estimate flow dependent cost. Several studies, like Kimms (2006), Camargo et al. (2009), and Campbell (2013), which focused on cost modeling, also rely on economies of scale. Aside from that, most of the research examined at a variety of network architectures intended for air transportation, such as Sasaki et al. (1999b) for one-stop flights, Campbell et al. (2003) for isolated hubs, and Campbell et al. (2005a, 2005b) for a partially connected network.

In addition, Torkestani et al., 2018) and Teymourian et al. (2011) conducted research on the reliability of airport hubs.

Postal delivery, freight shipping, time restricted service, and emergency services are all provided via service-oriented networks. O'Kelly & Lao (2010) proposed an issue of hub location based on modular transportation for timely deliveries as an objective of the problem. Aykin & Brown (1992) combined the concept of hub location with the mail delivery services. Ambrosino & Sciomachen (2016), Ernst & Krishnamoorthy, (1999), Osorio-Mora et al. (2020), and Racunica & Wynter, (2005) conducted research on the problems concerning freight transportation. Such potential applications should primarily focus on the network's ability to provide high-quality service (Campbell, 2009; Khaleghi & Eydi, 2021).

In a similar vein, rapid transit networks and urban/public transportation are implementations that don't need the creation of completely connected hubs at the network's nodes. Accordingly, researchers such as Mahéo et al., (2019), Martins De Sá et al. (2015), Owsiński et al. (2015), Gelareh & Nickel (2011) studied at incomplete hub networks that were consistent with the applications they were studying. Investigation on road network were conducted by Meier & Clausen (2013), Catanzaro et al. (2011), Lin (2010), Campbell (2009), Li & Liu, (2013), and Cunha & Silva (2007), Hu et al. (2020) amongst others.

In telecommunications applications, there are real-world problems that can be solved while taking into consideration hub location problems. Application studies in this area are listed as Lee et al. (1996), Klincewicz et al. (1998), Carello et al. (2004), Yaman & Carello (2005), Kim & O'Kelly (2009), Thomadsen & Larsen (2007), (Labbé & Yaman, 2008), Bollapragada et al. (2006) and Sen & Krishnamoorthy (2018). Furthermore, reliability studies conducted within the context of telecommunications have a significant presence in the literature. Grover and Tipper (2005), Kim (2008), Kim & O'Kelly (2009), Kim (2012), Yıldız and Karaşan (2015) and Rostami et al. (2018) are among the researchers who have concentrated on such studies based on hub location problems.

Extended versions of hub location problems also stand out when focusing on real-life applications. Hub location problem variations also considered topological differences

(Alumur et al., 2009; Karimi, 2018; Sheu et al., 2008; Yaman, 2008), modularity assumptions (Alumur et al., 2012, 2016; Hoff et al., 2017; Momayezi et al., 2021; Vasconcelos et al., 2011; Yaman & Carello, 2005), or routing costs (Aykin, 1995; Çetiner et al., 2010; Dukkanci & Kara, 2017; Ratli et al., 2020; Rieck et al., 2014; Wasner & Zäpfel, 2004).

Distribution and logistics network have recently emerged as new possible applications for hub location topics, as demonstrated by Groothedde et al. (2005), Ishfaq & Sox (2012), Puerto et al., 2011, 2013), Lin et al. (2013), Mokhtar et al., (2019), Zhao et al. (2021), Zhao et al. (2019), Abbasi et al., (2019), Mokhtar et al. (2019), Mokhtar et al. (2019), and Mokhtarzadeh et al., (2021).

3. MODEL AND PROBLEM ANALYSIS

3.1. Incomplete Hub Location Problems

As stated in the preceding chapter, the primary focus of this work is the uncapacitated single-allocation incomplete p-Hub Median the kind hub location problems. The focus on incomplete hub structures derives from the reality that a fully connected network design does not exist, particularly in telecommunications and transportation networks. In other words, there is no direct connection or transit between all hub network pairings since connecting each hub site to each hub location separately would incur extra costs and lead in an ineffective flow of data, products, and so on. Furthermore, in terms of transportation costs, partial networks do not impose a significant cost burden when compared to fully connected networks (Alumur et al. 2009).

In general, models based on the assumption of direct linkages between hub sites were given in the literature. O'Kelly and Miller (1994) developed the use of partial hub networks. Various network design approaches are given in this research. Campbell et al. (2005a; 2005b) concentrated on hub arc distribution problems and provided different models. In these investigations, rather than identifying the hubs, mathematical formulas for identifying arcs between hubs were constructed, and optimum solutions were offered. Alumur and Kara (2008) conducted a research that included a case study on incomplete hub coverage issues. As a specific instance of hub coverage problems, the model in this study incorporates the constraint that only enables visits to three hub sites. Even in instances with the most restrictive service level constraints, they concluded that a complete hub network topology is not needed. Calik et al. (2009) presented a heuristic approach for single-allocation incomplete hub covering problems using the tabu search method in a similar context. Davari et al. (2013) developed a heuristic approach to handle incomplete hub coverage problems using a simulation-based VNS technique in a similar research. In addition, in this study, demands were considered to be unpredictable, hence fuzzy variables were used. Alumur et al. (2009) created models with imperfect network architectures for single-allocation hub location problems such p-hub median, p-hub center, and hub coverage. This research involves in-depth examinations of incomplete hub topologies.

Sá et al. (2018a) provided a new solution scheme to not fully connected hub location problems based on the benders decomposition technique. In the multiple-allocation hub location model that they investigated, direct connections between non-hub nodes are also enabled. Furthermore, they believed that there was uncertainty about node needs and fixed costs. They provided a heuristic solution technique to large-scale and complicated problems with more than 100 nodes, based on iterated local search and variable neighborhood descent algorithms. An equivalent analysis was conducted on imperfect network problems by taking service time into consideration (Martins de Sá, Morabito, & de Camargo, 2018b). Xu et al. (2018) employed this strategy to solve capacitated incomplete hub network problems to cope with the complexity. Dai et al. (2019) presented incomplete hub networks with an iterated heuristic solution strategy that focused on node pair features. They provided good results in terms of solution time and quality.

3.2. Clustering, Flow and Centrality Based Solution Approaches In Hub Location Problems

USAp-HMP problems are classified as NP-hard (Wolf, 2007). This type large problems can not solve in polynomial time because of its complexity. Thus, it is very difficult to solve large-scale hub location problems. Because it is vital to highlight links between hubs, which serve as a third decision variable. Researchers can propose solution methodologies in this respect by creating various heuristic approaches. Furthermore, there are research that provide precise problem-solving approaches. Furthermore, if the problem is incomplete, finding a solution becomes considerably more difficult. As a result, an incomplete version of USAp-HMP is included in the NP-hard class (Alumur et al., 2009).

Various researchers presented clustering-based solutions to hub location problems, rather than conventional solution methodologies (Marwah, Parti, & Kalra, 2005; Peker, Kara, Campbell, & Alumur, 2016; J. Yu, Liu, Chang, Ma, & Yang, 2009; V. F. Yu, Kuo, & Dat, 2014). These techniques identify critical characteristics for discovering hubs and develop methodologies based on these factors. Marwah et al. (2005) developed a three-phase technique. They attempted to define optimum hub sites utilizing the fuzzy C-Means clustering method and methodologies that considers the zones of effect of hubs. Peker et al. (2016) provided several prioritized strategies based on node centrality and demand quantities and sought to specify significant nodes using these methods. Clustering formations were constructed in this study by focusing on the zones of effect of major nodes and the distances between nodes. Clustering configurations were constructed in this study by focusing on the zones of effect of major nodes and the distances between nodes. Sun et al. (2017) made numerous inferences after comparing the approach presented by Peker et al. (2016) to other problem-solving strategies for hub location problems. Across CAB, AP, and TR data sets, comparisons were done based on solution gap, computation time, and memory consumption criteria. The generic contraction approach suggested by Dai et al. (2018), it reduces the number of nodes throughout the network size depending on particular parameters, and the problem is scaled down to minimize solution time. In this work, flow-based and centrality-based methodologies were used to scale down the problems. Another research looked at a strategy that focuses on costs connecting access nodes. The quantitative strategy prioritizes the quality of the network connection to be made between nodes beyond randomization, and an iterated VNS algorithm was developed in this regard Dai et al. (2019). Furthermore, the network architecture chosen to solve the problem has an incomplete structure. Yuan and Yu (2018) created an enhanced lagrangian dual algorithm to optimize a multi-mode network architecture, which included a genetic algorithm. Yuan and Yu (2018)'s research adds to the creation of an integrated model for modeling and optimization using a cluster-based multi-mode central location model with balancing constraints.

The fundamental aim and background for this research are to identify variables influencing the establishment of hub sites and to develop simple and efficient ways based on these aspects. As a result, our study was motivated by research that revealed broad features of hub zones. For example, Rodrguez-Déniz et al. (2013) emphasize the relevance of traffic between nodes in identifying hub types. The ability of a node or region to generate traffic is measured in terms of demand (i.e. passengers departing and arriving in a city) or economic measurements. Martin and Voltes-Dorta (2008) investigate the impact of geographical demand frequency and passenger linkages on hub categorization. Similarly, this research argues that solutions to p-hub median problems may be identified based on specific features. From this viewpoint, nodes are prioritized based on several parameters in order to reduce the problem scale and produce sub-sets. In this regard, the

USAIp-HMP issue was addressed using the CPLEX solver for all variants of the CAB data set (α=0.2 to α=1, p=2 to p=5, q=(p-1) to p*(p-1)/2). The next subsection discussed methods and the extrapolations from this examination.

3.3. Problem Formulation and Data

CAB data set solutions were produced to examine optimum hub locations and features of these hubs. Furthermore, the best solutions in the literature for the AP and TR81 data sets were investigated. CAB, AP, and TR are well-known data sets that are used to solve hub location problems. These data sets were used in only a few studies in the literature. TR (Turkish Postal) data set is made up of 81 nodes that correspond to 81 locations in Turkey and illustrates Turkey's postal distribution system (Alumur et al., 2009). The CAB (Civil Aeronautics Board) data collection has 25 nodes and is based on civilian airline transport in the United States (O'kelly, 1987). The AP (Australian Post) data set depicts Australia's postal delivery service and is accessible in the literature in many problem sizes ranging from 10 to 200 nodes (Ernst & Krishnamoorthy, 1999). The discount ratio factor value was employed in this analysis for the CAB data set (α= 0.2, 0.4, 0.6, 0.8 and 1). These parameters are chosen based on their frequency of expression in the literature. The number of hubs was denoted as p=2, 3, 4, 5 for the CAB data set, p=4, 6, 8, 10 for the TR data set, and p=5, 10, 15 for the AP data set. Furthermore, a solution is provided for all variants in the CAB data set based on the number of interconnections among hubs (q=p-1 to q=p*(p-1)/2). Although TR and AP data sets are expected to have a large number of variants, not all variations were considered. For example, in the AP data set, the problem was addressed for p=10, 15, 20, 25, 30, 35, 40, and 45 values. In addition, Figure 3.1 provides data visualizations.

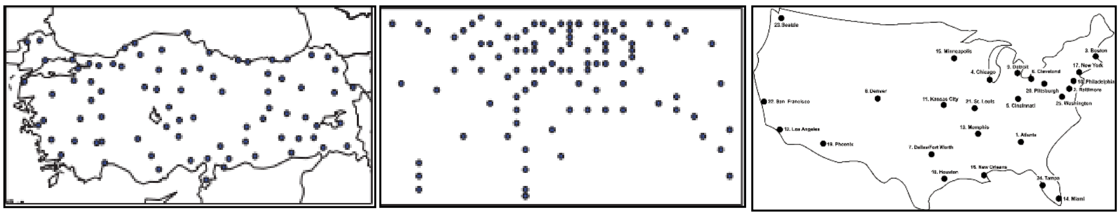


Figure 3.1. Test Data Representation on Map

The USAIpHMP mathematical model employs several definitions, such as N representing a number of nodes shows cities, regions, or centers. If the node i is allocated to the hub k, the decision variable x_{ik} equals 1, otherwise it is 0. f_{ij}^{k} denotes the overall flow rate utilizing hubs i and j as node k sources. The parameter indicates the discount factor used based on the shipping charge between the two nodes (economy of scale factor). The γ likewise represents the reduction factor used for the flow to the centers and δ represents the discount factor used for the flow from the centers. There is also $\alpha \leq \delta$ and $\alpha \leq \gamma$. Another parameter, c_{ik}, is the cost between nodes i and k. Furthermore, w_{ij} represents the amount of flow between nodes i and j.

Let $O_i = \sum_{j \in N} w_{ij}$ represent the total amount of flow from node i and $D_i = \sum_{j \in N} w_{ji}$ indicate the amount of flow to node i. In the presented model, the number of hubs is restricted to p, and the number of hub-to-hub connections is presented by q. The parameters, decision variables and model is as follows (Alumur et al., 2009):

Parameters and Decision Variables		
w_{ij}	:	Amount of flow between nodes i and j
O_i	:	Total order for node i ($\sum_{j \in N} w_{ij}$)
D_i	:	Total demand for node i ($\sum_{j \in N} w_{ji}$)
f_{ij}^k	:	overall flow rate utilizing hubs i and j as node k sources.
c_{ik}	:	cost between nodes i and k
α		Economies of scale coefficient
x_{ik}	:	1 if the node i is allocated to the hub k, 0 otherwise
z_{kl}	:	1 if the hub k connected to hub l 0 otherwise

$$min \sum_{i \in N} \sum_{k \in H} \delta c_{ik} O_i x_{ik} + \sum_{k \in H} \sum_{l \in H} \sum_{i \in N} \alpha c_{kl} f_{kl}^i + \sum_{i \in N} \sum_{k \in H} \gamma c_{ik} D_i x_{ik} \quad (3.1)$$

$s.t.$

$$\sum_{k \in H} x_{ik} = 1 \qquad \forall i \in N \quad (3.2)$$

$$\sum_{k \in H} x_{kk} = p \tag{3.3}$$

$$x_{ik} \leq x_{kk} \qquad \forall i \in N, \forall k \in H \tag{3.4}$$

$$z_{kl} \leq z_{kk} \qquad \forall k, l \in H: k < l \tag{3.5}$$

$$z_{kl} \leq z_{ll} \qquad \forall k, l \in H: k < l \tag{3.6}$$

$$\sum_{k \in H} \sum_{l \in H: l > k} z_{kl} = q \tag{3.7}$$

$$\sum_{l \in H: l \neq k} f_{lk}^{i} + O_i x_{ik} = \sum_{l \in H: l \neq k} f_{kl}^{i} + \sum_{j \in N} w_{ij} x_{ik} \qquad \forall i, j \in N, k \in H \tag{3.8}$$

$$f_{kl}^{i} + f_{lk}^{i} \leq O_i z_{kl} \qquad \forall k, l \in H: k < l, i \in N \tag{3.9}$$

$$f_{kl}^{i} \geq 0 \qquad \forall k, l \in H: k < l, i \in N \tag{3.10}$$

$$x_{ik} \in \{0,1\} \qquad \forall i \in N, \forall k \in H \tag{3.11}$$

$$z_{kl} \in \{0,1\} \qquad \forall k, l \in H \tag{3.12}$$

The objective function (3.1) is to reduce overall transportation costs. The transportation costs of flows from nodes to hubs are included in the first section of the equation. The second part displays the cost of transit between hubs as well as the alpha (α) coefficient, which represents the economics of scale. The third part of the objective function concludes with the transportation cost of flow from hubs to nodes. Equation (3.2) guarantees that each non-hub node is assigned to a single hub location. Only p hubs can be installed, according to the restriction (3.3). According to constraint (3.4), non-hub nodes can only be allocated to the hub node. The constraints (3.5) and (3.6) state that hub connections can only be formed between two hub locations. According to Equation (3.7), there are only q hub connections. The equality provided in (3.8) is meant to verify that the incoming and outgoing flows to a hub node are equal. The variables f only take a positive value with the restriction (3.9) on opened hub links. The remaining constraints represent non-negativity constraints and binary variables.

3.4. Analysis of Optimal Results on CAB Data Sets

The given model was solved using CPLEX solver on an Intel-Core i5-3210M 2.50 GHz with 6 GB RAM for the CAB data set, and the best solutions are shown in Table 3.1. Table 3.1 displays the coefficient of scale economies (α), the number of hubs (p), the number of connections between hubs (q), the solution time, the optimal solution's hub locations, the values of the optimal objective function, and the percentage changes in the objective function for the problem based on changes in the values of q. In this model, the solution time is limited to 5,000 seconds. Increases in the value of p and similarly in the value increase the solution time. Furthermore, the number of hub connections (q) has a considerable impact on problem resolution time. Increasing the alpha value and/or limiting the number of connections between hubs shortens the problem's solving time. To demonstrate, the problem-solving time for α=0.8, p=5, and q=4 is 4957.11 seconds, however the time is reduced by 210.56 seconds if q=10, that is, if q is increased from 4 to 10. Although when solving small-scale problems, such as the CAB data set, changes in the value of q in a system structure for 5 hubs increase solution time by 23 times. As a result, addressing an incomplete network problem is substantially more difficult than solving a complete network problem.

The assessment of the increases in transportation costs reveals that the largest percentage in terms of the number of connections between hubs is 5.85 percent. This proportion can be regarded acceptable given that the costs of establishing connections between hub locations and other factors impacting overall cost were not examined in this study. In this regard, connecting all hub locations is frequently redundant and unsuitable for real - life conditions. As a result, the study focused on incomplete network architectures. In this regard, the next part thoroughly examined incomplete p-hub median problems and made various generalizations using certain well-known data sets from the literature.

Variations in optimal hub locations are reported in some instances when the α value increase. Differences in optimum hub locations tend to be more sensitive to the α value of, especially when p is small. For example, whereas Chicago, Los Angeles, and New York are best hub sites for $\alpha = 0.2$ and $p = 3$, Philadelphia replaces New York for $\alpha = 0.4$ and $p = 3$. This fluctuation becomes increasingly obvious as the α value approximate to 1. For α=1 and p=3, the best hub locations are Chicago, Denver, and Pittsburgh.

Furthermore, fluctuations in the designation of optimum hub sites are anticipated when the value of q changes. For example, for =0.6 and p=3, when the number of connections between hubs is 2 (q=2), Chicago, Los Angeles, and Philadelphia are identified as hub locations; when the number of links between hubs is 3 ($q = 3$), Baltimore, Chicago, and Los Angeles are classified as optimal hub locations, with 0.158% increase in transportation costs Furthermore, for small value ($\alpha = 0.4$), all selected hub locations are equal in all combinations of q, however the designation of optimum hub locations becomes dependent to the value of q if $\alpha > 0.4$. There is a relationship between α value and the number of hub connections in this way. Furthermore, an increase in the α value of has an effect independent of q on the selection of optimum hub locations. The convergence of optimum hub areas is caused by an increase in the α value. For α=0.2 and p=2, Los Angeles and Pittsburgh are assigned as hub sites, but for α=1 and p=2, Denver and Pittsburgh are chosen. As a result, the distance between two hub locations was reduced by roughly 38%. An identical case exists for all combinations of α, p, and q. According to Peker et al. (2016), the optimum hub locations are not extremely sensitive to the α value of. However, it is possible to argue that this is not the case with incomplete hub networks.

The optimum hub placement is affected by whether the network topology connecting the hubs is complete or partial. When comparing the scenario where all hubs are connected to the case where the number of connections equals q=p-1, the proportion of identical ideal hub areas is roughly 47 %. As the number of hub sites increases, this ratio falls to around 20%. At small rates, optimum hub sites on full and partial networks are often equal. For example, comparing the case for p=5 and q=4 to the case for p=5 and q=10 reveals that the best solution has the same hub positions. However, for α=1, this is not be the case since, for small α value, optimum hub locations are assigned to areas that are far apart and hosting more flows. However, as α value grows, the closeness of hub districts to one another becomes more important than the amount of flows across hub locations. Simply expressed, whereas the value of q has little impact at the small α values, the identification of hub sites becomes increasingly sensitive to the value of q at large α value levels.

Table 3.1. Optimal hub locations for the CAB data with different values of the p, q and α

α	p	q	CPU time (sec)	Hub Locations	% increase transportation cost*	Transportation Costs
0.2	2	1	0.64	12-20-	0.000	8.55E+15
0.2	3	2	7.81	4-12-17-	2.000	6.55E+15
0.2	3	3	6.27	4-12-17-	0.000	6.55E+15
0.2	4	3	204.92	4-12-14-17	2.595	5.52E+15
0.2	4	4	11.61	4-12-17-24	0.481	5.40E+15
0.2	4	5	3.67	4-12-17-24	0.026	5.38E+15
0.2	4	6	1.63	4-12-17-24	0.000	5.38E+15
0.2	5	4	303.65	4-7-12-14-17	4.035	4.79E+15
0.2	5	5	62.39	4-7-12-14-17	2.282	4.71E+15
0.2	5	6	27.67	4-7-12-14-17	0.826	4.64E+15
0.2	5	7	11.8	4-7-12-14-17	0.322	4.61E+15
0.2	5	8	2.13	4-7-12-14-17	0.034	4.60E+15
0.2	5	9	1.77	4-7-12-14-17	0.000	4.60E+15
0.2	5	1	1.83	4-7-12-14-17	0.000	4.60E+15
Average			46.27		0.900	
0.4	2	1	1.51	12-20-	0.000	9.41E+15
0.4	3	2	59.95	4-12-18-	0.089	7.71E+15
0.4	3	3	62.05	4-12-18-	0.000	7.70E+15
0.4	4	3	641.83	4-12-14-17	3.165	6.95E+15
0.4	4	4	109.53	1-4-12-17	0.831	6.78E+15
0.4	4	5	44.56	1-4-12-17	0.041	6.73E+15
0.4	4	6	41.27	1-4-12-17	0.000	6.73E+15
0.4	5	4	1260.41	4-7-12-14-17	5.859	6.42E+15
0.4	5	5	657.09	4-7-12-14-17	3.338	6.25E+15
0.4	5	6	211.86	4-7-12-14-17	1.154	6.11E+15
0.4	5	7	61.05	4-7-12-14-17	0.446	6.07E+15
0.4	5	8	31.78	4-7-12-14-17	0.051	6.05E+15
0.4	5	9	23.44	4-7-12-14-17	0.005	6.04E+15
0.4	5	1	18.88	4-7-12-14-17	0.000	6.04E+15
Average			230.37		1.070	
0.6	2	1	4.86	12-20-	0.000	1.03E+15
0.6	3	2	261.41	4-12-18-	0.158	8.84E+15
0.6	3	3	197.66	2-4-12-	0.000	8.83E+15
0.6	4	3	1821.84	4-12-14-18	4.137	8.37E+15
0.6	4	4	542.36	1-4-12-17	1.044	8.11E+15
0.6	4	5	366.05	1-4-12-17	0.052	8.02E+15
0.6	4	6	219.09	1-4-12-17	0.000	8.02E+15
0.6	5	4	3721.03	4-12-14-17-	5.523	7.92E+15
0.6	5	5	1416.48	4-11-12-14-	3.202	7.73E+15
0.6	5	6	546.02	4-7-12-14-17	1.395	7.59E+15
0.6	5	7	374.58	4-7-12-14-17	0.540	7.53E+15
0.6	5	8	294	4-7-12-14-17	0.062	7.49E+15
0.6	5	9	178.94	4-7-12-14-17	0.006	7.49E+15
0.6	5	1	154.73	4-7-12-14-17	0.000	7.49E+15
Average			721.36		1.151	
0.8	2	1	11.55	12-20-	0.000	1.11E+15
0.8	3	2	616.96	2-4-12-	0.277	9.92E+15
0.8	3	3	295.56	2-4-12-	0.000	9.90E+15
0.8	4	3	2546.08	4-12-17-20	2.520	9.53E+15
0.8	4	4	1001.17	1-4-12-18	1.250	9.41E+15
0.8	4	5	393.41	1-4-12-18	0.136	9.30E+15
0.8	4	6	319.23	1-4-12-18	0.000	9.29E+15
0.8	5	4	4957.11	4-11-12-17-	3.729	9.17E+15

α	p	q	CPU time (sec)	Hub Locations	% increase transportation cost*	Transportation Costs
0.8	5	5	2723.75	4-7-12-17-20	2.735	9.08E+15
0.8	5	6	1455.67	1-4-11-12-18	1.910	9.00E+15
0.8	5	7	990.12	1-4-7-12-18	0.445	8.87E+15
0.8	5	8	668.17	1-4-7-12-18	0.176	8.85E+15
0.8	5	9	566.39	1-4-7-12-18	0.034	8.83E+15
0.8	5	1	210.56	1-4-7-12-18	0.000	8.83E+15
Average			1196.84		0.944	
1	2	1	23.64	8-20-	0.000	1.16E+15
1	3	2	1053.77	4-8-20-	0.107	1.07E+15
1	3	3	537.86	4-8-20-	0.000	1.07E+15
1	4	3	2797.94	4-8-17-20	1.478	1.05E+15
1	4	4	1261.19	4-7-8-20	0.805	1.04E+15
1	4	5	957.23	4-7-8-20	0.111	1.04E+15
1	4	6	492.33	4-7-8-20	0.000	1.03E+15
1	5	4	6785.24	4-8-11-17-20	3.261	1.04E+15
1	5	5	3842.73	4-7-8-17-20	1.614	1.02E+15
1	5	6	2172.92	4-7-8-17-20	0.910	1.01E+15
1	5	7	1280.52	1-4-6-8-18	0.232	1.00E+15
1	5	8	1067.88	1-4-6-8-18	0.153	1.00E+15
1	5	9	782.42	1-4-6-8-18	0.139	1.00E+15
1	5	1	424.02	1-2-4-7-8	0.000	1.00E+15
Average			1284.19		0.629	

One of the most important criteria for a node to be identified as a hub is the number of flows (O_i+D_i) that pass through it. New York (17), Chicago (4), and Los Angeles (12) would have the largest demand and supply in the CAB data set, accounting for 17.0 %, 10.0 %, and 7.3 % demand and supply, respectively. With percentages of 80%, 100%, and 63% for New York, Chicago, and Los Angeles, respectively, these nodes are identified as optimum hub placements for all p, q, and values (except p=2). Another explanation for Chicago's presence in practically all optimal result groups is its proximity to the network structure's center. As a consequence, Chicago is chosen as the hub site in all optimum solutions. In this regard, the number of neighbors, the usage frequency of the region's node, the closeness to network center points, as well as demand and supply, are all important factors for designating areas as hub sites. Another explanation for Chicago's presence in practically all optimal result groups is its proximity to the network structure's center. As a consequence, Chicago is chosen as the hub site in all optimum solutions. In this regard, the number of neighbors, the usage frequency of the region's node, the closeness to network center points, as well as demand and supply, are all important factors for designating areas as hub sites. On the other hand, being identified as a hub site is extremely plausible for a region such as Los Angeles, which is far from the network's center and part of a large volume of flows. However, this is generally true at extremely low values of alpha like α=0.2. Furthermore, despite the fact that Boston ranked fourth out of 25 areas in terms of overall demand and supply, it was not recognized as a hub since it was near to a significant node like New York and was located in a remote portion of the network structure. That is, being near an important node increases the chance of getting chosen as a hub.

Even when all of the aforementioned characteristics are considered, such as demand, supply, nodal usage patterns, distance to critical nodes, and closeness to the network's center, it is difficult to predict the best hub positions. Large-scale difficulties, in particular, and the abundance of nodes with comparable features make this endeavor considerably more difficult. Furthermore, optimum hub locations vary greatly depending on p, q, and α values. For instance, the optimum hub sites for p=5, q=4, and α=0.2 are Chicago, Dallas, Los Angeles, Miami, and New York. However, if = 1, the ideal hub regions are Chicago, Denver, Kansas, New York, and Pittsburgh. To put it another way, three hub places were swapped with various nodes. Similarly, best hub sites for p=5, q=4, and α=1 are Chicago, Denver, Kansas, New York, and Pittsburgh. If, on the other hand, q=10, indicating a comprehensive network, the optimum places are switched to Chicago, Atlanta, Baltimore, Dallas, and Denver.

3.5. Summarizing the Findings and Inferences

Specific conclusions were drawn based on the optimum results supplied from USAp-HMP. In addition to these conclusions, the features of nodes identified as optimum hub locations and the situations under which these nodes persisted in optimal outcomes were examined. As a result, the following conclusions were reached:

- Nodes with considerable demand and supply are more likely to be identified as hubs. This is especially likely if these nodes are located in isolated parts of the network structure. Figure 3.2(C) depicts an instance of this scenario. Chicago, New York, and Los Angeles have the biggest quantity demand and are generally chosen in optimum hub sites. Nonetheless, the alpha α value is essential at this point because this arrangement is not frequently valid at high α levels. (α=0.8, α=1).
- Vertices with lower demand and supply but close proximity to nodes with high demand and supply might also be chosen as hubs. This is especially true in regions

where nodes are used often. As shown in Figure 3.2(C), Philadelphia is an example of this sort of hub site. Even if there is not a large frequency of node usage in Tampa, it might be present in the cluster of optimum hubs in some circumstances because it is close to a very important node, namely Miami. This is usually the case due to the convergence of hub regions on each other depending on the α value increases.

- Hub vertices converge on each other as the α value increase. Demand amounts in respect to nodes become less important as a consideration in this situation. The significance of nodes with the capacity to serve as a linkage will also be increased, and they will most probably be labeled as hubs. Figure 3.2(A, B, C) illustrates this situation. When α=0.2 as (Figure 3.2(B)), hub sites are dispersed evenly across the whole network. If, on the other hand, α=1, hub locations are concentrated in a narrow network layer (Figure 3.2(A)). As a result, the relevance of the nodes in Figure 3.2 (C) that operate as the connector is increased.
- When all permutations of p, q, α and are examined, it is clear that some locations are never identified as hubs. This is due to factors such as being located in a distant portion of the overall network, a low level of demand, and distance to critical nodes. Figure 3.2(D) depicts an example of this scenario. Without in any instance will the locations displayed in this figure be within the optimum hub group (p=2 & p=5, q=(p-1) and p*(p-1) / 2), =0.2 & =1). The optimum results presented in table 3.1 show that these places occur in no permutations of the optimal solutions.
- The amount of connections between hubs effects the best solution. However, for large α values, the best solution is more sensitive to economies of scale coefficient. Furthermore, the limited number of connections between hubs allows for the placement of hub nodes in close proximity to one another. In Table 3.1, the comparison of the case for α=1, p=5 and q=5 to the case for α=1, p=5 and q=10 shows that hub changes occur in the optimal solution as illustrated by New York (17) & Baltimore (2) and Pittsburgh (20) & Atlanta (1). In other words, coupled with the increase in the value of q, the average distance between hub regions diminishes.

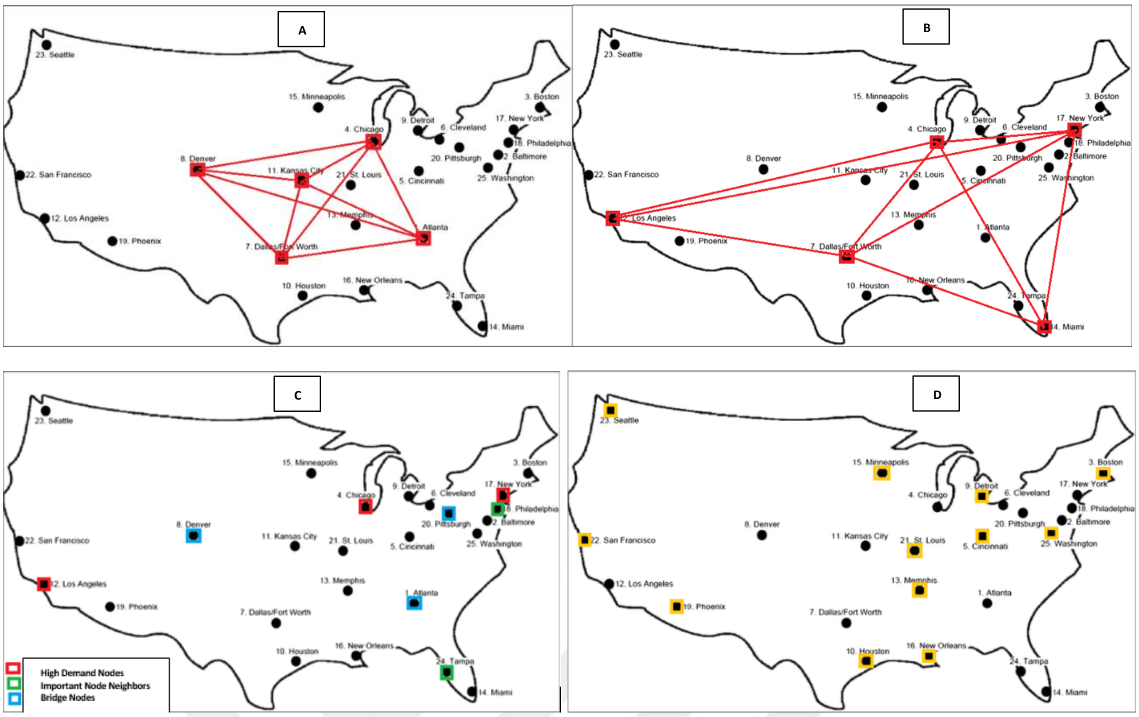


Figure 3.2. Analysis of nodes according to optimal results

Upon the increase of the value of q from 5 to 10, average distance between hubs is lowered by 3.24%. In a similar vein, upon the comparison of the case for α=1, p=4 and q=3 to the case for α=1, p=4 and q=6, it is observed that hub locations converge on each other by 4.24%.

In the scope of the foregoing conclusions, the purpose of setting up sub-sets, particularly for large-scale problems, is to minimize solution time and find the best solution using small-scale problems. Reducing size down the problem, on the other hand, must be based on a specific systematic strategy. The core concept behind this strategy is to examine each node based on certain parameters and analyze the likelihood of nodes to be identified as hubs. Following that, sub-sets including prospective hub locations are generated, and the solution set is reduced down. Because there are likely to be multiple nodes with similar features in large-scale problems, achieving the best solution becomes more difficult, and in this respect, features distinctive to becoming a hub are considered from a range of viewpoints. Moreover, in partial networks, the lack of links between some hub sites makes this procedure considerably harder. The core assumption of this strategy is to scale down the problem in a controlled manner, with the key guideline being to eliminate possibilities that are likely to impact the best solution. In other words, when vertices are classified into sub-sets based on particular node properties, it is critical to guarantee that nodes that exist among optimum solutions are included in these sub-sets. In this context, the following criteria for selecting potential hub locations are provided, along with explanations for each factor:

Centrality: Centrality metrics, which are more commonly utilized in social network analysis, are used to identify the critical nodes. The identification of nodes that are likely to reach and influence more people or facilitate the spread of information at a quicker rate is critical in concerns such as advertisement and data transmission. These measurements, which share comparable properties, may be applied to the identification of hub areas and can make significant contributions to the classification of potential hub locations in the optimal solution of large-scale problems. Following a study of previous research on hub locations, it is discovered that there are just a few studies that focus on centrality measurements (M. E. O'Kelly, 1992; Peker et al., 2016; Bernd Wagner, 2007; J. Yu et al., 2009). Furthermore, there is no in-depth investigation of the usage of centrality metrics in these researches.

Flow: The demand and supply amounts of nodes are important factors for determining hub positions. In this sense, the flow across nodes is another aspect that may be utilized to regulate the scale of the problem. This element was taken into account in instinctive problem-solving methodologies in several research. (Kratica et al., 2007; Peker et al., 2016). However, this feature alone is insufficient for identifying critical vertices.

Distribution: In addition to node centrality and supply/demand amounts, node dispersion within the network are important aspects. Hub locations are expected to be dispersed uniformly over the network in accordance with the α value, which is the coefficient of economies of scale. Another thing to consider in this regard is the dispersion of nodes.

Proximity: The closeness or proximity to nodes, which are essential in terms of centrality, o-d flow amounts, and dispersion, is another crucial consideration for the selection of hub sites. Because places adjacent to nodes that are significant in terms of supply and demand quantities and the centrality measures can also be identified as hub places, thanks to the increasing in the value of α, the closeness should be considered mostly on basis of values of α, p, and q.

For USAIp-HMP, centrality-based clustering methods are described that are focused on splitting down the problem and incorporating it into the problem-solving process in terms of the four aforementioned elements. The next section discuss about these methods and their basic considerations.

3.5. Determination of Candidate Hub Locations Based on Centrality Measures

Several algorithmic techniques based on centrality measurements were developed in this thesis. These techniques take into account all of the conclusions reached from the previous phase of this study. As a constraint, the outputs of the developed techniques are subsets of the major mathematical model. Each subset that is formed is expected to include at least one node that is part of the optimum solution. However, meeting this criterion is difficult since fluctuations in the values of α, p, and q cause changes in optimum positions. Algorithms created in this context are expected to be adaptable and robust to parameter changes.

3.5.1. Centrality based clustering algorithm (CBCA)

There are numerous measurements of centrality in the literature. Many other forms of centrality measurements, which were previously employed solely for social network research and development, have lately entered the picture. Closeness centrality, as established by Freeman and Mulholland (1979), offers the sum of a node's shortest available distances from other nodes across the network. Closeness centrality is based on the idea of distance and focuses on the proximity of nodes across the network. The importance of proximity centrality is related to the frequency of node interactions. In a similar spirit, Wasserman and (Brandes & Pich, 2007) state that proximity centrality considers how near a node is to all other nodes in the network. According to Degenne and Forsé (1999), it is a global metric that calculates proximity not only to surrounding nodes but also to all other nodes in the network. The following formula indicates how long it takes for data or product to transit from node i to other accessible nodes:

$$C_c(i) = \left[\sum_{j}^{N} d(i,j)\right]^{-1} \tag{3.13}$$

According to other nodes in the network, a node's importance is measured by its betweenness centrality. In terms of its function as a gateway between other nodes that are not directly connected to one another, it demonstrates the significance of a node. According to Freeman et al. (1979), nodes with high betweenness values have the ability to regulate the flow between all other nodes in the network. Lazega & Burt (1995) demonstrated how individuals working as brokers between unrelated people in an organization consolidated their influence. One of the most challenging centrality indices to compute is betweenness centrality. The number of these routes that travel through the node is determined using the betweenness centrality metric, which identifies the optimal routes between all node pairs in the network. The computation becomes more complicated as the network is expanded. As a result, a preliminary computation based on neighboring who are somewhat close to one another may be produced. σ_{ij} is supposed to be the number of options leading from node i to node j, and $\sigma_{ij}(k)$ is the number of shortest routes leading from node i to node j passing through node k. This information is used to evaluate the betweenness centrality of node k, which is between any node i and any node j across a network. The following expression could be used to compute the value of betweenness centrality for node k:

$$C_b(k) = \sum_{i \neq j \neq k \in N} \frac{\sigma_{ij}(k)}{\sigma_{ij}} \tag{3.14}$$

The eigenvector centrality is a different well-liked centrality metric (Bonacich, 1972). The fundamental eigenvector of the adjacency matrix describing the network is eigenvector centrality. An eigenvector centrality measure's defining equation is as follows:

$$\lambda v = Av \tag{3.15}$$

A represents the graph's adjacency matrix, is the eigenvalue constant, and v represents the eigenvector. It is possible to deduce from the equation that a node with a high eigenvector score is next to nodes with a high level of relevance. In the literature, eigenvector centrality is defined mathematically as eigen-centrality (Friedkin, 1991; Hubbell, 1965; Katz, 1953).

The metrics mentioned above are often used in the literature. The usage of these criteria, which are commonly employed to identify high-priority nodes on a network, can be beneficial in hub placement problems (especially in solving large-scale problems). This section provided a clustering procedure for each of the three centrality measurements. The results of applying this technique to each centrality metric were compared to the best results achieved using the CAB data set. The effectiveness of centrality measures was assessed using comparisons based on best hub locations, which are included inside sub-sets established in combination with each centrality metric. The initial part of the technique involves determining the level of relevance of hub nodes. This sort of study in the literature assesses the relevance of nodes from two viewpoints. The first of these viewpoints is about the node's demand quantity, and the second is about the node's position in relation to other nodes in the network.

The quantity of inflows to and outflows from the node (O_i + D_i) is determined in the first step of the method. The quantity of demand and supply that a node has, regardless of location, is the most essential aspect in establishing a node's classification as a hub (Kratica et al., 2007; Peker et al., 2016). Peker et al. (2016) provided various flow-based measures for determining the prominence of nodes. However, the measures that address centrality and demand are the most effective of the 16 recommended metrics. In this regard, the next step of the method ((O_i + D_i) * Centrality Measures (Closeness or Betweenness or Eigen-Vector Centrality)) calculated the overall rating for each node by multiplying the sum of the supply and demand quantities of each node by the centrality measures. However, choosing hub sites must take into account whether α value is high or low. At low levels α , the amount of flow determines a node's relevance, but at high levels α, node distance and dispersion across the network take central position. In this case, the following formulation was utilized to determine the nodes' relative relevance in the suggested algorithm:

$$External\ Importance\ of\ Node\ (EIoN) = ((O(i) + D(i)) * External\_Centrality\_Measure * \alpha + \left((O(i) + D(i)) * (1 - \alpha)\right) \quad i \epsilon N \qquad (3.16)$$

A node's relevance inside the network is determined using EIN. The node is a potential to be a hub since its importance increases with its EIoN value. According to this framework, evaluations are made according to significance rank. The average distance between the selected nodes is determined by selecting the first $(2 * p)$ number of nodes that are classified in descending order according to EIoN value. The radius for the $(2 * p)$ number of significant nodes is shown by the estimated average distance value. By taking into account the radius value that is set so that each significant node acts as the network's center, $(2 * p)$ number of circles are produced (Peker et al., 2016). Internal centrality values are computed for vertices that reside inside every circles. The following is the formulation created for this purpose:

$$Internal\ Importance\ of\ Node\ i\ in\ Cluster\ k\ (IIoN) = \big(O(i) + D(i)\big) * \ Internal\_Centrality\_Measure_k \qquad i \epsilon N \ \ k \epsilon K \qquad (3.17)$$

As a consequence, the nodes in each group that have the highest level of centrality and are hence candidates to serve as local hubs are identified. Here, the purpose is to explore nodes with high levels of centrality, considerable supply and demand, and proximity to significant nodes throughout the network. Examples of nodes in this context are shown in Figure 3.2(C) as green-hued nodes. It can be shown through an analysis of the CAB data set that Philadelphia has a high local centrality and that New York is one of the major hub places. The hub location is set as Philadelphia, same as it was for α=0.4, p=3, and q=2. IIoN is crucial for include this category of places in the solution set as well.

The CBCA method arranges each node in the network according to the significance values determined by the EIoN formulation, and then generates a Ci set with number of $(2 * p)$ significant vertices. Based on the radius value and the region that each node covers, each node that is assigned to Ci makes up the Cu and Cc sets. The process continues if there isn't another node in Ci that covers the same region as the corresponding element. The contained node is selected to the Cu set, assuming that it does not already exist in Ci, if just one node is covered by this Ci-existing node. Last but not least, IIoN is computed for all nodes in the region covered by the node in Ci if there are several nodes there. The procedure moves forward if the node with the highest centrality value outside of the one in Ci set is assigned to the Cc set. These actions continue until all sub-sets from the Ci set with significant centers have been used.

The algorithm generates three different results. These sets identify possible hub vertices with various properties. The pseudo code for CBCA is described in full below. The vertices with high demand that are far from the network's core are represented by the Cu set. The Cc set consists of nodes with high centrality values and strong demand, as well as places where nodes are used often. The Ci set stands for the collection of nodes that are considered essential at the start of the algorithm but are excluded from the Cu and Cc sets for a variety of reasons, including lower IIoN and closeness to the key node.

Algorithm CBCA:

```
EIoN=Ø,    IIoN=Ø,    ∀iϵN,    ∀kϵK,        Ci=Ø, Cu=Ø, Cc=Ø
Define k
for i in N do
   EIoN (i)=((O(i)+D(i))*External_Centrality_Measure(i)*α+((O(i)+D(i))*(1-α))
EIoN = EIoN U {i}
end for
sort(EIoN) descending order and append to Ci top 2*p nodes
while (m<2*p) do
        TC=Ø
        k=Ci[m]
        TC={k|dik<=radius  ∀iϵN}
        if(|TC|=0) then
                continue
        elif(|TC|=1 and i ∉ Ci) then
                Cu=CuU{i}
        else  do
                for j in (TC) do
                        (IIoN)=(O(j)+D(j))*(Internal_Centrality_Measure(j))
                        IIoN=IIoN U {j}
                        Cc=Cc U max(IIoN)
                        if(max(IIoN(i))ϵCi)   then
                                continue
                        end if
                end for
        end if
end while
```

Assignments are carried out such that at least one node is to be assigned from each sub-set as a hub, considering algorithm results and sub-sets (Ci, Cu, and Cc) into consideration. This formulation of USAIp-HMP is described in the preceding section. Formulation is as follows, with extra restrictions:

$$min \sum_{i \in N} \sum_{k \in H} \delta c_{ik} \, O_i x_{ik} + \sum_{i \in H} \sum_{j \in H} \sum_{k \in N} \alpha c_{ij} \, f_{kl}^{i} + \sum_{i \in N} \sum_{k \in H} \gamma c_{ki} \, D_i x_{ik} \qquad (3.18)$$

$s.t.$

$(3.2) - (3.12)$

$$\sum_{k \epsilon C_u} x_{kk} \geq 1 \tag{3.19}$$

$$\sum_{k \epsilon C_c} x_{kk} \geq 1 \tag{3.20}$$

$$\sum_{k \epsilon C_i} x_{kk} \geq 1 \tag{3.21}$$

The constraint (3.19) permits at least one of the isolated network nodes with high demand to be chosen as a hub. The restriction (3.20) ensures that the hubs are chosen from the nodes that are centrally placed and have a substantial quantity of flow from the node-dense locations. Hub areas are dispersed around the network as one node is chosen from each significant node location. The vertices that are perceived as important (based on centrality values and demand quantities) are offered for solution with the last constraint (3.21). So long as all sets have at least one hub, the formulation can locate the optimum solution (within the optimal set). The solution is not optimum, though, if the node engaged in it does not exit in at least one of the sets.

3.5.2. Candidate hub findings of centrality based algorithms

We suggested an approach that can be used for all three centrality measures in the previous sections. These centrality measures each have unique properties. In complicated networks, the proximity centrality metric is typically employed to pinpoint hub locations. The maximum closeness centrality rating is assigned to the node with the shortest overall distance to all other nodes. As a result, it plays a significant role in determining the best node for the flow of information or goods. Nevertheless, hub positioning problems are significantly influenced by the demands of the nodes. In order to get more insightful findings, thus it is necessary to take into account both the centrality measure and the node demand simultaneously. Additionally, the betweenness centrality metric is more crucial for identifying the network nodes that are on the shortest pathways and those that may act as bridging. It is typically used to find nodes in the most important places. Because hub areas are converging in this study, the betweenness centrality measure is mostly used because of the rising alpha value and rising significance levels of the nodes with high betweenness centrality values.

A metric that goes beyond centrality measures is eigenvector centrality. In addition to the quantity of neighbours a node has, eigenvector centrality also takes into account the significance of the neighbours it is connected to. In fact, eigenvector centralization is one of the best requirements for simulating actual life. In social media networks, this characteristic is sometimes referred to as popularity. Similar to how closeness centrality measure was supposed to be utilized, the eigenvector measure criterion was applied in this study to determine several significant vertices. Based on each criterion and with different parameter values (p = (2 to 5), q = (p-1 to p * (p-1) / 2) and α = (0.2, 0.4, 0.6, 0.8, 1)) algorithms were run. A overview of the solutions found is given in Table 3.2 and the outcomes attained. Sets of closeness, betweenness, and eigen-vector based algorithms were compared with the best outcomes for each alpha value (0.2-1.0).

The outcomes of the CBCA procedure for each centrality value are shown in Table 3.3. The inclusion of optimal hub zones by sub-sets is expressed quantitatively based on each centrality metric, with sub-sets denoted in the Ci, Cu, and Cc columns. In other words, it indicates whether nodes that are part of the optimum hub set are also represented in the CBCA-derived subgroups. In order to find the best hub vertices, this method operates a centrality measure using CBCA and assesses its effectiveness to decide which is more reliable. Table 3.2 displays the proportion of sub-sets formed by CBCA using the three centrality metrics of closeness, betweenness, and eigenvector via various values that contain the optimum hub set vertices. Additionally, a CAB data set with 25 vertices was employed for this research.

Running the CBCA algorithm on all problem sets produced subsets (CAB data). This study's findings are that closeness centrality can yield decent subsets for low alpha values, but betweenness centrality produces superior outcomes for large alpha values (α= 1). Only when α=0.2 could the eigen-vector measure provide suitable subsets of. Additionally, even when α=1, the subsets generated by the closeness and eigen-vector criterion do not cover the optimal set at about 50%. The closeness and eigen-vector scales become less accurate as alpha rises. However, closeness centrality often produced superior outcomes (approximately 79.16 percent on average). Table 3.2 has an overview of the above. For a more comprehensive example. Due to economies of scale in hub location problems, it is well acknowledged that value is typically lower than 1, and for 1, closeness centrality performs better than betweenness centrality. Additionally, eigenvector often created less persistent sub-sets. For CBCA, only the adoption of closeness centrality was favored due to all of these factors.

Table 3.2. Experiments with different centrality measures for CAB data sets

Alfa	Betweenness Centrality (%)	Closeness Centrality (%)	Eigenvector Centrality (%)
0.2	79.29	96.43	90.00
0.4	79.88	86.31	79.88
0.6	78.10	85.83	79.88
0.8	72.62	76.43	72.74
1	90.36	50.83	47.98
Average	80.05	79.16	74.09

In nearly all of the CPLEX findings, the best outcomes were attained in an acceptable amount of time (5000 sec). The lack of optimum node areas in the computed subgroups only prevented optimal outcomes from being produced in problems of type p = 2 and p=5, α=1. The original formulation can be solved using CPLEX in a matter of seconds (maximum 23 sec), and since p = 2 problems are network connectivity problems, it is not significant for the suggested technique. With the exception of the α= 0.2 category, the CBCA method produced optimum results faster than the initial problem. In other words, the effectiveness of this method grows as the alpha value rises. The average CPU increase can reach up to 33 percent when the alpha value is 1.

3.5.3. Bounded version of CBCA (BCBCA)

As was demonstrated in the preceding chapter, the issue is fixed by adding the CBCA subsets as a restriction to the initial model. The solution must be precisely optimum if each subset contains at least one optimum vertex. All p hubs must be chosen from the union of the subsets Ci, Cu, and Cc according to the constrained version of the CBCA algorithm (BCBCA). This is most noticeable when there are significantly less limitations (3.19), (3.20), and (3.21) than there are in p. This limitation allows for a large reduction in the solution set and a corresponding reduction in solution time. The restriction to be applied to the original formulation further restricts the problem-solving space using the subset produced via $CBCA_{cl}$.

$$\sum_{j=CI\cup CU\cup CC} x_{kk} = p \tag{3.22}$$

When the findings of Table 3.3 are analyzed, it becomes clear that while the BCBCA algorithm performs substantially better in terms of solution time than the CBCA method, it performs worse in terms of solution quality. This is so that no node other than the subsets produced by the BCBCA algorithm can access the solution. Therefore, inside the sub-sets, only the optimal answer can be found. However, there is not much of a difference between the outcomes and optimum solutions. The 42 percent of solutions that were optimally solved had an average maximum gap of about 3.27 percent.The quality of the solution declines as the alpha value rises. since there are fewer eligible vertices in the ideal set of subgroups. Except for p = 2, practically all findings for the 0.2 alpha value showed that BCBCA had attained the optimal structure. Another crucial observation is that when the p value rises, the distance from the optimum solution narrows. This shows that the impact of changing hub locations on costs is less pronounced at larger p values.

Table 3.3. Results obtained for CAB data sets of CBCA and BCBCA algorithms.

				Clusters			CBCA					BCBCA				
α	p	q	Cplex CPU time	Cc	Cu	Ci	CPU time (sec)	Hub Locations	Obj	GAP(%)	Time Imp. (%)	CPU time (sec)	Hub Locations	Obj	GAP(%)	Time Imp. (%)
0.2	2	1	0.64	2,5,12		17,4,3	1.81	12,17	opt	0.00	-182.81	0.11	12,17	9.52E+15	10.21	82.81
0.2	3	2	7.81	17,4,3,25,14		2,5,12,20,24	14.83	4,12,17	opt	0.00	-89.88	0.22	4,12,17	opt	0.00	97.18
0.2	3	3	6.27	17,4,3,25,15		2,5,12,20,25	7.86	4,12,17	opt	0.00	-25.36	0.17	4,12,17	opt	0.00	97.29
0.2	4	3	204.92	18,9,17,2,6	12,14,24,22	4,3,25	40.47	4,12,14,17	opt	0.00	80.25	1.19	4,12,14,17	opt	0.00	99.42
0.2	4	4	11.61	18,9,17,2,6	12,14,24,23	4,3,25	10.78	4,12,17,24	opt	0.00	7.15	0.53	4,12,17,24	opt	0.00	95.43
0.2	4	5	3.67	18,9,17,2,6	12,14,24,24	4,3,25	3.77	4,12,17,24	opt	0.00	-2.72	0.41	4,12,17,24	opt	0.00	88.83
0.2	4	6	1.63	18,9,17,2,7	12,14,24,25	4,3,25	1.52	4,12,17,24	opt	0.00	6.75	0.25	4,12,17,24	opt	0.00	84.66
0.2	5	4	303.65	18,9,17,2,8	12,14,24,22,7,10	4,3,25	142.95	4,7,12,14,17	opt	0.00	52.92	10.5	4,7,12,14,17	opt	0.00	96.54
0.2	5	5	62.39	18,9,17,2,9	12,14,24,22,7,11	4,3,25	55.36	4,7,12,14,17	opt	0.00	11.27	5.42	4,7,12,14,17	opt	0.00	91.31
0.2	5	6	27.67	18,9,17,2,10	12,14,24,22,7,12	4,3,25	12.56	4,7,12,14,17	opt	0.00	54.61	1.36	4,7,12,14,17	opt	0.00	95.08
0.2	5	7	11.8	18,9,17,2,11	12,14,24,22,7,13	4,3,25	8	4,7,12,14,17	opt	0.00	32.20	0.8	4,7,12,14,17	opt	0.00	93.22
0.2	5	8	2.13	18,9,17,2,12	12,14,24,22,7,14	4,3,25	2.44	4,7,12,14,17	opt	0.00	-14.55	0.73	4,7,12,14,17	opt	0.00	65.73
0.2	5	9	1.77	18,9,17,2,13	12,14,24,22,7,15	4,3,25	2.52	4,7,12,14,17	opt	0.00	-42.37	0.33	4,7,12,14,17	opt	0.00	81.36
0.2	5	10	1.83	18,9,17,2,14	12,14,24,22,7,16	4,3,25	1.36	4,7,12,14,17	opt	0.00	25.68	0.22	4,7,12,14,17	opt	0.00	87.98
0.4	2	1	1.51	2,5,12		17,4,3	4.38	4,12	1.04E+16	9.54	-190.07	0.42	4,12	1.04E+16	9.54	72.19
0.4	3	2	59.95	2,5,12,20,24		17,4,3,25,14	23.59	4,12,18	opt	0.00	60.65	0.61	4,12,17	7.71E+15	0.03	98.98
0.4	3	3	62.05	2,5,12,20,25		17,4,3,25,15	67.91	4,12,18	opt	0.00	-9.44	0.02	4,12,17	7.71E+15	0.12	99.97
0.4	4	3	641.83	18,9,17,2,6	12,14,24,22	4,3,25	469.61	4,12,14,17	opt	0.00	26.83	7.14	4,12,14,17	opt	0.00	98.89
0.4	4	4	109.53	18,9,17,2,7	12,14,24,23	4,3,25	69.65	1,4,12,17	opt	0.00	36.41	1.7	4,12,14,17	6.81E+15	0.41	98.45
0.4	4	5	44.56	18,9,17,2,8	12,14,24,24	4,3,25	25.84	1,4,12,17	opt	0.00	42.01	0.64	4,12,17,24	6.78E+15	0.76	98.56
0.4	4	6	41.27	18,9,17,2,9	12,14,24,25	4,3,25	15.31	1,4,12,17	opt	0.00	62.90	0.64	4,12,17,24	6.78E+15	0.81	98.45
0.4	5	4	1260.41	18,9,17,2,10	12,14,24,22,7,10	4,3,25	1059.03	4,7,12,14,17	opt	0.00	15.98	57.78	4,7,12,14,17	opt	0.00	95.42
0.4	5	5	657.09	18,9,17,2,11	12,14,24,22,7,11	4,3,25	601.53	4,7,12,14,17	opt	0.00	8.46	48.39	4,7,12,14,17	opt	0.00	92.64
0.4	5	6	211.86	18,9,17,2,12	12,14,24,22,7,12	4,3,25	131.84	4,7,12,14,17	opt	0.00	37.77	6.19	4,7,12,14,17	opt	0.00	97.08
0.4	5	7	61.05	18,9,17,2,13	12,14,24,22,7,13	4,3,25	52.44	4,7,12,14,17	opt	0.00	14.10	5.34	4,7,12,14,17	opt	0.00	91.25
0.4	5	8	31.78	18,9,17,2,14	12,14,24,22,7,14	4,3,25	25.52	4,7,12,14,17	opt	0.00	19.70	2.56	4,7,12,14,17	opt	0.00	91.94
0.4	5	9	23.44	18,9,17,2,15	12,14,24,22,7,15	4,3,25	16.67	4,7,12,14,17	opt	0.00	28.88	1.52	4,7,12,14,17	opt	0.00	93.52
0.4	5	10	18.88	18,9,17,2,16	12,14,24,22,7,16	4,3,25	9.91	4,7,12,14,17	opt	0.00	47.51	1.23	4,7,12,14,17	opt	0.00	93.49
0.6	2	1	4.86	2,5,12		17,4,3	3.45	12,25	opt	0.00	29.01	0.55	2,4	1.08E+15	5.02	88.68
0.6	3	2	261.41	2,5,12,20,24		17,4,3,25,14	211.45	4,12,18	opt	0.00	19.11	0.84	2,4,12	8.84E+15	0.01	99.68
0.6	3	3	197.66	2,5,12,20,224		17,4,3,25,14	187.62	2,4,12	opt	0.00	5.08	1.09	2,4,12	opt	0.00	99.45
0.6	4	3	1821.84	18,9,17,2,6	12,14,24,22	4,3,25	1544.33	4,12,14,18	opt	0.00	15.23	14.16	4,12,14,18	opt	0.00	99.22
0.6	4	4	542.36	18,9,17,2,7	12,14,24,23	4,3,25	514.07	1,4,12,17	opt	0.00	5.22	4.17	4,12,14,17	8.16E+15	0.67	99.23
0.6	4	5	366.05	18,9,17,2,8	12,14,24,24	4,3,25	252.13	1,4,12,17	opt	0.00	31.12	2.03	4,12,14,17	8.13E+15	1.29	99.45
0.6	4	6	219.09	18,9,17,2,9	12,14,24,25	4,3,25	193.73	1,4,12,17	opt	0.00	11.58	1.97	4,12,14,17	8.13E+15	1.34	99.10
0.6	5	4	3721.03	18,9,17,2,10	12,14,24,22,7,10	4,3,25	3413.48	4,12,14,17,20	opt	0.00	8.27	79.23	4,6,12,14,17	7.93E+15	0.08	97.87
0.6	5	5	1416.48	18,9,17,2,11	12,14,24,22,7,11	4,3,25	1770.56	4,11,12,14,17	opt	0.00	-25.00	73.19	4,7,12,14,17	7.79E+15	0.72	94.83
0.6	5	6	546.02	18,9,17,2,12	12,14,24,22,7,12	4,3,25	701.19	4,7,12,14,17	opt	0.00	-28.42	20.59	4,7,12,14,17	opt	0.00	96.23
0.6	5	7	374.58	18,9,17,2,13	12,14,24,22,7,13	4,3,25	365.3	4,7,12,14,17	opt	0.00	2.48	14.06	4,7,12,14,17	opt	0.00	96.25
0.6	5	8	294	18,9,17,2,14	12,14,24,22,7,14	4,3,25	271.56	4,7,12,14,17	opt	0.00	7.63	7.39	4,7,12,14,17	opt	0.00	97.49
0.6	5	9	178.94	18,9,17,2,15	12,14,24,22,7,15	4,3,25	217.41	4,7,12,14,17	opt	0.00	-21.50	5.25	4,7,12,14,17	opt	0.00	97.07

				Clusters			CBCA					BCBCA				
α	p	q	Cplex CPU time	Cc	Cu	Ci	CPU time (sec)	Hub Locations	Obj	GAP(%)	Time Imp. (%)	CPU time (sec)	Hub Locations	Obj	GAP(%)	Time Imp. (%)
0.6	5	10	154.73	18,9,17,2,16	12,14,24,22,7,16	4,3,25	122.35	4,7,12,14,17	opt	0.00	20.93	3.59	4,7,12,14,17	opt	0.00	97.68
0.8	2	1	11.55	2,5,12		17,4,3	3.58	2,4	opt	0.00	69.00	0.58	2,4	1.13E+15	2.20	94.98
0.8	3	2	616.96	2,5,12,20,24		17,4,3,25,14	39.65	2,4,12,	opt	0.00	93.57	5.14	2,4,12	opt	0.00	99.17
0.8	3	3	295.56	2,5,12,20,24		17,4,3,25,15	28.35	2,4,12,	opt	0.00	90.41	1.55	2,4,12	opt	0.00	99.48
0.8	4	3	2546.08	18,9,17,2,6	12,14,24,22	4,3,25	2459.83	4,12,17,20	opt	0.00	3.39	26.53	4,6,12,18	9.46E+15	0.73	98.96
0.8	4	4	1001.17	18,9,17,2,6	12,14,24,23	4,3,25	947.27	1,4,12,18	opt	0.00	5.38	21.52	4,12,17,25	9.51E+15	1.09	97.85
0.8	4	5	393.41	18,9,17,2,6	12,14,24,24	4,3,25	431.54	1,4,12,18	opt	0.00	-9.69	17.83	4,12,18,24	9.48E+15	1.89	95.47
0.8	4	6	319.23	18,9,17,2,6	12,14,24,25	4,3,25	226.1	1,4,12,18	opt	0.00	29.17	17.03	4,12,18,24	9.47E+15	1.92	94.67
0.8	5	4	4957.11	18,9,17,2,6	12,14,24,22,7,10	4,3,25	4480.11	4,11,12,17,20	opt	0.00	9.62	146.38	4,6,12,17,24	9.33E+15	1.68	97.05
0.8	5	5	2723.75	18,9,17,2,6	12,14,24,22,7,11	4,3,25	2556.45	4,7,12,17,20	opt	0.00	6.14	84.56	4,6,7,12,18	9.11E+15	0.33	96.90
0.8	5	6	1455.67	18,9,17,2,6	12,14,24,22,7,12	4,3,25	1518.26	1,4,11,12,18	opt	0.00	-4.30	57.95	4,6,7,12,18	9.05E+15	0.52	96.02
0.8	5	7	990.12	18,9,17,2,6	12,14,24,22,7,13	4,3,25	983.55	1,4,7,12,18	opt	0.00	0.66	45.19	4,7,12,18,24	8.97E+15	1.11	95.44
0.8	5	8	668.17	18,9,17,2,6	12,14,24,22,7,14	4,3,25	441.38	1,4,7,12,18	opt	0.00	33.94	23.52	4,7,12,18,24	8.92E+15	0.82	96.48
0.8	5	9	566.39	18,9,17,2,6	12,14,24,22,7,15	4,3,25	380.44	1,4,7,12,18	opt	0.00	32.83	22.67	4,7,12,18,24	8.90E+15	0.74	96.00
0.8	5	10	210.56	18,9,17,2,6	12,14,24,22,7,16	4,3,25	216.07	1,4,7,12,18	opt	0.00	-2.62	17.91	4,7,12,18,24	8.90E+15	0.77	91.49
1	2	1	23.64	18,2,6	4	17,25,9	2.05	2,4	1.17E+15	0.00	91.33	0.41	4,25	1.17E+15	0.79	98.27
1	3	2	1053.77	18,9,2,6,17	14,24	4,25,3	4.89	2,4,24	1.17E+15	0.00	99.54	7.48	2,4,24	1.17E+15	8.18	99.29
1	3	3	537.86	18,9,2,6,17	14,24	4,25,3	17.23	4,18,24	1.15E+15	0.00	96.80	1.66	4,18,24	1.15E+15	6.68	99.69
1	4	3	2797.94	2,4,20,18,12,21	14,24	17,25,9,3	2194.16	4,8,24,25	1.08E+15	0.00	21.58	43.42	4,12,24,25	1.09E+15	3.68	98.45
1	4	4	1261.19	2,4,20,18,12,22	14,24	17,25,9,3	1094.15	4,8,17,24	1.06E+15	0.00	13.24	23.2	12,21,24,25	1.08E+15	3.45	98.16
1	4	5	957.23	2,4,20,18,12,23	14,24	17,25,9,3	701.91	4,8,17,24	1.06E+15	0.00	26.67	25.73	12,21,24,25	1.07E+15	3.22	97.31
1	4	6	492.33	2,4,20,18,12,24	14,24	17,25,9,3	249.64	4,8,24,25	1.06E+15	0.00	49.29	15.8	4,12,24,25	1.07E+15	3.33	96.79
1	5	4	5324.45	18,4,2,6,17,12,21,20	14,24	25,9,3	5635.09	4,8,17,20,24	1.04E+15	0.00	-5.83	217.25	4,6,12,14,25	1.07E+15	3.20	95.92
1	5	5	3842.73	18,4,2,6,17,12,21,21	14,24	25,9,3	4157.13	4,8,9,17,24	1.03E+15	0.00	-8.18	110.34	4,12,21,24,25	1.05E+15	3.01	97.13
1	5	6	2172.92	18,4,2,6,17,12,21,22	14,24	25,9,3	1694.44	4,8,9,17,24	1.02E+15	0.00	22.02	54.75	4,12,21,24,25	1.03E+15	1.83	97.48
1	5	7	1280.52	18,4,2,6,17,12,21,23	14,24	25,9,3	911.86	4,8,13,24,25	1.01E+15	0.00	28.79	31.17	4,12,21,24,25	1.03E+15	2.50	97.57
1	5	8	1067.88	18,4,2,6,17,12,21,24	14,24	25,9,3	856.13	4,8,13,24,25	1.01E+15	0.00	19.83	37.55	4,12,21,24,25	1.03E+15	2.57	96.48
1	5	9	782.42	18,4,2,6,17,12,21,25	14,24	25,9,3	644.98	4,8,13,24,25	1.01E+15	0.00	17.57	21.55	4,12,21,24,25	1.02E+15	1.63	97.25
1	5	10	424.02	18,4,2,6,17,12,21,26	14,24	25,9,3	418.33	4,8,13,24,25	1.01E+15	0.00	1.34	12.3	4,12,21,24,25	1.02E+15	1.77	97.10

3.5.4. Hybrid Centrality Based Clustering Algorithm (HCBCA)

The study of the findings from the preceding section shows that depending on the α value of, the centrality measures' level of relevance varies. The value of (external centrality*(O_i+D_i)) that is generated using the CBCA algorithm's closeness centrality measure proves to be insufficient for determining the degree of significance of nodes when the value of is increased. The relevance of betweenness centrality is increased, particularly at high α levels, according to examination of the size of the influence (in percentages) of centrality measures on the construction of the optimal set based on different values. In this sense, proximity centrality and betweenness centrality both need to be used in order to enhance the algorithm. The likelihood of the optimal set being present in the sub-sets produced in this way is higher. An illustration that applies to this scenario is shown in Figure 4(C). Atlanta is included in the optimal set in the CAB data set for p=5, q=6, and different values of, namely α=0.2, α=0.4, and α=0.6. Denver and Kansas both have similar situations. These nodes have relatively high betweenness centrality scores. The inclusion of sub-sets derived by the BCBCA method into the original model as a constraint might sometimes prevent the creation of an optimum set, especially if α>0.8. Thus, the hybridization CBCA (HCBCA) method was suggested by including the betweenness centrality delivering satisfactory results for α>0.8 into the technique. HCBCA is an algorithm that considers more than one kind of centrality metric, as opposed to only one. The following scores are calculated for each node in the first stage of the algorithm to determine external centrality measures based on closeness centrality and betweenness centrality:

$$\text{External Closeness Metric} = (\text{external\_closeness})_i*(O_i+D_i) \quad \forall i \epsilon N \tag{3.23}$$

$$\text{External Betweenness Metric } (\text{external\_betweenness})_i*(O_i+D_i) \quad \forall i \epsilon N \tag{3.24}$$

Following the identification of $(2 * p)$ significant nodes based on betweenness and closeness centrality measurements, it is defined which nearby nodes should be encompassed by significant nodes in terms of the radius value, which represents the depends on the dimensions between important key nodes. However, with CBCA and BCBCA, this operation is only carried out for significant nodes that are defined for the closeness centrality metric. Nodes that rank highly in terms of betweenness centrality are preserved separately in the Cb set. As contrast to CBCA, the proximity centrality metric defines all surrounding nodes as a sub-set when they are covered by significant nodes. The method subsequently continues on to regulating for previously constructed iterated sets. If a given element already exists in a prior set, it is not added to the present set. In this regard, it is possible that for each scenario, the procedure and the number of sub-sets to be obtained will vary. An important node is added to the Cu sub-set if it has only one neighbor or none at all in the region it covers. Finally, a vertex is eliminated from the Cb set if it appears in the Cu set or Cc set as well. An illustration of the algorithm's process mode is shown in Figure 3.3. By focusing on significant nodes based on 'external closeness,' red circles were produced (Chicago, Dallas, Los Angeles, Philadelphia and Miami). Each node within the circle is considered as a level if there are more than two nodes there. For instance, Detroit, St. Louis, Cincinnati, and Cleveland are all included in the circle located on Chicago.

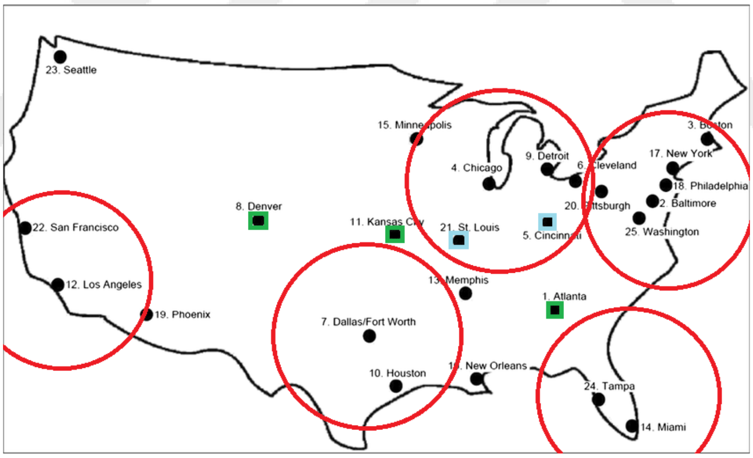


Figure 3.3. HCBCA algorithm node evaluation

The Cb sub-set, particularly displays significant vertices for betweenness centrality, has the green and blue colored Denver, Kansas, Atlanta, St. Louis, and Cincinnati vertices. The blue-colored St. Louis and Cincinnati, however, do not appear in the Cb set since they are contained within a circle. These sub-sets are produced in this situation:

C_{c1}={4,5,6,9,21}, C_{c2}={2,3,17,18,28}, C_u={7,10,12,14,22,24} and C_b={1,8,11}.

Each of these is included as a constraint in the original model. Following is the algorithm's pseudo code:

Algorithm HCBCA:

```
EIoN(cl)=Ø,   EIoN(btw)=Ø,      ∀iϵN,      ∀kϵK,     Ci=Ø, Cu=Ø, Cb=Ø
Define k
for i in N do
EIoN(cl) (i)=((O(i)+D(i))*External_Centrality_Measure(i)*α+((O(i)+D(i))*(1-α))
EIoN(btw) (i)= EIoN U {i}
end for
sort(EIoN(cl)) descending order and append to Ci top 2*p nodes
sort(EIoN(btw)) descending order and append to Cb top 2*p nodes
while (m<2*p and Ci≠ Ø) do
        Cc(m)=Ø
        TC=Ø
        k=Ci[m]
        if (kϵCb) then
          remove k from Cb
        end if
        TC={k|dik<=radius  ∀iϵN}
        if(|TC|=0) then
                Cu=CuU{k}
        elif(|TC|=1 and i∉Ci) then
                Cu=CuU{i}
                Cu=CuU{k}
                if (kϵCb) then
                   remove k from Cb
        else  do
                for j in (TC) do
                        if (j∉Ci,Cu) then
                          Cc(m)=Cc(m)= U {j}
                          if (kϵCb) then
                            remove k from Cb
                          end if
                        end if
                Cc= Cc U {k}
                end for
        end if
        m=m+1
end while
```

Constraints 3.6, 3.7, and 3.8 are incorporated into the model in a similar way to the preceding section. Extra limitations must be established, though, if several Cc sets are generated.

In Table 3.4, the USAIp-HMP results for the subgroups derived by the method are presented. When these findings are considered, the hybrid technique has a greater success rate in finding the best solution than the CBCA and BCBCA algorithms. For every type of problem, the HCBCA algorithm has produced the best outcome. The nodes in the ideal solution are present in each of the produced subgroups. The places within the optimal result set are indicated by nodes printed in red in Table 3.4. The typical reduction in solution times are roughly 25 percent.

3.5.5. Bounded version of HCBCA (HBCBCA)

Unlike HCBCA, p hubs in HBCBCA are required to pick exclusively from nodes inside the computed subgroups. That is, the problem is just addressed by vertices that are members of the Cb, Cu, and Cc groups. In this case, the problem is downscaled based on different parameters, and the solution time is minimized. As the restriction, HCBCA subgroups are added to the original formulation. In other words, unlike HCBCA, an additional restriction is added to the model. Along with this limitation, possible vertices are reduced into sub-sets, as seen below:

$$\sum_{j \in CB \cup CS \cup CC} x_{kk} = p \tag{3.25}$$

The study of HBCBCA revealed that, like HCBCA, optimal outcomes were attained. The advantage of HBCBCA over HCBCA is that it drastically cuts down on the amount of time needed to solve problems. Additionally, there was no change in the efficiency of the solutions. The best solutions were found for every variant of the problem. Outcomes from the CAB data set for both the HCBCA and HBCBCA are displayed in Table 3.4. The amount of improvement is roughly 90% for HBCBCA, whereas HCBCA reduces the solution time by about 20% over all problem variants. For HCBCA, the largest reduction in solution time was achieved for p=3, q=2, and =1, while the least reduction was achieved for p=5, q=6, and α=1. In the case of p=5, q=5, and α=1, the greatest improvement in the solution time for HBCBCA was 97.64 percent, while the least improvement was 64.79 percent in the case of p=5, q=8, and α=0.2. The values of p (number of hubs) and q (number of connections between hubs) in particular have a major impact on how rapidly a problem may be solved. There are significant disparities between the maximal and minimal improvement rates in this regard.

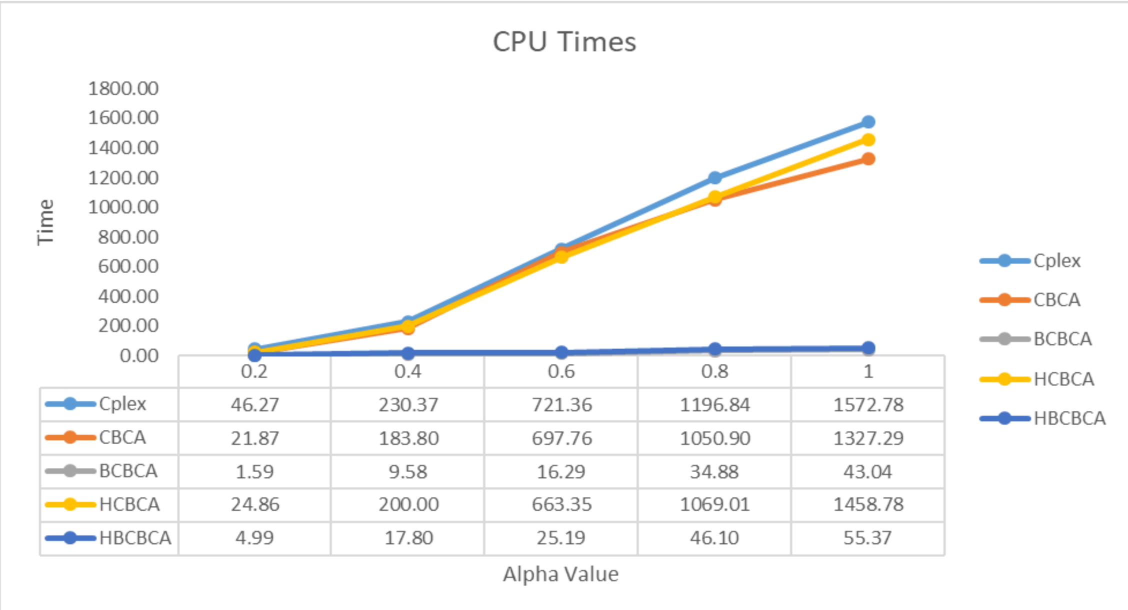
Table 3.4. Results obtained for CAB data sets of HCBCA and HBCBCA algorithms

			Clusters				HCBCA				HBCBCA			
α	p	q	$Cc_{(1)}$	$Cc_{(2)}$	Cu	Cb	CPU time (sec)	CPU Time Imp. %	Hub Locations	Obj.	CPU time (sec)	CPU Time Imp. %	Hub Locations	Obj.
0.2	2	1	2,3,17,18,20,25	4,5,6,9,15,21	12	11,22,8	0.44	31.25	12-20-	opt	0.13	79.69	12-20-	opt
0.2	3	2	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1	4.15	46.86	4-12-17-	opt	2.05	73.75	4-12-17-	opt
0.2	3	3	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1	2.76	55.98	4-12-17-	opt	0.8	87.24	4-12-17-	opt
0.2	4	3	2,3,17,18,20,25	4,5,6,9,21	12,20,14,24	8,1,11	60.34	70.55	4-12-14-17	opt	4.58	97.76	4-12-14-17	opt
0.2	4	4	2,3,17,18,20,25	4,5,6,9,21	12,20,14,24	8,1,11	9.16	21.10	4-12-17-24	opt	1.11	90.44	4-12-17-24	opt
0.2	4	5	2,3,17,18,20,25	4,5,6,9,21	12,20,14,24	8,1,11	2.99	18.53	4-12-17-24	opt	0.55	85.01	4-12-17-24	opt
0.2	4	6	2,3,17,18,20,25	4,5,6,9,21	12,20,14,24	8,1,11	1.54	5.52	4-12-17-24	opt	0.3	81.60	4-12-17-24	opt
0.2	5	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	184.16	39.35	4-7-12-14-17	opt	31.63	89.58	4-7-12-14-17	opt
0.2	5	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	57.94	7.13	4-7-12-14-17	opt	20.13	67.74	4-7-12-14-17	opt
0.2	5	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	12.61	54.43	4-7-12-14-17	opt	5.33	80.74	4-7-12-14-17	opt
0.2	5	7	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	8.22	30.34	4-7-12-14-17	opt	1.66	85.93	4-7-12-14-17	opt
0.2	5	8	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	1.64	23.00	4-7-12-14-17	opt	0.75	64.79	4-7-12-14-17	opt
0.2	5	9	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	1.33	24.86	4-7-12-14-17	opt	0.59	66.67	4-7-12-14-17	opt
0.2	5	10	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	0.77	57.92	4-7-12-14-17	opt	0.28	84.70	4-7-12-14-17	opt
Avg.							*24.86*	*34.77*			*4.99*	*81.12*		
0.4	2	1	2,3,17,18,20,25	4,5,6,9,15,21	12	11,22,8	0.94	37.75	12-20-	opt	0.13	91.39	12-20-	opt
0.4	3	2	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	17.56	70.71	4-12-18-	opt	8.09	86.51	4-12-18-	opt
0.4	3	3	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	6.99	88.73	4-12-18-	opt	4.58	92.62	4-12-18-	opt
0.4	4	3	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	537.49	16.26	4-12-14-17	opt	19.16	97.01	4-12-14-17	opt
0.4	4	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	94.34	13.87	1-4-12-17	opt	4.25	96.12	1-4-12-17	opt
0.4	4	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	36.75	17.53	1-4-12-17	opt	1.36	96.95	1-4-12-17	opt
0.4	4	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	21.6	47.66	1-4-12-17	opt	0.91	97.80	1-4-12-17	opt
0.4	5	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	1225.4	2.78	4-7-12-14-17	opt	102.4	91.88	4-7-12-14-17	opt
0.4	5	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	556.01	15.38	4-7-12-14-17	opt	82.66	87.42	4-7-12-14-17	opt
0.4	5	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	189.33	10.63	4-7-12-14-17	opt	10.81	94.90	4-7-12-14-17	opt
0.4	5	7	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	51.09	16.31	4-7-12-14-17	opt	7.56	87.62	4-7-12-14-17	opt
0.4	5	8	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	29.33	7.71	4-7-12-14-17	opt	3.23	89.84	4-7-12-14-17	opt
0.4	5	9	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	21.69	7.47	4-7-12-14-17	opt	2.11	91.00	4-7-12-14-17	opt
0.4	5	10	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	11.42	39.51	4-7-12-14-17	opt	1.95	89.67	4-7-12-14-17	opt
Avg.							*200.00*	*28.02*			*17.80*	*92.19*		
0.6	2	1	2,3,17,18,20,25	4,5,6,9,15,21	12	11,22,8	3.85	20.78	12-20-	opt	1.14	76.54	12-20-	opt
0.6	3	2	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	31.18	88.07	4-12-18-	opt	12.77	95.11	4-12-18-	opt
0.6	3	3	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	15.41	92.20	2-4-12-	opt	5.44	97.25	2-4-12-	opt
0.6	4	3	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	1842.98	-1.16	4-12-14-18	opt	77.35	95.75	4-12-14-18	opt
0.6	4	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	516.12	4.84	1-4-12-17	opt	28.05	94.83	1-4-12-17	opt
0.6	4	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	329.08	10.10	1-4-12-17	opt	16.30	95.55	1-4-12-17	opt
0.6	4	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	217.63	0.67	1-4-12-17	opt	12.15	94.45	1-4-12-17	opt
0.6	5	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	3188.27	14.32	4-12-14-17-20	opt	90.75	97.56	4-12-14-17-20	opt
0.6	5	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	1619.71	-14.35	4-11-12-14-17	opt	41.88	97.04	4-11-12-14-17	opt
0.6	5	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	617.03	-13.01	4-7-12-14-17	opt	29.16	94.66	4-7-12-14-17	opt
0.6	5	7	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	357.77	4.49	4-7-12-14-17	opt	17.27	95.39	4-7-12-14-17	opt
0.6	5	8	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	264.91	9.89	4-7-12-14-17	opt	11.75	96.00	4-7-12-14-17	opt

							HCBCA				HBCBCA			
			Clusters											
α	p	q	$Cc_{(1)}$	$Cc_{(2)}$	Cu	Cb	CPU time (sec)	CPU Time Imp. %	Hub Locations	Obj.	CPU time (sec)	CPU Time Imp. %	Hub Locations	Obj.
0.6	5	9	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	166.33	7.05	4-7-12-14-17	opt	4.98	97.22	4-7-12-14-17	opt
0.6	5	10	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	116.64	24.62	4-7-12-14-17	opt	3.67	97.63	4-7-12-14-17	opt
Avg.			-	-	-	-	*663.35*	*17.75*	-	-	*25.19*	*94.64*	-	-
0.8	2	1	2,3,17,18,20,25	4,5,6,9,15,21	12	1,13,16,24,22,8	6.58	43.03	12-20-	opt	1.19	89.70	12-20-	opt
0.8	3	2	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	61.19	90.08	2-4-12-	opt	43.71	92.92	2-4-12-	opt
0.8	3	3	2,3,17,18,25	4,5,6,9,21	12,20,14,24	22,8,1,11	32.85	88.89	2-4-12-	opt	11.10	96.24	2-4-12-	opt
0.8	4	3	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	2010.03	21.05	4-12-17-20	opt	96.24	96.22	4-12-17-20	opt
0.8	4	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	912.34	8.87	1-4-12-18	opt	39.75	96.03	1-4-12-18	opt
0.8	4	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	371.11	5.67	1-4-12-18	opt	29.27	92.56	1-4-12-18	opt
0.8	4	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22	8,1,11	266.47	16.53	1-4-12-18	opt	21.94	93.13	1-4-12-18	opt
0.8	5	4	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	5819.32	-17.39	4-11-12-17-20	opt	176.85	96.43	4-11-12-17-20	opt
0.8	5	5	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	2379.61	12.63	4-7-12-17-20	opt	80.55	97.04	4-7-12-17-20	opt
0.8	5	6	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	1648.14	-13.22	1-4-11-12-18	opt	47.09	96.77	1-4-11-12-18	opt
0.8	5	7	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	555.39	43.91	1-4-7-12-18	opt	34.16	96.55	1-4-7-12-18	opt
0.8	5	8	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	429.07	35.78	1-4-7-12-18	opt	26.73	96.00	1-4-7-12-18	opt
0.8	5	9	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	266.03	53.03	1-4-7-12-18	opt	23.68	95.82	1-4-7-12-18	opt
0.8	5	10	2,3,17,18,20,25	4,5,6,9,21	12,14,24,22,7,10	8,1,11	207.98	1.23	1-4-7-12-18	opt	13.13	93.76	1-4-7-12-18	opt
Avg.			-	-	-	-	*1069.01*	*27.86*	-	-	*46.10*	*94.94*	-	-
1	2	1	17,18,4,2,25,6,9			3,20,1,5,13,16,24,12,22,8	8.61	63.58	8-20-	opt	5.91	75.00	8-20-	opt
1	3	2	2,3,17,18,20,25	4,5,6,9,21	14,24	22,8,1,11	59.37	94.37	4-8-20-	opt	26.32	97.50	4-8-20-	opt
1	3	3	2,3,17,18,20,25	4,5,6,9,21	14,24	22,8,1,11	39.12	92.73	4-8-20-	opt	14.15	97.37	4-8-20-	opt
1	4	3	2,3,17,18,25	4,9,20,5,6,14,24,12,11,21		22,8,1,7,10	2227.08	20.40	4-8-17-20	opt	100.27	96.42	4-8-17-20	opt
1	4	4	2,3,17,18,25	4,9,20,5,6,14,24,12,11,21		22,8,1,7,10	1051.89	16.60	4-7-8-20	opt	67.18	94.67	4-7-8-20	opt
1	4	5	2,3,17,18,25	4,9,20,5,6,14,24,12,11,21		22,8,1,7,10	700.12	26.86	4-7-8-20	opt	32.02	96.65	4-7-8-20	opt
1	4	6	2,3,17,18,25	4,9,20,5,6,14,24,12,11,21		22,8,1,7,10	425.64	13.55	4-7-8-20	opt	24.19	95.09	4-7-8-20	opt
1	5	4	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	5759.66	N/A	4-8-11-17-20	opt	244.44	N/A	4-8-11-17-20	opt
1	5	5	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	4019.3	-4.59	4-7-8-17-20	opt	87.13	97.73	4-7-8-17-20	opt
1	5	6	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	2184.73	-0.54	4-7-8-17-20	opt	51.37	97.64	4-7-8-17-20	opt
1	5	7	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	1647.48	-28.66	1-4-6-8-18	opt	44.09	96.56	1-4-6-8-18	opt
1	5	8	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	1136.36	-6.41	1-4-6-8-18	opt	35.97	96.63	1-4-6-8-18	opt
1	5	9	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	764.17	2.33	1-4-6-8-18	opt	22.63	97.11	1-4-6-8-18	opt
1	5	10	2,17,18	4,25,6,9,3,14,12,21,20		22,8,1,11,7	399.41	5.80	1-2-4-7-8	opt	19.48	95.41	1-2-4-7-8	opt
Avg.			-	-	-	-	*1458.78*	*21.14*	-	-	*55.37*	*88.13*	-	-

In Figures 3.4 and 3.5, the suggested CBCA and three versions of this algorithm were evaluated to one another and to the original solution in terms of how fastly they solved the CAB instance. In Figure 3.4, the X axis reflects the rise in the value while the Y axis shows the solution time. Average values were considered for each α value response to changes in p and q. To put it another way, the average problem-solving time was calculated for 14 distinct scenarios after evaluating them from p=1 to p=5 within each α value based on different q cases. In a similar spirit, Figure 3.5 shows % improvement in problem-solving time over the original version on the y axis, while the x axis contains results.

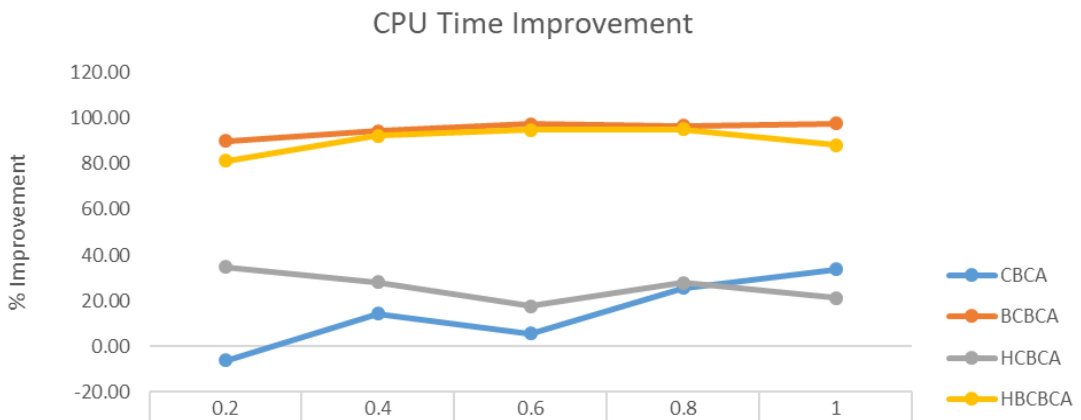
Reviewing both figures reveals that the increase in value causes the solution time to rise. This is the first obvious outcome. In a similar manner, an additional element prolonging the solution time is the increase in the value of p. Similar to this, the problem takes longer time to solve as fewer interconnections exist between hub regions. Another finding shows that, on average, all suggested solutions can address the issue faster than the original model. In terms of solution time, BCBCA and HBCBCA approaches obtain to the optimal solutions more rapidly than unrestricted variants. Even if the HBCBCA approach creates more sub-sets than BCBCA, this is still a drawback. For locating the best solution, HBCBCA provided a superior option than BCBCA.



	0.2	0.4	0.6	0.8	1
Cplex	46.27	230.37	721.36	1196.84	1572.78
CBCA	21.87	183.80	697.76	1050.90	1327.29
BCBCA	1.59	9.58	16.29	34.88	43.04
HCBCA	24.86	200.00	663.35	1069.01	1458.78
HBCBCA	4.99	17.80	25.19	46.10	55.37

Figure 3.4. Comparison of the methodologies for CAB data set in terms of solution time

The comparison of proportionate improvements in solution time between the CBCA technique and other CBCA-based approaches is shown in Figure 3.5. In this case, it was determined that the CBCA approach was just marginally less effective than the original solution for α=0.2. Improvements in terms of solution time are visible in all other instances. Improved solution times of up to 80% on average were obtained, especially by BCBCA and HBCBCA approaches that limited the list of possible hub vertices from subsets. Despite having the fastest results in terms of solution times, the BCBCA approach did not always get at the best solution. Average percentage gaps for values of α=0.2, 0.4, 0.6, 0.8, and 1 are, in order, 0.72 percent, 0.83 percent, 0.65 percent, 0.98 percent, and 3.27 percent.



	0.2	0.4	0.6	0.8	1
CBCA	-6.21	14.41	5.77	25.54	33.86
BCBCA	89.78	94.34	97.30	96.42	97.63
HCBCA	34.77	28.02	17.75	27.86	21.14
HBCBCA	81.12	92.19	94.64	94.94	88.13

Figure 3.5. CPU time improvements for the CAB data set

3.5.6. AP100 and TR81 data sets

The centrality-based solution approaches developed in this thesis have been implemented in AP100 and TR81 instances. To the author's knowledge, no comparable results for USAIp-HMP over these instances are available in the literature. Although Alumur et al (2009) reported some results for the TR81 sample, only solution times are included in this study. Also, in this study, not all 81 nodes are included in the candidate hub set. The problem is solved based on the identified 16 candidate hub locations and the results are presented. Therefore, no studies reported for incomplete-p hub median problems were found in the literature for AP100 and TR81 samples. As a result, for the AP100 instances and its variants comparisons were only made over full networks. All problems for the TR81 sample was resolved using the 16 possible hub locations that Alumur et al. (2009) revealed analysis. The HBCBCA technique, which was one of the approaches provided for each data set, was then used to do evaluations. Assessments based on the AP100 and TR81 data sets did not use the CBCA, BCBCA, or HCBCA methodologies. It can be asked the author for the outputs of all these methods.

The outcomes obtained from the AP100 data set are displayed in Table 3.5. Regrettably, results for p=10 & q=10 and p=15 & q=15 were not generated within the 24 hours provided for problem solution. However, all results for additional variations were identified in less than a day. The highest time required to solve a problem is around 11 hours in the case of p=10 and q=15. The scenario with p=10 and q=15 has a maximum problem-solving timeframe of about 11 hours. For strategic decision-making processes such problems identifying hub locations, it is a reasonable period of time. When comparing solutions obtained for whole hub network problems, there are gaps of 3.23 percent, 1.94 percent, and 2.49 percent for p=5, p=10, and p=15, respectively, based on best-known data. This condition indicates that the technique developed for complex network topologies also produced successful outcomes. The solution times are also good for incomplete networks. A sizeable portion of the vertices in the AP100 data set are concentrated in a very limited region. Finding the best hub nodes is therefore getting increasingly challenging. In real life, several factors might be considered (geographical characteristics, accessibility, infrastructure, skilled labor force etc.). Future models will have additional options for identifying hub sites if these indications can be included. When the number of nodes with identical properties rapidly increases, finding the best solution becomes much more challenging when just transportation costs are taken into account. Additionally, the solution time and a more adaptable solution method architecture for getting the best outcomes would both be substantially shorter if the methodology created on the basis of centrality measurements could be merged into heuristic procedures. The method developed as a consequence of this investigation provides opportunities to several additional study types.

Table 3.5. HBCBCA solutions on AP100 data sets and comparisons with best known solutions for complete networks

α	γ	δ	p	q	CPU time (sec)	Hub Locations	Obj.	Best Solution	Gap(%)
0.75	3	2	5	4	5330.18	7-29-66-70-77	146072.675		
0.75	3	2	5	5	3256.91	7-29-66-70-77	143587.146		
0.75	3	2	5	6	1150.86	7-29-66-70-77	141932.012		
0.75	3	2	5	7	831.84	7-29-63-70-77	141651.716		
0.75	3	2	5	8	623.44	7-29-63-70-77	141466.717		
0.75	3	2	5	9	535.09	7-29-63-70-77	141360.593		
0.75	3	2	5	10	344.91	7-29-63-70-77	141359.052	136929.44	3.23
0.75	3	2	10	10	> 24 hours				
0.75	3	2	10	15	40587.86	7-10-20-29-45-50-57-66-70-85	109399.043		
0.75	3	2	10	20	18337.19	7-10-20-29-45-50-57-66-70-85	108759.327		
0.75	3	2	10	25	7968.56	7-10-20-29-45-50-57-66-70-85	108764.285		
0.75	3	2	10	30	4433.89	7-10-20-29-45-50-57-66-70-85	108573.454		
0.75	3	2	10	35	3506.69	7-10-20-29-45-50-57-66-70-85	108541.218		
0.75	3	2	10	40	2337.52	7-10-20-29-45-50-57-66-70-85	108531.903		
0.75	3	2	10	45	1408.14	7-10-20-29-45-50-57-66-70-85	108529.966	106459.57	1.94
0.75	3	2	15	15	>24 hours				
0.75	3	2	15	30	35987.53	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	93561.593		
0.75	3	2	15	45	10714.89	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	92942.591		
0.75	3	2	15	60	7312.37	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	92870.065		
0.75	3	2	15	75	3468.47	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	92802.205		
0.75	3	2	15	90	1770.88	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	92789.625		
0.75	3	2	15	105	1305.17	7-10-13-16-20-23-28-36-45-50-57-66-70-76-85	92788.126	90534.785	2.49

Similar findings were observed for the TR81 dataset in Table 3.6. The TR81 data set was examined only for α=0.6 since it is difficult to analyze all cases for incomplete networks, particularly large-scale networks. As a result, the most likely value for economies of scale was considered. Other parameters included p=4, p=6, p=8, and p=10, and potential q values were examined using four cases for each p value. No result was obtained since the solution time for p=8 & q=7 and p=10 & q=10 exceeded 24 hours. In all other permutations, however, solutions were found within acceptable time scales. For α=0.6, p=6, and q=5, the maximum solution time is roughly 1 hour.

Table 3.6. HBCBCA solutions for TR81 data set and comparison with CPLEX solutions of original model with constant 16 candidate hub nodes

			CPLEX(with constant 16 candidate Hub)		HBCBCA			
α	p	q	CPU time (sec)	Best Solution	CPU time (sec)	Hub Locations	Best Solution	% decrease in transportation costs
0.6	4	3	1473.0	**104591121665**	3217.78	6-27-34-45	**104591121665**	0
0.6	4	4	506.61	**103052650660**	1326.83	6-27-34-45	**103052650660**	0
0.6	4	5	353.30	**102516696225**	658.47	6-27-34-45	**102251669225**	0
0.6	4	6	222.60	**102516696225**	621.75	6-27-34-45	**102251669225**	0
0.6	6	5	1932.7	85798778815	3846.35	1-6-21-34-45-52	**83974091686**	2.12
0.6	6	8	444.66	82545052632	2461.41	1-6-21-34-45-52	**81879662104**	0.80
0.6	6	12	57.91	81619281874	157.11	1-6-21-34-45-52	**80849830424**	0.94
0.6	6	15	35.91	81619281874	97.77	1-6-21-34-45-52	**80849830424**	0.94
0.6	8	7	>24 hrs	-	>24 hrs(%2.41)	1-6-7-21-25-34-35-58	**78435408047**	-
0.6	8	12	1653.5	73874198593	2065.16	1-6-15-16-21-34-35-52	**72810740136**	1.43
0.6	8	20	239.06	72814022038	748.69	1-6-15-16-21-34-35-52	**72116125036**	0.95
0.6	8	28	82.92	72774040019	541	1-6-15-16-21-34-35-52	**72096167564**	0.93
0.6	10	10	>24 hrs	-	>24 hrs			-
0.6	10	20	643.52	**66492791109**	3451.89	1-6-16-21-25-27-34-35-42-55	**66492791109**	0
0.6	10	30	117.38	**65919583336**	354.7	1-6-7-16-21-25-34-35-42-55	**65919583336**	0
0.6	10	45	37.33	**65855831725**	127.59	1-6-7-16-21-25-34-35-42-55	**65855831725**	0

3.5.7. Benchmark instances

In this section we focus on experiments on medium and large-scale data sets in order to clarify presented techniques. Because the CPLEX solver does not produce comparable solutions within 24 hours (86400 seconds), especially in large data sets (large networks with more than 100 nodes), we chose to employ Peker et al. (2016)'s technique (RCBS) by implementing it in the Python programming language. This part contains comparison analyses aiming at proving the efficacy of the algorithms we propose on large data sets and validating the results presented in the previous sections. The AP and CAB data sets, which are widely addressed in the hub location literature, as well as a randomly generated URAND data set, were employed for this purpose.

Table 3.7. Comparison on Medium and Large CAB, AP and URAND instances. A symbol "–" for gaps entries indicates that the instance was not solved to proven optimality within the time limit.

		CPLEX		RCBS		BCBCA		HBCBCA	
	n, p, q, α, γ, δ	gap (%)	time	gap (%)	time	gap (%)	time	gap (%)	time
AP40	40, 2, 1, 0.75, 3, 2	0.00	3.30	7.54	0.32	0.00	**0.12**	**0.00**	0.18
	40, 3, 2, 0.75, 3, 2	0.00	2867.51	1.73	2.55	7.45	**0.87**	**0.00**	0.87
	40, 4, 4, 0.75, 3, 2	0.00	3482.19	5.87	54.13	1.08	**3.66**	**0.00**	9.36
	40, 5, 5, 0.75, 3, 2	0.00	7221.36	5.12	26.66	1.35	**11.48**	**0.00**	17.71
	40, 5, 7, 0.75, 3, 2	0.00	1444.27	5.26	4.92	1.72	**2.51**	**0.00**	3.09
	40, 10, 25, 0.75, 3, 2	0.00	819.20	1.75	38.75	2.55	**17.69**	**0.00**	29.55
	40, 10, 30, 0.75, 3, 2	0.00	452.38	1.98	11.23	2.39	**8.34**	**0.00**	13.83
	40, 10, 35, 0.75, 3, 2	0.00	398.28	2.07	9.58	2.16	**1.17**	**0.00**	2.56
AP50	50, 2, 1, 0.75, 3, 2	0.00	15.58	9.96	0.44	0.00	**0.27**	**0.00**	0.33
	50, 3, 2, 0.75, 3, 2	0.00	14330.25	10.41	3.02	8.56	**1.13**	**0.00**	1.45
	50, 4, 4, 0.75, 3, 2	0.00	15474.86	2.71	82.14	2.10	**9.68**	**0.00**	12.1
	50, 5, 5, 0.75, 3, 2	0.00	26673.88	3.13	39.38	2.66	**22.08**	**0.00**	28.64
	50, 5, 7, 0.75, 3, 2	0.00	8506.72	3.44	5.02	2.78	**4.36**	**0.00**	4.55
	50, 10, 25, 0.75, 3, 2	0.00	14029.34	5.32	52.05	3.33	**27.69**	**0.00**	42.56
	50, 10, 30, 0.75, 3, 2	0.00	6860.59	5.41	18.14	3.57	**16.55**	**0.00**	21.38
	50, 10, 35, 0.75, 3, 2	0.00	3159.55	5.45	12.06	3.71	**6.18**	**0.00**	9.18
CAB100	100, 3, 2, 0.80, 1, 1	16.60	86400.00	3.50	564.18	4.58	**90.02**	**1.13**	130.48
	100, 4, 4, 0.80, 1, 1	17.00	86400.00	4.62	1214.55	4.44	**257.15**	**0.80**	345.19
	100, 5, 5, 0.80, 1, 1	22.92	86400.00	5.47	1711.25	3.71	**974.99**	**0.00**	1300.52
	100, 5, 7, 0.80, 1, 1	19.58	86400.00	5.34	1668.90	3.58	**903.03**	**0.00**	1245.63
	100, 10, 25, 0.80, 1, 1	11.16	86400.00	2.83	3618.57	4.75	**2004.18**	**0.16**	2547.19
	100, 10, 30, 0.80, 1, 1	9.49	86400.00	2.36	3275.41	4.62	**1966.57**	**0.18**	2284.36
	100, 10, 35, 0.80, 1, 1	9.05	86400.00	2.12	3002.85	4.47	**1900.24**	**0.21**	2005.17
AP200	200, 3, 2, 0.75, 3, 2	-	86400.00	7.36	10800.00	2.48	**162.43**	**0.00**	395.61
	200, 4, 4, 0.75, 3, 2	-	86400.00	3.58	10800.00	2.13	**247.33**	**0.04**	569.35
	200, 5, 5, 0.75, 3, 2	-	86400.00	3.15	10800.00	2.07	**588.31**	**0.02**	1847.47
	200, 5, 7, 0.75, 3, 2	-	86400.00	2.86	10800.00	1.95	**462.39**	**0.02**	1682.19
	200, 10, 25, 0.75, 3, 2	-	86400.00	6.64	10800.00	3.89	**1541.55**	**0.53**	3937.84
	200, 10, 30, 0.75, 3, 2	-	86400.00	5.77	10800.00	3.48	**1126.74**	**0.28**	3582.69
	200, 10, 35, 0.75, 3, 2	-	86400.00	5.61	10800.00	3.12	**987.55**	**0.21**	3268.25
URAND200	200, 3, 2, 0.75, 1, 1	-	86400.00	1.38	10800.00	3.51	**209.56**	**0.38**	266.04
	200, 4, 4, 0.75, 1, 1	-	86400.00	3.67	10800.00	3.85	**367.82**	**0.12**	452.27
	200, 5, 5, 0.75, 1, 1	-	86400.00	2.78	10800.00	2.78	**1100.07**	**0.49**	1529.08
	200, 5, 7, 0.75, 1, 1	-	86400.00	2.35	10800.00	2.35	**1215.33**	**0.35**	1384.75
	200, 10, 25, 0.75, 1, 1	-	86400.00	4.57	10800.00	4.78	**2388.28**	**1.66**	2967.71
	200, 10, 30, 0.75, 1, 1	-	86400.00	4.44	10800.00	4.52	**2119.74**	**1.35**	2619.98
	200, 10, 35, 0.75, 1, 1	-	86400.00	4.30	10800.00	4.33	**1994.19**	**1.22**	2341.57

Table 3.7 summarizes the results of medium and large-scale cases of partial uncapacited p-hub median problems generated by RCBS, BCBCA, and HBCBCA. The numbers in bold type in the 'gap' columns indicate that the investigated method produces a better solution than other algorithms. In the 'time' columns, the values in bolded provide a shorter solution time than the other algorithms.

The features of the instance types are listed in the first column of Table 3.7. The column contains the following information: network size, number of hubs, number of connections between hubs, economic of scale coefficient (in transit between hubs), node-to-hub transportation cost coefficient, and hub-to-node distribution cost coefficient. The remaining columns (from 2 to 9) provide the solution times of the CPLEX, RCBS, BCBCA, and HBCBCA techniques for each scenario, as well as the percentage gaps based on the best solutions. While CPLEX provides the best solutions for AP40 and AP50 data sets within reasonable time restrictions. However, the exact answers could not be reached within the one-day limited time for larger data sets. As a result, in CPU time comparisons, we only consider cases where the original formulation provides the best answer within one day (86400 sec). When CPLEX cannot find the optimal solution, we do the lower bound of the original formulation to compare solution quality (from CPLEX). We also established a time restriction of 3 hours for the RCBS, BCBCA, and HBCBCA methodologies.

Table 3.7 shows that the reported findings are identical to those derived over the CAB25 dataset with extensive analysis, as stated below.

- Clearly, when n (number of nodes) and p (number of hubs) increase, so does the average computing complexity (as in the CAB25 and TR81 instances).
- The HBCBCA algorithm finds the best solutions for networks with up to 50 nodes. Gaps of 0.35 % and 0.16 % were reported in the CAB100 and AP200 datasets, accordingly. This ratio was found to be around 0.80 % in the URAND200 data set. Given the vast of data sets and the difficulty of the problems, these gaps are tolerable.
- The BCBCA algorithm outperforms the HBCBCA method among all data sets. In the AP40 and AP50 data sets, the gap values were 2.33 % and 3.33 %, correspondingly. Large size problems, such as CAB100, AP200, and

URAND200, exhibited average gaps of 4.3%, 2.7%, and 3.7%, respectively. This is because the HBCBCA method extends the probable hub locations in the subsets while accounting for the betweenness centrality indicator. The BCBCA method, on the other hand, outperforms the HBCBCA algorithm in terms of solution time due to the lesser number of nodes in the subsets. The BCBCA method, on the other hand, surpasses the HBCBCA algorithm in terms of solution time owing to the decreased number of nodes in the subsets. In terms of solution time for all cases, the BCBCA algorithm outperforms the HBCBCA method by an average of 32%.

- In terms of solution times, the BCBCA algorithm appears to be superior in all instances. This is due to the minimal number of nodes in the subsets created by the BCBCA method. As a result, the number of decision variables in the problem reduces and solution time reduce. CPLEX results for AP40, AP50, and CAB100 datasets are also extremely long. The HBCBCA algorithm, on the other hand, has found optimum solutions in less time for AP40, AP50, and CAB100 examples.
- The solution time of the RCBS algorithm is relatively long, especially in large size problems. The percentage gaps acquired within the 3-hour time limitations appear to be larger than the HBCBCA method in networks with 200 nodes.

3.6. Statistical Analysis

According to the results of the above investigation, the HBCBCA algorithm outperforms all other algorithms in terms of generating optimal or best solutions among all problem dimensions. However, the Wilcoxon signed rank test is used to determine if the HBCBCA algorithm's findings differ significantly from those of other algorithms in terms of percentage gap values. The Wilcoxon signed rank test stands out for non-parametric analysis, as it does not require any population distribution to compare two paired and comparable samples. (Li et al. 2020; Li and Cheng 2017). This methodology is used in in the study to compare the results of two similar algorithms. The null hypothesis indicates that the outcomes of the algorithm pair have no statistical significance. The alternative hypothesis is that the algorithm pairs' findings are statistically significant.

For the Wilcoxon's test procedure, SPSS software was utilized, and the significance level was set to $\alpha = 0.05$. The p and z values represent the statistical variables employed in the analysis of the findings. The null hypothesis is false if the p value is less than the given significance level (0.05 in this case and at the 95 percent confidence level). The standard values from -1.96 to +1.96 are used to determine z values. The null hypothesis is rejected if the obtained z value exceeds these values.

The findings in Table 3.8 demonstrate that the HBCBCA algorithm outperforms other algorithms for medium-sized problems statistically. P values less than 0.05 were reported for hub location problems in both full and partial hu network topologies. Similarly, because the p values are considerably below 0.05, it is evident that there is a statistically significant difference between the BCBCA algorithm and the RCBS method (0.02 and 0.01). Three techniques are investigated through pair comparison in large-scale problems for both complete and partial hub network architecture. In such scenarios, the HBCBCA algorithm obviously outperforms. Wilcoxon tests for these problems demonstrate that the HBCBCA method is much superior to alternative algorithms, with all p values less than 0.05 (all p-values approximately 0.00). However, in the whole large-scale hub location problems, the p-value of 0.40 for the comparison of BCBCA and RCBS was more than the significance level of 0.05, indicating that statistical superiority was not confirmed. However, given that the R(+) value is 116 and the R(-) value is 74, it is clear that the BCBCA method outperforms the RCBS algorithm.

As a consequence, the evaluations in Table 3.8 reveal that the HBCBCA algorithm outperforms the BCBCA and RCBS algorithms statistically for both medium and large size problems (in different hub connection structures). In full large-scale hub locating issues, the BCBCA method does not show a statistically significant difference from the RCBS algorithm. However, in large-scale incomplete hub location problems, the null hypothesis is rejected, indicating that the BCBCA method beats RCBS statistically.

Table 3.8. Results of the Wilcoxon signed-rank test

Problem Scale	Graph Topology	Algorithm Pairs	R+	R-	N+	N-	z-value	p-value	Significance
Medium-Scale	Incomplete	BCBCA-RCBS	113.00	23.00	12	4	-2.33	0.02	Yes
		HBCBCA-RCBS	105.00	0.00	14	0	-3.29	0.00	Yes
		HBCBCA-BCBCA	136.00	0.00	16	0	-3.51	0.00	Yes
	Complete	BCBCA-RCBS	116.00	20.00	14	2	-2.48	0.01	Yes
		HBCBCA-RCBS	120.00	0.00	15	1	-3.40	0.00	Yes
		HBCBCA-BCBCA	136.00	0.00	16	0	-3.51	0.00	Yes
Large-Scale	Incomplete	BCBCA-RCBS	218.00	13.00	18	3	-3.56	0.00	Yes
		HBCBCA-RCBS	229.00	2.00	20	1	-3.58	0.00	Yes
		HBCBCA-BCBCA	184.00	6.00	18	1	-3.94	0.00	Yes
	Complete	BCBCA-RCBS	116.00	74.00	10	9	-0.85	0.40	No
		HBCBCA-RCBS	231.00	0.00	21	0	-4.02	0.00	Yes
		HBCBCA-BCBCA	231.00	0.00	21	0	-4.02	0.00	Yes

4. METAHEURISTIC ALGORITHMS FOR p-HUB MEDIAN PROBLEMS

In this chapter, algorithms known as SA, GA, GVNS, and R-GVNS are given as alternative solutions to the USAp-HMP problem. All aspects of the solution, including its representation and construction, as well as the development of initial solutions, new neighborhoods, and local search techniques, as well as the operator actions provided by the algorithm, are completely detailed. Experimental test results are given in Chapter 5.

4.1. Genetic Algorithm

Throughout history, humanity has sought solutions to problems by taking into account the physical events and the movements of biological entities in nature. In this union, many inventions and techniques have been developed as a result of human beings' examination of nature and living things in nature. Genetic algorithm is one of these techniques and is used by many researchers in solving various problems. Genetic algorithm is based on the rules of natural selection and genetics. These rules are based on the principle that dominant generations are protected their own lives, while weak generations are perished. Genetic algorithm is an optimization technique that aims to search for the best generations by using these two rules together (Sen, 2004).

The term “genetic algorithm” was used for the first time in the literature by Bagley (Bagley 1967). Later, Holland (1975) who worked on machine learning, imitated the processes of genetic functions in living things and adapted them to problem solutions. Then, it has been demonstrated on various experiments that the genetic algorithm can be used for optimization problems in the work presented by De Long (1975). In recent years, it has become an effective optimization tool due to its ability to reduce the complexity of problems and to obtain good solutions in a short time (Gupta and Imtanavanich, 2010). Especially in solving optimization problems, genetic algorithm has proven its competence and is one of the most used heuristic algorithms.

The genetic algorithm basically contains three different operators, which direct the actions of population multiplication, crossover, and mutation. Afterward the application of these operators, a new population is obtained. The new population replaces the old population. Every sequence has a fitness value. New generations are selected according to the obtained fitness value. Chromosomes with good fitness values are more likely to be passed on to future generations. The evolution process gradually increases the average fitness of the population and ensures that better fitness values are obtained in the following generations (Taşkın and Emel, 2009). According to Goldberg's definition, genetic algorithm is a heuristic search technique based on parameter coding, which tries to find a solution using the random search technique (Elmas, 2016).

In order to demonstrate the superiority of the genetic algorithm, some of its advantages over classical optimization techniques can be listed as follows:

The genetic algorithm is a population-based evolutionary algorithm. Therefore, instead of producing a single solution for a problem, it produces a set of solutions consisting of different solutions. In this way, with the diversification feature, many solutions are evaluated at the same time in the search space and as a result, the probability of reaching the best solution (or best solutions) increases. In addition, the solution set are completely or partially independent of each other. Each solution is represented as a vector on multidimensional space.

The good solution obtained in the genetic algorithm increases the chance of its use for new generations but does not guarantee it. Although the process of selecting individuals from the population is random, the fitness function values of the individuals have an increasing effect on the selection chance (Özkan, 2008). Thus, while the probability of choosing solutions with good fitness values is high, it is possible to evaluate solutions with bad fitness values. Thus, the risk of being trapped in the local optimum is reduced.

Genetic algorithm is a technique that can evaluate more than one solution simultaneously. Possible solutions or new generations are created on individuals with the best fitness value. In addition, the genetic algorithm is a probabilistic approach and searches different regions of the solution space at the same time (Özkan, 2008). In classical basic meta-heurisitic approaches, the best solution is sought only in certain regions of the solution space.

When searching the solution space, the genetic algorithm is considered the fitness value of the existing individuals and it is a blind search technique. It only has information about the solution process, the purpose of the problem is provided automatically. The genetic algorithm is efficient solution technique complex problems. It is an optimization technique that is frequently used especially in solving complex problems that are difficult to reach exact solutions or cannot be defined with a mathematical model. Like all heuristics, it does not guarantee optimality, but can achieve good results with short solution times.

4.2. Genetic Algorithm Conceptional Framework

Since the purpose of the genetic algorithm is to produce solutions to problems by imitating the evolutionary process, the concept definitions in the algorithm are based on a biological framework. To be successful in the implementation of the genetic algorithm in the application phase of the problems requires a good knowledge of the conceptual definitions. The concepts of genetic algorithm are given sequentially in the subsections.

4.2.1. Gene

Genes are parts of a living thing (individual) that carry the hereditary characteristics and contain the description of each character for the living thing. Genes come together to form the chromosome, that is, each of the possible solutions (Yücel, 2016). Genes represent a feature of the solution in the genetic algorithm. In the programming structure in which the genetic algorithm is used, the gene structures depend on the definition of the programmer. For this reason, the gene content may change according to the written program. (Diamond, 2007).

4.2.2. Chromosome (Individual)

There is a collection of individuals that make up a generation, and each of these individuals represents a chromosome. Chromosomes are formed by the combination of one or more genes and carry the features found in their genes. In the genetic algorithm, each chromosome represents a solution and must be correctly identified in the problem as it is responsible for the generation of new solutions.

4.2.3. Population

A population is a collection of individuals formed by living things in the same species and represents the set of possible solutions. The number of chromosomes (individuals) in the population is determined by the programmer according to the type of the problem and is considered constant for all iterated generations.

Determining the population size is one of the main challenges in genetic algorithm applications. If the number of individuals in the population is defined too many, the solution time of the problem is considerably longer. On the other hand, if the population size is small, it becomes difficult to retain good solutions within the generation. Therefore, balance is a very important factor when identifying population size.

4.2.4. Genetic operators

The basic genetic algorithm approach includes three different types of genetic operators. These operators are selection, crossover, and mutation.

Selection: This operator is selection to crossover the chromosomes in the population. The selection criterion is usually based on the fitness function values. Chromosomes with a high fitness value are more likely to be selected for reproduction. Thus, stronger generations can be obtained. Since at this step it is decided which of the chromosomes will be passed on to the next generation, it has a great impact on the performance of the algorithm.

Crossover: The crossover operator refers to the exchange of genes between chromosomes in the mating pool. There are different approaches to crossover in the literature. These approaches are specially designed and vary according to the nature of the problem being addressed.

The main purpose of the crossover process is to obtain new individuals from chromosomes with high fitness value to obtain better offspring. As a result of this process, which is performed with a parametric ratio determined at the beginning of the algorithm, new chromosomes with different properties are obtained. There are some points to be considered in determining the crossover ratio. For example, a high crossover rate may cause individuals with a high fitness function to leave the generation quickly. On the other hand, the low crossover rate is an obstacle to the differentiation of individual characteristics in the new generation. Therefore, the crossover ratio must be accurately determined, and a balance must be maintained between the current generation and the new generation. In crossover, it is expected to create better solutions by combining good features in individuals. There are basically four different crossover operators in the literature:

Single-Point Crossover: Single point crossover is the most commonly used crossover method in the literature. A random point is selected along the length of the matched string and values to the right of the specified point are swapped diagonally between individuals. If the changed genes are genetically good traits of individuals, better values will be obtained by taking these traits for next generations.

Two-Point Crossover: Two-point crossover works similar to a single-point crossover. In two-point crossover, two new individuals are obtained by replacing the sub-sequences between two points.

k-point Crossover: The k-point crossover method is an advanced version of the two-point crossover. Chromosomes are fragmented and new individuals are obtained from the pairs obtained by skipping the gene pairs of the selected length.

Uniform Crossover: In a uniform crossover, the chromosome is not divided into parts, but rather each gene is treated independently. In this type of crossover operator, we effectively flip a coin for each chromosome to see if it will be included in the offspring. We can also tilt the coin in favor of one parent in order to have more genetic material from that parent in the child.

In genetic algorithms developed for some types of problems, solutions for all possible chromosomes are not valid. Therefore, specially designed crossover operators are used to avoid violating the constraints of the problem. For example, in the traveling salesman problem, a chromosome contains an ordered list of cities that should be visited. As a result of using the crossover operators above, a chromosome may not contain all the cities that need to be visited. For such problems, different crossover operators can be used in the literature. Partially mapped crossover, cycle crossover, order crossover operator, order-based crossover operator, position-based crossover operator, voting recombination crossover operator, alternating-position crossover operator, sequential constructive crossover operator, simulated binary crossover operator are some of them.

Mutation: This operator randomly selects some genes in the chromosome and provides an exchange between these genes. This operator, which enables minor changes in gene sequence, is designed to recover useful genes lost in previous steps and to ensure diversity in the population.

The mutation operator exchanges one or more genes, resulting in different types of individuals. As with the crossover operator, the mutation rate must be determined correctly. If the mutation rate is too large, it will be very difficult to reach the optimum result because the gene sequence of individuals will differ. Since the mutation rate is lower than necessary, it will reduce the differentiation within the generation, especially the local search procedure will not be fully provided.

4.2.5. Selection

The step of a genetic algorithm known as selection is the process by which individual genomes are selected from a population for further reproduction (using the crossover operator).

The following are the details of how a generic selection technique may be carried out:

- During the evaluation of the fitness function for each individual, the generated fitness values are subsequently subjected to standardization. The term "normalization" refers to the process of dividing the value of an individual's fitness by the total value of all fitness values, with the goal of having the sum of all the generated fitness values equal 1.
- The aggregated normalized fitness values are computed as follows: the aggregated fitness value of an individual is the sum of that individual's own fitness value as well as the fitness values of all the individuals who originated before them; the accumulated fitness of the individual who originated before that should be 1; otherwise, whenever anything went wrong with the verification stage.
- It is decided upon a number Rr that is random between 0 and 1.
- The individual whose cumulative normalized value is first to be more than or equal to Rr is the one that will be chosen as the winner.

It's possible that the technique described above will need a lot of processing power for many different problems. One option that is both less complicated and more expedient is the so-called stochastic acceptance.

This technique of selection is known as fitness proportional selection or roulette-wheel selection when the operation is repeated until there are sufficient individuals chosen. It is referred to as stochastic universal sampling when, rather than there being a single pointer that is spun several times, there are multiple points that are evenly spaced on a wheel that is spun just once. The term "tournament selection" refers to the process of repeatedly picking the best individual from a subset that is determined at random. The process of truncation selection involves selecting just the top half, top third, or other fraction of the individuals.

There are further selection algorithms that do not take into account each and every individual for the selection process. Instead, these algorithms choose just those people whose fitness value is greater than a predetermined (arbitrary) constant. Other algorithms choose candidates from a limited pool, admitting just a certain proportion of the whole population on the basis of their level of physical fitness. Methods such as Roulette Wheel Selection, Rank Selection, Steady State Selection, Tournament Selection, Elitism Selection, Boltzmann Selection are the most used selection processes in the literature.

4.2.6. Stopping criterion

As in all meta-heuristic algorithms, determining the termination criterion is one of the most important problems in genetic algorithms. The termination criterion is a condition that determines in which state or situations the genetic algorithm process should stop. This criterion can be determined depending on the number of iterations or running time, as well as stopping the algorithm when it reaches the best-known solution. The determination of the stopping criterion may vary depending on the difficulty and type of the problem and is also related to the intuition of the decision maker. In terms of terminating conditions, there are a variety of options that may be combined and chosen dependent on the application. The most frequent termination criteria are population and time based, solution based, and a combination of these elements. Solution convergence, fitness value limit, generation limit, population convergence, and time limit are some of the metrics used to determine the boundaries for stopping criteria.

4.3. Genetic Algorithm Implementation to Hub Location Problems

Genetic Algorithm (GA) is search algorithm which is inspired from theory of evolution and using heuristic search. It introduced by James Holland in his seminal work ”Adaptation in natural and artificial systems” in 1975. The workflow of GA consists of 6 steps:

1.Initialize Population

2.Compute Fitness

3.Selection

4.Crossover

5.Mutation

6.Local Optimization

The basics of the chromosomal representation, fitness assessment, and genetic operators that were employed are presented in this section. In the GA, each chromosome in the randomly generated population pool is turned into a hub-spoke network, which is then used to connect the nodes of the network. Afterwards, the chromosomes are exposed to an evolutionary process until a minimum cost hub-spoke network is formed or the termination condition is reached, whichever comes first. Genetic and selection operators on chromosomes are used to carry out the evolutionary process, just as they would be in a conventional GA, as seen in Figure 2. Fitness-based selection is accomplished via the use of tournament selection with elite survival. This component of the USAHLP contains two new solution representation schemes as well as two new problem specific crossover operators that are suggested for the USAIPHLP. In addition, the GA makes use of three mutation operators in a probabilistic manner.

```
Set the parameters of the genetic algorithm
Read problem specific data
Generate Initial Population
       While (until the termination criterion is met)
              Generation of the Solutions ()
              Evaluation ()
              Selection ()
              Crossover ()
              Mutation ()
        end While
end process
```

We developed our algorithm according to these steps. Firstly, for Initialize Population, we need the population format which also called chromosome in GA. In our approach, GA chromosome consists of three arrays: Hub Array, Allocation Array and Connection Array as represented in the Figure 4.1. Hub Array is represented by binary values: 1 and 0. 1 is for hub and 0 is for spokes. The Allocate Array is using for assignments of the spokes to hubs. If a spoke is assigned to hub, this hubs value is presented to Allocate Array. The last sequence is related to the connections between the hubs and we named the connection array. It can take 0 and 1 values on the elements in the arrays and indicates whether there is a connection between the two hubs. If the element of the array has a value of 0, there is no connection between the two hubs and vice versa. Illustrations related to this are given in the previous section.

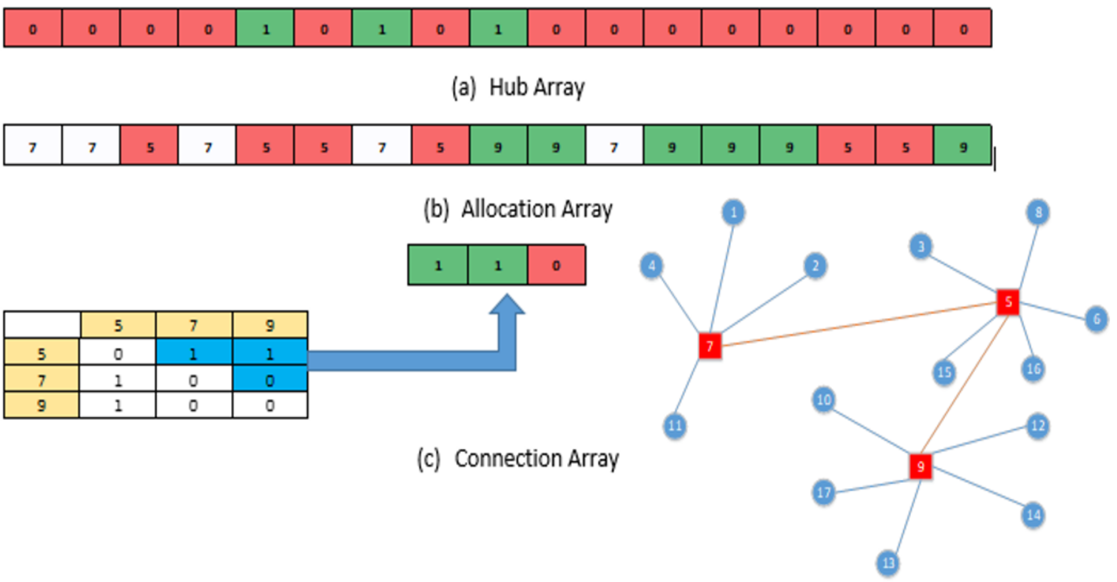


Figure 4.1. Solution Representation of Hub Network and Allocation Scheme

For each non-hub node, we create the array of located hub facilities and arrange it in non-decreasing order of their distances from the current node. Hub nodes are assigned to themselves. We can notice that optimal solution usually does not include allocation of each origin/destination node to its nearest hub. Indexes of hubs closer to non-hub nodes appear often in the optimal solution, while the indexes of far away hubs are rare. For this reason, we direct our search to "closer" hub facilities, while the "distant" ones are considered with small probability. Sorting the array of hubs in non-decreasing order of distances from each non-hub node, we make sure that closer hubs have higher priority in assigning them to non-hub nodes.

Genetic operators for hub location problems

Our algorithm uses tournament selection method for selection of two parents. After a pair of parents is selected, crossover operator is applied to them producing two offsprings. Since genetic code of each parent consists of two segments with different nature, standard one-point crossover operator directly used. In each segment of parents' genetic codes a crossover point is randomly chosen and genes are exchanged after chosen position. Crossover is performed with the rate C_{prob}=0.85, which means that around 85% individuals take part in producing offsprings. We apply single-point crossover operator on HubArray and AllocationArray of the input strings by considering the same crossover point selected at random. The offsprings are generated by combining the left and right parts, which is followed by a phase for adjusting the offsprings, if necessary. After applying the crossover on HubArrays, if any of the offsprings does not have p number hub, then both of the offsprings will be rearranged. For any offsprings of the AllocationArrays, if a node i is assigned to another node which is not a hub anymore because of crossover on HubArrays, then node i is reassigned to the closest hub with respect to distance values or randomly. The same applies to connection arrays. If the sum of the bits in the chromosome obtained by the crossover is not equal to the total number of connections q (the total number of connections), the two closest hubs are connected to each other, or the connections are determined as random.

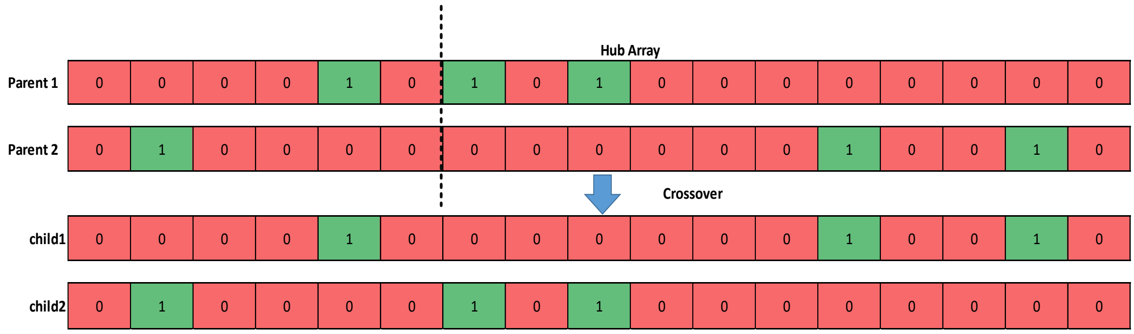


Figure 4.2. Crossover operation based on two individuals

In Figure 4.2, the nodes 5, 7 and 9 are hubs for parent1 and nodes 2, 13 and 16 are hubs for parent2, initially. After applying the crossover on HubArrays, the hubs of the first offspring child1 are 5, 13 and 16; and the hubs of the second offspring child2 are 2, 7 and 9.

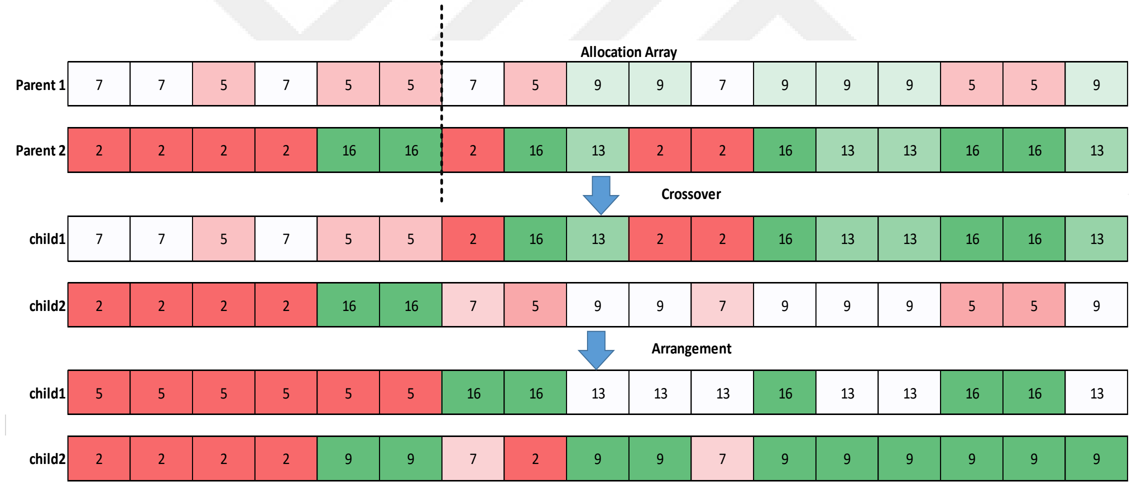


Figure 4.3. Individual gene arrangement process

After applying the crossover on AllocationArrays, the assignments of nodes 1, 2, 4, 7, 10 and 11 in child1 need rearrangement based on distance values and randomly, since node 7 and node 2 are not hubs any more in the HubArray child1 and similar adjustment is done for the second AllocationArray child2.The rearranged nodes are assigned according to the distance value with probability of 0.9 and assigned randomly with probability of 0.1.

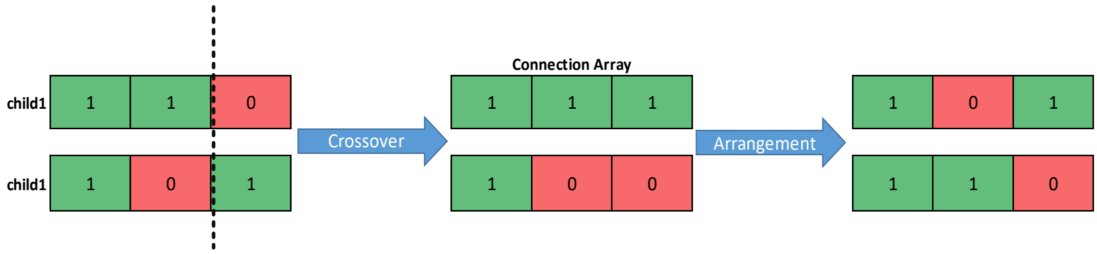


Figure 4.4. Crossover process on hub connections

Considering the connections between hubs, there are connections between 5-7 and 5-9 for parent 1. For Parent2, there are connections between 2-16 and 13-16. After applying the crossover on Connectionarrays, all connections to child1 are complete, and there is one connection to child2. In this study, the nearest hubs are connected to each other with probability of 0.90, and the hub connections with probability of 0.10 chance are performed randomly and we rearrange Connectioarrays according to this logic. The reason why random connections are considered is due to the fact that in p-hub median problems, optimal results are not always guaranteed to be the closest hubs.

The mutation operator is applied only for the assignments of nodes. We consider two mutation operators, called shift and exchange. Shift operator selects a spoke and reassigns it to another hub selected at random. If there is only one hub in the string, this mutation operator is not applied. Exchange selects two spokes at random and switches their assignments. Since we need at least two pairs of hubs and nodes, if there is only a single hub or only a single spoke, i.e., all other nodes are hubs in the string, then the exchange move is not applicable.

Figure 4.5. Mutation process on individuals

Offsprings generated by crossover operator are subject to mutation. Mutation operator is performed by changing a randomly selected gene in the genetic code. Mutation rates differ for the hub array, allocation array and connection array of the genetic code:

Algorithm: GA for USAIp-HMP (pop, gen, C_{prob}, M_{prob}, t_{size})

```
Generate an initial population randomly (pop)
Calculate fitness values for each individual
while ((convergence is not achieved) and (number of generation ≤ gen))
    for i=1 to pop/2 do
        while true (parent1≠parent2)
            select number of t_size chromosome randomly
            choose the best one (parent1)
            select number of t_size chromosome randomly
            choose the best one (parent2)
            break
        procedure crossover(parent1, parent2, P_cross)
        procedure mutation(child1, child2, P_mut)
        calculate  chromosomes fitness functions
        insert offsprings into the new generation
        apply elitism for the new generation
    end for
end while
return the best chromosome and best fitness
procedure crossover(parent1, parent2, C_prob)
if random (0,1) ≤  C_prob     then
    select crossover point randomly
    apply single point crossover on Hubarray, Allocationarray, Connectionarray
    obtain child1 and child2
    rearrangement procedure
end if
procedure mutation(child1, child2, M_prob)
if random (0,1) ≤  M_prob     then
select mutation operator randomly
end if
if mutation operator=shift then
    select a gene randomly
    select a hub randomly
    assign the gene to the hub (if the hub include the gene)
else
    select two gene from different hubs
    exchange these genes between the hubs
end if
```

4.4. Simulated Annealing Algorithm (SA)

SA, developed by Kirkpatrick et al. in 1983. It is a local search method to handle optimization problems. On the resemblance between the annealing process in physical systems that reduce the energy state of the solid and the solution process in combinatorial optimization problems, the SA technique, Metropolis et al. have developed the Metropolis et al. (Kirkpatrick et al., 1983).

Annealing is a kind of heat treatment that is used to reduce the energy levels of a material. When the temperature is increased to the melting temperature of the solid, it is said to be melting. The energy and freedom of the high-temperature solid are very high. The operation of cooling slowly is conducted in the following procedure. Because of the cooling process, when thermal balance is achieved, the molecules in the solid create a more stable structure. The passing of time is a critical factor in this process. If the cooling is carried out too quickly, irregularities and distortions in the crystal structure arise (Kirkpatrick et al., 1983).

When a solid is in distinct states in a physical system, the different states of the solid correspond to different possible solutions in a combinatorial optimization problem, and the energy of the system corresponds to the objective function of the problem. The ground state represents the optimum solution on a global scale, while the quasi-stable state represents the best solution on a local scale. The SA method begins with an initial solution and a temperature value that is somewhat high in order to prevent a local minimum. When a solution with a local neighborhood is found, the algorithm creates the next solution, and the temperature lowers in accordance with a predetermined rule. A new solution that accurately depicts the system's energy level while also improving the objective function is always considered as a valid solution. Partial solution suggestions that allow for a rise in the temperature of the system or a certain degree of divergence or degradation from the goal function of the system, on the other hand, are also acknowledged as acceptable. This is due to the fact that you are not restricted to the finest option available locally. Until the algorithm ending rule is realized: if the new solution is approved, the algorithm continues to work with the new solution; if the new solution is prohibited, the algorithm continues to work with an already existing solution. When the algorithm is stopped because it has satisfied the criteria (minimum temperature value, objective function change, or the number of iterations), the operations continue until the condition is met again.

The condition of accepting a new solution of the algorithm is realized according to the Boltzmann rule. If the neighboring solution is better than the current solution, it always replaces the new solution. However, if the neighboring solution is worse than the current solution, a choice is made according to the Boltzmann rule. The reason for considering worse solutions than the current solution is to get rid of the local optimum. The difference between the objective function value of the current solution and the neighboring solution value shows ΔE. If the acceptance condition calculated with the ΔE value is met, the result obtained with the local neighborhood is accepted as the current solution. The Boltzmann rule is defined as:

$$P(\Delta E, T) = e^{\frac{-\Delta E}{T}} \quad (4.1)$$

As the temperature (T) value increases, the P value decreases, which indicates that the chance of adopting more no neighbor solution reduces with increasing temperature (T). When performing at a certain temperature level, the user determines how many new solutions will be evaluated overall. Nevertheless, after a steady state has been attained (and no further progress in the solutions has been seen), the temperature is often reduced. Both current solutions and the best solutions that may be found should be represented using this procedure (Rosocha, Vernerova, & Verner, 2015). As the T value approaches zero, the majority of the solutions that result in an increase in the E value, that is, the value of the objective function, are ruled out of consideration. Consequently, even if there are periodic deteriorations in the objective function, the process is prevented from being trapped in local optima, and it is feasible to achieve the process's overall optimal performance provided the objective function is maintained.

4.4.1. General and problem specific parameters in SA

Some parameters of the SA algorithm should be established externally by the user at the beginning of the algorithm running process in order to function properly. Due to the fact that such parameters have a significant impact on the performance of the algorithm, they should be properly examined.

The basic choices that must be taken for the SA algorithm include the initial temperature (T(0)), cooling plan (number of repeats, rate of temperature drop), and termination criteria, among others. These kinds of decisions may be found in all types of problems

SA algorithm in a same way (not problem specific). However, since we have more than one option for each of these criteria.

- **Initial Temperature:** To be efficient algorithm, the initial temperature should be high enough to ensure that adopting the preliminary solutions has a high possibility of accepting. If, on the other hand, to choose a very high beginning temperature, the computational cot will be very high and the algorithm give poor performance (Temiz, 2010). It is possible to catch local minimums when the beginning temperature is set to a low degree. This means that choosing an appropriate beginning temperature is quite crucial factor to good performance and efficiency of the algorithm (Kendall, 2000).
- **Cooling Process:** The cooling process in SA is one of the aspects that has a considerable impact on the overall performance of the algorithm, as previously stated. This parameter adjusts the temperature value of the algorithm in accordance with the temperature reduction principle that has been defined.

  In the literature, it is possible to come across references to the use of various cooling methods. In this investigation, the geometric cooling strategy that was applied is just as follows:

$$T_n = T_{n-1} \times \alpha \qquad (4.2)$$

  In the equation T_n represent the reduced temperature in iteration n and T_{n-1} shows previous period temperature. α is the reduction factor (cooling rate) to decrease the temperature in each iteration. α value must be in 0-1 interval. However, a value close to 1 is usually chosen because a sudden decrease in temperature increases the chance of stuck in the local optimum.

  The geometric cooling process was first proposed by Kirkpatrick in 1983 and Kirkpatrick (1983) used the cooling rate in the range of 0.8< α <0.95 in this study. Experiments show that α gives better results when selected between 0.8 and 0.99. However, it should be taken into account that as the value of α increases, the number of iterations will also increase (Kendall, 2000).

In developing a cooling strategy, there are two critical factors. Using the ending criteria when the temperature converges to 0 is the first of these steps. Because waiting for the temperature to reach zero is unproductive because it needs too many iterations and there is little probability of changing the best solution, waiting for the temperature to hit zero is useless. In order to do this, the algorithm's terminating condition must be fulfilled when the temperature converges to 0. Another problem to consider is the amount of neighborhood searches that must be conducted at each temperature setting. It is referred to as the number of repetitions in the literature when referring to this. When a sufficient number of motions are approved in accordance with a predetermined upper limit, the number of repeats is decided. It is anticipated that, in this manner, the issue would approach an equilibrium state that corresponds to the thermal equilibrium seen during physical annealing. The number of repetitions might be treated as a constant or as a function of the magnitude of the issue being solved. This means that for a certain number of consecutive temperature changes, the solution found at each value of the temperature parameter does not change and the process is terminated.,

In the literature, many cooling functions have been derived apart from the geometric cooling schedule. These are Exponential multiplicative cooling, Logarithmical multiplicative cooling, Linear multiplicative cooling, Quadratic multiplicative cooling and their additive type ones. However, since we see that the cooling mechanism we use is not sensitive to the solution time and quality of the algorithm, we use the simplest geometric cooling mechanism.

- **Stopping Criteria:** When the existing solution does not change any longer from one iteration to the next throughout a sufficiently high number of iterations, it is determined that the algorithm has been terminated. Specifically, the stopping criteria determines when the SA algorithm should be terminated. There are many such systems, each of which is based on some specified value, the actual results of the search, or factors when continuing the search is regarded too costly or doubtful to result in significant improvements. Basically, the stopping criterion can be examined under two headings. The first of these is Fixed termination criteria, and the other is the adaptive approach.

Maximum time, minimum temperature and a fixed number of iterations can be defined for the fixed termination category. Adaptive approaches, on the other hand, are mostly related to the improvement capabilities of the algorithm on the solution. For example, the number of improvements to the solution, the number of improvements in one iteration, the acceptance rate for the last k candidate moves in a given threshold, nothing at all of the last k candidate movements resulted in a new best solution being discovered etc.

4.4.2. SA implementation to hub location problems

In this part, we focus on simulated annealing-based metaheuristic algorithm for solving uncapacitated single allocation incomplete p-hub median problems. SA algorithm was developed by Metropolis et al. (1953) in 1953 and it is in the metaheuristic algorithms class. Due to its simple solution structure, it is frequently used in the literature to solve combinatorial optimization problems and can achieve effective results. Kirkpatricketal (1983) and Cerny (1985) enhanced SA algorithm in their studies. The SA algorithm start with a random initial solution to solve optimization problems. Then tries to develop current solution with neighborhood structures. If the newly produced solution is better than the current solution, the current solution replaced with the newly found solution. If the new solution worse than the current solution, it can replace or not replace the existing solution based on a probability value calculated with the formula $\exp^{-\Delta E/T}$ (if the problem type is maximization, then this formula change $\exp^{\Delta E/T}$). In this formula, ΔE indicates that the difference between the objective function value of the current solution and the new solution. T denote current temperature. For each temperature, the algorithm searches better solutions within a certain iteration limit, and then the temperature is reduced at a specific rate. At higher temperatures the solution to replace current solution is more likely to be adopted. Therefore, poor solutions can also be accepted. However, as the temperature decreases, the likelihood of accepting poor solutions decreases. As the iteration progress, the algorithm converges or reaches to the optimal solution.

4.4.3. Solution Representation

Three types of one-dimensional arrays are used to illustrate the solution. The first array shown in Figure 4.6 (a) indicate the hub nodes. If any element in the specified array has a value of “one”, it indicates that the node selected as a hub point. Otherwise, this node is called non-hub node if it corresponding to “zero” elements.

0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

(a) Hub Array

7	7	5	7	5	5	7	5	9	9	7	9	9	9	5	5	9

(b) Allocation Array

1	1	0

	5	7	9
5	0	1	1
7	1	0	0
9	1	0	0

(c) Connection Array

Figure 4.6. Solution Representation for SA

The second array shown in Figure 4.6 (b) denote the allocation of non-hub nodes to hub nodes and is called the allocation array. The network structure of this array is shown in Figure 4.6 (c). As can be seen in this figure three nodes (nodes 5, 7 and 9) are established as hubs and corresponding elements in the first array in figure4 (a) take the value of 1. For instance, as can be seen in Figure 4.6 (b), first node (node 1) allocated to the hub 7. Similarly, nodes 2, 4 and 11 assigned to hub 7, since the second, fourth and eleventh elements corresponds with 7. In addition, each hub in the array allocated to itself.

4.4.4. Initial Solution Generation

The initial solutions are generated randomly in our algorithms. To this end, we randomly select p out of |N| nodes as hub nodes. However, non-hub nodes to be assigned to hub nodes are determined according to the nearest neighborhood strategy. Based on this operator, which is originally proposed by O’Kelly (1987), for a given set of hub nodes, each non-hub node is allocated to its nearest open hub. A similar approach applies to connections of hub nodes in incomplete network structures. Each hub node is connected other by its closeness situation according to the other hubs. If there is more than one connection for a hub node, the connections will be established with most close distance. Figure 4.7 shows a visual representation of this situation.

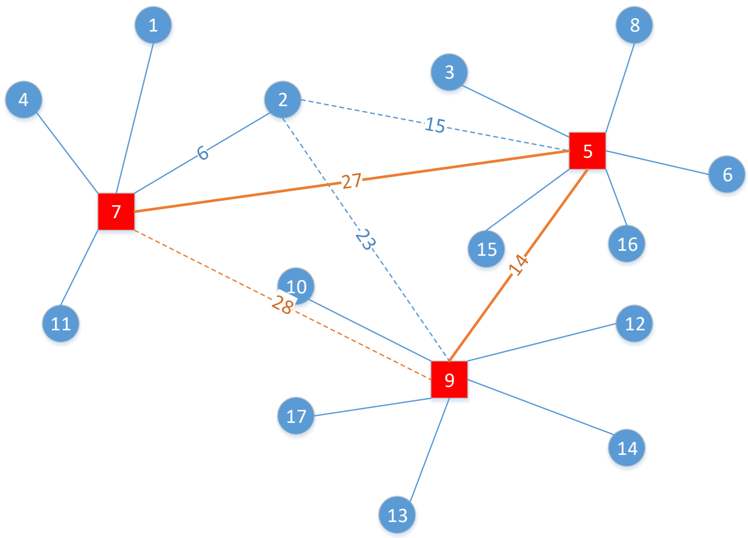


Figure 4.7. Nearest Neighboring Strategy

4.4.5. Neighborhood Structures

We used five different operators to generate neighboring solutions for USAIp-HMP with SA algorithm. Each operator is designed to generate new neighboring solutions randomly from the current solution, and their definitions are as follows;

- **Internal_Swap_Hub:** This operator is used to change one of the hub node in the current solution. Firstly, a random hub node and non-hub node is selected. However, the non-hub node must be connected to the selected hub node. In the other words, we change hub nodes from the same cluster. The selected non-hub node is then defined as a hub. Also, hub node is assigned to the new hub node in the new solution.

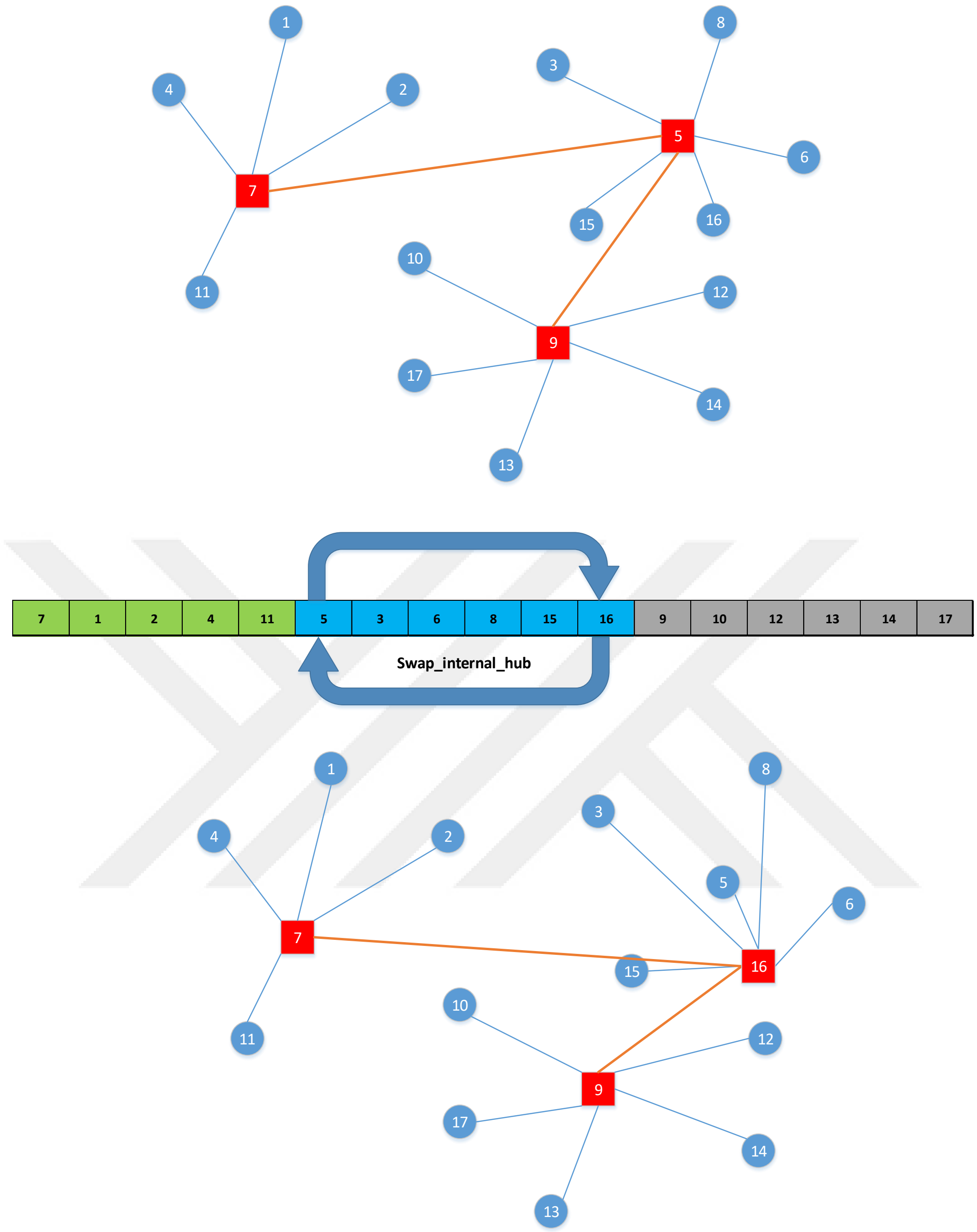


Figure 4.8. Swap internal hub operator

- **External_Swap_Hub:** This operator is similar to previous one but has a small difference. In this neighboring policy, a hub node is selected randomly, but for the candidate node that replace the selected hub node, must be connected to another hub node. In the other words, it must be another cluster.

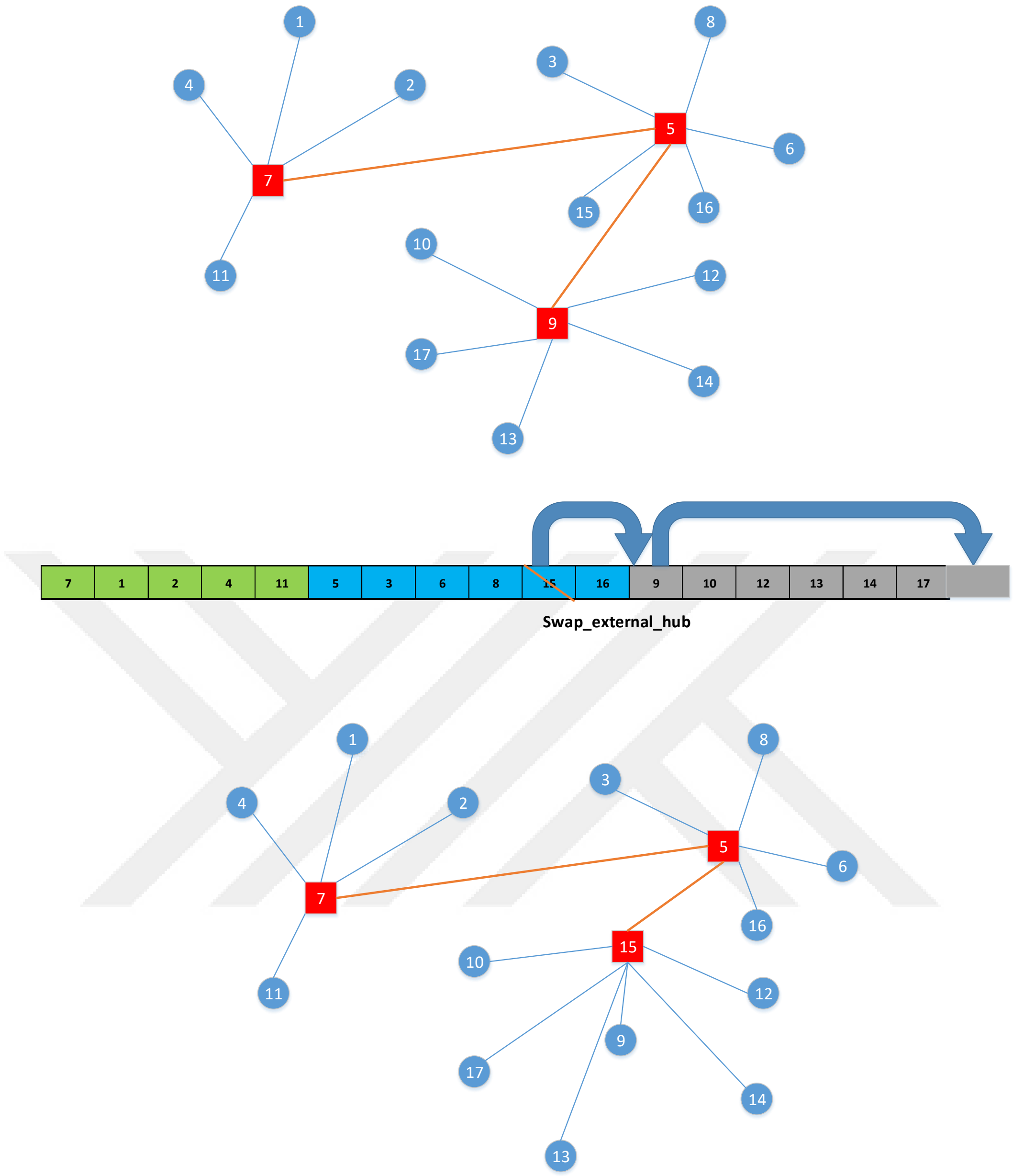


Figure 4.9. Swap external hub operator

- **Swap_Node:** This operator changes the allocation of a randomly selected non-hub node to a hub node other than its current hub.

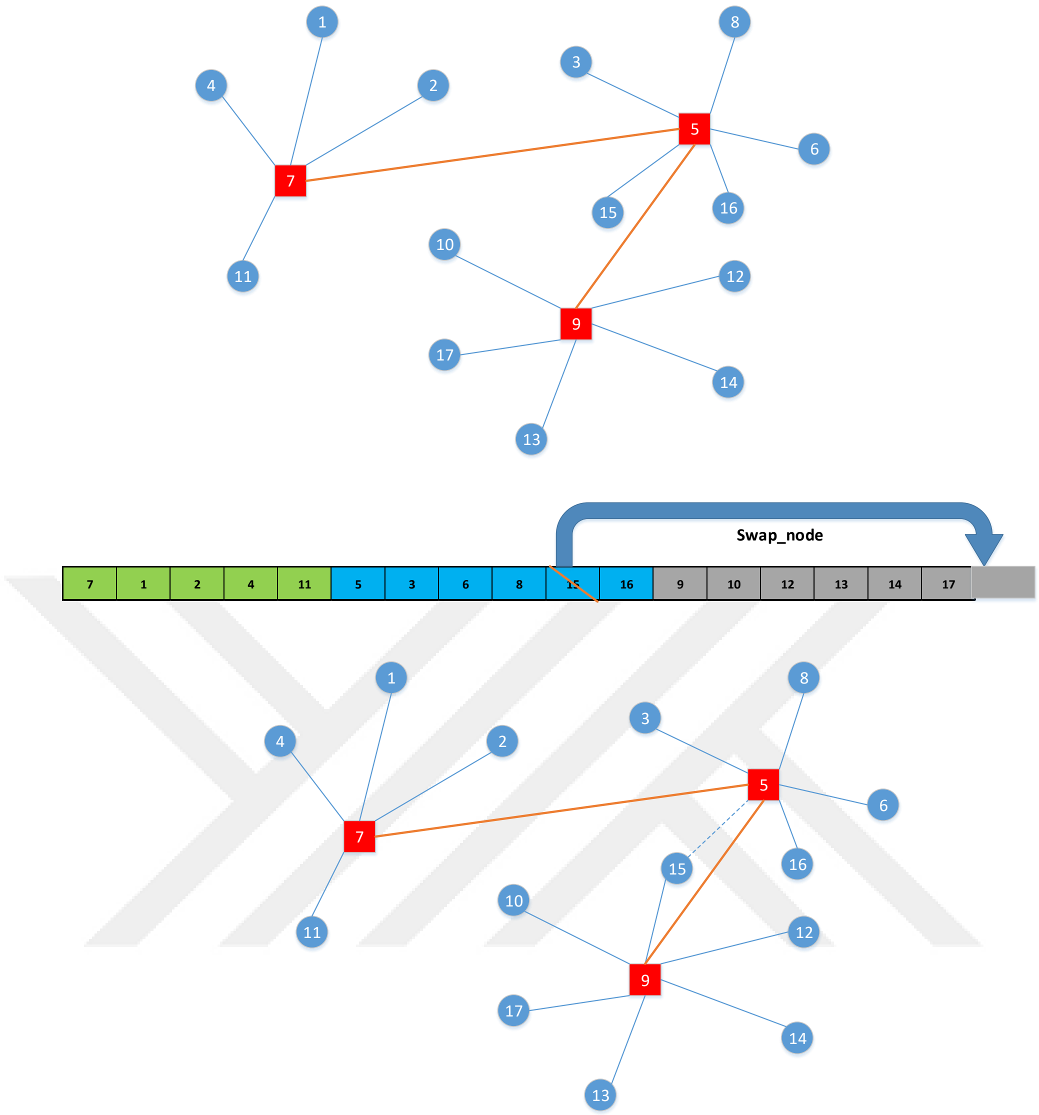


Figure 4.10. Swap node operator

- **Nearest_Allocation:** In the nearest allocation neighborhood operator, the non-hub node is assigned to its nearest open hub. In the algorithm, initial solutions are established according to this strategy. In particular, displace in hub points change the distance of non-hub nodes to their nearest hub nodes. For example, after the application of the swap_internal_hub operator non-hub node 12 close to the hub node 16 and move away the hub node 9. In this case, if this neighborhood structure is applied, node 12 will be assigned to the hub 16.

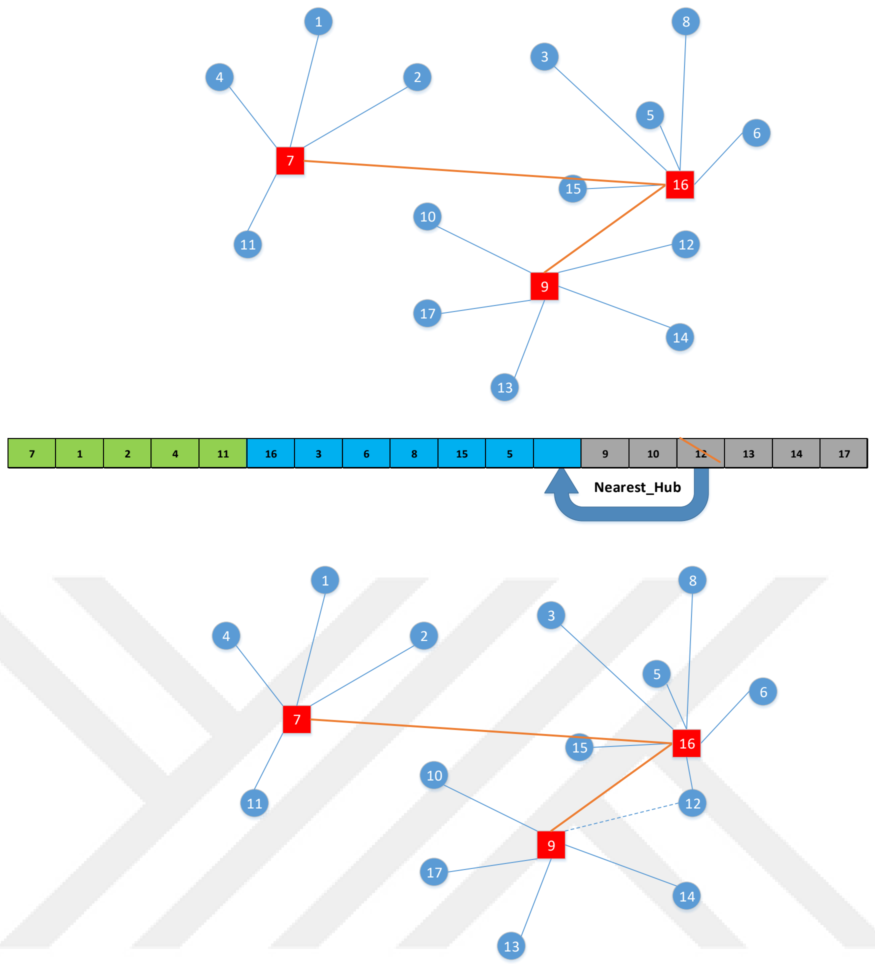


Figure 4.11. Nearest hub operator

- **Swap_Hub_Link:** Especially in incomplete hub networks, the connection of hub nodes with each other is underestimated. In most cases, the closest hub nodes are connected, but this is not always true. Therefore, it is also necessary to search the neighborhood with the connections of the hub nodes. This operator is used to change the connection of a hub point to another hub node.

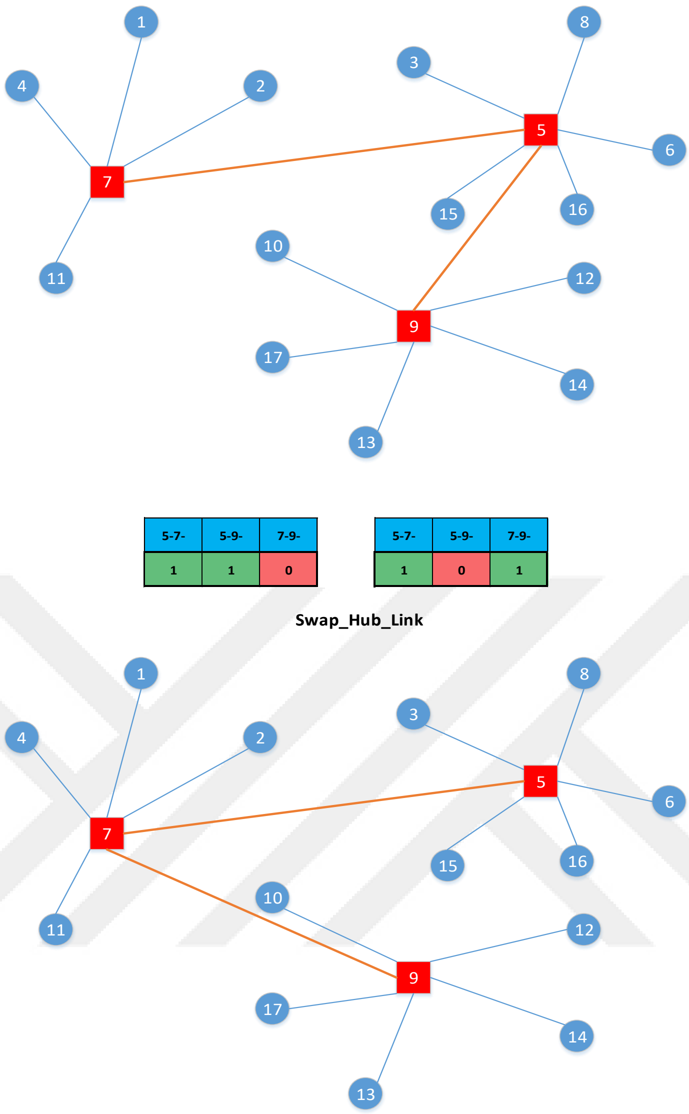


Figure 4.12. Swap hub link operator

4.4.6. Parameters used in the SA

In developed SA algorithm, there are a total of five parameters, T_s, T_e, C_r, N_i, and r. T_s denotes the initial temperature and T_e represent the final temperature in the algorithm. The C_r controls cooling rate and determines speed of cooling. N_i shows the total number of iterations at each temperature. As the last parameter r determines rate of increase in the number of iterations, depending on improvement in the objective function value (increase for maximization problems, decrease for minimization problems). For instance, when N_i is 100 and algorithm achieves an improvement to the objective function at sixtieth iteration, the number of iterations extended to N_i+(60-60*r) and the local search increased at the same time.

4.4.7. The Overall SA Algorithm

In the USAIPHM problem, it is assumed that the locations of all candidate hub nodes are known. Firstly, the SA algorithm generate initial solution such as S_p. In the first step, hub nodes determine randomly between the candidate hub nodes. The objective function value of the solution obtained as a result of the neighborhood operators applied in each iteration is calculated. The difference between newly generated solution and the current solution objective function values is calculated as $\Delta E = f(S'_p) - f(S_p)$. If $\Delta E > 0$, and $S_p \leftarrow S'_p$. Besides, if the new solution is better than the new solution, the best solution will be the new solution ($S_{best} \leftarrow S'_p$ and $f(S_{best}) \leftarrow f(S'_p)$). If $\Delta E \leq 0$, another random number is generated in the range [0,1] to check whether the number is greater than the value come from exp(ΔE/T) formula. If the randomly generated number is greater than probability value, the new solution replaces the current solution even though it is poor than current solution. In the other words, we accept worse quality solution with probability exp(ΔE/T) to stand out from the local minima. At the end of each local search iteration, the temperature level is reduced, and we use geometric temperature reduction process as in $T = TxC_r$ equation. The pseudo codes of the SA algorithm developed for USAIp-HM problem are shown in Algorithm SA for USAIp-HMP.

Algorithm: SA for USAIPHMP (T_s, T_e, C_r, N_i, r)

Generate an initial solution randomly (S_p)
Calculate objective function value for initial solution ($f(S_p)$.)
$T \leftarrow T_s$; $S_{best} \leftarrow S_p$; $f(S_{best}) \leftarrow f(S_p)$;
while $T>T_e$ do
  $h \leftarrow$ random (0,1)
  if ($0 \leq h < 0.2$) then
    Generate a new solution S_p' from the S_p with swap_hub_internal operator
  else if ($0.2 \leq h < 0.8$) then
    Generate a new solution S_p' from the S_p with swap_node operator
  else
    Generate a new solution S_p' from the S_p with swap_link operator
  end if
  calculate $f(S_p')$;
  $\Delta E \leftarrow f(S_p') - f(S_p)$
  if ($\Delta E>0$) then
    $S_p \leftarrow S_p'$; $f(S_p) \leftarrow f(S_p')$;
  else
    $h \leftarrow$ random (0,1)
    if ($h > \exp(\Delta E/T)$ then
      $S_p \leftarrow S_p'$; $f(S_p) \leftarrow f(S_p')$;
    end if
  end if
  if ($f(S_p) < f(S_{best})$) then
    $S_{best} \leftarrow S_p$; $f(S_{best}) \leftarrow f(S_p)$;
  end if
  $T=TxC_r$
  Generate a new solution S_p from the S_{best} with swap_external_hub operator
end while
return S_{best},$f(S_{best})$

4.5. General Variable Neighborhood Search Algorithm (GVNS)

The variable neighborhood search, often known as VNS, is a kind of metaheuristic that was first developed by Mladenovic and Hansen in (1997). The most fundamental version of the variable neighborhood search algorithm is referred to as Basic VNS, and it is comprised of the following steps: alternatingly executing one local search procedure (used to improve a solution) and one so-called shaking procedure (used to hopefully resolve local minima traps); additionally, changing the neighborhood. The fundamental VNS completes its tasks after a pre-defined stopping condition (such as a maximum number of iterations or a specified amount of CPU time) has been satisfied. From this fundamental VNS concept, several other variations of VNS have been established (for recent surveys, see for example (Hansen and Mladenovic, (2002); Hansen et al. (2010)), and they have been effectively used to the solution of a wide variety of optimization problems (see e.g., (Brimberg et al. (2015); Hansen et al. (2010); Lazic et al. (2015); Mladenovic et el. (2016)). The so-called General VNS is a variation of VNS that is well-known and often employed (GVNS). It is developed from the fundamental VNS strategy by exchanging a straightforward local search with an advanced improvement process that investigates different structures located in the neighborhood. Variable neighborhood descent (VND) is the foundation for the majority of the optimization strategies that are implemented inside GVNS. This includes sequential VND, nested VND, and cyclic VND, amongst others. Although the Nested VND algorithm offers a powerful local search procedure, it is not considered efficient in terms of solution time. Therefore, sequential VND approach is adopted in the thesis.

Sequential variable neighborhood descent

Within the context of deterministic local search, the sequential method is the typical approach for the investigation of many neighborhoods (Seq-VND). The following problem-specific questions naturally occur throughout the process of building an effective VND heuristic: Which neighborhoods will be utilized? What will be their search sequence? What search approach will be employed while switching neighborhoods? Thus, we can write basic steps of the Seq-VND on the following:

1. Sort all of the neighborhoods that will be utilized in the search.

2. Conduct local searches with them while keeping their order in consideration.

3. When a better option is discovered, proceed to the first neighborhood on the ranking.

4. Stop since the last neighborhood has no better alternatives.

The best answer that was found is a "local minimum" with regard to all of the structures in the neighborhood. In addition, it is abundantly evident that the cardinality of the Seq VND neighborhood is equivalent to the total of the cardinalities of all of its constituents, hence the cardinality of the neighborhood itself is equivalent to the sum.

4.6. General Variable Neighborhood Search Algorithm Implementation to Hub Location Problems

Variable neighborhood search (VNS) is a metaheuristic proposed by Mladenovic´ and Hansen (1997). Local search proceeds from an initial solution and using a sequence of local changes, improves the value of the objective function until a local optimum is found. VNS systematically exploits the idea of change of neighborhood during the search in a dynamic way. To construct different neighborhood structures and to perform a systematic search, one needs to supply the solution space with some (Quasi)-metrics and then induce neighborhoods from them. VNS algorithm neighboring structure same as SA.

4.6.1. Initialization

The algorithm starts with a randomly generated initial set of hubs and then allocates all non-hub nodes to their closest hub. Then a cluster as the set of nodes allocated to the same hub is found. This USAIPHMP naturally decomposes into three subproblems: the hub location problem, the allocation problem and hub connection problem. The VNS exploits these three parts through use of proposed nine different neighborhood transitions. As previously stated, determining optimum allocations is an NP-hard problem although when hub locations are known. If the locations of hubs and node-to-hub allocations are known, the optimum pathways between each pair of nodes are identified as the least costly potential routes.

4.6.2. Cost Calculation

Assume that the sets of hub nodes, hub links and spoke links are represented by H, arc1 and arc2, respectively. The variable cost per unit flow C_{km} between each pair of hubs (k,m) is obtained by Dijkstra algorithm with computational complexity of O(q + plog(p)), where p, q are the numbers of hubs and hub links, respectively. The current sets of hubs that are connected to spoke node i and j by collection links and distribution links are represented by C_{ik} and C_{mj}. Based on the variable cost between hub nodes, the variable costs between hub node k-m and spoke node i-j are computed as follows:

$$TC = C_{ik} + \min\{C_{km}\} + C_{mj} \quad (4.3)$$

There are two different hub network structures in Figure 2. In the first way, H1 and H4 hubs are connected to each other, while H1 and H4 are not connected in the second way. In the first network structure, the flow between the i and j nodes must be provided between H1 and H4. Because the closest connection point is provided between these two hubs. In the second way, since there is no connection between H1 and H4, the shortest distance is found with Dijkstra algorithm. In this context, out of routes such as H1-H3-H4 or H1-H2-H4, there is one that has the least cost. This increases the difficulty level of the problem.

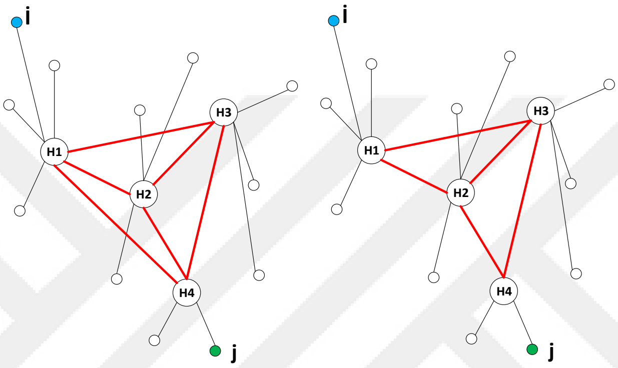


Figure 8: Comparison of Complete and Incomplete Networks Total Cost Calculation

4.6.3. Searching

Nine neighborhoods search (samehubnode, closestnodeinsert, samehubnodecl, nodeinsert, samehubnodebtw, samehubnodeflow, differenhubnode, samehublink_swap, differenthublink_swap) are examined in the Sequential VNS.The usual strategy for exploration of several neighborhoods within deterministic local search is sequential :

- Make an order of all ‘max neighborhoods that will be used in the search (usually in non-decreasing order of their cardinality).
- Perform local searches with them respecting their order.
- The first time a better solution is found, return to the first neighborhood in the list.
- Stop when there is no better solution in the last neighborhood.

4.6.4. Modifying access arcs and hub-node conversions

There are different types of operators used in arranging node-to-hub or hub-to-node connections and hub-node conversions. The samehubnode operator changes a node connected to a randomly selected hub location to be a hub. The current hub location becomes a node assigned to the new hub location. In Figure 4.13, the random H1 hub region is selected and node A connected to H1 is changed to hub. H1 is allocated to the hub A.

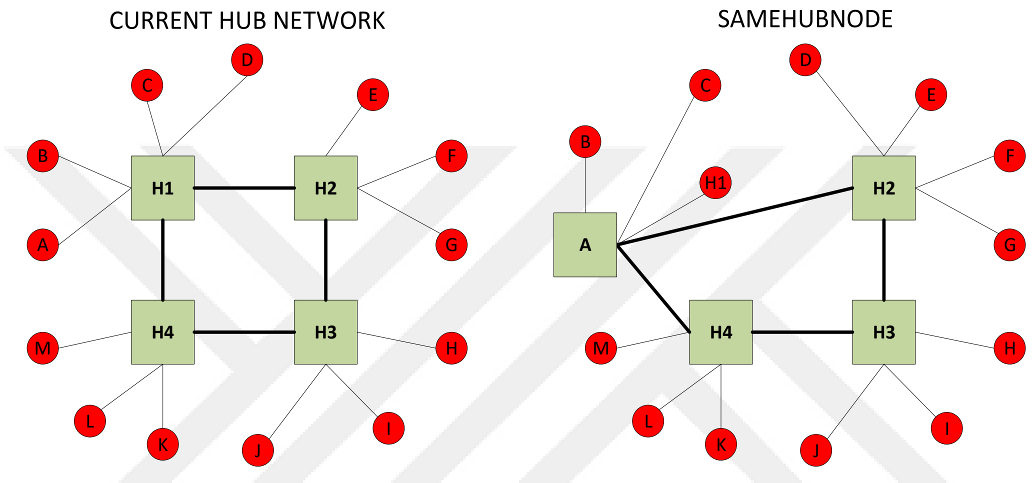


Figure 4.13. Samehubnode operator

The closestnodeinsert operator works according to the rule of allocating any node to the second closest hub location. In Figure 4.14, while node M is connected to the H4 hub, it is assigned to the second closest hub A. Thus, by assigning a random node to a hub location that is far from itself, an unnecessary calculation process that is far from the optimal solution is not performed. However, the operation performed with the closestnodeinsert operator is likely to improve the objective function.

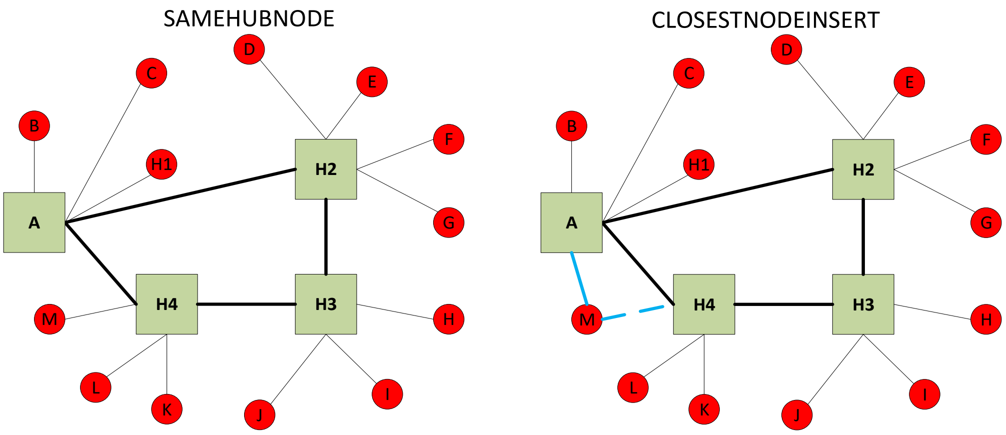


Figure 4.14. Closestnodeinsert operator

The samehubnodecl operator looks for neighborhoods according to the logic of assigning the node with the highest closeness_centrality * flow value among any selected hub location and its allocated nodes. If the node with the highest value is the current hub location, the second-tier node is determined as the hub. In Figure 4.15, it was chosen as the hub because node E has the highest closeness_centrality * flow value among nodes D, E, F, G and H2.

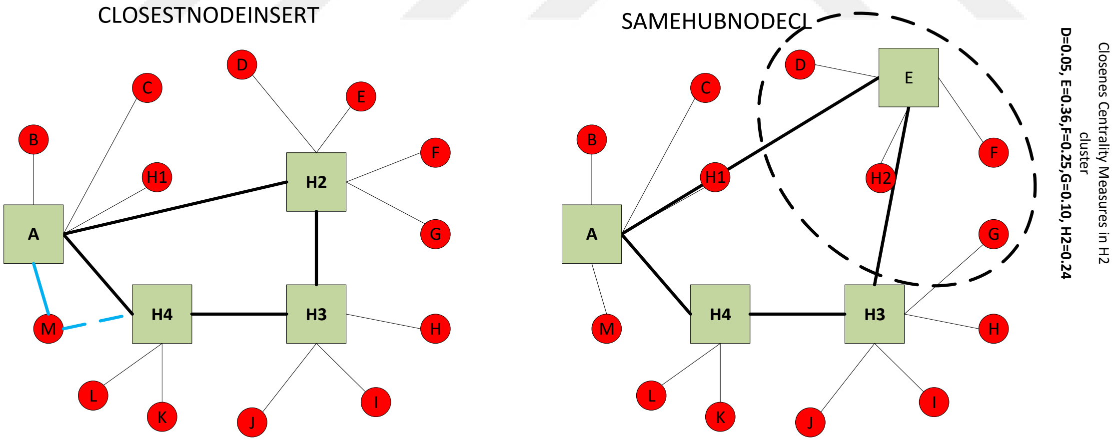


Figure 4.15. Samehubenodecl operator

Unlike the closestnodeinsert operator, the nodeinsert operator works by assigning a randomly selected node to a randomly selected hub location. In Figure 4.16, node C is assigned to E hub instead of A hub.

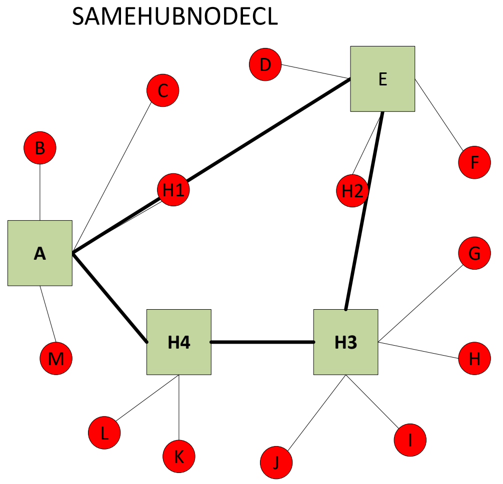


Figure 4.16. Nodeinsert operator

The samehubnodebtw operator looks for neighborhoods according to the logic of assigning the node with the highest betweenness_centrality * flow value among any selected hub location and its allocated nodes. If the node with the highest value is the current hub location, the second-tier node is determined as the hub. In Figure 4.17, it was chosen as the hub because node I has the highest betweenness_centrality * flow value among nodes G,H,I,J and H3.

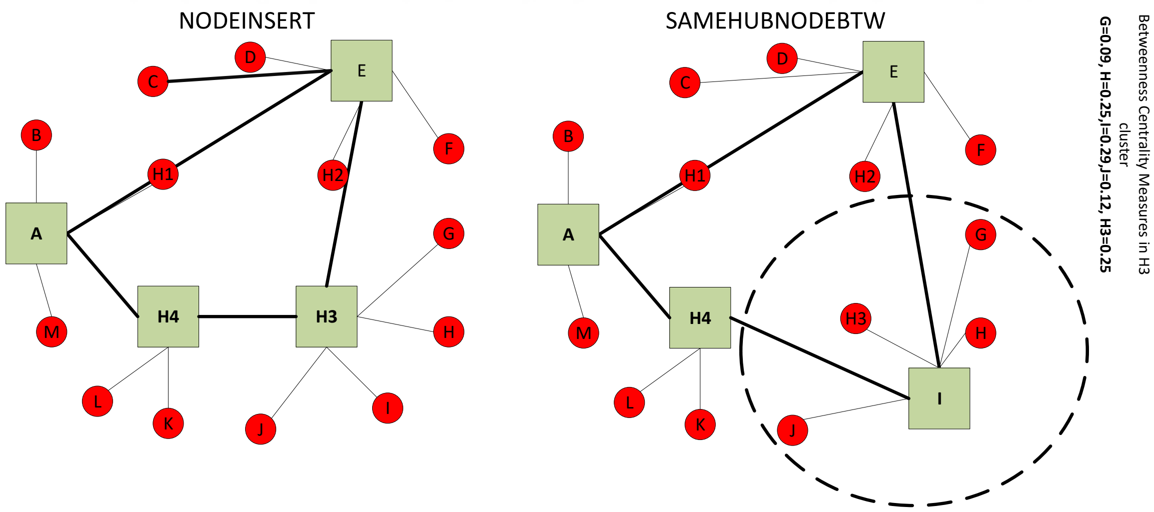


Figure 4.17. Samehubnodebtw operator

Unlike the integrated neighborhood search structures betweenness and closeness, the samehubnodeflow operator considers only the sum of demand and supply values of a node. Among the selected hub and its allocated nodes, the node with the highest flow value is assigned as the hub. In Figure 4.18, the flow on node B has a value of 450 and since this value is the highest, node B became the hub. In addition, with node B becoming a hub, node C closer to it has been allocated to the B hub.

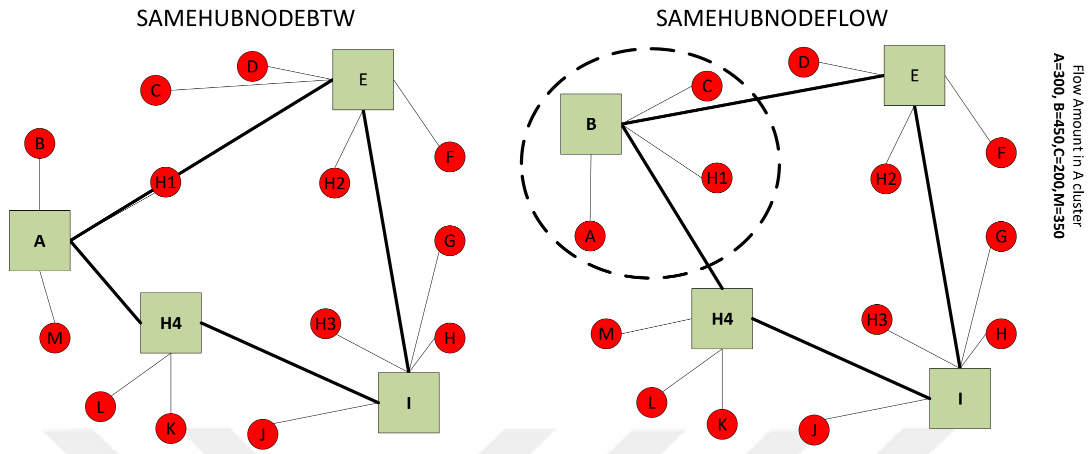


Figure 4.18. Samehubnodeflow operator

The node to be assigned as the new hub on the differenthubnode operator is not connected to the current hub node. Thus, a diversification of solution is provided. In Figure 4.19, H1 connected to node B becomes hub again, but instead of hub location B to which it is connected, E hub becomes node and allocated to H1 hub.

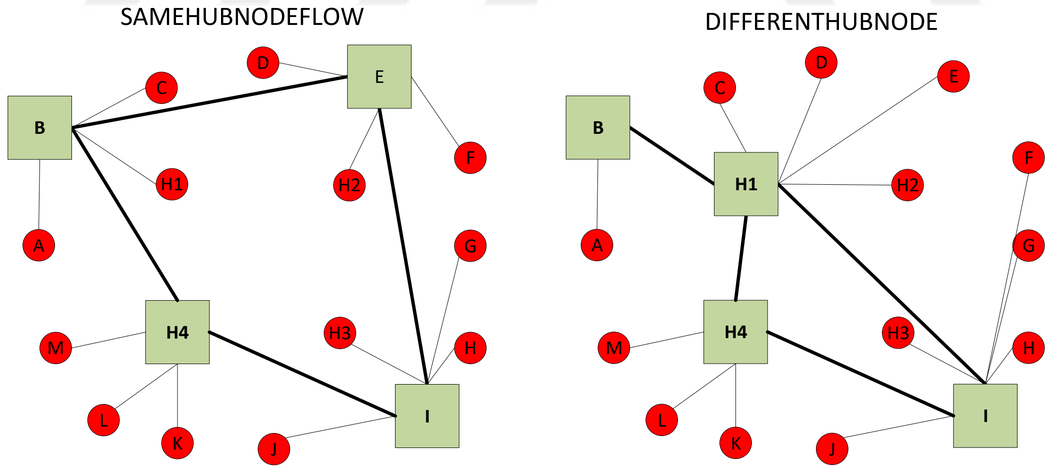


Figure 4.19. Differenhubnode operator

4.6.5. Modifying Hub Arcs

In order for an arc to be defined as a hub link, both ports must have a hub node. The samehublink_swap operator is based on changing only one of these two hub nodes. The selected arc gets connected to a hub point in the current state, while another hub region is changed. While H1 and H4 are connected in Figure 4.20, H4 has been connected to the B hub.

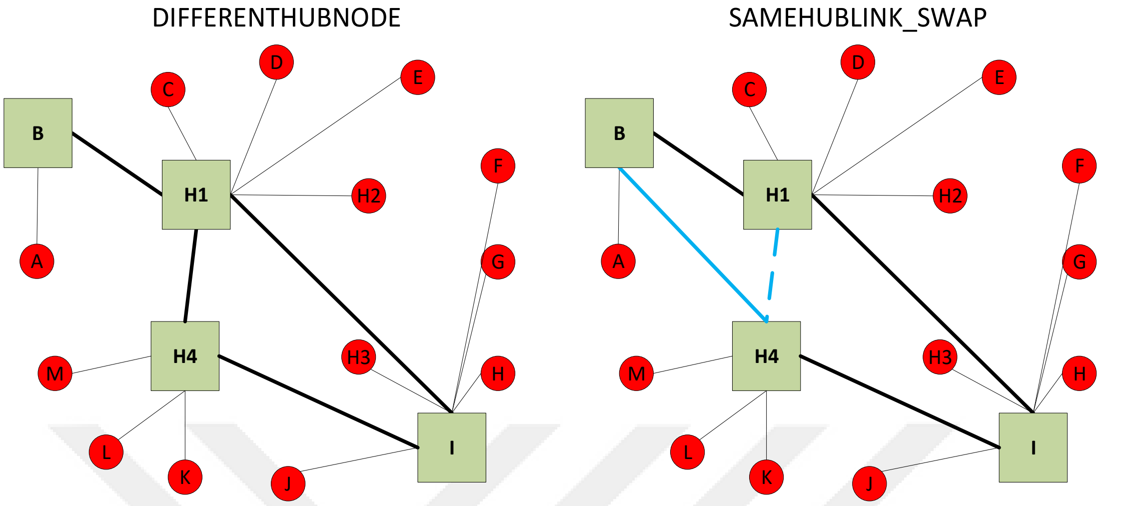


Figure 4.20. Samehublink_swap operator

The differenthublink_swap operator removes a link that it chooses and instead establishes an arc connection between a different pair of hubs independent of the connected hubs. In Figure 4.21, instead of H1 and H4, the B and I hub nodes are connected to each other.

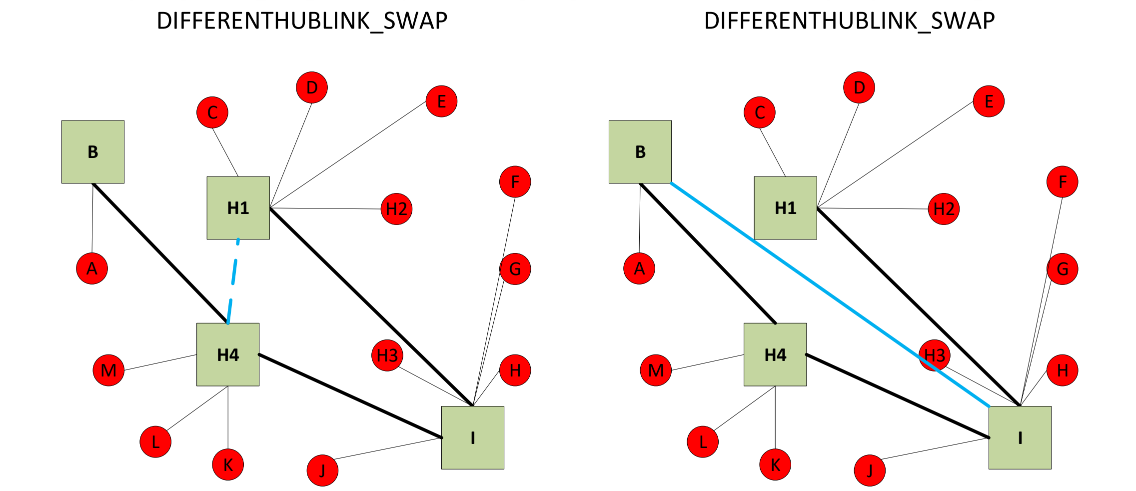


Figure 4.21. Differenthublink_swap operator

4.6.6. Analysis of Neighborhood Search Operators

In this subsection, we examine the sequence in which local searches are performed in sequential VND. The seven different neighborhoods are used in the examination of sequential GVNS techniques. In this analysis, the decrease effect of the proposed neighborhood structures on the objective function value is observed based on iteration. However, neighborhood structures related with connections between hubs are not taken into account. Because connection exchanges between hubs are used in in-complete hub location problems and they are usually used in the last orders in local search procedure. In this analysis, in which seven different neighborhood structures are considered, the main purpose is to determine the order of neighborhood structures in Seq-VND. The neighborhood search structure, which provides the most decrease effect on the objective function in 100 iterations, starts from the first order and continues in a decreasing manner. Different versions of the CAB, AP, and TR datasets are tested to ensure accurate results. The first of these is the economies of scale coefficient. Analyzes are performed using the 0.2, 0.4, 0.6, 0.8 and 1 values of the economies of scale coefficients. In addition, examples for small, medium, and large data sets are discussed. Complete and incomplete problems are also evaluated and presented separately. For the analysis to be fair, each neighborhood structure is run for 100 iterations over the same initial solution and the changes on the objective function value are observed.

In general, the proposed novel samehubnodecl neighborhood structure seems to have the greatest convergence effect on the optimal outcome. In addition, the standard deviation value is very low, as it offers an exchange over the nodes connected to the same hub. Another efficient neighborhood search operator is the closestnodeinsert. By default, nodes are assigned to the nearest hubs in the initial solution. But this is not always optimal, especially for incomplete p-hub median problems. The relevant neighborhood operator is assigned to the nearest secondary hub and proposes a new solution. The nodeinsert neighborhood operator assigns a node assigned to a randomly selected hub to another randomly selected node. There is no specific rule in this assignment. However, as a result of the analysis, it can be seen that this operator gives good results. The remarkable point for this operator is that the convergence process to the optimal occurs in smaller steps, not instantaneously. However, the standard deviation value is a little high. This is because hub assignments are random, in some cases they are assigned to remote hubs. Another remarkable point is that the standard deviation value of the differenthubnode operator is very high. The hub location of any chosen node has a significant impact on the objective function. However, this search operator usually extracts very different cost values. Considering the experimental results, it has been observed that the most appropriate neighborhood search operator order for seq-GVNS is samehubnodecl, closestnodeinsert, nodeinsert, samehubnode, sahubnodeflow, samehubnodebtw, difhubnode, samehublink_swap and differenthublink_swap and this order has been integrated into the algorithm. The outputs of the detailed analyzes on the performance of neighborhood operators for different data sets are presented in the following Figures.

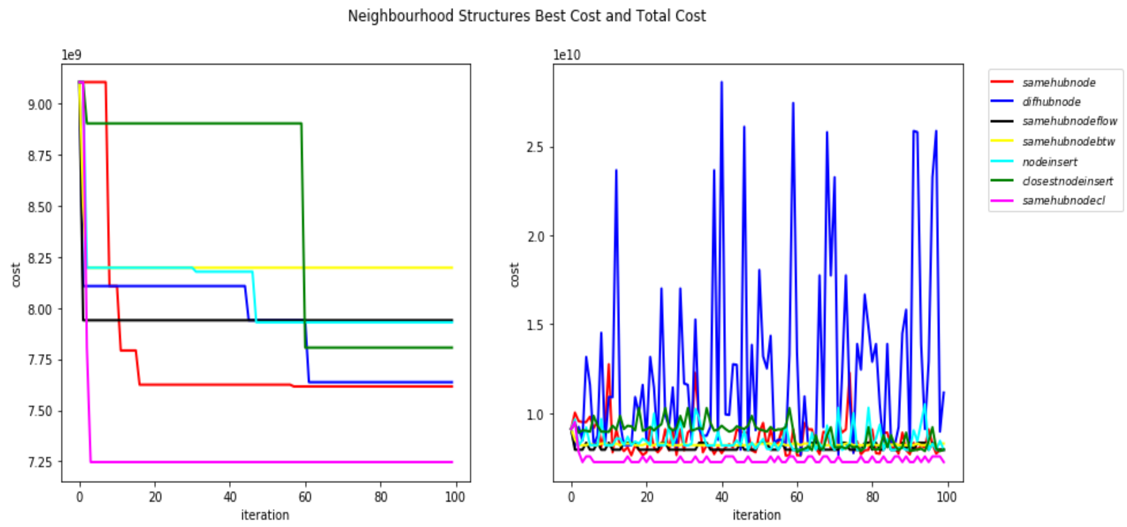


Figure 4.22. Neighbourhood operators for CAB25 *p*=5 *q=4 alpha*=0.2

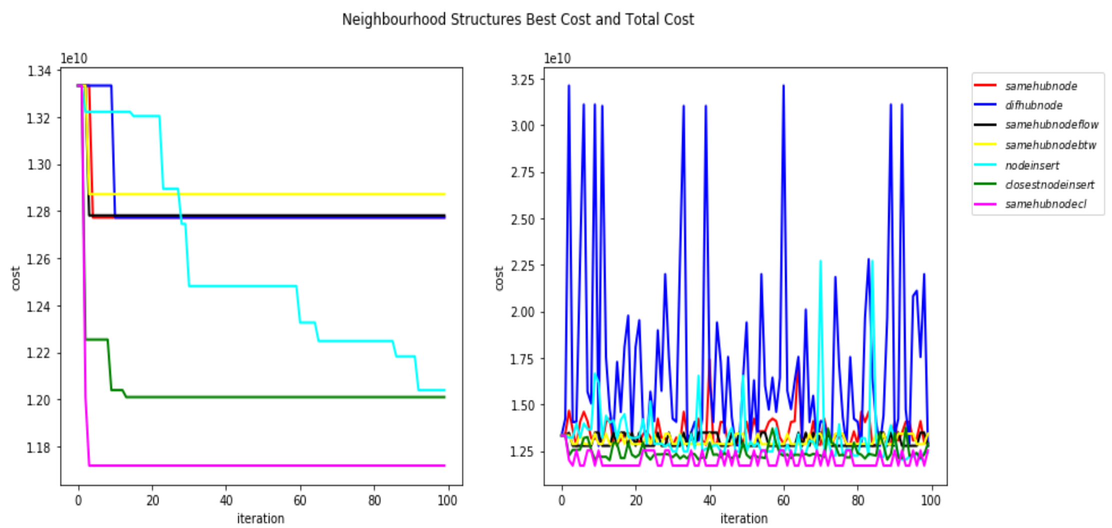


Figure 4.23. Neighbourhood operators for CAB25 *p*=5 *q=4 alpha*=0.8

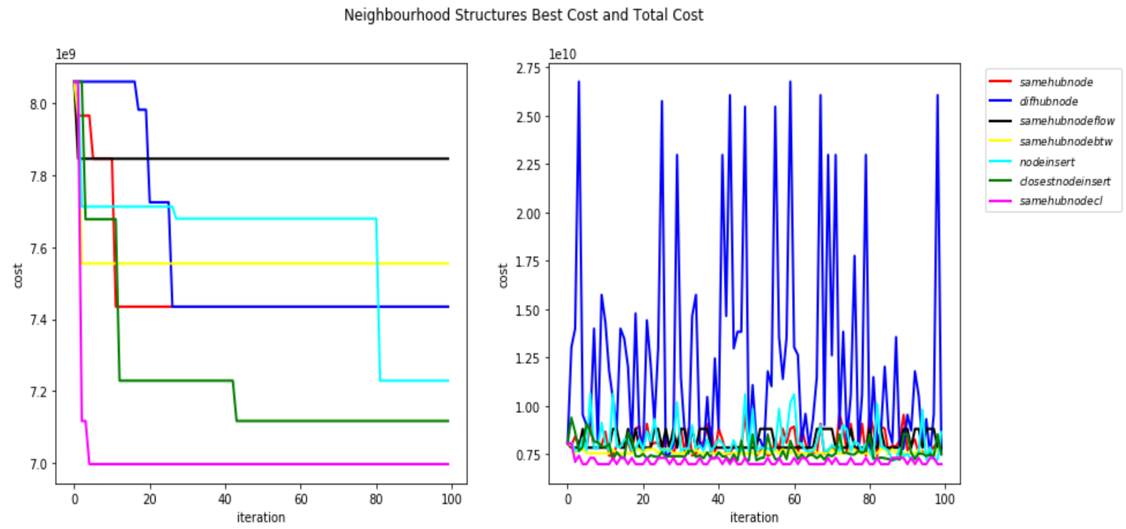


Figure 4.24. Neighbourhood operators for CAB25 *p*=5 *q=8 alpha*=0.2

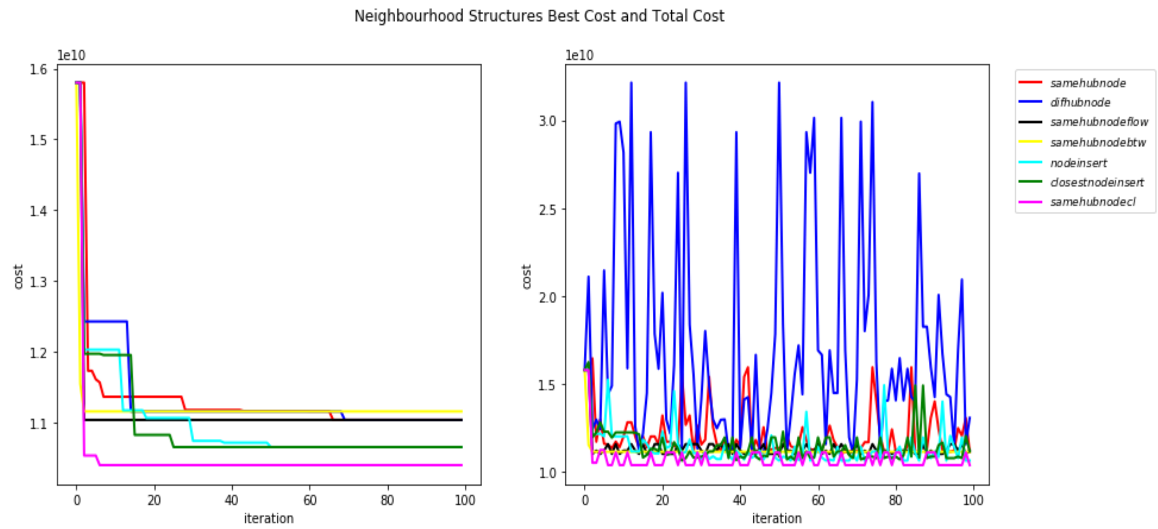


Figure 4.25. Neighbourhood operators for CAB25 *p*=5 *q=8 alpha*=0.8

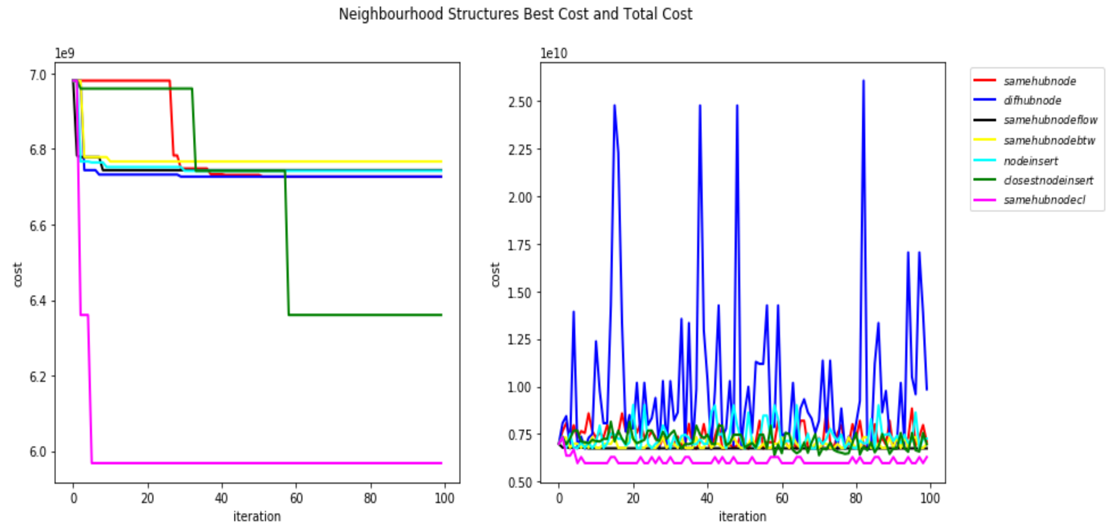


Figure 4.26. Neighbourhood operators for AP50 *p*=5 *q=8 alpha*=0.2

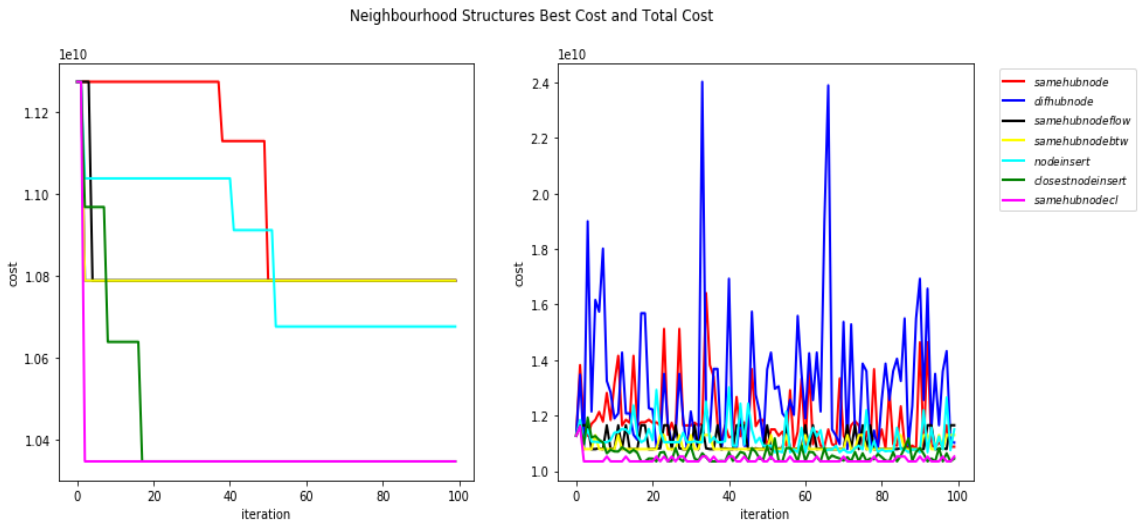


Figure 4.27. Neighbourhood operators for AP50 *p*=5 *q=8 alpha*=0.2

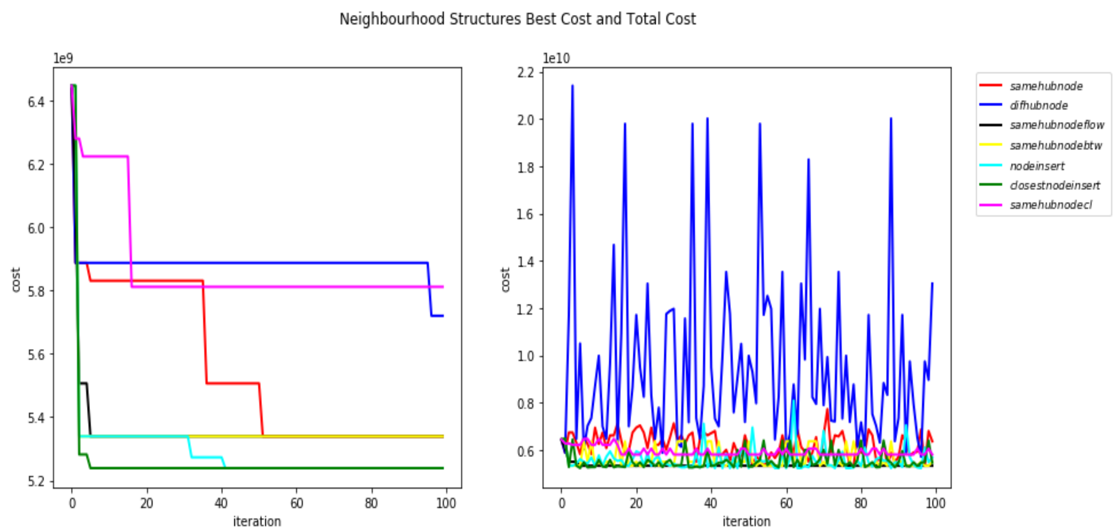


Figure 4.28. Neighbourhood operators for AP50 *p*=5 *q*=*8 alpha*=0.75

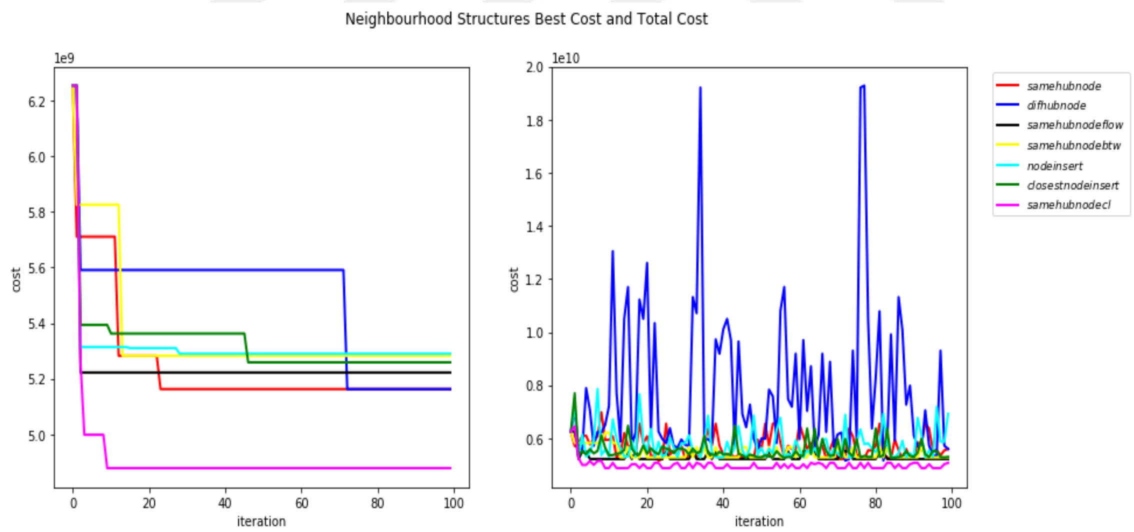


Figure 4.29. Neighbourhood operators for AP100 *p*=5 *q*=*4 alpha*=0.75

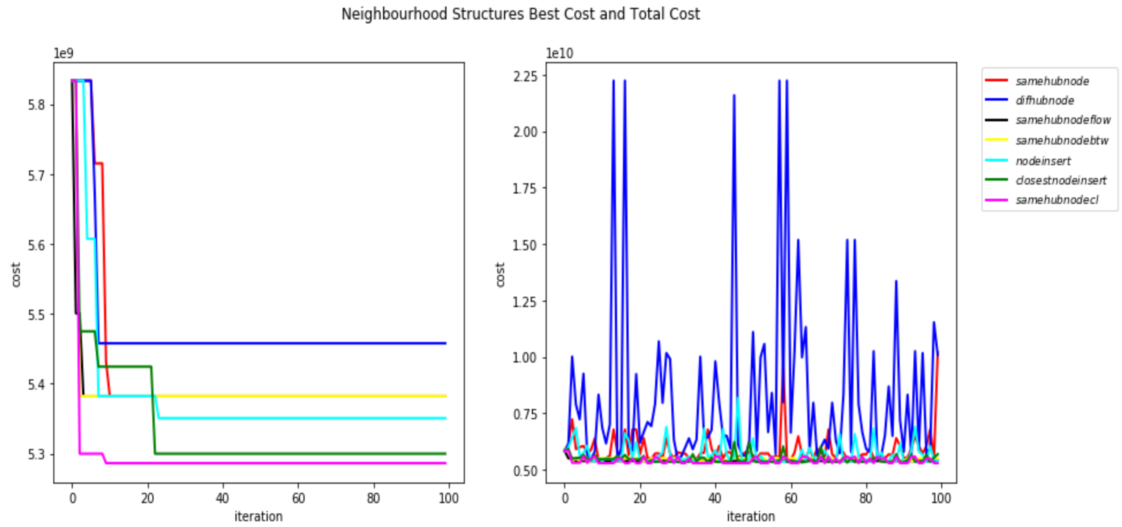


Figure 4.30. Neighbourhood operators for AP100 *p*=10 *q=20 alpha*=0.75

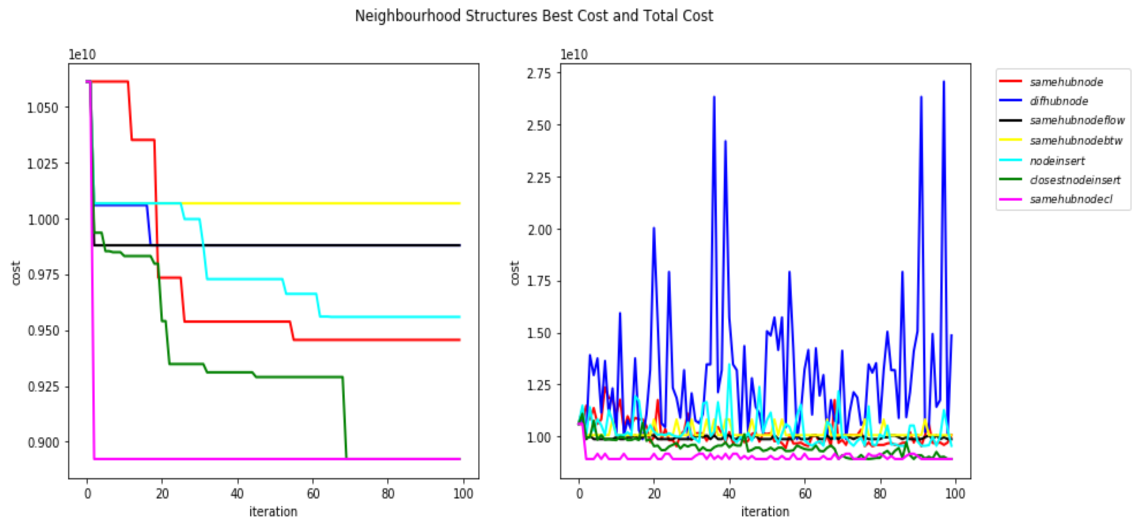


Figure 4.31. Neighbourhood operators for AP100 *p*=10 *q=45 alpha*=0.75

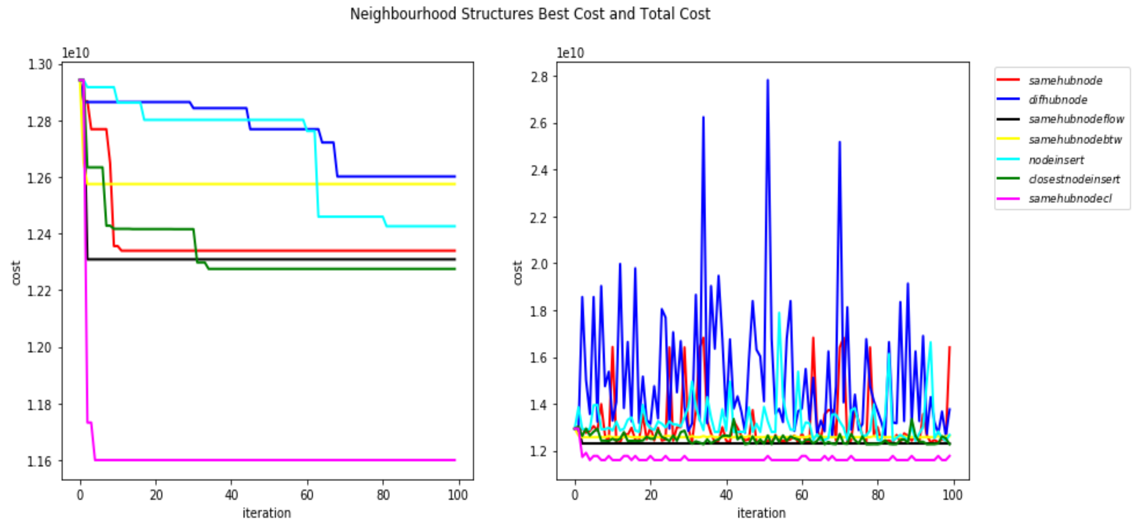


Figure 4.32. Neighbourhood operators for AP200 *p*=20 *q=100 alpha*=0.75

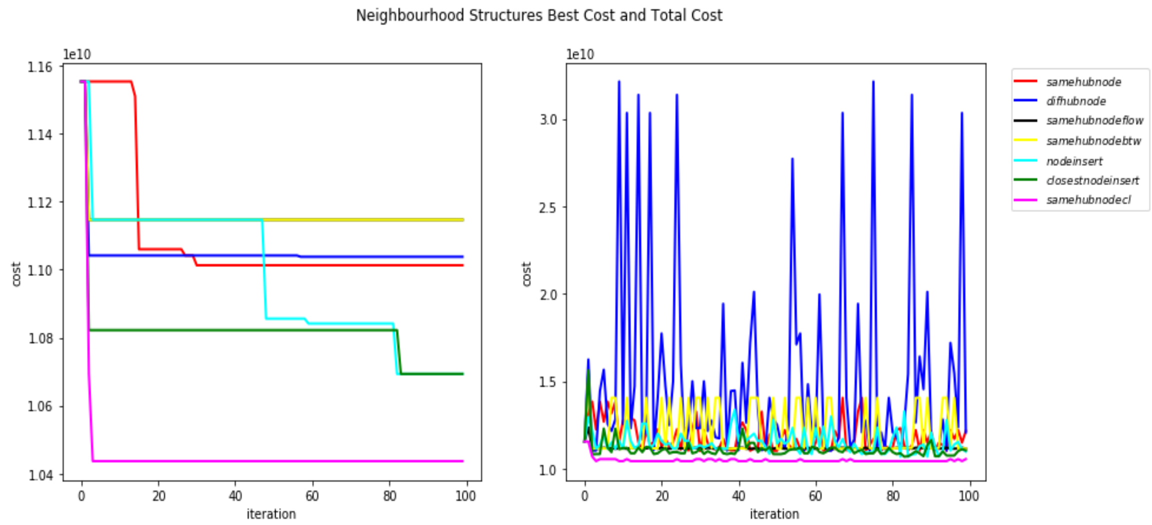


Figure 4.33. Neighbourhood operators for AP200 *p*=20 *q=150 alpha*=0.75

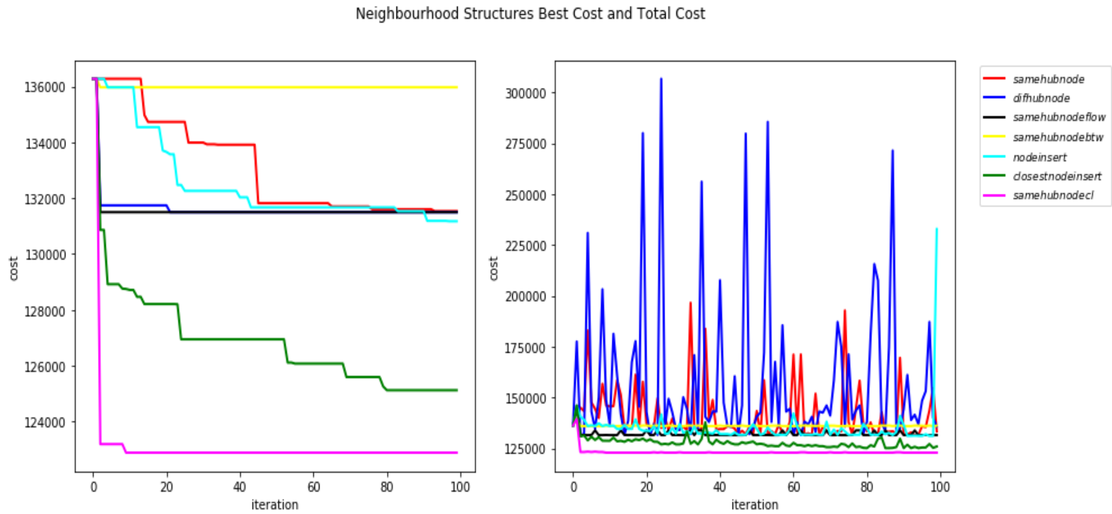


Figure 4.34. Neighbourhood operators for TR *p*=5 *q=5 alpha*=0.8

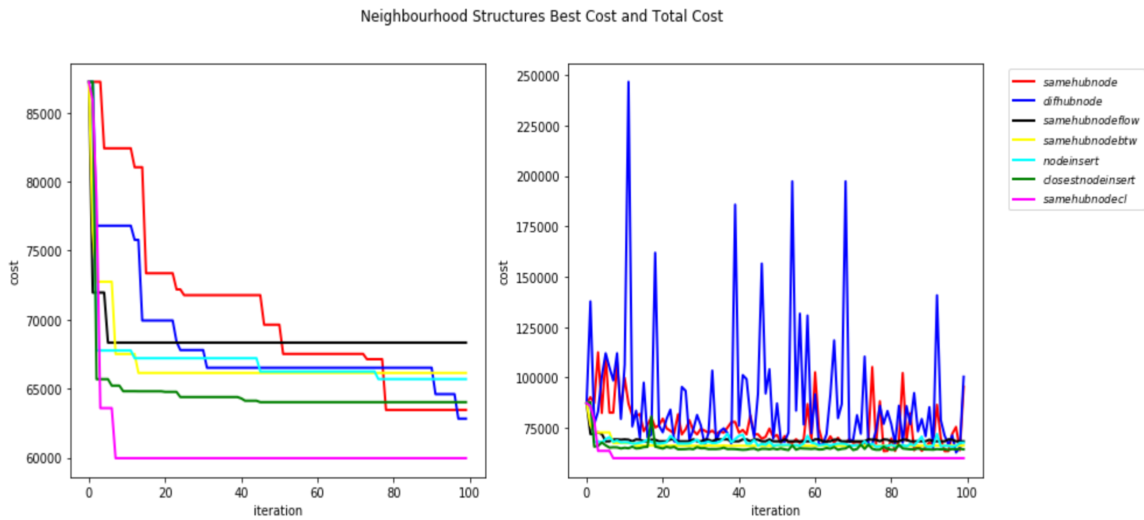


Figure 4.35. Neighbourhood operators for TR *p*=8 *q=10 alpha*=0.8

4.6.7. Shaking

In the Shaking step we generate a random assignment from the kth neighborhood of the current solution. The new assignment is constructed as follows. Choose at random k different hubs $h_1; h_2; ... ; h_k$ and k different non-hub locations $l_1; l_2; ... ; l_k$ and interchange hi with l_i for each $i; 1 <= i <= k$. This means that l_i becomes a hub in the cluster where h_i previously was, and the location hi becomes a non-hub in the cluster where l_i was. Figure 4.36 represent the shaking procedure in detailed.

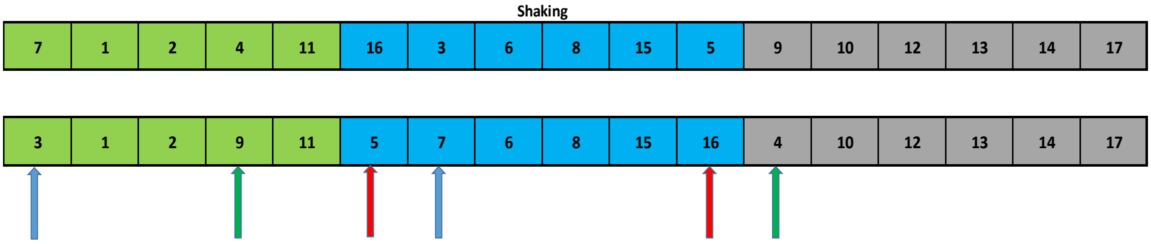


Figure 4.36. Shaking procedure for GVNS

4.7. Contributions of the GVNS Algorithm

The following structural algorithmic aspects are included in the proposed GVNS. These elements have been explicitly specified for the methodologies adopted. Each component is either an entirely novel strategy or an updated version of an existing strategy that has been implemented.

The proposed GVNS algorithm for the hub location problems differs from the existing approaches in the literature in several aspects. The most important of these are neighborhood search operators. Among the neighborhood operators, operators that take into account the closeness and betweenness centrality metrics are suggested in this thesis for the first time in the literature. The centrality properties are very useful when evaluated together with the flow values between the o-d pairs. In the local search process, it ensures that the hub locations within the optimal set are quickly obtained. In addition, more than one alternative approach to the change of hub locations also minimizes the computational complexity brought by randomness. Besides, in incomplete hub location problems, the connections between hub locations need to be updated every iteration. The two neighborhood operators proposed for this are very important to obtain the optimal result and are used for the first time based on the model discussed in the literature. In this context, 6 of the 9 proposed neighborhood structures were used for the first time for the local search process and efficient results were obtained. The pseudo code of the Sequential GVNS is given below.

Algorithm: Sequential GVNS

```
Identify it_max and loc_max: Generate the initial solution
Best Solution=Initial Solution
loc=1
for i=1 to it_max do:
    j=1
    while j<9 do:
        if j=1 do:
            apply neighborhood operator samehubnodecl () on current network;
        else if j=2 do:
            apply neighborhood operator closestnodeinsert () on current network;
        else if j=3 do:
            apply neighborhood operator nodeinsert () on current network;
        else if j=4 do:
            apply neighborhood operator samehubnode () on current network;
        else if j=5 do:
            apply neighborhood operator samehubnodeflow () on current network;
        else if j=6 do:
            apply neighborhood operator samehubnodebtw () on current network;
        else if j=7 do:
            apply neighborhood operator difhubnode () on current network;
        else if j=8 do:
            apply neighborhood operator samehublink_swap () on current network;
        else if j=9 do:
            apply neighborhood operator differenthublink_swap () on current network;
        j=j+1
        end
        if there is a cost reduction with new network:
            best solution = new solution, current network = new network and j=1
            loc=1
        else:
            loc= loc+1
        end
        if loc<locmax do:
            apply shaking () on current network;
        else:
            current network = new network and j=1
    end
    loc_max=1
end
```

4.8. Reduced General Variable Neighborhood Search (R-GVNS)

GVNS's performance has been greatly improved, making it one of the most competitive algorithms for tackling a broad variety of combinatorial optimization problems. Although VNS cannot guarantee global optimum owing to its heuristic nature, it is typically capable of finding a high-quality solution to a given problem within a restricted computational expense. In this thesis, based on the GVNS algorithm, we present the R-GVNS method in order to eliminate the main obstacles that existing metaheuristic methods face in solving complex problems. As shown in Figure 1, the first phase of our R-GVNS method includes certain problem characteristics specified on a subset of optimally solved small p-hub median problem instances with known optimum location and allocation structure. We extract problem features as well as statistical indicators from graphs of solved p-hub median problem cases and mapping each node to a candidate solution in the feature space. These attributes will then be utilized to forecast the 'likelihood' that a node in the relevant graph corresponds to the optimal solution for an unsolved test p-hub median problem instance, as illustrated in Figure 4.37. After the hub candidate locations are identified based on various features, the feature extraction procedure of the hub connections is applied for incomplete hub problems. In a complete hub network, edge classification is made with the indicators used according to the number of connections between hubs. The most suitable edges are defined as hub connections. However, the point to be noted here is that the defined edge connections are connected to all hub nodes. In the example presented in the Figure, all nodes on the network are evaluated on the basis of the determined graph and statistical metrics, and it is determined that the nodes highlighted in red are likely to be hubs. All hubs are directly connected by assigning the nearest non-hub nodes to the determined hubs. In the second stage, an evaluation is made based on edge features between hubs. This evaluation is based on graphs and statistics, as in the previous stage. The metrics considered in these evaluations are presented in the next section. Then, hub connections are established according to the defined hub connection number and a hub network is obtained.

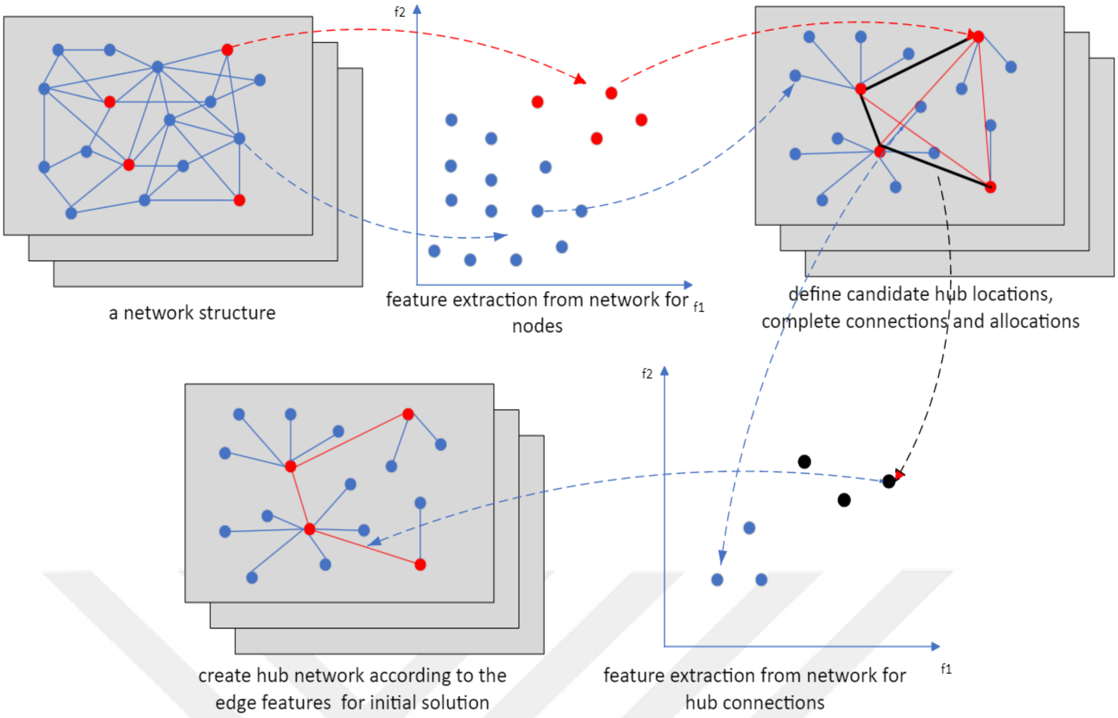


Figure 4.37. R-GVNS framework representation

Researchers are frequently faced in the big data era with optimization problems involving huge amounts of decision variables, such as sophisticated global supply chain network analysis. The large problem size presents many challenges to existing solution algorithms, particularly commercial Mixed Integer Programming (MIP) solvers like CPLEX, which typically try to cope to solve or even find good solutions for such large-scale optimization problems in an acceptable computing effort. Furthermore, in many real-world applications, such as drone location planning, we must give customers with a high-quality service level in a fraction of a second. This is difficult to do, particularly when the problem size is high, requiring the deployment of an effective problem reduction approach that may drastically restrict the search space while still capturing an optimum (or near-optimal) solution in the lower dimensional space.

This thesis represents a considerable contribution by providing a systematic investigation of the resilience of our problem reduction approach when such nontrivial changes occur in test datasets. We demonstrate experimentally that our problem reduction approach generalizes pretty well to a diverse set of location models with varying characteristics or dimensions (as p-hub center, hub covering etc.). We also highlight areas where our problem reduction approach may underperform, namely on test instances that are purposefully designed to be considerably different in terms of problem characteristics from small size and non-complex networks. The main purpose is to provide quality initial solutions to the VNS algorithm to be run, and to ensure that it converges to the optimal in a shorter time. We show that quality solutions can be obtained with less computational cost with the determined decision variables, and we think that this approach will serve as a guide for the application of all combinatorial optimization problems.

In addition, in this thesis, we show that the problem reduction technique applied for complete samples is also applicable for incomplete hub networks. Our experimental results show that the incomplete solution set can also be defined through the properties of the nodes in the optimal solution. Since hub connections are also important in incomplete problems, edge features between hub pairs are also needed. In this context, we also define flow and distance-based edge feature for each hub pair for incomplete problems.

There are a lot of combinatorial optimization problems, and most of them include a lot of decision variables, most of which are "irrelevant" to the best possible solution. The purpose of problem reduction is to separate out most of the variables that are not even important to the problem at hand and take them out of the equation altogether, with the expectation that the subsequent problem will be faster and easier to solve. However, determining which of these factors are not important is not a simple process in and of itself. The majority of the problem reduction methods that are currently available in optimization algorithms are exact approaches. These methodologies only remove decision variables that cannot be part of an optimal solution, and they do so on the basic principle of mathematical knowledge and/or the calculations of an objective limits. An exact method ensures that the smaller problem will always have a unique solution that is optimum, but it is sometimes computationally costly and does not come with tools that can dramatically reduce on the problem's overall dimension.

We use instances with known optimal solutions (as CAB25) to analyze and define problem-specific features. In this way, nodes are rated on the basis of the defined features to distinguish the decision variables included in the optimal solution from those that are not (variables that are less likely to be in the optimal solution set). To describe each decision variable, we first extract computationally efficient problem attributes, and then we calculate statistical measures based on random sampling of solutions that are feasible. On the basis of these features, we make a prediction for each decision variable on the probability that it is part of an optimum solution. Our problem reduction approach may also be used as a preprocessing strategy, allowing for the elimination of decision variables from a previously unknown problem instance that are not anticipated to be a component of a solution that is optimum. In this context, the following algorithm shows the general steps of the methodology.

Algorithm: Problem Reduction Steps

Step 1: Solve optimally small or medium size p-hub median problems.

Step 2: Analyze optimal solutions and define some features of nodes in optimal set.

Step 3: Obtain a function according to the node properties in the optimal set.

Step 4: Prioritize the nodes based on the features of the nodes in the optimal node set.

Step 5: Hub-to-hub edge weighting is achieved through the determined prioritized nodes for the purpose of minimum transportation cost.

Step 6: For the solution of larger and complex problems, define the sets obtained by prioritization as candidate sets and integrate them into meta-heuristic algorithms.

4.8.1. Identifying Features for Each Node

Graph evaluation has been made in terms of several features (flow and centrality) as centrality-based solution approaches in previous section. However, we aim to create new candidate hub sets based on different features in order to make the separation in the clusters stronger. To classify each node, we first directly calculate 13 features from the graphs, and then we use two quantitative measurements that are derived from probabilistically generated samples of solutions that are possible.

4.8.2. Node Features from Graph

Since the aim of the hub location problem is to establish a network with minimum cost and considering that the main factors affecting this cost are flow and distance, the objective values are evaluated on these two basic factors. Consider that there are n nodes in a network defined as G(V, E). We define each node on this network based on the following 13 features.

$$P_1(n_i) = \sum_{i=1}^{n} f_{(i,j)} + \sum_{i=1}^{n} f_{(j,i)} \tag{4.4}$$

$$P_2(n_i) = \sum_{i=1}^{n} d_{(i,j)}(f_{(i,j)} + f_{(j,i)}) \tag{4.5}$$

$$P_3(n_i) = \frac{\sum_{i=1}^{n} f_{(i,j)} + \sum_{i=1}^{n} f_{(j,i)}}{\sum_{i=1}^{n} f_{(k,j)} + \sum_{i=1}^{n} f_{(j,k)}} \quad j,k = 1,2,\dots,n \tag{4.6}$$

$$P_4(n_i)=C_i \tag{4.7}$$

$$P_5(n_i)=B_i \tag{4.8}$$

$$P_6(n_i) = \left(\sum_{i=1}^{n} f_{(i,j)} + \sum_{i=1}^{n} f_{(j,i)}\right) C_i \tag{4.9}$$

$$P_7(n_i) = \left(\sum_{i=1}^{n} f_{(i,j)} + \sum_{i=1}^{n} f_{(j,i)}\right) B_i \tag{4.10}$$

$$P_8(n_i) = \left(\sum_{i=1}^{n} f_{(i,j)} + \sum_{i=1}^{n} f_{(j,i)}\right) C_i B_i \tag{4.11}$$

$$P_9(n_i) = \frac{maxf_{(i,j)} - minf_{(i,j)}}{maxf_{(j,k)} - minf_{(j,k)}} \quad j,k = 1,2,\dots,n \tag{4.12}$$

$$P_{10}(n_i) = \frac{maxd_{(i,j)} - mind_{(i,j)}}{maxd_{(j,k)} - mind_{(j,k)}} \quad j,k = 1,2,\dots,n \tag{4.13}$$

$$P_{11}(n_i) = \frac{maxf_{(i,j)} - \sum_{k=1}^{n} \frac{f_{(i,j)}}{n}}{maxf_{(j,k)} - minf_{(j,k)}} \quad j = 1,2,\dots,n \tag{4.14}$$

$$P_{12}(n_i) = \frac{maxd_{(i,j)} - \sum_{k=1}^{n} \frac{d_{(i,j)}}{n}}{maxd_{(j,k)} - mind_{(j,k)}} \quad j = 1,2,\dots,n \tag{4.15}$$

$$P_{13}(n_i) = \frac{1 median\ objective\ value\ of\ n_i}{\sum_{i=1}^{N} 1 median\ objective\ value\ of\ n_i} \tag{4.16}$$

The first feature shows the total flow entering and leaving the corresponding node i. Considering that nodes with high flow frequencies are usually included in the optimal hub set, we think that this feature is important for the candidate hub set. Another important parameter, the distance criterion, is also important in hub locations. Because the distance criterion is directly related to the transportation costs. Therefore, the second feature consists of the total flow multiplied by the distance. It is based on the total flow entering the node i, the total flow leaving the node i and the distance of the node i from the related nodes. The third feature is defined by the ratio of the total flow entering and leaving the i node to the total flow in the entire network. The fourth and fifth features are the indicators that give the closeness centrality and betweenness centrality values of the i node, respectively. The sixth property is based on the product of the closeness centrality value of the total flow of the node i. The same process is repeated with the betweenness centrality in the seventh feature. In the eighth feature, both betweenness centrality and closeness centrality values are considered. The closeness and betweenness centrality values of high-flow nodes are also large, increasing the probability of being included in the optimal hub set. The ninth property is found by dividing the difference between the maximum and minimum flows of the node i by the maximum and minimum flow over the entire network. In the tenth feature, the same operation is considered on the basis of distances. P_{11} is obtained by dividing the difference between the maximum flow and the average flow of the node i by the difference between the maximum and minimum flow in the network. P_{12} repeats the same operation based on distances. The last graph feature, P_{13}, shows the 1-median objective function of the node i. In addition, since the solution of the 1-median problem is quite easy, it does not pose a problem in terms of computational difficulty.

4.8.3. Statistical Metrics for Nodes

Statistical metrics aim to measure the probability that each node belongs to the optimal hub set, based on randomly generated samples of feasible hub location networks. In this context, feasible hub node shares in the optimal hub network structure can be easily determined.

Generating random feasible p-hub median networks is a fairly simple process. Assuming that there are m nodes in a network, it is sufficient to randomly select p network locations and assign the remaining nodes to the nearest hubs locations to obtain a feasible solution. We produce t random feasible networks $\{N^1, N^2, \dots, N^t\}$, and calculate the objective functions based on each feasible solution $\{O^1, O^2, \dots, O^t\}$. The time complexity of establishing a feasible sample set is Θ(mt).

We use the binary variable x^k to define the statistical metric. N^k represents the randomly generated k_{th} network structure, where $x_i^k = 1$ indicates that node i is the hub in the k_{th} sample, otherwise $x_i^k = 0$. In the first of the statistical criteria, a calculation is performed based on the ordering of the network structures. The objective function values of randomly generated hub networks are sorted in ascending order and r^k is used to define the k_{th} sample ordering. The following measure is used based on the order for the node n_i,

$$P_r(n_i) = \sum_{k=1}^{t} \frac{x_i^k}{r^k} \tag{4.17}$$

where $i = 1,2,\dots,m$. Nodes with a high ranking-based score that occur often in high-quality sample network structures are more likely to be part of an optimal hub set. Then, on a graph, we divide the maximum ranking-based score to normalize each ranking-based score,

$$P_{14}(n_i) = \frac{P_r(n_i)}{maxP_r(n_p)} \quad i = 1,2,\dots,n \tag{4.18}$$

This normalization prevents a feature with a significant value from taking precedence over a classification function. The second kind of statistical measure that we have devised is a correlation-based measure. This measure computes the Pearson correlation coefficient between each variable x_i and the objective values that are found throughout the sample networks:

$$P_c(n_i) = \frac{\sum_{k=1}^{t}(x_i^k - \bar{x}_i)(o^k - \bar{o})}{\sqrt{\sum_{k=1}^{t}(x_i^k - \bar{x}_i)^2}\sqrt{\sum_{k=1}^{t}(o^k - \bar{o})^2}} \tag{4.19}$$

where $\bar{x}_i = \sum_{k=1}^{t} x_i^k /t$, and $\bar{o} = \sum_{k=1}^{t} o^k /t$. Since p-HMP is a minimization problem, nodes in an optimum network should have a high degree of negative correlation with the objective values. This will increase the likelihood that the network will be optimal. In a similar manner, we normalize the correlation-based score by the smallest correlation value that can be found in a graph:

$$P_{15}(n_i) = \frac{P_c(n_i)}{minP_c(n_p)} \; p = 1,2,\dots,n \tag{4.20}$$

Sun et al. (2019) estimated the time and space complexity $\Theta(mt^2)$ of computing the binary representation of these two statistical measures directly from x. This calculation cost a significant amount of time. However, they have adopted a way to further reduce the computation complexity that has to be accomplished. Calculating the Pearson correlation coefficient may be achieved with minimal effort using the two equations that are provided in the following.

$$\sum_{k=1}^{t} \left(x_i^k - \bar{x}_i\right)^2 = \bar{x}_i(1 - \bar{x}_i)t \tag{4.21}$$

$$\sum_{k=1}^{t} (x_i^k - \bar{x}_i)(o^k - \bar{o}) = (1 - \bar{x}_i)S_i^1 - \bar{x}_i S_i^0 \tag{4.22}$$

$$S_i^1 = \sum_{1<k<t;x_i^k=1}^{t} (o^k - \bar{o});\; S_i^0 = \sum_{1<k<t;x_i^k=0}^{t} (o^k - \bar{o}); \tag{4.23}$$

4.8.4. Identifying Features for Hub Connections

In hub location problems, determining hub locations alone is not sufficient for a holistic solution of the problem. Hub allocation and hub connection decisions are also included in the solution of the problem. The node assignment procedure is usually done by considering the nearest hub, and this is probably optimal for non-hub nodes (not always). However, hub connection decisions are important in incomplete hub networks. For example, considering four links q=4 on a network with p=5 hubs, which edges should be selected is an incomplete hub network. In this context, the second stage is the identification of candidate connections over certain metrics for hub connections. For this, four graph-based indicators and two statistical metrics are considered.

4.8.5. Edge Features from Graph

Recall that the objective of the G (V, E) hub p-hub median problem is to search for a hub network structure within certain allocation strategies (single and multiple allocation) to minimize total transportation costs. We determine the basic 15 metrics directly related to the objective function of the problem in the node definition section. At the same time, we extract a similar set of definitions for the edges that provide the inter-hub connections. Eight of these definitions are graph-based and the remaining two are statistical metrics. Statistical metrics are exactly the same as in the node definition section.

$$P_1(a_{ij}) = \frac{f_{(i,j)} - minf_{(i,k)}}{maxf_{(i,k)} - minf_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.24}$$

$$P_2(a_{ij}) = \frac{d_{(i,j)} - mind_{(i,k)}}{maxd_{(i,k)} - mind_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.25}$$

$$P_3(a_{ij}) = \frac{d_{(i,j)}f_{(i,j)} - min\, f_{(i,k)}min\, d_{(i,k)}}{maxf_{(i,k)}maxd_{(i,k)} - minf_{(i,k)}min\, d_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.26}$$

$$P_4(a_{ij}) = \frac{d_{(i,j)}f_{(i,j)} - min\, f_{(i,k)}\, d_{(i,k)}}{maxf_{(i,k)}d_{(k,j)} - minf_{(i,k)}\, d_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.27}$$

$$P_5(a_{ij}) = \frac{d_{(i,j)}f_{(i,j)} - \ min\, d_{(i,k)}f_{(i,k)}}{maxd_{(i,k)}f_{(i,k)} - min\, d_{(i,k)}f_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.28}$$

$$P_6(a_{ij}) = \frac{f_{(i,j)} - \sum_{k=1}^{n}\frac{f_{(i,k)}}{n}}{maxf_{(i,k)} - minf_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.29}$$

$$P_7(a_{ij}) = \frac{d_{(i,j)} - \sum_{k=1}^{n}\frac{d_{(i,k)}}{n}}{maxd_{(i,k)} - mind_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.30}$$

$$P_8(a_{ij}) = \frac{f_{(i,j)}d_{(i,j)} - \sum_{k=1}^{n} \frac{f_{(i,k)}d_{(i,k)}}{n}}{maxf_{(i,k)}d_{(i,k)} - minf_{(i,k)}d_{(i,k)}} \quad k = 1,2,\dots,n \tag{4.31}$$

4.8.6. Statistical Metrics for Hub Arcs

In order to determine the direct connections between HUBs, first of all, the parametric value of q used in incomplete p-hub median problems should be known. This value represents the number of hub connections externally identified into the problem. Hub-to-hub connections are usually established with the lowest cost in mind. For this, the nodes to which hubs are allocated must also be defined. Since the hub locations are determined in the first stage of the problem, the flow amount on the hubs can be easily obtained by completing the nodes closest to the determined hub locations. The important point at this stage is to ensure that the flow over the hubs is distributed at minimum cost. In this context, we can define direct transport edges for the lowest cost transportation based on the amount of flow and distance between each hub pair.

In order to calculate the first statistical metric, cost-based objective function values are calculated over the randomly generated t feasible hub edges with connection number q. We produce t random feasible networks $\{N^1, N^2, \dots, N^t\}$, and calculate the objective functions based on each feasible solution $\{O^1, O^2, \dots, O^t\}$. The time complexity of establishing a feasible sample set is Θ(mt).

The statistical metric is defined by the binary variable x^k. For each sample, N^k represents the random generation of a network topology with $x_{ij}^k = 1$ as the hub edge, and $x_{ij}^k = 0$ for nodes that are not hub edges. The order of the network structures is used as the basis for the first statistical feature. Randomly generated hub networks' objective function values are ordered ascendingly, and the k_{th} sample ordering is defined by r^k. Based on the node a_{ij}'s order, the following metric is applied:

$$P_r(a_{ij}) = \sum_{k=1}^{t} \frac{x_{ij}^k}{r^k} \quad j = 1,2,\dots,n \tag{4.32}$$

where $i = 1,2,3, \ldots, n$. Network structure samples with high ranking-based scores are more likely to have an optimal hub connection configuration. To normalize each ranking-based score, we divide the highest ranking-based score by a graph.

$$P_9(a_{ij}) = \frac{P_s(a_{ij})}{maxP_s(a_{kl})} \quad k, l = 1,2, \ldots, n \tag{4.33}$$

The purpose of the normalization process is to prevent large-value features from coming to the fore in edge classification.

The second statistical metric presented is obtained by correlation-based computations. We calculate the Pearson correlation coefficient between the objective function value and the variable x_{ij} for the entire network.

$$P_c(a_{ij}) = \frac{\sum_{k=1}^{t}(x_{ij}^k - \bar{x}_{ij})(o^k - \bar{o})}{\sqrt{\sum_{k=1}^{t}(x_{ij}^k - \bar{x}_{ij})^2}\sqrt{\sum_{k=1}^{t}(o^k - \bar{o})^2}} \tag{4.34}$$

where $\bar{x}_{ij} = \sum_{k=1}^{t} x_{ij}^k / t$, and $\bar{o} = \sum_{k=1}^{t} o^k / t$. Edges that are strongly negative correlated with the objective values are likely to be in an optimum network since HLP is a minimization issue. Therefore, we use the smallest correlation value in a graph to standardize the correlation-based ranking. In the hub location feature identification phase, the method developed to make correlation-based metric calculations easier is applied in the same way for edge classification. Transactions and pseudo code are not replayed in this section.

4.8.7. R-GVNS Implementation on Hub Location Problems

The algorithm steps of the R-GVNS algorithm are similar to the processing mechanism of the GVNS algorithm. But it basically has two differences with the GVNS algorithm. The first one is the initial solution creation strategies, and the other one is the search strategies over the reduced decision variables.

Unlike GVNS, a hub set is obtained within the framework of the first determined features. For this, the sorting and prioritizing algorithm is used. According to the connection characteristics between the obtained hub locations, the sorting and prioritizing algorithm is applied for each possible edge. However, the weight of each feature varies in different iterations. Thus, we also highlight the diversification feature in the search space. In addition, after the application of each neighborhood operator, the candidate hub set and the candidate edge set change. This is because, after a neighborhood structure is applied, a node that is not in the specified set enters the solution set.

Algorithm: Sorting and Prioritizing Nodes

```
Initialization (flows, distances, max flow between nodes, max distances between nodes, number of p)
while i > network size do:
        compute
        p_i^1 p_i^2, p_i^3, p_i^4, p_i^5, p_i^6, p_i^7, p_i^8, p_i^9, p_i^10, p_i^11, p_i^12, p_i^13
end
Generate random numbers for each feature between 0-1
Assigned generated random number to the features
Normalize the assigned weights for features
Compute node weights based on weighted features
Construct candidate hub set
```

In the first step of the R-GVNS algorithm, it_{max} and loc_{max} parameters are defined as in GVNS. Then, the candidate hub set is obtained with the sorting and prioritizing nodes algorithm. Number of p hub selection is obtained randomly over the obtained hub set. The connections between the obtained hub locations are leveled with the sorting and prioritizing edges algorithm and an edge set of size 2*q is obtained. The initial solution is obtained by creating randomly selected connections from the edge set. At this stage, non-hub nodes are assigned to the nearest hub location. Neighborhood search operators are operated iteratively over the initial solution obtained. However, since hub location changes can occur outside the candidate set in each iteration, the update process is performed. In addition, the weights of node and edge features vary in each iterational update.

Algorithm: Sorting and Prioritizing Edges

```
Initialization (flows, distances, max flow between nodes, max distances between nodes, number of p)
while i > network size do:
        compute
        p_i^1 p_i^2, p_i^3, p_i^4, p_i^5, p_i^6, p_i^7, p_i^8                ,
end
Generate random numbers for each feature between 0-1
Assigned generated random number to the features
Normalize the assigned weights for features
Compute edge weights based on weighted features
Construct candidate edge set
```

Algorithm: Sequential R-GVNS

```
Identify itmax and locmax
Apply Sorting and Prioritizing Nodes Algorithm
Apply Sorting and Prioritizing Edges Algorithm based on cadidate hub set
Construct Initial Solution according to the candidate hub and edge set
Best Solution=Initial Solution
loc=1
for i=1 to itmax do:
        j=1
        while j<9 do:
                if j=1 do:
                    apply neighborhood operator samehubnodecl () on current network;
                else if j=2 do:
                    apply neighborhood operator closestnodeinsert () on current network
                else if j=3 do:
                    apply neighborhood operator nodeinsert () on current network;
                else if j=4 do:
                    apply neighborhood operator samehubnode () on current network;
                else if j=5 do:
                    apply neighborhood operator samehubnodeflow () on current network;
                else if j=6 do:
                    apply neighborhood operator samehubnodebtw () on current network;
                else if j=7 do:
                    apply neighborhood operator difhubnode () on current network;
                else if j=8 do:
                    apply neighborhood operator samehublink_swap () on current network;
                else if j=9 do:
                    apply neighborhood operator differenthublink_swap () on current network;
                j=j+1
                update node and edge candidate set
                end
                if there is a cost reduction with new network:
                    best solution = new solution, current network = new network and j=1
                    loc=1
                else:
                    loc= loc+1
                end
                if loc<locmax do:
                    apply shaking () on current network;
                 update node and edge candidate set
                else:
                    current network = new network and j=1
        end
        locmax=1
end
```

5. COMPUTATIONAL RESULTS

5.2. Experimental Testbed

The Civil Aeronautics Board (CAB) established the collection of data, which was based on the traffic of airline passengers between cities in the United States. It includes 60 instances, each of which may have up to 25 nodes and as many as four hubs. Also, instances containing 10 to 25 nodes were generated by us for CAB data. It is assumed that the costs of collection and distribution, χ and δ, are each equal to one, whereas α may take on any value between 0.2 and 1. In addition, the CAB dataset has been expanded up to 100 nodes in the current literature.

The data set used by AP originated from Australia Post's attempt to solve a real-world problem with hub location. The largest instance has 200 nodes, which each show a single postcode region. The settings for this instance are $\chi = 3$, $\alpha = 0.75$, and $\delta = 2$. Through the process of grouping the largest initial instance, it is possible to get smaller size instances with 10, 20, 25, 50, or 100 nodes. In the scenarios that have been evaluated, the number of hubs, also known as mail sorting and consolidation centers, may reach up to 10.

The Turkish Network (TR) dataset is connected to the 81-node Turkish postal network that was presented by Tan and Kara [10]. This dataset is one of the largest datasets available in the literature pertaining to hub locations. It takes into account the mail traffic as well as the distances between cities. In addition to this, it offers a set hub cost and a fixed link cost. In the computational framework that we have developed, we do not need the fixed hub cost that was provided in this dataset. We take into account that $\chi = 1$, $\alpha = 0.80$, and $\delta = 0.90$.

The three basic data sets considered in this thesis, CAB, TR and AP, have different properties. The first of these are the coefficients used in collection; distribution, and hub-to-hub transportation. In addition, nodes (hub candidates) are located at closer distances from each other in the AP dataset, while these distances are greater in the CAB dataset. In addition, the flows in the AP and TR datasets are not symmetrical. In contrast, flows between nodes in the CAB dataset are symmetrical.

Information on CAB, AP and TR datasets, which optimal solution is known in the literature, can be obtained from the Table 5.1. Accordingly, the solution limits of the problems with complete and incomplete structures differ each other. For complete problems, exact solution results can be presented up to 80 nodes for each data, while this limit is 25 for problems with incomplete topological feature. The maximum number of hubs considered for each data set is 16.

Table 5.1. Optimal Solutions of Data Sets According to Network Size and Number of Hubs

Data Sets	Min Network Size	Max Network Size	Min Number of Hubs	Max Number of Hubs	Flows
CAB (complete)	10	80	2	16	Symmetrical
AP (complete)	10	80	2	16	Not Symmetrical
TR (complete)	10	80	2	16	Not Symmetrical
CAB (incomplete)	10	25	2	8	Symmetrical
AP (incomplete)	10	25	2	8	Not Symmetrical
TR (incomplete)	10	25	2	8	Not Symmetrical

5.2. Lower Bounds

As can be seen from the Table 5.1., optimal solutions for incomplete p-hub median problems for networks with more than 25 nodes are not known. In addition, in networks with a complete structure, solutions up to 80 networks can be achieved. In this context, performance evaluation for larger sized samples is done in two ways. First, incomplete problems are evaluated based on complete solutions lower bound. This is because a complete solution gives an objective function value equal to or smaller than an incomplete solution with the same instance size. For instance, consider three hub locations as in the following figure. In the first figure, when all hub locations are connected, depending on the triangle inequality, the shortest path is provided directly between the two connected hubs. But in the second figure, two specific hubs are not connected according to their incomplete hub positions. Therefore, although the flow between these two hubs is not changed, the distance increased. As a result, transportation costs tend to increase. In the figure, the values above the edge represent the distances between hubs, and the values below shows the flows. As can be seen in the figure, the total costs have increased from 1300 to 1700, since the transportation between hubs A and B will be carried out via C.

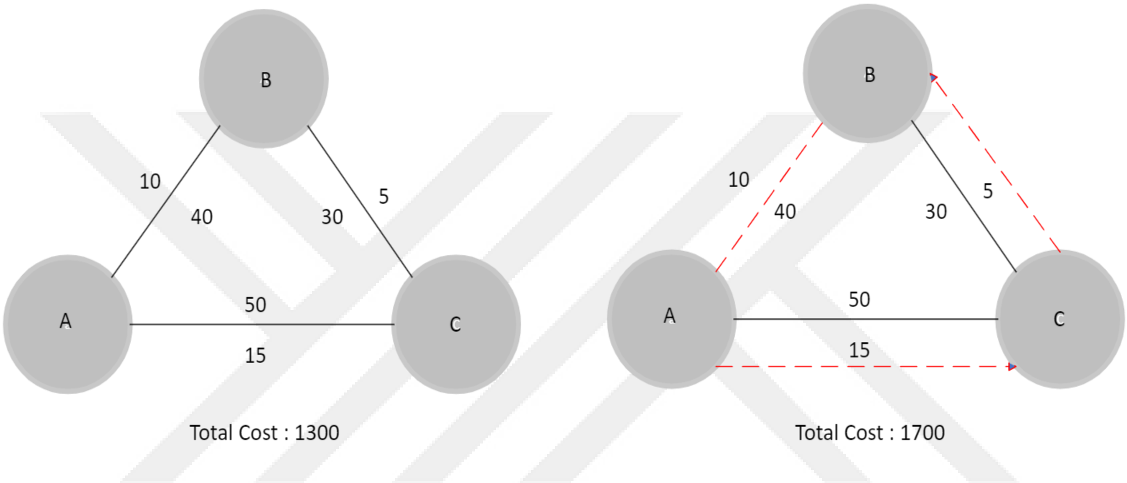


Figure 5.1. Transportation cost difference between complete and incomplete hub structure

In this context, complete networks can be taken as reference for incomplete network examples over 25 whose optimal solution is unknown. This is because in networks with triangle inequality, the costs in complete networks are always less than incomplete networks.

Another performance evaluation approach is based on comparison with the results of similar studies in the literature. However, there is no heuristic approach developed in the literature for incomplete p-hub median problems. For this reason, GA and SA algorithms are used for performance evaluation besides VNS algorithm in this thesis.

5.3. Experimental Test Results

In this section, the calculation results of the developed VNS, GA, and SA algorithms are presented. In addition, performance comparisons between these algorithms and comparisons with CPLEX results are given. All tests were carried out on an Intel(R) Core

(TM) i7-9850H CPU @ 2.60GHz with 32GB RAM. Algorithms were coded in Python programming language in ANACONDA ecosystem in Spyder interface and tested on CAB, AP, URAND and TR datasets.

In this section, to test the performance of the implemented algorithms, we first focus on the complete hub location problems. Optimal solutions up to 80 nodes can be obtained for AP, CAB, URAND and TR datasets. In addition, on datasets up to 400 nodes solved. Data sets are examined in three categories. Small-size (10 to 30 nodes), medium-size (50 nodes) and large-size (80 to 400 nodes) data are analyzed in separate tables, respectively. Two key performance metrics are considered in the experimental results. The first of these is the convergence to the optimal solution, and the other is the solution time. In the result tables, the name of the test data is written under the title of data set, and N. Size indicates the size of the data considered. For example, if N. Size is 10, this network consists of 10 nodes. p and q represent the number of hubs and the number of connections between hubs, respectively. The opt column shows that the optimal solution of the problem variant is known, while obj shows the objective function value of the solution. CPU time is the column that shows the solution times of the algorithms in seconds. The gap (%) indicates the percentage difference of the solution obtained by the relevant algorithm from the optimal solution. It indicates as $Gap\ (\%)_{best} = 100 * \frac{sol_{best} - opt.sol.}{opt.sol.}$. All methods were run 20 times in each instance (for small and medium size problems).

The parameters of the integrated SA algorithm are adjusted with a good balance between time and solution quality. In the experimental set, different parameter combinations were tried in many tests problem examples and the best experimental results were considered in this direction. For small size problems, T_s=200 as the starting temperature and T_e=1 as the stopping criterion was chosen. For large-sized problems, it was defined as Ts=400 and T_e=1. A value of 0.99 was used for the cooling parameter Cr. It was defined as N_i = 25, which represents the number of calls at each temperature level. The reason for keeping this value relatively small is that instead of applying a randomly chosen neighborhood structure in our SA algorithm, all neighborhood structures are tried in order at each temperature level. The higher the N_i value, the higher the time consumption. In case of any improvement in the objective function in the local search process, the r parameter, which determines the rate of increase in the number of iterations, was identified with a value of 0.75. The fact that the solutions obtained with the SA algorithm are highly dependent on the parameters in terms of time and solution quality requires precise determination of the defined parametric values.

GA parameters are determined by considering the studies that give the best solution available in the literature. The maximum number of generations is 500 for smaller problems and 5000 for larger problem instances. The algorithm also stops if the best individual or best objective value remains unchanged for successive generations, up to 200 iterations for smaller problems and 2000 iterations for larger problem instances, respectively. In GA, the initial population is randomly generated. The initial population consists of 150 individuals for small-size problems and 300 individuals for large-size problems. Based on the implemented "nearest neighbor allocation" and minimization objective function, we prefer "closer" hubs for each non-hub node. In the implemented GA method, the tournament approach is used as the selection method. In this selection method, a certain number of chromosomes (mostly two chromosomes) are randomly selected from the existing population and their fitness values are compared. The mutation rate M_{prob} and crossover rate C_{prob} are 0.05 and 0.85, respectively. The crossover strategy is applied both single-point and two-point. Crossover point selected randomly. Offspring produced by a crossover operator enter the mutation process. The mutation operator occurs on a randomly selected gene in the genetic code. Both crossover and mutation decisions are made according to whether the generated random number is less than the determined rates. The population changes, except for the best individuals who pass directly on to the next generation. Elitist individuals preserve the population's highly cohesive genes, and objective function values are not calculated based on iteration.

In order to prevent the loss of the best chromosome as a result of the change experienced during crossover and mutation, as the selection operator, the elitism method, which copies this chromosome and hides it, is used. It is known that the elitism method increases the performance of the genetic algorithm by preserving the best and following the process steps.

For each test case we defined loc_{max}=p shaking and stopping condition defined as n iterations without improvement in GVNS algorithm. Besides, it was defined as it_{max}=5n (n is the network size). Hub location problems where the k_{max} value is small than the p value (especially for incomplete networks) is kept higher. For this, the equation was $k_{max} = max\{p, 5\}$ used. Because, in cases where the p value is small, if the problem size is large, the local search procedure is not sufficient. The algorithmic parameters used in the R-GVNS algorithm are also the same as those used in GVNS. However, since the initial solutions of the R-GVNS algorithm usually converge to the optimal, it is natural that there is no improvement in high number of iterations. Therefore, in the R-GVNS algorithm stopping condition defined as 2*n iterations without improvement. The R-GVNS algorithm defines $hub\_set = 2 * p$ set of candidate hubs in the initial solution and update operations before each shaking procedure. This set may vary according to the random value that each feature will receive. However, due to the inclusion of different nodes in the solution during the local search process, the determined initial hub sets may change. Changes in hub sets require updating edge sets as well. In this context, both hub set and edge set are redefined after the change of hub locations after local search neighborhoods. For the edge set, $edge\_set = 2 * q$ candidate edges are determined. Two neighborhood operators that affect the connection structure between hubs are applied over the determined candidate edge set.

5.3.1. Small Size Complete Network Structure Instances

As seen in Table 5.2, all algorithms provide optimal solution for ap dataset up to 30 nodes. In this respect, percentage gap values are 0 for all algorithms. On the other hand, the solution time performance of the algorithms is different from each other. It can be seen on the Figure 5.2.

The developed R-GVNS algorithm seems to be quite superior in terms of solution time with an average of 0.06 sec. Although GA and GVNS meta-heuristics have close average solutions times, GA has slightly better performance than GVNS. The SA algorithm, on the other hand, has a worse performance with 0.220 sec than other algorithms in terms of solution time. It can also be seen that the average CPLEX solution time is 3445.4 seconds. In this context, meta-heuristic approaches can provide the optimal solution in a very short time. Especially the R-GVNS algorithm seems to be quite efficient for small size complete AP datasets. In addition, the standard deviation values obtained as a result of running each algorithm 20 times are also included in the Figure. Accordingly, in terms of percentage standard deviation, the R-GVNS algorithm is quite stable compared to other algorithms and gives better results.

Table 5.2. Small Size Complete USAp-HMP Solutions and Comparisons for AP data sets

						CPLEX		SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
ap	10	2	1	0.2	opt	56820888	0.67	0	0.001	0	0.003	0	0.002	0	0.000
ap	10	2	1	0.4	opt	61509402	1.30	0	0.001	0	0.002	0	0.003	0	0.000
ap	10	2	1	0.6	opt	65814135	2.31	0	0.002	0	0.003	0	0.002	0	0.000
ap	10	2	1	0.8	opt	69625203	3.30	0	0.009	0	0.007	0	0.005	0	0.000
ap	10	4	1	0.2	opt	56820888	6.29	0	0.006	0	0.002	0	0.001	0	0.000
ap	10	4	1	0.4	opt	61509402	8.26	0	0.011	0	0.003	0	0.002	0	0.000
ap	10	4	1	0.6	opt	65814135	8.31	0	0.011	0	0.004	0	0.004	0	0.000
ap	10	4	1	0.8	opt	69625203	10.26	0	0.016	0	0.010	0	0.010	0	0.000
ap	15	2	1	0.2	opt	60730465	0.14	0	0.007	0	0.002	0	0.004	0	0.000
ap	15	2	1	0.4	opt	65441825	3.49	0	0.012	0	0.003	0	0.003	0	0.000
ap	15	2	1	0.6	opt	70153185	4.32	0	0.012	0	0.005	0	0.002	0	0.000
ap	15	2	1	0.8	opt	74764746	6.96	0	0.017	0	0.012	0	0.011	0	0.000
ap	15	4	6	0.2	opt	46714887	2.52	0	0.015	0	0.015	0	0.013	0	0.001
ap	15	4	6	0.4	opt	53222196	3.03	0	0.027	0	0.017	0	0.018	0	0.001
ap	15	4	6	0.6	opt	59699472	10.30	0	0.030	0	0.011	0	0.010	0	0.000
ap	15	4	6	0.8	opt	65884769	16.46	0	0.038	0	0.028	0	0.025	0	0.001
ap	20	2	1	0.2	opt	63339209	0.83	0	0.016	0	0.006	0	0.002	0	0.000
ap	20	2	1	0.4	opt	68166157	1.43	0	0.030	0	0.008	0	0.004	0	0.000
ap	20	2	1	0.6	opt	72852695	10.33	0	0.033	0	0.012	0	0.009	0	0.000
ap	20	2	1	0.8	opt	77286131	12.41	0	0.042	0	0.003	0	0.007	0	0.000
ap	20	4	6	0.2	opt	47908576	17.24	0	0.038	0	0.013	0	0.027	0	0.001
ap	20	4	6	0.4	opt	56086122	21.50	0	0.078	0	0.020	0	0.081	0	0.003
ap	20	4	6	0.6	opt	62496458	29.29	0	0.084	0	0.031	0	0.083	0	0.003
ap	20	4	6	0.8	opt	68748402	35.34	0	0.108	0	0.135	0	0.070	0	0.003
ap	25	2	1	0.2	opt	65444350	0.94	0	0.044	0	0.015	0	0.032	0	0.001
ap	25	2	1	0.4	opt	70030641	1.48	0	0.087	0	0.022	0	0.090	0	0.004
ap	25	2	1	0.6	opt	74530345	8.46	0	0.095	0	0.036	0	0.094	0	0.004
ap	25	2	1	0.8	opt	78888379	7.61	0	0.131	0	0.165	0	0.086	0	0.003
ap	25	4	6	0.2	opt	49937686	61.52	0	0.125	0	0.043	0	0.089	0	0.004
ap	25	4	6	0.4	opt	57246299	57.65	0	0.271	0	0.068	0	0.280	0	0.011
ap	25	4	6	0.6	opt	63521517	76.41	0	0.217	0	0.082	0	0.216	0	0.009
ap	25	4	6	0.8	opt	69796736	88.60	0	0.249	0	0.313	0	0.163	0	0.007
ap	25	8	28	0.2	opt	33939754	54.67	0	0.266	0	0.091	0	0.189	0	0.008
ap	25	8	28	0.4	opt	43590857	73.03	0	0.484	0	0.122	0	0.502	0	0.020
ap	25	8	28	0.6	opt	52654024	73.81	0	0.442	0	0.166	0	0.440	0	0.018
ap	25	8	28	0.8	opt	61236693	78.33	0	0.531	0	0.367	0	0.346	0	0.014
ap	30	2	1	0.2	opt	65954078	341.52	0	0.309	0	0.106	0	0.220	0	0.009
ap	30	2	1	0.4	opt	70873273	380.89	0	0.577	0	0.146	0	0.598	0	0.024
ap	30	2	1	0.6	opt	75554278	420.56	0	0.520	0	0.195	0	0.517	0	0.021
ap	30	2	1	0.8	opt	80235282	344.54	0	0.651	0	0.518	0	0.425	0	0.017
ap	30	4	6	0.2	opt	51809864	298.16	0	0.221	0	0.175	0	0.157	0	0.006
ap	30	4	6	0.4	opt	58598652	237.72	0	0.501	0	0.126	0	0.119	0	0.005
ap	30	4	6	0.6	opt	65037807	253.76	0	0.461	0	0.173	0	0.158	0	0.006
ap	30	4	6	0.8	opt	70936680	222.13	0	0.554	0	0.396	0	0.361	0	0.014
ap	30	8	28	0.2	opt	37271396	229.77	0	0.507	0	0.173	0	0.161	0	0.006
ap	30	8	28	0.4	opt	46105498	291.33	0	0.910	0	0.229	0	0.242	0	0.010
ap	30	8	28	0.6	opt	54607960	300.58	0	0.849	0	0.319	0	0.344	0	0.014
ap	30	8	28	0.8	opt	62678464	324.56	0	0.933	0	0.872	0	0.609	0	0.024

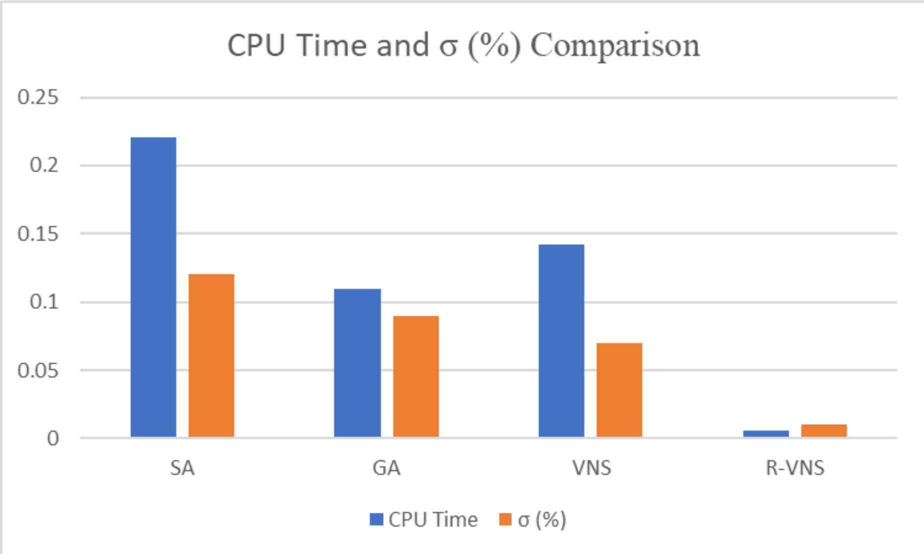


Figure 5.2. Algorithms performance comparisons for USAp-HMP for small size AP data sets

Similar results are obtained for SA, GA, VNS and R-GVNS algorithms for CAB and TR datasets up to 30 nodes. The Table 5.3 contains examples of various variants of the cab dataset. Optimal solutions obtained with CPLEX are in the obj column. Accordingly, all algorithms provide the optimal solution. VNS is slightly faster than the other two regular meta-heuristic approaches (SA and GA). On the other hand, the R-GVNS approach provides the optimal solution very quickly with average solution time of 0.01 compared to other algorithms. The solution times and percentage standard deviations of the four algorithms can be seen in the Figure 5.3. Although the results are similar to the AP datasets, the percentile standard deviations are lower for the CAB dataset. In other words, algorithms run 20 times provide more stable results for the cab dataset. The average CPLEX solution time increased by 32% in the CAB dataset compared to the AP dataset. Although the solution times of other algorithms have increased to the AP dataset, the average solution time of the R-GVNS algorithm is considerably lower than the AP datasets. The reason is that the distinctive metrics (centrality, statistical features, etc.) used in the R-GVNS algorithm can be identified more clearly in the CAB data set. In other words, graph features such as flows, and node distributions have more distinctive features for nodes. Therefore, candidate hubs are more likely to be included in the optimal solution. Since the TR dataset has closer features to the AP dataset in terms of the features it contains (both datasets are related to the postal service), the results are similar in terms of solution time and solution gaps. In terms of deviation from the best value found, close results are obtained for GA and GVNS. At the same time, the solution time difference is very low for these two algorithms. The R-GVNS algorithm is considerably superior to other algorithms in terms of both the deviation from the best value and the solution time performance metrics. In general, considering the small-sized complete problems of up to 30 nodes, it can be seen that the meta-heuristic approaches considered reach the optimal solution. In all datasets, GA and GVNS algorithms give better results than SA on both solution time and standard deviation criteria. In addition, meta-heuristic approaches are quite superior to CPLEX for cases where the network size exceeds 20 nodes. Another remarkable point is that the increase in the economies of scale parameter α increases the solution time of the problem.

Table 5.3. Small Size Complete USAp-HMP Solutions and Comparisons for CAB data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
cab	10	2	1	0.2	opt	16229500	0.28	0	0.009	0	0.010	0	0.009	0	0.000
cab	10	2	1	0.4	opt	18668515	0.23	0	0.007	0	0.007	0	0.007	0	0.000
cab	10	2	1	0.6	opt	21107529	0.27	0	0.006	0	0.006	0	0.006	0	0.000
cab	10	2	1	0.8	opt	23546544	0.23	0	0.007	0	0.006	0	0.006	0	0.000
cab	15	2	1	0.2	opt	56649272	2.65	0	0.032	0	0.033	0	0.026	0	0.001
cab	15	2	1	0.4	opt	64492041	2.41	0	0.026	0	0.028	0	0.031	0	0.001
cab	15	2	1	0.6	opt	72152968	2.22	0	0.032	0	0.027	0	0.037	0	0.002
cab	15	2	1	0.8	opt	78861493	2.47	0	0.047	0	0.052	0	0.044	0	0.002
cab	15	4	6	0.2	opt	29351869	1.83	0	0.058	0	0.059	0	0.060	0	0.003
cab	15	4	6	0.4	opt	41332600	1.81	0	0.061	0	0.060	0	0.059	0	0.003
cab	15	4	6	0.6	opt	53101450	1.91	0	0.096	0	0.087	0	0.082	0	0.003
cab	15	4	6	0.8	opt	64654950	1.92	0	0.077	0	0.061	0	0.088	0	0.004
cab	20	2	1	0.2	opt	1.05E+08	17.46	0	0.111	0	0.116	0	0.125	0	0.005
cab	20	2	1	0.4	opt	1.17E+08	20.49	0	0.099	0	0.102	0	0.109	0	0.005
cab	20	2	1	0.6	opt	1.28E+08	19.24	0	0.102	0	0.094	0	0.107	0	0.005
cab	20	2	1	0.8	opt	1.39E+08	21.18	0	0.088	0	0.093	0	0.103	0	0.005
cab	20	4	6	0.2	opt	67429582	10.71	0	0.105	0	0.091	0	0.105	0	0.004
cab	20	4	6	0.4	opt	85212850	10.98	0	0.081	0	0.090	0	0.074	0	0.003
cab	20	4	6	0.6	opt	1.03E+08	15.50	0	0.179	0	0.155	0	0.175	0	0.007
cab	20	4	6	0.8	opt	1.19E+08	12.91	0	0.462	0	0.471	0	0.445	0	0.021
cab	25	2	1	0.2	opt	4.52E+08	99.03	0	0.177	0	0.185	0	0.177	0	0.008
cab	25	2	1	0.4	opt	4.75E+08	61.25	0	0.315	0	0.285	0	0.292	0	0.014
cab	25	2	1	0.6	opt	4.98E+08	126.2	0	0.514	0	0.515	0	0.482	0	0.019
cab	25	2	1	0.8	opt	5.2E+08	98.34	0	0.832	0	0.843	0	0.821	0	0.035
cab	25	4	6	0.2	opt	2.38E+08	56.56	0	0.220	0	0.228	0	0.212	0	0.009
cab	25	4	6	0.4	opt	3.04E+08	61.52	0	0.309	0	0.313	0	0.245	0	0.012
cab	25	4	6	0.6	opt	3.69E+08	59.83	0	0.425	0	0.352	0	0.448	0	0.020
cab	25	4	6	0.8	opt	4.32E+08	47.34	0	0.969	0	1.065	0	0.946	0	0.045
cab	25	8	28	0.2	opt	1.4E+08	51.42	0	0.297	0	0.269	0	0.266	0	0.012
cab	25	8	28	0.4	opt	2.16E+08	62.96	0	0.377	0	0.360	0	0.361	0	0.015
cab	25	8	28	0.6	opt	2.9E+08	74.27	0	0.587	0	0.519	0	0.492	0	0.020
cab	25	8	28	0.8	opt	3.59E+08	58.29	0	0.494	0	0.420	0	0.475	0	0.020
cab	30	2	1	0.2	opt	8.59E+08	315.6	0	0.631	0	0.721	0	0.516	0	0.021
cab	30	2	1	0.4	opt	9.11E+08	325.3	0	0.946	0	0.760	0	0.630	0	0.028
cab	30	2	1	0.6	opt	9.63E+08	290.9	0	0.733	0	0.719	0	0.754	0	0.037
cab	30	2	1	0.8	opt	1.01E+09	211.2	0	0.924	0	0.990	0	0.780	0	0.035
cab	30	4	6	0.2	opt	5.52E+08	267.2	0	0.398	0	0.257	0	0.237	0	0.010
cab	30	4	6	0.4	opt	6.61E+08	299.5	0	0.576	0	0.489	0	0.470	0	0.022
cab	30	4	6	0.6	opt	7.54E+08	429.1	0	0.766	0	0.556	0	0.551	0	0.023
cab	30	4	6	0.8	opt	8.47E+08	622.9	0	1.326	0	0.544	0	0.504	0	0.022
cab	30	8	28	0.2	opt	2.58E+08	219.1	0	0.671	0	0.620	0	0.673	0	0.032
cab	30	8	28	0.4	opt	3.94E+08	156.1	0	0.466	0	0.422	0	0.389	0	0.017
cab	30	8	28	0.6	opt	5.3E+08	154.6	0	0.780	0	0.681	0	0.729	0	0.030
cab	30	8	28	0.8	opt	6.67E+08	216.0	0	1.474	0	0.786	0	0.581	0	0.024

The Figure 5.3 shows the average standard deviation and solution time measures of four different solution approaches for the CAB data set. As can be seen from the figure, the R-GVNS algorithm is quite superior to the other algorithms. In particular, the average solution time is quite short, and it has obtained the optimal solution for all small-sized problems.

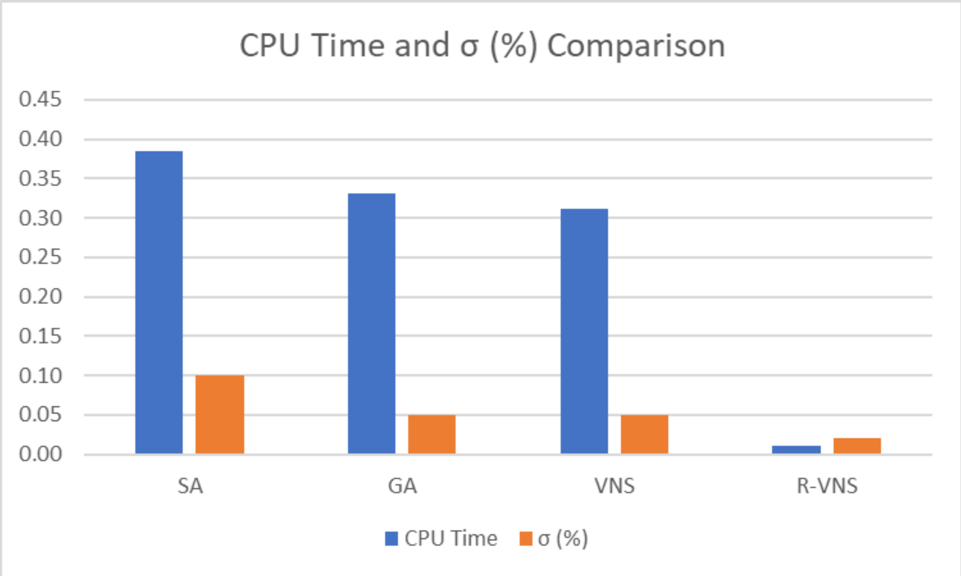


Figure 5.3. Algorithms performance comparisons for USAp-HMP for small size CAB data sets

The Table 5.4. contains the results of the small instance TR data set. In addition, the Figure 5.4 provides summary information over average values for the TR dataset. In this context, the R-GVNS algorithm gives good results for the TR dataset compared to other algorithms. All algorithms at least once obtain the optimal solution for all problem variants. The R-GVNS algorithm reaches the optimal solution for all algorithmic runs for the TR dataset. It can be seen that all approaches are quite efficient in terms of solution times. In cases where the number of hubs is high, there are great differences when compared to CPLEX solution time. The average solution times of the SA, GA, VNS, and R-GVNS meta-heuristics are 0.12, 0.08, 0.08, and 0.01 seconds, respectively. In contrast, the average CPLEX solution time is about 100 seconds. The absolute minimum amount of time needed for the CPU was around 0.24 seconds, while the absolute maximum was 375 seconds. It is clear from referring to Table 5.4 that the scenarios with higher values of turned out to be more challenging. Additionally, we found that the cases in which we caused the hub networks to be sparse were much more challenging than the instances in which the hub networks were almost complete. The most difficult of these 44 instances was when p was set to 8, and it consumed roughly 375 seconds of time on the computer's CPU. As was discussed in a prior analysis, Hub nodes, which have lower values of the parameter, are situated closer to the region's boundaries. Because of this, the higher cost values in the TR data set are likely to appreciate greater benefits as a result of economies of scale when the hub nodes are located closer to the peripheral regions. When there is a greater number of hub facilities in the center of the area, the value of increases.

.In the scenarios with fully connected hub networks, there was no noticeable rise in the proportion of additional expenses associated with transportation. This proportion goes up, which is to be anticipated, since the hub network is being driven to become sparser.

Table 5.4. Small Size Complete USAp-HMP Solutions and Comparisons for TR data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
tr	10	2	0	0.2	opt	917310421	0.24	0	0.03	0	0.01	0	0.02	0	0.000
tr	10	2	0	0.4	opt	967657939	0.27	0	0.04	0	0.01	0	0.03	0	0.000
tr	10	2	0	0.6	opt	1011037997	0.27	0	0.03	0	0.03	0	0.02	0	0.002
tr	10	2	0	0.8	opt	1054418055	0.20	0	0.02	0	0.03	0	0.00	0	0.001
tr	15	2	0	0.2	opt	1329775360	2.21	0	0.04	0	0.02	0	0.04	0	0.002
tr	15	2	0	0.4	opt	1418834904	2.95	0	0.05	0	0.03	0	0.03	0	0.002
tr	15	2	0	0.6	opt	1480108317	2.87	0	0.03	0	0.02	0	0.01	0	0.001
tr	15	2	0	0.8	opt	1535320089	2.89	0	0.02	0	0.04	0	0.01	0	0.001
tr	15	4	0	0.2	opt	769792452	2.45	0	0.02	0	0.02	0	0.01	0	0.001
tr	15	4	0	0.4	opt	955133029	1.94	0	0.03	0	0.02	0	0.01	0	0.001
tr	15	4	0	0.6	opt	1132243028	1.95	0	0.02	0	0.01	0	0.01	0	0.001
tr	15	4	0	0.8	opt	1309353027	2.03	0	0.01	0	0.04	0	0.00	0	0.001
tr	20	2	0	0.2	opt	2489678706	20.82	0	0.02	0	0.02	0	0.02	0	0.001
tr	20	2	0	0.4	opt	2600327229	21.76	0	0.04	0	0.03	0	0.03	0	0.002
tr	20	2	0	0.6	opt	2710975752	16.59	0	0.03	0	0.02	0	0.01	0	0.001
tr	20	2	0	0.8	opt	2821624275	21.56	0	0.06	0	0.03	0	0.03	0	0.003
tr	20	4	0	0.2	opt	1566511700	12.16	0	0.07	0	0.06	0	0.05	0	0.003
tr	20	4	0	0.4	opt	1883009244	13.86	0	0.06	0	0.03	0	0.03	0	0.003
tr	20	4	0	0.6	opt	2180103894	14.27	0	0.07	0	0.04	0	0.03	0	0.003
tr	20	4	0	0.8	opt	2444347156	13.45	0	0.05	0	0.06	0	0.03	0	0.002
tr	25	2	0	0.2	opt	3964284368	95.61	0	0.04	0	0.04	0	0.03	0	0.002
tr	25	2	0	0.4	opt	4371513562	103.80	0	0.05	0	0.04	0	0.02	0	0.002
tr	25	2	0	0.6	opt	4717338597	82.23	0	0.04	0	0.03	0	0.02	0	0.002
tr	25	2	0	0.8	opt	5063163632	105.42	0	0.05	0	0.04	0	0.04	0	0.002
tr	25	4	0	0.2	opt	2612187626	46.00	0	0.08	0	0.03	0	0.07	0	0.003
tr	25	4	0	0.4	opt	3292674555	58.05	0	0.09	0	0.05	0	0.05	0	0.004
tr	25	4	0	0.6	opt	3895729299	67.66	0	0.20	0	0.17	0	0.11	0	0.009
tr	25	4	0	0.8	opt	4431785781	59.85	0	0.09	0	0.05	0	0.04	0	0.004
tr	25	8	0	0.2	opt	1607887096	65.58	0	0.14	0	0.08	0	0.06	0	0.007
tr	25	8	0	0.4	opt	2367065968	65.25	0	0.05	0	0.03	0	0.02	0	0.002
tr	25	8	0	0.6	opt	3113998210	60.87	0	0.18	0	0.11	0	0.10	0	0.009
tr	25	8	0	0.8	opt	3804774016	45.94	0	0.17	0	0.11	0	0.07	0	0.008
tr	30	2	0	0.2	opt	5538105307	263.95	0	0.10	0	0.08	0	0.07	0	0.004
tr	30	2	0	0.4	opt	6072848590	256.51	0	0.07	0	0.05	0	0.04	0	0.003
tr	30	2	0	0.6	opt	6578819683	266.08	0	0.06	0	0.05	0	0.05	0	0.003
tr	30	2	0	0.8	opt	7084790775	275.55	0	0.39	0	0.41	0	0.23	0	0.017
tr	30	4	0	0.2	opt	3930606052	243.60	0	0.81	0	0.10	0	0.72	0	0.037
tr	30	4	0	0.4	opt	4707022968	227.96	0	0.16	0	0.14	0	0.08	0	0.007
tr	30	4	0	0.6	opt	5435868004	211.51	0	0.18	0	0.11	0	0.08	0	0.009
tr	30	4	0	0.8	opt	6132758306	197.70	0	0.11	0	0.11	0	0.10	0	0.004
tr	30	8	0	0.2	opt	2471551504	189.56	0	0.08	0	0.13	0	0.08	0	0.004
tr	30	8	0	0.4	opt	3480977763	236.07	0	0.33	0	0.19	0	0.17	0	0.015
tr	30	8	0	0.6	opt	4446896579	260.82	0	0.55	0	0.42	0	0.39	0	0.023
tr	30	8	0	0.8	opt	5333009484	375.99	0	0.37	0	0.33	0	0.32	0	0.017

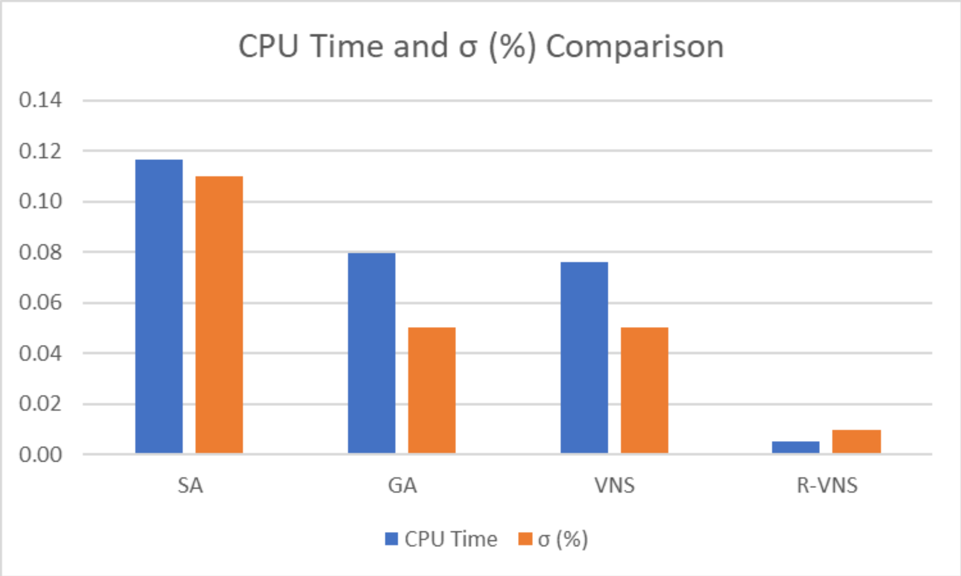


Figure 5.4. Algorithms performance comparisons for USAp-HMP for small size TR data sets

The figure shows comparisons of solution times for all small-size complete datasets. Comparisons of AP, CAB and TR datasets with different algorithms are included. In this figure, the variation of the solution times on the basis of the data set is also presented. It can be seen that the solution times of the CAB dataset increase for each algorithm. For small-sized instances, the maximum solution time is approximately 0.4 seconds with the SA algorithm.

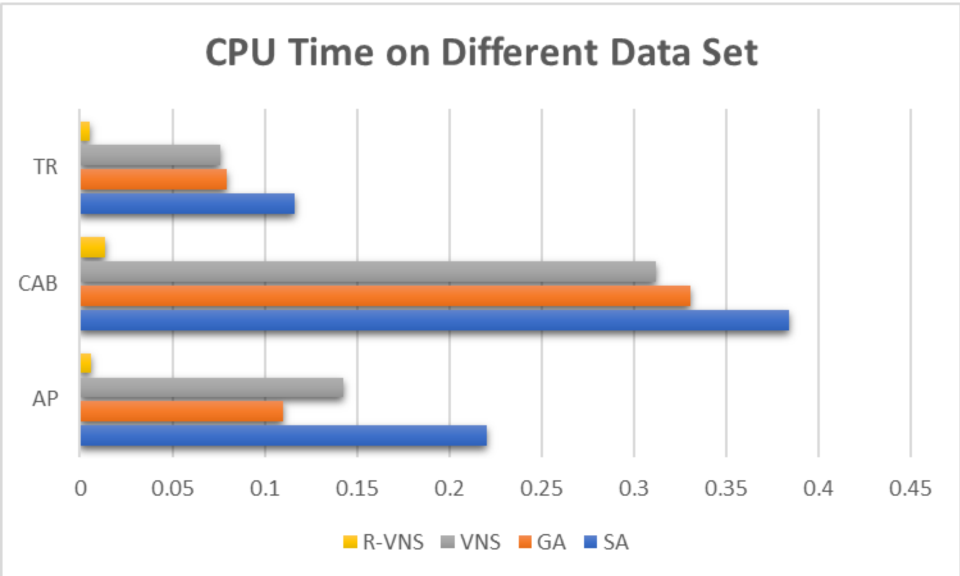


Figure 5.5. CPU time performance for different small size data sets

The R-GVNS algorithm has the minimum solution time for all datasets. In addition, the maksimum solution times for all datasets belong to the SA algorithm. It can also be observed that the average solution times of GA and VNS algorithms are close.

5.3.2. Medium Size Complete Network Structure Instances

For complete p-hub median problems, medium-sized data sets and networks of 50 nodes are considered. For the CAB, AP and TR datasets, the results of the algorithms run for the 50-node size are given in the Table 5.5. The SA algorithm has worse results compared to the GA and GVNS algorithms, both in terms of solution time and deviation from the best solution. SA and GA have similar results as in small-sized samples. R-GVNS, on the other hand, has a superior performance compared to other algorithms. The R-GVNS algorithm converges to the optimal solution faster due to the small number of candidate hub locations. In addition, R-GVNS ensures that the standard deviation is low by considering other hub locations besides the determined candidate hub positions.

All algorithms can obtain the optimal solution in all medium size complete hub location problems. The CPLEX solution times of medium size instances are approximately 7 times longer than small size instances. The results of the AP data set are represented in the Table 5.5. The average solution time of SA algorithm is 1.56 seconds, GA is 1.00 seconds, VNS is 0.93 seconds, and R-GVNS is 0.15 seconds. Medium size instances also reach optimal solutions in a very short time with the current implemented algorithms. A summary of these brief results is presented in the figure. As in the solution results of the small samples, the SA algorithm has the worst results in terms of standard deviation values and solution times. R-GVNS, on the other hand, is superior to other algorithms in terms of both metrics.

Table 5.5. Medium Size Complete USAp-HMP Solutions and Comparisons for AP data sets

						CPLEX		SA		GA		VNS		R-GVNS	
Data Set	Network Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
ap	50	2	1	0.75	opt	178484	337.55	0	0.122	0	0.110	0	**0.092**	0	**0.021**
ap	50	3	3	0.75	opt	159569	365.46	0	0.168	0	0.154	0	**0.122**	0	**0.045**
ap	50	4	6	0.75	opt	143378	438.11	0	0.325	0	**0.145**	0	0.151	0	0.096
ap	50	5	10	0.75	opt	132366	761.36	0	0.364	0	0.216	0	**0.193**	0	**0.102**

Increasing the α value for medium-sized problem also increases the difficulty of solving the problem. In addition, with the increase of the α value, the hub locations are concentrated in the central regions on the network and converge to each other. The scale economy coefficient has significant effect on the distribution of hub locations on the network.

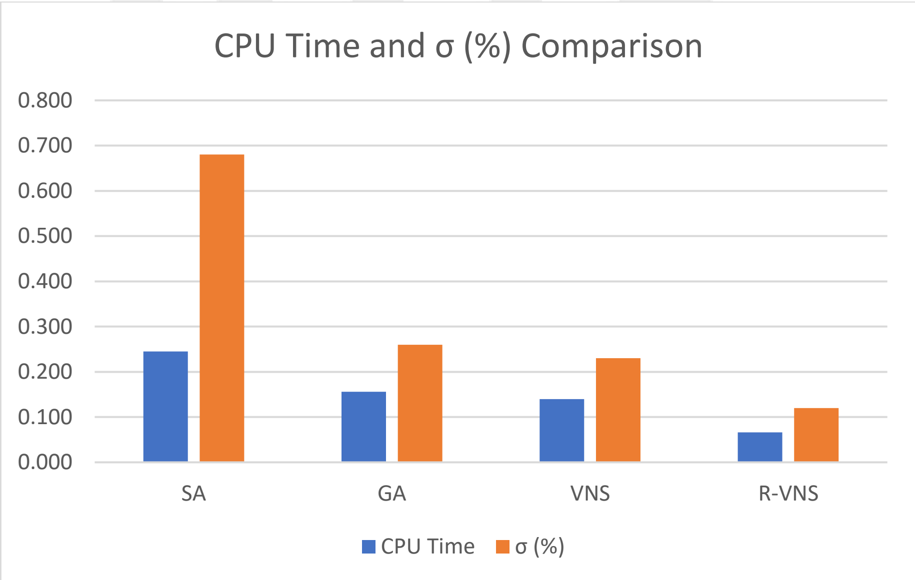


Figure 5.6. Algorithms performance comparisons for USAp-HMP for medium size AP data sets

The difference in solution times with the increase in the number of hubs for the TR dataset is more emphatic. Especially in scenarios with 8 hubs, the CPU time difference between the R-GVNS algorithm and other conventional algorithms can be clearly seen. All approaches achieve the optimal result at least once within the TR dataset. However, as can be seen in the figure, the superiority of the R-GVNS algorithm can be seen in terms of standard deviation values. The fact that the R-GVNS algorithm is more stable and has lower deviation values is directly related to the ratio of the candidate hub locations (included in the solution) to be in the optimal solution.

Table 5.6. Medium Size Complete USAp-HMP Solutions and Comparisons for TR data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
tr	50	2	0	0.2	opt	26558031387	544.67	0	0.13	0	0.18	0	0.08	0	0.01
tr	50	2	0	0.4	opt	29186653305	649.04	0	0.76	0	0.47	0	0.05	0	0.01
tr	50	2	0	0.6	opt	31085111101	894.49	0	0.52	0	0.44	0	0.42	0	0.07
tr	50	2	0	0.8	opt	32791758475	1927.65	0	0.31	0	0.47	0	0.31	0	0.06
tr	50	4	0	0.2	opt	18127018532	368.97	0	0.35	0	0.27	0	0.26	0	0.07
tr	50	4	0	0.4	opt	22188861604	977.16	0	0.60	0	0.23	0	0.41	0	0.08
tr	50	4	0	0.6	opt	25996176518	1455.31	0	0.26	0	0.32	0	0.37	0	0.05
tr	50	4	0	0.8	opt	28931912034	1305.26	0	1.87	0	0.63	0	1.23	0	0.12
tr	50	8	0	0.2	opt	12091388764	306.74	0	1.04	0	1.05	0	0.82	0	0.15
tr	50	8	0	0.4	opt	16628590467	485.05	0	2.95	0	2.44	0	1.97	0	0.21
tr	50	8	0	0.6	opt	20973448393	582.77	0	1.95	0	1.19	0	1.71	0	0.19
tr	50	8	0	0.8	opt	25172212877	729.70	0	5.07	0	0.66	0	1.72	0	0.21
tr	50	16	0	0.2	opt	8375191940	282.56	0	1.91	0	1.63	0	1.73	0	0.25
tr	50	16	0	0.4	opt	13241494092	322.39	0	2.61	0	1.99	0	1.63	0	0.28
tr	50	16	0	0.6	opt	18026116335	610.52	0	2.20	0	1.11	0	0.47	0	0.32
tr	50	16	0	0.8	opt	22677331819	1178.75	0	2.39	0	2.89	0	1.64	0	0.29

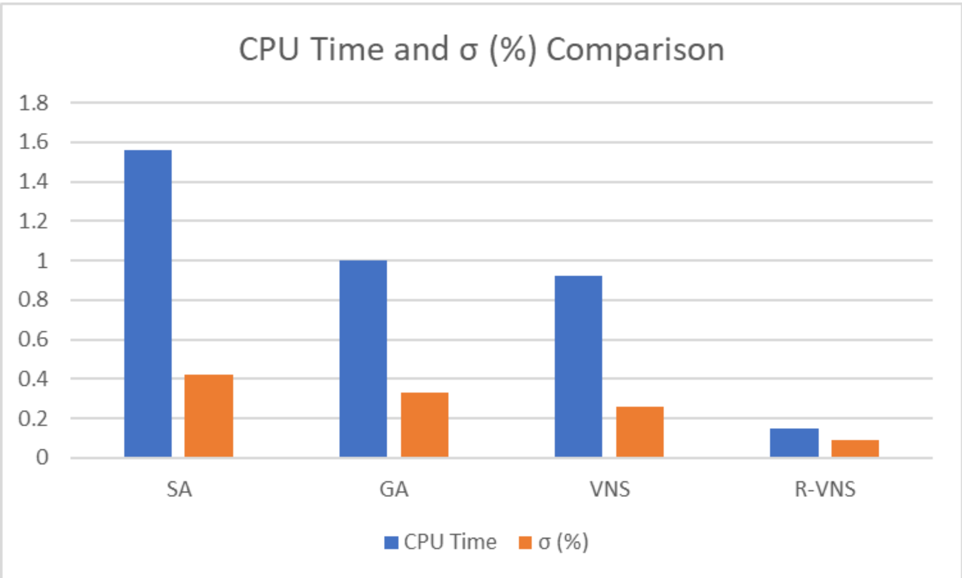


Figure 5.7. Algorithms performance comparisons for USAp-HMP for medium size TR data sets

SA, GA, VNS and R-GVNS results for the CAB-50 dataset, which is one of the medium-sized problem types, are reported in the Table 5.7. The results obtained show that all algorithms have found the optimal result at least once. In addition, the R-GVNS algorithm is superior in terms of solution time and standard deviation values. The shortest CPU time for all scenarios is achieved with R-GVNS. In addition, as can be seen from the Figure, the average deviation values in the CAB data set seem higher than in other data sets. The SA algorithm gives the highest σ (%) value of about 1.6. Considering the GA and VNS deviation values, it gives very close results.

Table 5.7. Medium Size Complete USAp-HMP Solutions and Comparisons for CAB data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
cab	50	2	0	0.2	opt	1816984711	246.820	0	0.021	0	0.103	0	0.014	0	0.001
cab	50	2	0	0.4	opt	1926529756	312.707	0	0.091	0	0.097	0	0.008	0	0.001
cab	50	2	0	0.6	opt	2027091343	176.420	0	0.115	0	0.214	0	0.100	0	0.009
cab	50	2	0	0.8	opt	2124663628	275.465	0	0.086	0	0.067	0	0.040	0	0.012
cab	50	4	0	0.2	opt	1233592814	238.946	0	0.115	0	0.045	0	0.683	0	0.094
cab	50	4	0	0.4	opt	1425665957	156.205	0	0.240	0	0.072	0	0.067	0	0.021
cab	50	4	0	0.6	opt	1606273388	513.166	0	0.067	0	0.057	0	0.119	0	0.068
cab	50	4	0	0.8	opt	1786880820	812.569	0	0.361	0	0.076	0	0.284	0	0.085
cab	50	8	0	0.2	opt	733005910	252.788	0	0.123	0	0.336	0	0.115	0	0.094
cab	50	8	0	0.4	opt	993547738	842.629	0	0.645	0	0.450	0	0.236	0	0.102
cab	50	8	0	0.6	opt	1244059039	840.646	0	0.420	0	0.665	0	0.841	0	0.163
cab	50	8	0	0.8	opt	1493504570	2686.593	0	3.411	0	0.137	0	0.283	0	0.087
cab	50	16	0	0.2	opt	455208274	629.579	0	0.367	0	0.373	0	0.203	0	0.115
cab	50	16	0	0.4	opt	736176464	413.452	0	0.697	0	3.748	0	0.723	0	0.206
cab	50	16	0	0.6	opt	1014217128	530.407	0	0.445	0	0.265	0	2.854	0	0.174
cab	50	16	0	0.8	opt	1287232739	865.401	0	1.717	0	0.714	0	0.356	0	0.216

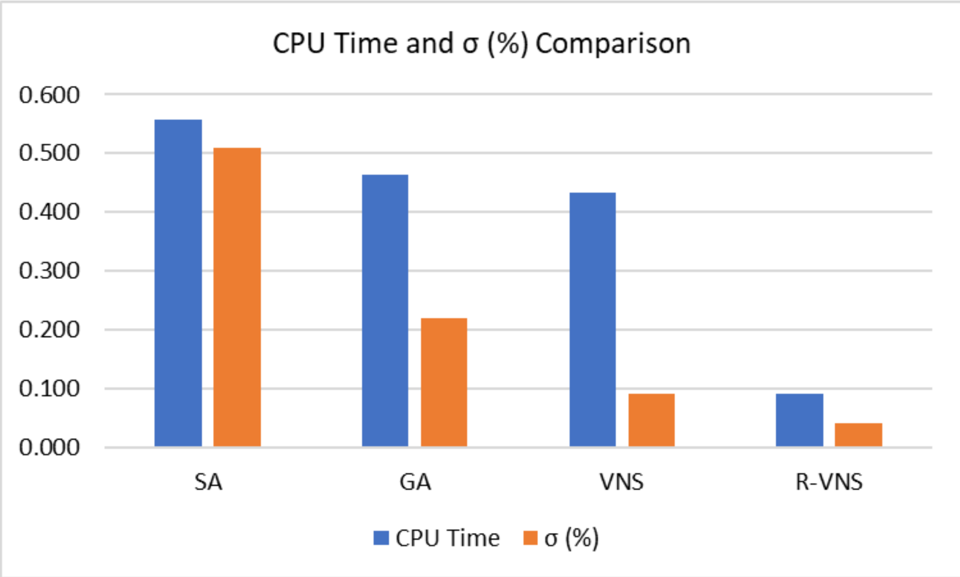


Figure 5.8. Algorithms performance comparisons for USAp-HMP for medium size CAB data sets

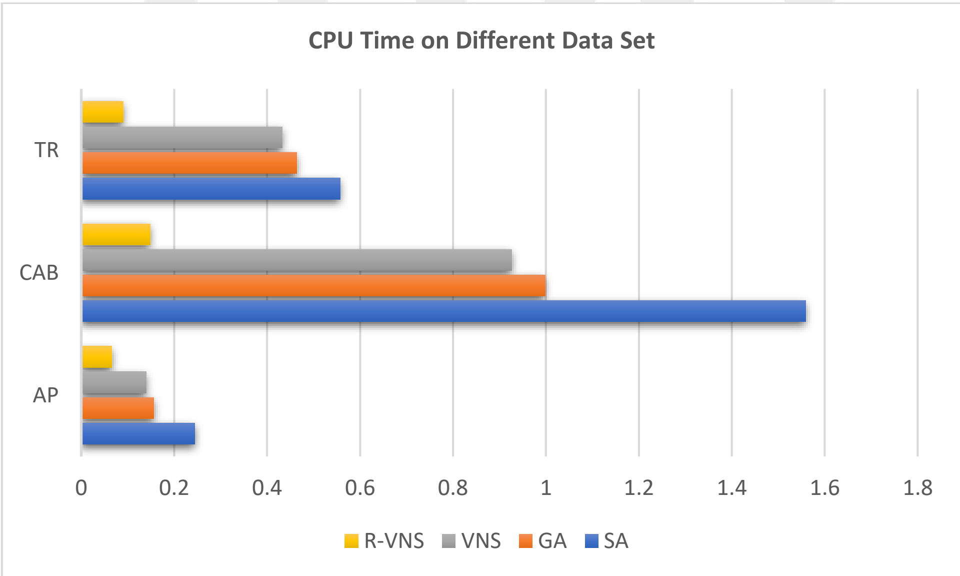


Figure 5.9. CPU time performance for different small medium data sets

5.3.3. Large Size Complete Network Structure Instances

In this section, the computational experiments to test the performance of the proposed SA, GA, VNS and R-GVNS algorithms are illustrated. In order to investigate the efficiency of the R-GVNS algorithm, the AP and URAND data sets has been used. For each instance, 20 different seeds generated randomly were employed in order to execute the test of the algorithms. From the results presented in the first part of Table 5.8, it follows that R-GVNS is capable of obtaining the best-known solutions for 12 large-size AP problems, while the R-GVNS solutions are almost always equal best-known but for the instance with sizes (n =100 and p = 20), (n =200 and p = 10), (n =200 and p = 20) . Furthermore, regarding the running time, the results with R-GVNS are significantly better than those obtained by the other implemented algorithms. In this regard, the results obtained by VNS have a similar behavior with respect to the ones provided by GA. In GA, it obtains best known solutions in 12 cases like R-GVNS, but the deviation values are slightly higher than the R-GVNS algorithm. Especially in n=200 and p=20 scenarios, it gives worse results with 1.2 deviation value compared to R-GVNS. But in general, there is no big difference in AP datasets in terms of solution performance between R-GVNS, GA, and VNS algorithms. The SA algorithm, on the other hand, has a relatively low competitive power compared to these algorithms.

For the URAND dataset, SA, GA, VNS and R-GVNS algorithms have been implemented for networks with n=100 and n=200 sizes. In addition, larger dimensional problems such as n=300 and n=400 are also used in this section. In scenarios where 28 different scenarios are considered, the R-GVNS algorithm obtains the best-known solutions in 23 cases. SA, GA and VNS were able to obtain the best-known value in 6, 10 and 16 solutions, respectively. However, while these algorithms usually capture the best solutions for network sizes of 200 and below, they rarely reach the best solution when the network size exceeds 200 nodes.

Table 5.8. Large Size Complete USAp-HMP Solutions and Comparisons for AP

					CPLEX	SA			GA			VNS			R-VNS		
Data Set	N. Size	p	q	alpha	Obj.	SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time
ap	100	2	1	0.75	-	**180223**	0	2.50	**180223**	0	0.76	**180223**	0	0.48	**180223**	0	0.04
ap	100	3	3	0.75	-	**160847**	0	2.84	**160847**	0	1.08	**160847**	0	1.02	**160847**	0	0.08
ap	100	4	6	0.75	-	**145896**	0	3.98	**145896**	0	1.22	**145896**	0	1.12	**145896**	0	0.09
ap	100	5	10	0.75	-	**136929**	0.11	17.06	**136929**	0	3.45	**136929**	0	2.59	**136929**	0	0.16
ap	100	10	45	0.75	-	**106470**	0	23.39	**106470**	0	13.54	**106470**	0	19.23	**106470**	0	1.59
ap	100	15	105	0.75	-	92434	0.26	85.62	**90533**	0	55.34	**90533**	0.18	57.72	**90533**	0	5.36
ap	100	20	190	0.75	-	**80270**	0	234.72	**80270**	0	100.96	**80270**	0	79.40	80342	0.09	5.43
ap	200	2	1	0.75	-	**182459**	0	8.44	**182459**	0	2.72	**182459**	0	2.86	**182459**	0	0.28
ap	200	3	3	0.75	-	**162887**	0	10.78	**162887**	0	7.02	**162887**	0	4.98	**162887**	0	0.33
ap	200	4	6	0.75	-	147767	0.16	23.92	**147767**	0	17.36	**147767**	0	15.50	**147767**	0	1.03
ap	200	5	10	0.75	-	145384	0.61	75.35	**140062**	0	33.76	**140062**	0	31.20	**140062**	0	1.81
ap	200	10	45	0.75	-	116646	0.73	115.09	110918	0.09	65.92	111028	0.16	73.74	110972	0.11	10.11
ap	200	15	105	0.75	-	103716	1.56	240.46	**94459**	0	150.62	95215	0.12	138.34	**94459**	0	18.62
ap	200	20	190	0.75	-	93705	2.02	402.17	90307	1.21	202.02	89288	0.76	190.30	**85252**	0.35	27.45

Considering the average CPU times for the AP data set, the values of 89, 46, 44 and 5 seconds for SA, GA, VNS and R-GVNS, respectively, can be seen in the Table 5.8. These average values for the URAND dataset are 329, 190, 164 and 34 seconds, respectively. In this context, it can be seen that there is not much difference in the solution times of GA and VNS algorithms. However, the VNS algorithm gives slightly better CPU time performance than GA. R-GVNS gives best-known results approximately 8 times faster than GA and VN

Table 5.9. Large Size Complete USAp-HMP Solutions and Comparisons for URAND data sets

						SA			GA			VNS			R-GVNS		
Data Set	Network Size	p	q	alpha	best known	SA-Best	Avg. Gap	CPU Time	GA-Best	Avg. Gap	CPU Time	VNS-Best	Avg. Gap	CPU Time	VNS-Best	Avg. Gap	CPU Time
URAND	100	3	0	0.75	34532.88	**34532.88**	0.000	3.248	**34532.88**	0.000	1.308	**34532.88**	0.000	1.402	**34532.88**	0.000	0.109
URAND	100	4	0	0.75	32608.28	**32608.28**	0.000	5.061	**32608.28**	0.000	1.689	**32608.28**	0.000	1.404	**32608.28**	0.000	0.099
URAND	100	5	0	0.75	31107.70	31109.26	0.001	19.292	**31107.70**	0.000	4.482	**31107.70**	0.000	2.668	**31107.70**	0.000	0.200
URAND	100	10	0	0.75	27058.40	27063.81	0.004	24.312	**27058.40**	0.000	15.270	**27058.40**	0.000	21.578	**27058.40**	0.000	3.073
URAND	100	15	0	0.75	25408.56	25416.18	0.006	112.550	25440.32	0.025	64.743	**25408.56**	0.000	69.231	**25408.56**	0.000	8.546
URAND	100	20	0	0.75	24377.65	24514.16	0.112	275.199	24495.88	0.097	125.102	**24377.65**	0.000	108.228	**24377.65**	0.000	12.484
URAND	200	3	0	0.75	139223.25	**139223.25**	0.000	12.544	**139223.25**	0.000	8.240	**139223.25**	0.000	6.408	**139223.25**	0.000	0.348
URAND	200	4	0	0.75	132676.89	**132676.89**	0.000	37.339	**132676.89**	0.000	23.710	**132676.89**	0.000	21.127	**132676.89**	0.000	1.414
URAND	200	5	0	0.75	127220.02	135648.35	1.325	83.857	132248.22	0.790	34.353	128835.71	0.254	41.446	**127220.02**	0.000	3.224
URAND	200	10	0	0.75	112539.21	118672.60	1.090	230.356	116087.40	0.631	57.781	**112539.21**	0.000	114.896	**112539.21**	0.000	9.980
URAND	200	15	0	0.75	105690.52	110150.66	0.844	453.293	110368.52	0.885	334.960	108655.14	0.561	266.478	106245.40	0.105	21.427
URAND	200	20	0	0.75	102022.32	114611.87	2.468	779.527	106564.43	0.890	489.251	103537.35	0.297	343.619	**102022.32**	0.000	97.989
URAND	300	3	0	0.75	308765.08	**308765.08**	0.000	201.467	**308765.08**	0.000	138.529	**308765.08**	0.000	75.813	**308765.08**	0.000	0.921
URAND	300	4	0	0.75	293636.81	295457.36	0.124	240.097	297175.13	0.241	155.030	**293636.81**	0.000	90.276	**293636.81**	0.000	1.894
URAND	300	5	0	0.75	282116.88	283555.68	0.102	389.928	283442.83	0.094	184.982	284134.02	0.143	103.142	282130.99	0.001	3.587
URAND	300	10	0	0.75	251393.30	255905.81	0.359	459.029	266514.61	1.203	232.382	258897.39	0.597	130.707	**251393.30**	0.000	15.648
URAND	300	15	0	0.75	236781.77	261489.95	2.087	553.247	253912.93	1.447	308.426	239196.94	0.204	298.327	**236781.77**	0.000	36.217
URAND	300	20	0	0.75	228005.19	256540.04	2.503	867.467	271337.58	3.801	473.204	236384.38	0.735	444.813	231482.27	0.305	163.112
URAND	400	3	0	0.75	543717.32	550078.81	0.234	324.929	544008.21	0.011	269.9614	**543717.32**	0.000	60.743	**543717.32**	0.000	1.624
URAND	400	4	0	0.75	519217.48	524980.79	0.222	486.236	520203.99	0.038	255.3403	519762.66	0.021	286.371	**519217.48**	0.000	2.802
URAND	400	5	0	0.75	501421.52	550661.11	1.964	596.687	506109.81	0.187	298.1967	502875.64	0.058	299.474	504530.33	0.124	15.637
URAND	400	10	0	0.75	446361.10	499433.43	2.378	841.920	469839.69	1.052	353.3536	460533.06	0.635	345.270	**446361.10**	0.000	69.101
URAND	400	15	0	0.75	422284.78	475936.06	2.541	948.512	474373.61	2.467	504.0618	449796.63	1.303	568.644	**422263.67**	-0.001	198.290
URAND	400	20	0	0.75	407110.51	470212.64	3.100	1213.418	464126.34	2.801	972.7497	448656.14	2.041	888.167	409308.91	0.108	294.750

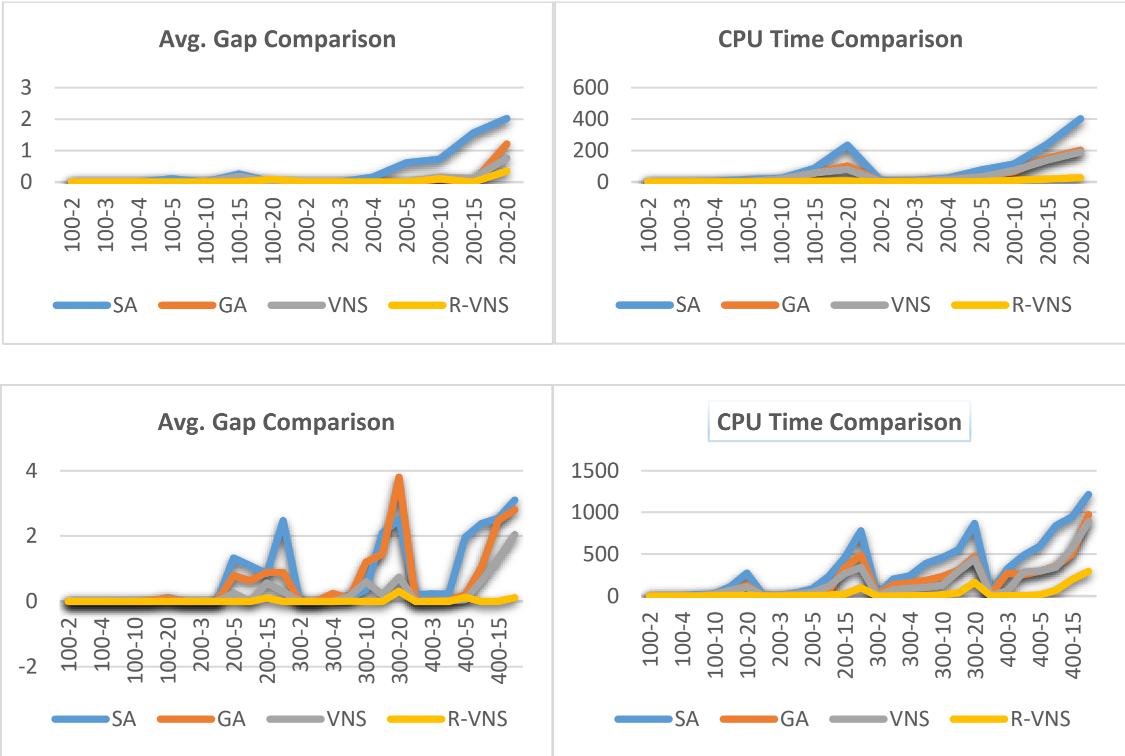


Figure 5.10. Average gap and CPU time comparisons for large size complete USApHMP for URAND and AP data sets

5.3.4. Small Size Incomplete Network Structure Instances

In this section we analyzed small size incomplete hub network instances to compare implemented meta-heuristics and CPLEX. In the hub location literature, small-size problems are represented by network structures of 10 to 25 nodes. In addition, the problem variant is extended by considering different number of hubs, hub connections, and economies of scale. For this, ap, cab and tr datasets are considered in the experimental results. Solution times and solution quality (best objective function value) are taken into account to test the performance of meta-heuristics. For AP, CAB and TR datasets, optimal solutions can be obtained up to a maximum of 25 nodes in incomplete structure. The average solution times of each meta-heuristic approach and the average standard deviations from the optimal solution are as shown in the figure. The algorithm with the largest standard deviation value is the SA algorithm with 0.23. Although VNS and GA algorithms have close standard deviation values, VNS gives slightly better results than GA. The R-GVNS algorithm, on the other hand, has the lowest deviation values. It is also superior to other algorithms in terms of solution time. The SA algorithm converged to the optimal with a gap of 0.21% and 0.15% in cases 25-4-4 and 25-8-12, respectively, but could not obtain the optimal result. GA, on the other hand, achieved a near-optimal result with 0.18% in 25-8-12 cases. The VNS algorithm, on the other hand, obtains the optimal solution for all small-sized incomplete p-hub median problems. For small datasets of up to 25 nodes, algorithms integrated into incomplete problems generally give good results. Considering the difficulty of solving incomplete hub location problems, the performance of the proposed meta-heuristic approaches and the developed R-GVNS algorithm is quite high. Considering the CPLEX solution time, the R-GVNS algorithm obtains the optimal solution in an average of 1123 times faster. It works very efficiently especially when the network size is more than 25 nodes and the number of hubs is increased. In the AP and TR datasets, the performance levels of the algorithms are similar in terms of both solution time and standard deviation metrics. In the CAB and TR datasets, the standard deviation value of the R-GVNS algorithm from the best solution was 0.09, lower than the AP datasets. As mentioned earlier, it is difficult to determine the nodes in the optimal solution, as the separation of node characteristics in AP datasets is quite unclear. In the figure and figure, data on the mean solution times and standard deviations of the algorithms run for the CAB and TR data sets are included. Unlike the AP dataset, the deviation values of the best solutions from the optimal solution are 0 in the TR and CAB data sets. In other words, all algorithms obtained the optimal solution at least once in 20 executions. For small-sized incomplete p-hub median problem variations, the operating performance of the algorithms is acceptable. They can solve small-sized problems in a very short time, where average CPLEX solution times are in the 150-200 second ban

Table 5.10. Small Size Inomplete USAp-HMP Solutions and Comparisons for AP data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
ap	10	2	1	0.2	opt	56820888	1.01	0	0.00	0	0.00	0	0.00	0	0.000
ap	10	2	1	0.4	opt	61509402	2.18	0	0.00	0	0.01	0	0.01	0	0.004
ap	10	2	1	0.6	opt	65814135	2.37	0	0.00	0	0.00	0	0.00	0	0.000
ap	10	2	1	0.8	opt	69625203	3.14	0	0.00	0	0.01	0	0.01	0	0.004
ap	15	2	1	0.2	opt	60730465	3.13	0	0.00	0	0.00	0	0.00	0	0.012
ap	15	2	1	0.4	opt	65441825	3.64	0	0.00	0	0.01	0	0.06	0	0.004
ap	15	2	1	0.6	opt	70153185	3.76	0	0.01	0	0.01	0	0.01	0	0.008
ap	15	2	1	0.8	opt	74764746	11.34	0	0.01	0	0.02	0	0.02	0	0.012
ap	15	4	5	0.2	opt	46764265	11.93	0	0.02	0	0.02	0	0.06	0	0.008
ap	15	4	5	0.4	opt	53260885	15.50	0	0.02	0	0.04	0	0.03	0	0.008
ap	15	4	5	0.6	opt	59757504	21.98	0	0.03	0	0.03	0	0.02	0	0.012
ap	15	4	5	0.8	opt	66244820	24.63	0	0.09	0	0.06	0	0.04	0	0.013
ap	20	2	1	0.2	opt	63339209	15.37	0	0.02	0	0.01	0	0.01	0	0.004
ap	20	2	1	0.4	opt	68166157	21.57	0	0.03	0	0.03	0	0.07	0	0.016
ap	20	2	1	0.6	opt	72852695	26.93	0	0.04	0	0.04	0	0.05	0	0.012
ap	20	2	1	0.8	opt	77286131	50.35	0	0.07	0	0.13	0	0.09	0	0.016
ap	20	4	5	0.2	opt	47913751	43.65	0	0.05	0	0.03	0	0.03	0	0.008
ap	20	4	5	0.4	opt	56086986	69.35	0	0.09	0	0.14	0	0.12	0	0.020
ap	20	4	5	0.6	opt	62770046	189.07	0	0.28	0	0.38	0	0.34	0	0.020
ap	20	4	5	0.8	opt	69139329	269.57	0	0.39	0	0.39	0	0.35	0	0.016
ap	25	2	1	0.2	opt	65444350	44.91	0	0.06	0	0.07	0	0.03	0	0.008
ap	25	2	1	0.4	opt	70030641	38.71	0	0.05	0	0.04	0	0.04	0	0.008
ap	25	2	1	0.6	opt	74530345	27.50	0	0.04	0	0.03	0	0.08	0	0.012
ap	25	2	1	0.8	opt	78888379	56.79	0	0.08	0	0.06	0	0.03	0	0.004
ap	25	4	5	0.2	opt	49958105	130.43	0	0.18	0	0.22	0	0.15	0	0.016
ap	25	4	5	0.4	opt	57407634	578.80	0.21	0.70	0	0.31	0	0.48	0	0.020
ap	25	4	5	0.6	opt	63763520	571.54	0	0.74	0	0.94	0	0.56	0	0.016
ap	25	4	5	0.8	opt	70119406	529.70	0	0.78	0	0.66	0	0.42	0	0.016
ap	25	8	12	0.2	opt	34337868	315.85	0.15	0.93	0	0.70	0	0.92	0	0.024
ap	25	8	12	0.4	opt	44311456	441.73	0	0.72	0	0.92	0	0.68	0	0.032
ap	25	8	12	0.6	opt	53674782	481.18	0	0.81	0.18	1.02	0	0.74	0	0.036
ap	25	8	12	0.8	opt	62597703	693.82	0	0.98	0	0.39	0	0.50	0	0.040

Table 5.11. Small Size Incomplete USAp-HMP Solutions and Comparisons for CAB data sets

					CPLEX			SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
cab	10	2	1	0.2	opt	16229500	0.69	0	0.008	0	0.005	0	0.005	0	0.000
cab	10	2	1	0.4	opt	18668515	0.80	0	0.004	0	0.005	0	0.002	0	0.000
cab	10	2	1	0.6	opt	21107529	0.60	0	0.005	0	0.001	0	0.003	0	0.000
cab	10	2	1	0.8	opt	23546544	0.69	0	0.005	0	0.002	0	0.009	0	0.004
cab	15	2	1	0.2	opt	56649272	2.58	0	0.023	0	0.012	0	0.003	0	0.000
cab	15	2	1	0.4	opt	64492041	1.77	0	0.031	0	0.036	0	0.055	0	0.004
cab	15	2	1	0.6	opt	72152968	4.22	0	0.014	0	0.030	0	0.011	0	0.004
cab	15	2	1	0.8	opt	78861493	3.99	0	0.046	0	0.077	0	0.018	0	0.004
cab	15	4	5	0.2	opt	29351869	5.37	0	0.047	0	0.019	0	0.067	0	0.004
cab	15	4	5	0.4	opt	41332600	8.89	0	0.039	0	0.021	0	0.009	0	0.004
cab	15	4	5	0.6	opt	53101450	7.91	0	0.083	0	0.090	0	0.021	0	0.008
cab	15	4	5	0.8	opt	64654950	17.84	0	0.050	0	0.060	0	0.024	0	0.004
cab	20	2	1	0.2	opt	1.05E+08	10.46	0	0.052	0	0.115	0	0.058	0	0.000
cab	20	2	1	0.4	opt	1.17E+08	12.28	0	0.098	0	0.146	0	0.028	0	0.004
cab	20	2	1	0.6	opt	1.28E+08	12.39	0	0.066	0	0.091	0	0.024	0	0.004
cab	20	2	1	0.8	opt	1.39E+08	8.75	0	0.061	0	0.066	0	0.052	0	0.004
cab	20	4	5	0.2	opt	67429582	31.75	0	0.078	0	0.045	0	0.051	0	0.008
cab	20	4	5	0.4	opt	85212850	94.04	0	0.093	0	0.085	0	0.029	0	0.004
cab	20	4	5	0.6	opt	1.03E+08	82.75	0	0.158	0	0.185	0	0.300	0	0.012
cab	20	4	5	0.8	opt	1.19E+08	179.83	0	0.463	0	0.020	0	0.125	0	0.016
cab	25	2	1	0.2	opt	4.52E+08	66.03	0	0.098	0	0.100	0	0.092	0	0.004
cab	25	2	1	0.4	opt	4.75E+08	67.46	0	0.319	0	0.105	0	0.028	0	0.004
cab	25	2	1	0.6	opt	4.98E+08	56.47	0	0.541	0	0.183	0	0.135	0	0.008
cab	25	2	1	0.8	opt	5.2E+08	78.46	0	0.882	0	0.340	0	0.146	0	0.012
cab	25	4	5	0.2	opt	2.39E+08	54.17	0	0.131	0	0.133	0	0.120	0	0.016
cab	25	4	5	0.4	opt	3.05E+08	118.05	0	0.280	0	0.201	0	0.163	0	0.012
cab	25	4	5	0.6	opt	3.7E+08	453.42	0	0.495	0	0.212	0	0.110	0	0.024
cab	25	4	5	0.8	opt	4.33E+08	1081.31	0	0.471	0	0.198	0	0.130	0	0.024
cab	25	8	12	0.2	opt	1.42E+08	196.37	0	0.127	0	0.055	0	0.167	0	0.020
cab	25	8	12	0.4	opt	2.19E+08	234.23	0	0.152	0	0.014	0	0.139	0	0.028
cab	25	8	12	0.6	opt	2.95E+08	318.15	0	0.564	0	0.456	0	0.387	0	0.032
cab	25	8	12	0.8	opt	3.67E+08	1027.89	0	0.506	0	0.391	0	0.356	0	0.036

Table 5.12. Small Size Incomplete USAp-HMP Solutions and Comparisons for TR data sets

						CPLEX		SA		GA		VNS		R-GVNS	
Data Set	N. Size	p	q	α	Opt.	Obj.	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time	Gap (%)	CPU Time
tr	10	2	1	0.2	opt	9.17E+08	0.146125	0	0.002	0	0.000	0	0.000	0	0.001
tr	10	2	1	0.4	opt	9.68E+08	0.442335	0	0.005	0	0.003	0	0.001	0	0.014
tr	10	2	1	0.6	opt	1.01E+09	1.504438	0	0.002	0	0.002	0	0.001	0	0.003
tr	10	2	1	0.8	opt	1.05E+09	1.189916	0	0.002	0	0.001	0	0.001	0	0.006
tr	15	2	1	0.2	opt	1.33E+09	5.849345	0	0.002	0	0.001	0	0.014	0	0.002
tr	15	2	1	0.4	opt	1.42E+09	8.022944	0	0.015	0	0.011	0	0.017	0	0.067
tr	15	2	1	0.6	opt	1.48E+09	5.916337	0	0.002	0	0.005	0	0.009	0	0.013
tr	15	2	1	0.8	opt	1.54E+09	7.482304	0	0.015	0	0.007	0	0.015	0	0.013
tr	15	4	5	0.2	opt	7.7E+08	10.06868	0	0.029	0	0.008	0	0.011	0	0.070
tr	15	4	5	0.4	opt	9.57E+08	8.0842	0	0.029	0	0.017	0	0.016	0	0.018
tr	15	4	5	0.6	opt	1.14E+09	11.04742	0	0.017	0	0.014	0	0.001	0	0.027
tr	15	4	5	0.8	opt	1.31E+09	21.27771	0	0.072	0	0.034	0	0.037	0	0.018
tr	20	2	1	0.2	opt	2.49E+09	18.77042	0	0.013	0	0.033	0	0.042	0	0.012
tr	20	2	1	0.4	opt	2.6E+09	19.40578	0	0.006	0	0.017	0	0.027	0	0.104
tr	20	2	1	0.6	opt	2.71E+09	15.92881	0	0.004	0	0.025	0	0.017	0	0.029
tr	20	2	1	0.8	opt	2.82E+09	23.639	0	0.064	0	0.003	0	0.004	0	0.068
tr	20	4	5	0.2	opt	1.57E+09	50.91395	0	0.060	0	0.023	0	0.015	0	0.030
tr	20	4	5	0.4	opt	1.89E+09	62.71584	0	0.021	0	0.137	0	0.057	0	0.093
tr	20	4	5	0.6	opt	2.18E+09	88.65559	0	0.052	0	0.146	0	0.029	0	0.454
tr	20	4	5	0.8	opt	2.45E+09	75.4128	0	0.123	0	0.144	0	0.106	0	0.210
tr	25	2	1	0.2	opt	3.96E+09	0.68371	0	0.041	0	0.047	0	0.027	0	0.038
tr	25	2	1	0.4	opt	4.37E+09	1.556987	0	0.093	0	0.059	0	0.035	0	0.029
tr	25	2	1	0.6	opt	4.72E+09	6.819088	0	0.042	0	0.046	0	0.038	0	0.076
tr	25	2	1	0.8	opt	5.06E+09	13.60224	0	0.212	0	0.120	0	0.103	0	0.004
tr	25	4	5	0.2	opt	2.61E+09	207.7769	0	0.331	0	0.237	0	0.214	0	0.130
tr	25	4	5	0.4	opt	3.3E+09	813.0506	0	0.159	0	0.163	0	0.491	0	0.405
tr	25	4	5	0.6	opt	3.9E+09	1002.749	0	0.909	0	1.020	0	0.808	0	0.535
tr	25	4	5	0.8	opt	4.46E+09	1357.37	0	1.171	0	1.042	0	0.960	0	0.388
tr	25	8	12	0.2	opt	1.63E+09	404.2831	0	0.393	0	0.124	0	0.222	0	0.490
tr	25	8	12	0.4	opt	2.4E+09	691.8419	0	1.017	0	0.533	0	0.718	0	1.022
tr	25	8	12	0.6	opt	3.15E+09	626.4491	0	0.618	0	0.466	0	0.513	0	1.333
tr	25	8	12	0.8	opt	3.86E+09	789.6327	0	0.983	0	0.946	0	0.633	0	0.716

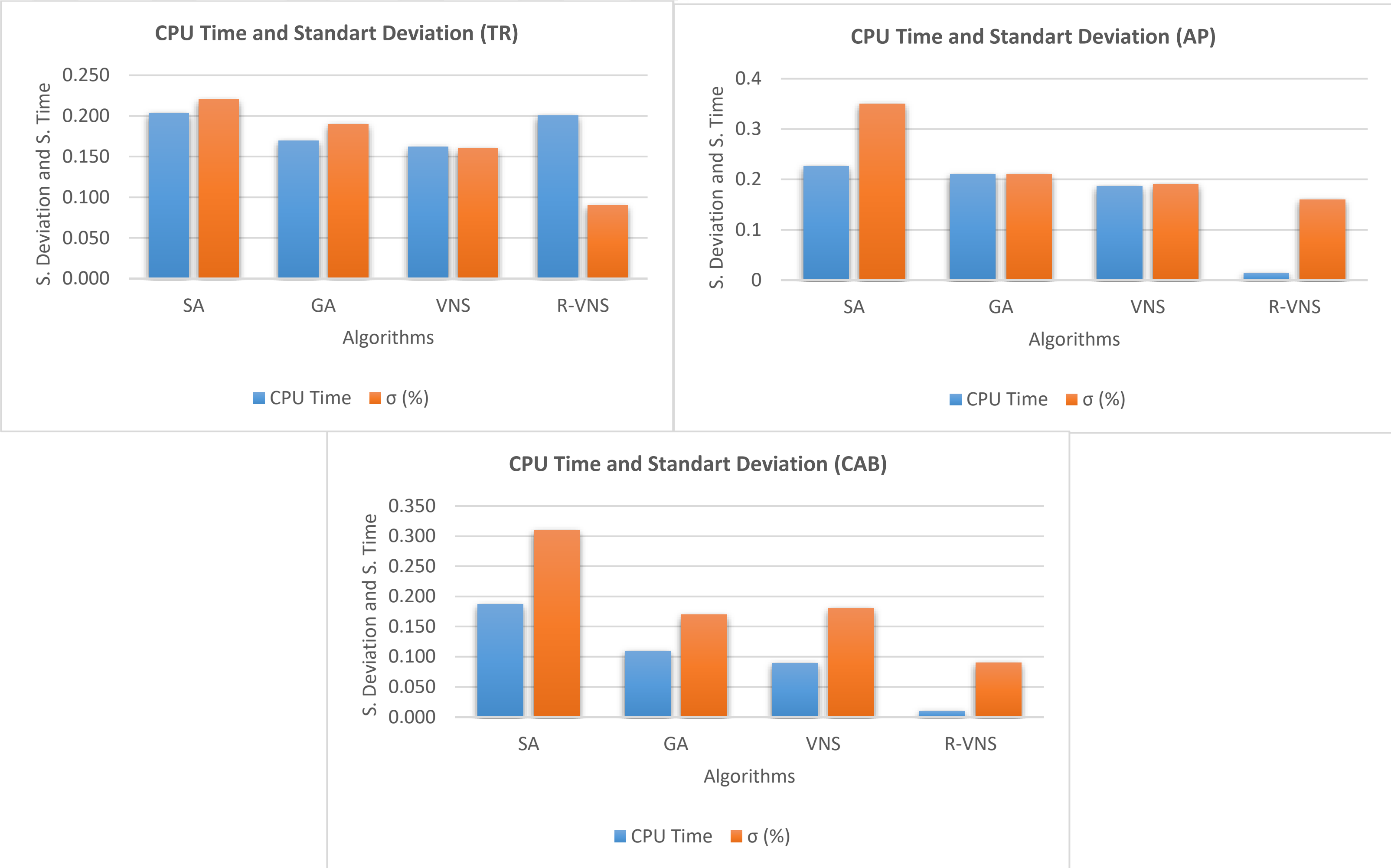


Figure 5.11. Standard deviation comparison for small size incomplete USAp-HMP

5.3.5. Medium Size Incomplete Network Structure Instances

In this section we report the results of our experimental evaluation of medium size incomplete problem implementations. For the AP, CAB and TR datasets, the proposed algorihms using the AP, TR and CAB datasets are tested and compared in this section, where networks of 50 nodes are considered. The R-GVNS algorithm is superior to other approaches in obtaining the best results. At the same time, when the solution times are compared, it can be seen that the R-GVNS algorithm achieves best results in a very short time. However, it gives good results in GVNS and GA, especially in problem variants where the number of hubs is low. When the deviation values from the optimal solutions obtained in the hub location problems with complete feature are examined, it can be seen that the R-GVNS algorithm has the minimum average deviation value. In this section, the best result of the R-GVNS algorithm among the other three algorithms is given as a percentage of improvement value. In these tables % imp. highlighted in the column. In this framework, the R-GVNS algorithm achieves the best results in three test problems in TR dataset, 2 in AP dataset, and 5 test problems in CAB dataset. In terms of solution times, it provides quality results in 5 to 10 times shorter time compared to other algorithms.

Table 5.13. Medium Size Incomplete USAp-HMP Solutions and Comparisons for TR data sets

					CPLEX			SA			GA			VNS			R-VNS		
Data	N. Size	p	q	alpha	Opt.	Obj.		SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	R-VNS-Best	Improvement	CPU Time
tr	50	4	4	0.8	-	-	86000	**2.94E+10**	0.015	0.226	**2.94E+10**	0.015	0.242	**2.94E+10**	0.015	0.123	**2.94E+10**	0.000	0.030
tr	50	4	5	0.8	-	-	86000	2.91E+10	0.006	0.324	**2.91E+10**	0.005	0.440	**2.91E+10**	0.005	0.119	**2.91E+10**	0.000	0.024
tr	50	8	8	0.8	-	-	86000	2.55E+10	0.013	0.571	2.55E+10	0.013	0.332	2.55E+10	0.013	0.476	**2.55E+10**	**0.001**	0.093
tr	50	8	10	0.8	-	-	86000	2.52E+10	0.003	3.033	**2.52E+10**	0.002	1.478	2.52E+10	0.003	1.285	2.52E+10	0.000	0.218
tr	50	16	20	0.8	-	-	86000	2.31E+10	0.019	1.766	2.31E+10	0.019	1.476	2.31E+10	0.019	1.530	**2.31E+10**	**0.002**	0.221
tr	50	16	40	0.8	-	-	86000	2.29E+10	0.011	2.418	2.29E+10	0.010	2.547	2.29E+10	0.010	0.868	**2.29E+10**	**0.001**	0.316
tr	50	16	90	0.8	-	-	86000	2.28E+10	0.006	5.874	**2.28E+10**	0.005	2.788	2.28E+10	0.006	3.918	2.28E+10	0.000	0.447
tr	50	16	100	0.8	-	-	86000	2.27E+10	0.001	3.137	**2.27E+10**	0.000	1.581	**2.27E+10**	0.000	2.758	**2.27E+10**	0.000	0.403

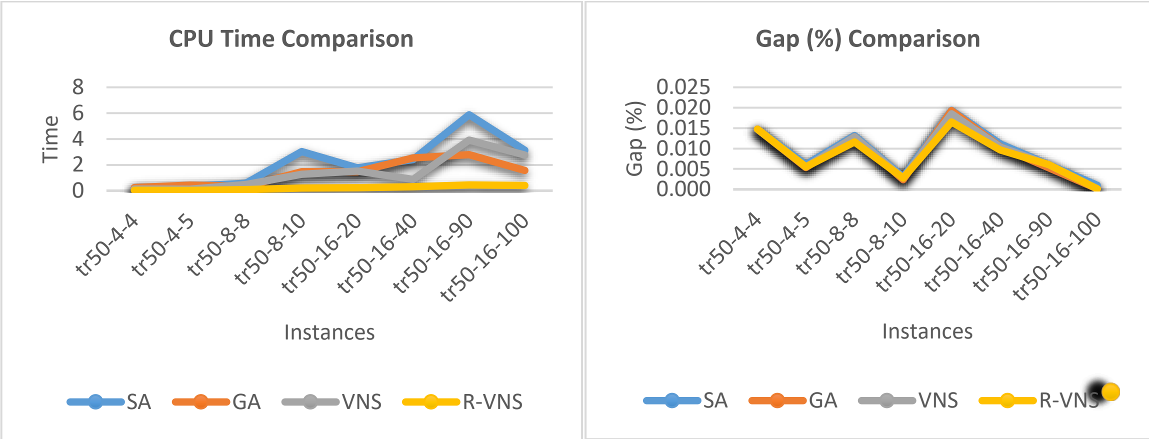


Figure 5.12. Average gap and CPU time comparisons for medium size incomplete USAp-HMP for TR data sets

Table 5.14. Medium Size Incomplete USAp-HMP Solutions and Comparisons for AP data sets

					CPLEX			SA			GA			VNS			R-GVNS		
Data	N. Size	p	q	α	Opt.	Obj.	CPU Time	SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	R-GVNS-Best	Imp. (%)	CPU Time
ap	50	3	2	0.75	-	-	86000	163102	0.022	0.256	**162710**	0.022	0.201	**162710**	0.022	0.098	**162710**	0.000	0.008
ap	50	4	4	0.75	-	-	86000	145737	0.016	0.321	**145374**	0.016	0.255	**145374**	0.016	0.165	**145374**	0.000	0.096
ap	50	4	5	0.75	-	-	86000	143671	0.002	0.555	**143378**	0.000	0.438	**143378**	0.000	0.403	**143378**	0.000	0.124
ap	50	5	4	0.75	-	-	86000	139716	0.052	1.684	**134553**	0.016	1.267	136536	0.030	1.019	**136000**	**0.394**	0.257
ap	50	5	5	0.75	-	-	86000	137523	0.038	1.894	136879	0.034	1.642	**136441**	0.031	1.688	**134476**	**1.461**	0.275
ap	50	5	6	0.75	-	-	86000	133386	0.007	2.600	133342	0.007	2.351	133342	0.007	2.375	**132779**	**0.424**	0.437
ap	50	5	7	0.75	-	-	86000	133238	0.006	4.281	132919	0.004	3.027	132712	0.003	3.121	**132440**	**0.205**	0.916
ap	50	5	8	0.75	-	-	86000	132373	0.000	3.562	132373	0.000	3.144	**132368**	0.000	3.103	132652	-0.214	0.834
ap	50	5	9	0.75	-	-	86000	**132368**	0.000	3.945	**132368**	0.000	3.519	**132368**	0.000	3.461	**132368**	0.000	0.839

Table 5.15. Medium Size Incomplete USAp-HMP Solutions and Comparisons for CAB data sets

					CPLEX			SA			GA			VNS			R-GVNS		
Data	N. Size	p	q	α	Opt.	Obj.	Time	SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	R-GVNS-Best	Imp (%)	CPU Time
cab	50	4	4	0.75	-	-	86000	1827262012.18	0.023	0.252	**1804423061**	0.010	0.048	1811897152	0.014	0.090	1810119008	-0.314	0.006
cab	50	4	5	0.75	-	-	86000	1803421447.96	0.009	0.446	1790077072	0.002	0.280	1792241463	0.013	0.149	**1774603435**	**0.994**	0.025
cab	50	8	8	0.75	-	-	86000	1532763614.28	0.026	0.370	1534925129	0.028	0.344	1530842184	0.025	0.468	**1528571136**	**0.149**	0.133
cab	50	8	10	0.75	-	-	86000	1514731104.92	0.014	2.513	1517776978	0.016	1.239	1506946111	0.019	1.174	**1491969842**	**1.004**	0.267
cab	50	16	20	0.75	-	-	86000	1334720603.81	0.037	1.467	**1319868586**	0.025	1.117	1323275256	0.028	1.658	**1317929676**	0.000	0.291
cab	50	16	40	0.75	-	-	86000	1315749647.36	0.022	3.628	1308523632	0.017	0.838	1307828463	0.016	3.337	**1296956604**	**0.838**	0.515
cab	50	16	90	0.75	-	-	86000	1310649001.96	0.018	4.017	**1296693055**	0.007	4.354	1298817834	0.009	3.091	1297991568	-0.100	1.632
cab	50	16	100	0.75	-	-	86000	1295481751.38	0.006	5.286	1291938505	0.004	3.784	1289807205	0.002	2.067	**1289263171**	**0.042**	1.022

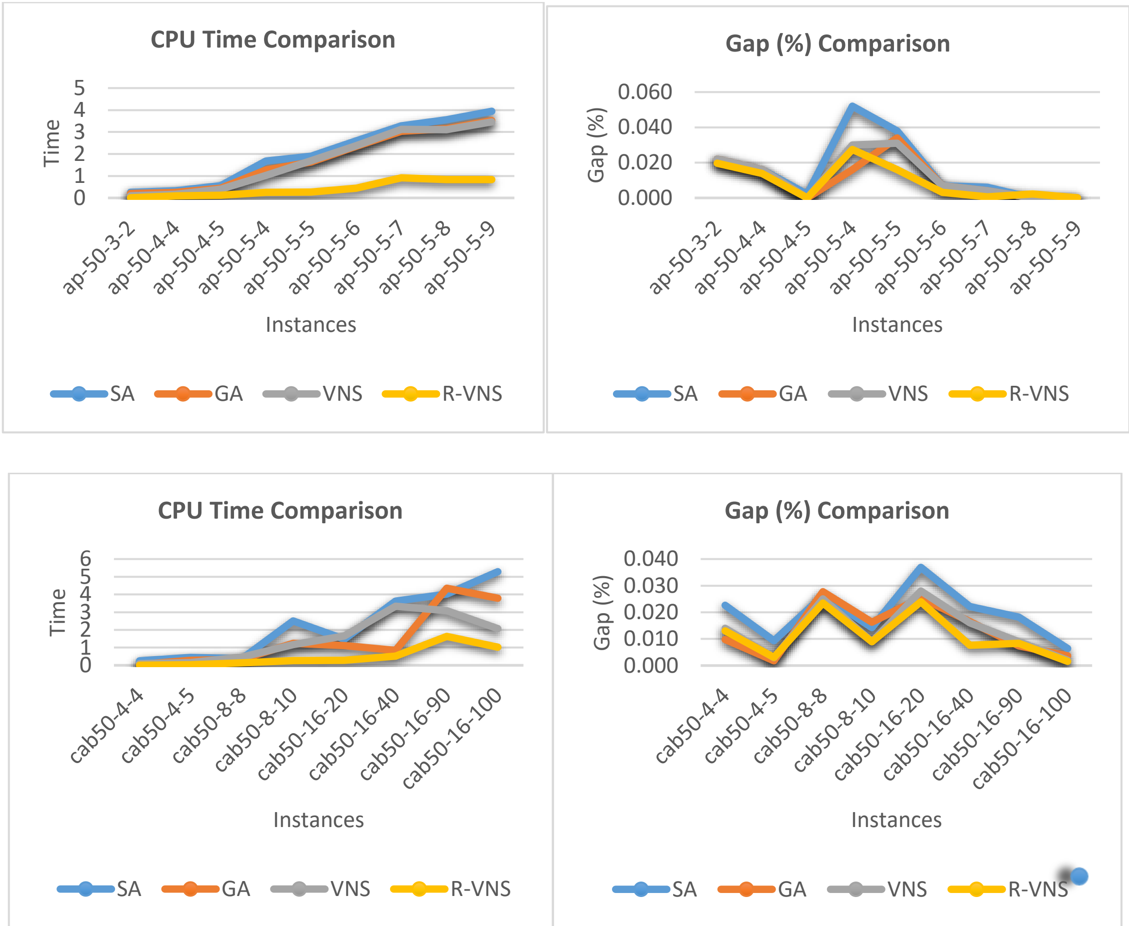


Figure 5.13. Average gap and CPU time comparisons for medium size incomplete USAp-HMP for AP and CAB data sets

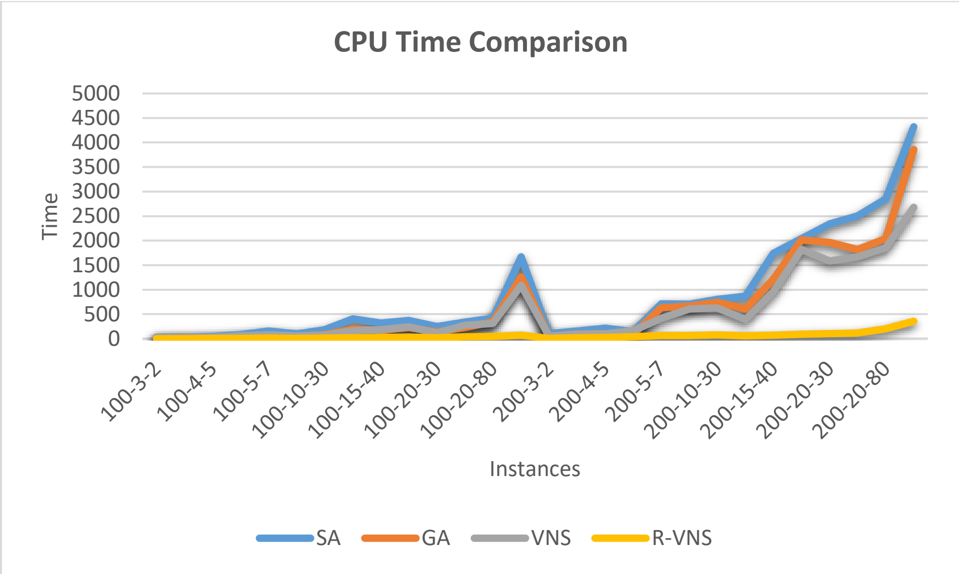
5.3.6. Large Size Incomplete Network Structure Instances

Large size hub location problems are tested with two different data sets. The first of these is the AP-100 and AP-200 datasets. Another is the URAND-100, URAND-200, URAND-300 and URAND-400 datasets. Incomplete solutions are tried to be obtained for these problems, where the exact solutions are not known even for the complete hub location problems. As with complete hub location problems, comparisons are presented with the table 5.16 and the two-performance metrics in the table. The first of these is the solution time. Another is the percentage gap of the best solution obtained after 20 runs from the best-known solution. Since there is no solution method developed in the literature for incomplete p-hub median problems and there are no objective function values presented, percentage gap comparisons are made with the best-known solutions of the complete problems. For instance, a complete problem solution with 100 nodes and 5 hub locations is compared with solutions with the same features with incomplete hub connections. The reason why complete solutions are taken as lower bound here is that incomplete solutions can never obtained the objective function value lower than the complete objective function value. In this context, incomplete solutions always have lower cost values than complete solutions. These values may converge or be equal to complete solutions as the number of connections between hubs increases.

Table 5.16 shows the incomplete p-hub median results of the AP-100 and AP-200 datasets. Different hub connection variations are tested between 3-20 as the number of hubs. As in the previous sections, each algorithm is run 20 times until the maximum number of hubs is five. However, due to the time complexity, the algorithms are run 10 times in the instances of the number of hubs greater than five. In this section where 28 different problem variants are tested, the R-GVNS algorithm achieves the best result in 24 samples. The SA algorithm, on the other hand, did not achieve the best result in any sample. VNS and GA give the best results for 11 and 3 problem variants, respectively. When evaluated in terms of solution times, the R-GVNS algorithm has a very superior performance compared to the others. Information about CPU time comparisons is in the Figure. As can be seen from the figure, the increase in p values has a significant effect on the solution time. It is seen that the solution times of the three proposed meta-heuristics, SA, GA, and VNS algorithms, increase dramatically in 200-node networks. The proposed R-GVNS algorithm, on the other hand, provides good solutions with more acceptable solution time increments. When the cost differences between the incomplete best solutions presented by the algorithms and the complete best-known solutions are examined, the GA and VNS algorithms have close results. However, as the problem size increases, the results of the VNS algorithm are slightly better than GA. The efficiency of the R-GVNS algorithm can be seen clearly. Our results also provide reference solutions for incomplete p-hub median problems for AP dataset.

Table 5.16. Large Size Incomplete USAp-HMP Solutions and Comparisons for AP data sets

					CPLEX	SA			GA			VNS			R-GVNS		
Data	N. Size	p	q	α	Obj.	SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	R-GVNS-Best	Gap (%)	CPU Time
ap	100	3	2	0.75	-	180796	0.318	27.47	180754	0.295	16.12	**180708**	0.269	15.06	**180708**	0.269	1.15
ap	100	4	4	0.75	-	147962	1.416	35.69	147807	1.310	24.40	**147763**	1.280	21.65	**147763**	1.280	3.65
ap	100	4	5	0.75	-	147032	0.779	48.76	146892	0.683	26.06	146828	0.639	29.66	**146094**	0.135	4.12
ap	100	5	5	0.75	-	140279	2.446	87.53	139671	2.002	48.85	139671	2.002	56.73	**139531**	1.900	8.62
ap	100	5	7	0.75	-	137674	0.543	154.60	137626	0.509	75.55	137569	0.467	75.89	**137204**	0.200	10.38
ap	100	10	15	0.75	-	119361	0.493	100.67	119334	0.471	53.68	**119128**	0.297	57.00	119605	0.698	9.03
ap	100	10	30	0.75	-	118917	0.119	183.68	118779	0.004	83.63	**118779**	0.004	74.30	119017	0.204	14.51
ap	100	15	20	0.75	-	81486	0.985	405.58	81434	0.921	193.50	81357	0.826	164.23	**81113**	0.523	21.68
ap	100	15	40	0.75	-	81063	0.461	321.80	81000	0.383	169.86	80939	0.307	182.04	**80858**	0.207	25.74
ap	100	15	90	0.75	-	81026	0.415	375.08	80934	0.302	230.61	80925	0.290	242.85	**80763**	0.090	36.11
ap	100	20	30	0.75	-	82208	1.880	247.25	82003	1.626	138.87	81897	1.495	117.65	**81406**	0.886	20.05
ap	100	20	50	0.75	-	81472	0.968	338.60	81329	0.791	228.96	81268	0.715	275.79	**81105**	0.514	39.86
ap	100	20	80	0.75	-	81134	0.549	415.92	81059	0.456	348.90	81020	0.408	315.70	**80939**	0.308	46.66
ap	100	20	150	0.75	-	81474	0.971	1671.14	81284	0.735	1256.30	81284	0.735	1086.70	**80877**	0.231	69.87
ap	200	3	2	0.75	-	163306	0.257	102.57	163286	0.245	49.42	**163239**	0.216	52.83	**163239**	0.216	10.10
ap	200	4	4	0.75	-	150077	1.563	157.29	149902	1.445	98.03	**149865**	1.420	89.59	**149865**	1.420	19.63
ap	200	4	5	0.75	-	148337	0.386	214.34	148342	0.348	101.79	148285	0.350	98.51	**147988**	0.150	25.34
ap	200	5	5	0.75	-	149087	2.547	141.26	**148423**	2.090	94.12	**148423**	2.090	165.39	**148423**	2.090	38.07
ap	200	5	7	0.75	-	146655	0.874	710.13	146195	0.557	631.25	146117	0.504	415.65	**145928**	0.374	64.88
ap	200	10	15	0.75	-	119523	2.467	703.36	117729	1.329	667.57	117638	0.851	603.68	**117520**	0.750	63.94
ap	200	10	30	0.75	-	117150	0.433	800.25	**116992**	0.297	518.95	**116992**	0.297	624.57	**116992**	0.297	79.12
ap	200	15	20	0.75	-	105532	1.751	865.68	104617	1.268	409.67	**104363**	0.624	391.83	**104363**	0.624	58.38
ap	200	15	40	0.75	-	104323	0.585	1742.52	104138	0.407	940.08	**103858**	0.137	980.17	104273	0.537	71.67
ap	200	15	90	0.75	-	104160	0.428	2041.23	**103803**	0.084	1311.81	**103803**	0.084	1821.94	103813	0.094	94.23
ap	200	20	30	0.75	-	95053	1.438	2340.49	94926	1.303	1400.09	94776	1.143	1581.62	**94587**	0.941	102.14
ap	200	20	50	0.75	-	94852	1.224	2502.22	94906	1.281	1520.95	94587	0.941	1668.76	**94303**	0.638	115.27
ap	200	20	80	0.75	-	96342	2.814	2851.63	95341	1.745	1642.96	95167	1.559	1851.47	**93938**	0.248	201.01
ap	200	20	150	0.75	-	94589	0.943	3600*	94353	0.691	2357.42	94172	0.498	2678.96	**93889**	0.196	356.84



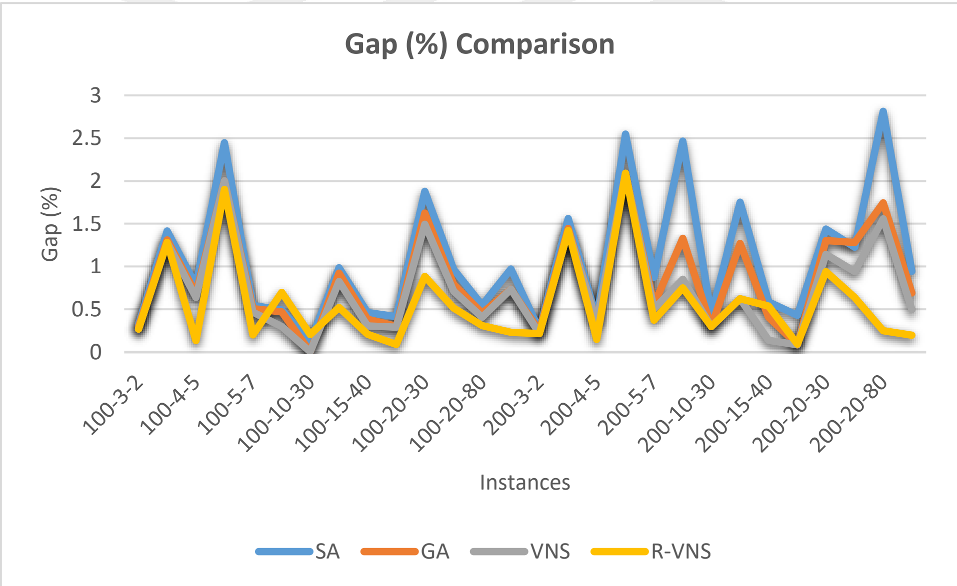


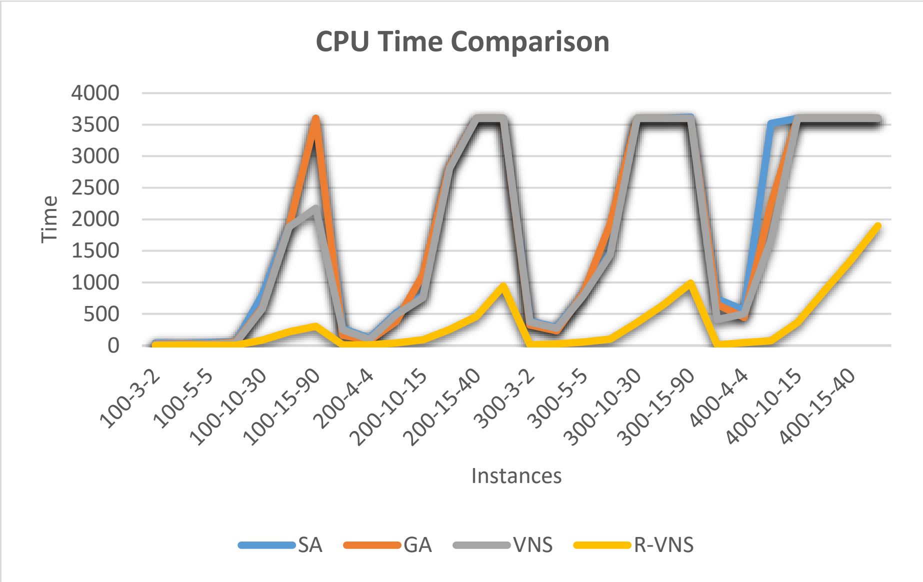
Figure 5.14. Average gap and CPU time comparisons for medium size incomplete USAp-HMP for AP data sets

Table 5.17 contains the results of four different algorithms proposed for URAND datasets. For the URAND data set, the economies of scale coefficient is defined as 0.75 as AP. Unlike the AP, the algorithms are also run for networks with 300 and 400 nodes in the URAND dataset. In addition, running time limit of 3600 seconds is set for each meta-heuristic. Therefore, GA, VNS, and SA algorithms are stuck on this time limit, especially for networks of 400 nodes. The incomplete structure best results are compared with the best results of the complete problems as in other datasets. The R-GVNS algorithm achieves the best results in all tests except the 400-10-15 scenario. On the other hand, GA and VNS are able to achieve the results obtained by R-GVNS in cases where the number of hubs is less. As can be seen in the figure, GA, VNS and SA have close running times for large-size problems in terms of solution time. However, the R-GVNS algorithm has a distinctively shorter resolution time. The average solution times for SA, GA, and VNS were 1828, 1781, and 1678 seconds, respectively, while the average solution time for R-GVNS is 330 seconds. It achieves quality results in approximately 6 times by comparison with other three algorithm solution times. When the percentages of distance from the lower bound (complete problem solutions) value are examined, the superiority of R-GVNS can be seen clearly. The VNS algorithm, on the other hand, converges to the optimal more than SA and GA. While the average percentage gap to the best solutions is 3.181%, 2.86% and 1.94% for SA, GA and VNS, respectively, this value is 1.30% for the R-GVNS algorithm. It is seen that conventional meta-heuristic algorithms may be insufficient, especially in large-sized Np-hard problems. Therefore, new approaches are needed as problem reduction-based approaches. By removing unnecessary decision variables from the problem with problem reduction strategies, both quality results (approximate optimal) can be obtained, and solutions can be reached in a short time.

Figure 5.16 highlights the differences between the average solution times of large-size URAND and AP data sets within the framework of complete and incomplete problems. As can be seen clearly in the Figure 5.16, solution times for incomplete problems are considerably longer than for complete problems. The faint black boxes indicate the difference between the algorithm that finds the solution in the shortest time and the algorithm that gives the solution in the longest time. This difference is very sensitive to changes in the size and topological properties of the problem. In addition, the % gap values showing the differences of the best solution from the lower bound have similar features. Because the difference between percentage gaps are directly related to the problem size and topological features.

Table 5.17. Large Size Incomplete USAp-HMP Solutions and Comparisons for URAND data sets

						SA			GA			VNS			R-GVNS		
Data	N. Size	p	q	alpha	best known	SA-Best	Gap (%)	CPU Time	GA-Best	Gap (%)	CPU Time	VNS-Best	Gap (%)	CPU Time	R-GVNS-Best	Gap (%)	CPU Time
URAND	100	3	2	0.75	-	35031.28	1.443	39.45	**34794.54**	0.758	22.66	**34794.54**	0.758	15.20	**34794.54**	0.758	1.161
URAND	100	4	4	0.75	-	34582.94	6.056	35.46	34465.62	5.696	29.43	33987.93	4.231	18.73	**33584.34**	2.993	3.157
URAND	100	5	5	0.75	-	34109.09	9.648	44.38	34042.32	9.434	26.16	33144.98	6.549	20.59	**32406.23**	4.174	2.860
URAND	100	10	15	0.75	-	27998.40	3.474	60.91	27915.20	3.166	54.13	27650.67	2.189	38.22	**27479.88**	1.558	5.808
URAND	100	10	30	0.75	-	27388.08	1.218	803.13	27306.01	0.915	602.76	27223.62	0.611	582.93	**27168.38**	0.406	85.416
URAND	100	15	40	0.75	-	25723.47	1.239	1905.64	25646.24	0.935	1894.21	25567.57	0.626	1881.08	**25514.83**	0.418	219.890
URAND	100	15	90	0.75	-	25606.03	0.777	3600.00	25520.20	0.439	3600.00	25485.28	0.302	2177.23	**25458.35**	0.196	304.224
URAND	200	3	2	0.75	-	141262.25	1.465	268.56	**140309.70**	0.780	160.31	**140309.70**	0.780	246.56	**140309.70**	0.780	9.986
URAND	200	4	4	0.75	-	139308.02	4.998	117.02	139193.41	4.912	81.89	136839.35	3.137	95.53	**135396.15**	2.050	10.074
URAND	200	5	5	0.75	-	139030.05	9.283	516.78	138980.30	9.244	378.85	135360.48	6.399	487.57	**132121.87**	3.853	40.248
URAND	200	10	15	0.75	-	116445.26	3.471	788.67	116004.94	3.080	1126.24	115073.35	2.252	759.13	**114332.96**	1.594	89.412
URAND	200	10	30	0.75	-	113913.43	1.221	2826.98	113573.39	0.919	2836.47	113228.78	0.613	2808.28	**112998.72**	0.408	251.716
URAND	200	15	40	0.75	-	106994.15	1.233	3600.00	106674.85	0.931	3600.00	106347.17	0.621	3600.00	**106126.39**	0.412	463.451
URAND	200	15	90	0.75	-	106498.65	0.765	3600.00	106166.89	0.451	3600.00	106014.32	0.306	3600.00	**105914.87**	0.212	941.995
URAND	300	3	2	0.75	-	313284.08	1.464	386.15	**311153.24**	0.773	325.17	**311153.24**	0.773	369.74	**311153.24**	0.773	10.176
URAND	300	4	4	0.75	-	309278.29	5.327	287.06	309507.37	5.405	232.68	304090.82	3.560	271.40	**300446.17**	2.319	26.406
URAND	300	5	5	0.75	-	307324.77	8.935	814.95	304511.17	7.938	829.82	298008.13	5.633	793.17	**291519.17**	3.333	53.546
URAND	300	10	15	0.75	-	258779.05	2.938	1466.98	258237.08	2.722	1933.56	255793.89	1.750	1441.56	**254154.11**	1.098	98.978
URAND	300	10	30	0.75	-	254460.74	1.220	3600.00	253702.35	0.919	3600.00	252932.89	0.612	3600.00	**252418.52**	0.408	365.346
URAND	300	15	40	0.75	-	239705.39	1.235	3600.00	238992.91	0.934	3600.00	238258.46	0.624	3600.00	**237766.88**	0.416	650.880
URAND	300	15	90	0.75	-	238504.61	0.728	3621.42	237778.15	0.421	3600.00	237443.96	0.280	3600.00	**237262.01**	0.203	991.257
URAND	400	3	2	0.75	-	551805.21	1.488	734.79	550122.14	1.178	647.97	**546644.67**	0.538	411.57	**546644.67**	0.538	12.359
URAND	400	4	4	0.75	-	544211.13	4.814	561.14	544217.43	4.815	445.53	535104.06	3.060	504.09	**529104.08**	1.904	46.092
URAND	400	5	5	0.75	-	541113.95	7.916	3518.00	540243.84	7.742	2243.29	526997.91	5.101	1671.94	**513675.30**	2.444	78.190
URAND	400	10	15	0.75	-	462055.83	3.516	3600.00	460662.95	3.204	3600.00	**452829.34**	1.449	3600.00	456376.48	2.244	365.143
URAND	400	10	30	0.75	-	451809.20	1.221	3600.00	450462.74	0.919	3600.00	449096.68	0.613	3600.00	**448185.06**	0.409	865.721
URAND	400	15	40	0.75	-	427510.43	1.237	3600.00	426228.77	0.934	3600.00	424921.91	0.624	3600.00	**424052.25**	0.419	1358.854
URAND	400	15	90	0.75	-	425404.41	0.739	3600.00	424289.43	0.475	3600.00	423494.21	0.286	3600.00	**423131.84**	0.201	1896.809



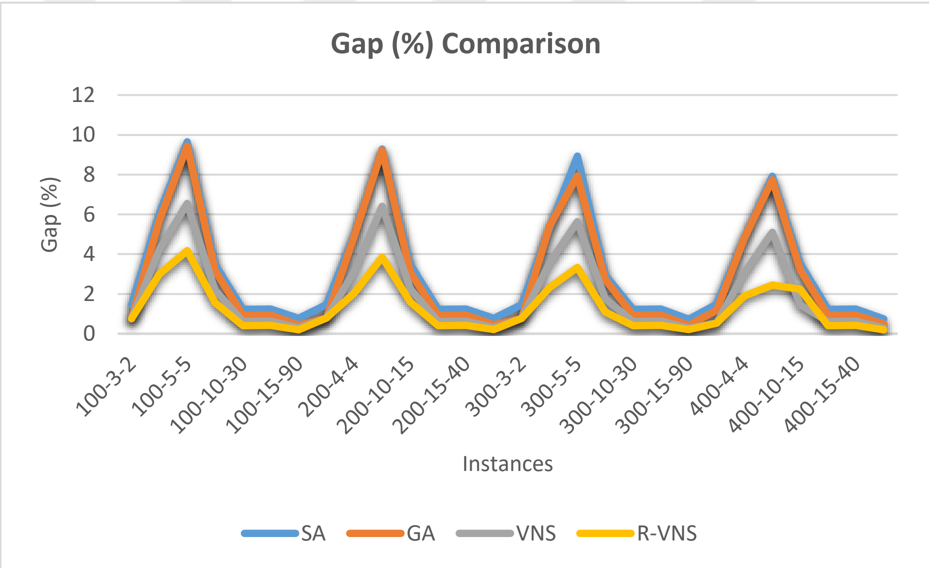
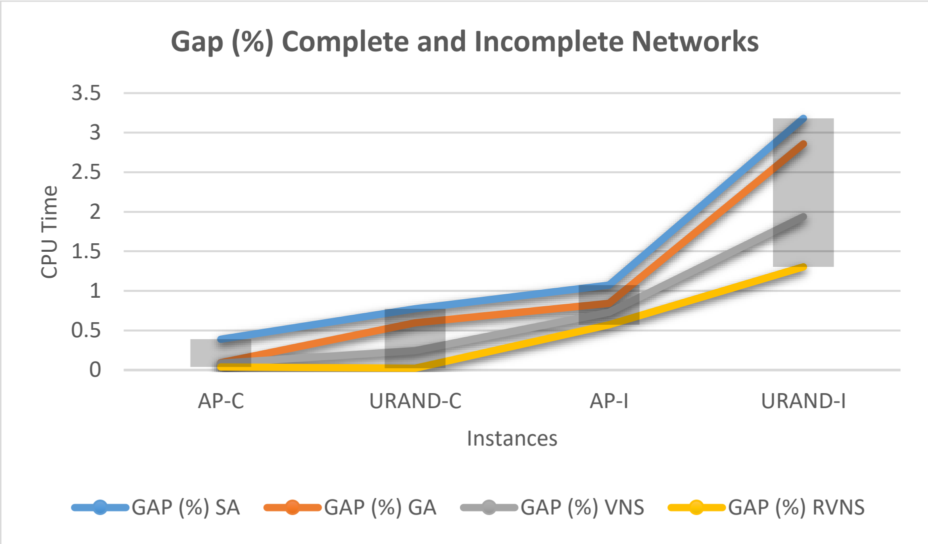


Figure 5.15. Average gap and CPU time comparisons for medium size incomplete USAp-HMP for URAND data sets



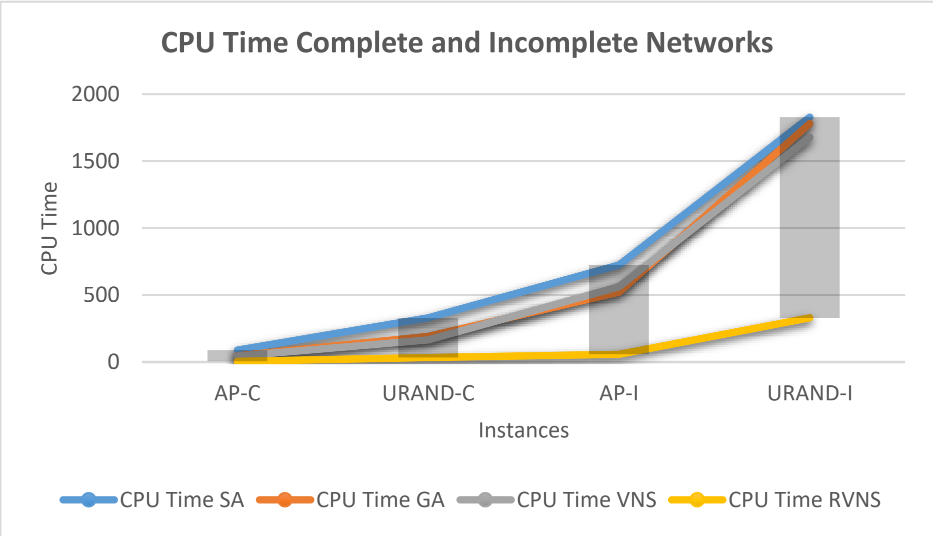


Figure 5.16. CPU time comparison for complete and incomplete hub networks

5.4. Statistical Analysis of Meta-Heuristic Algorithm Performances

Wilcoxon Signed-Rank Test, which we used to analyze whether there is a significant difference between the performances of centrality-based clustering algorithms in the previous sections, is also used in this section to examine the performance differences of the proposed meta-heuristic algorithms. When the meta-heuristic approaches used in solving the complete and incomplete p-hub median problems in the previous section are examined, it can be observed that the R-GVNS algorithm is quite effective. However, Wilcoxon signed-rank test is applied to analyze more deeply whether the R-GVNS algorithm has significant differences from the results of other algorithms in terms of solution performance. The Wilcoxon signed-rank test stands out for non-parametric analysis without the need for any population distribution to compare two matched and related samples. The main purpose of performing this statistical analysis is to examine the solution differences obtained by the two related meta-heuristic methods. The null hypothesis indicates that the proposed methods are not statistically significant among the results. The alternative hypothesis used indicates the existence of a statistically significant difference between the pairs of methods.

SPSS, one of the commercial software, was used for the Wilcoxon signed rank test application and the significance level was defined as $\alpha = 0.05$. In this analysis, p and z values, which are two important variables, are used for the analysis of the results. If the p-value is less than the defined significance level (0.05 for this study), the null hypothesis is false (at 95% confidence level). The z-values are evaluated based on standard values from -1.96 to $+1.96$. If the obtained z value is outside this value range, the null hypothesis is rejected.

Each of the solution approaches considered in the thesis is evaluated with the Wilcoxon sign rank test over two basic performance metrics. The first of these metrics is CPU time, and the other is the best solution results. In the Table5.18 , considering the CPU time criterion, the performance superiority of the algorithms over each other is analyzed for both complete and incomplete networks in small, medium, and large data sets, respectively.

Table 5.18. Wilcoxon Sign Rank Test Results to Compare Algorithms' Superiorities Based on CPU Time

Graph Topology	Network- Size	Algorithm Pairs	R+	R-	N+	N-	z-value	p-value	Sig.
Complete	Small	SA-GA	16.44	26.69	18	26	-2.322	0.020	Yes
		SA-VNS	16.08	24.91	12	32	-3.524	0.000	Yes
		SA-RVNS	0.00	22.50	0	44	-5.777	0.000	Yes
		GA-VNS	20.94	23.58	18	26	-1.377	0.168	No
		GA-RVNS	0.00	22.50	0	44	-5.777	0.000	Yes
		VNS-RVNS	0.00	22.50	0	44	-5.777	0.000	Yes
	Medium	SA-GA	13.73	19.96	11	24	-2.286	0.007	Yes
		SA-VNS	15.00	18.50	5	30	-3.931	0.000	Yes
		SA-RVNS	1.00	19.00	1	35	-5.216	0.000	Yes
		GA-VNS	19.67	16.32	12	22	-1.051	0.293	No
		GA-RVNS	2.00	19.50	3	32	-5.061	0.000	Yes
		VNS-RVNS	0.00	18.00	0	35	-5.159	0.000	Yes
	Large	SA-GA	0.00	21.50	0	42	-5.646	0.000	Yes
		SA-VNS	0.00	21.50	0	42	-5.647	0.000	Yes
		SA-RVNS	0.00	21.50	0	42	-5.646	0.000	Yes
		GA-VNS	0.00	21.50	0	42	-5.647	0.000	Yes
		GA-RVNS	0.00	21.50	0	42	-5.647	0.000	Yes
		VNS-RVNS	0.00	21.00	0	42	-5.58	0.000	Yes
Incomplete	Small	SA-GA	49.34	35.77	37	49	-0.356	0.018	Yes
		SA-VNS	35.23	50.31	26	65	-4.66	0.000	Yes
		SA-RVNS	35.01	54.92	37	56	-3.411	0.001	Yes
		GA-VNS	39.1	43.64	30	53	-2.588	0.010	Yes
		GA-RVNS	34.43	59	44	50	-2.706	0.007	Yes
		VNS-RVNS	37.77	55.53	42	52	-2.47	0.014	Yes
	Medium	SA-GA	6.75	14.19	4	21	-3.646	0.000	Yes
		SA-VNS	5.00	13.7	2	23	-4.103	0.000	Yes
		SA-RVNS	0.00	13	0	25	-4.372	0.000	Yes
		GA-VNS	12.64	13.29	11	14	-0.632	0.527	No
		GA-RVNS	0.00	13	0	25	-4.372	0.000	Yes
		VNS-RVNS	0.00	13	0	25	-4.372	0.000	Yes
	Large	SA-GA	21.75	24.210	4	43	-5.216	0.000	Yes
		SA-VNS	14.5	24.71	1	47	-5.612	0.000	Yes
		SA-RVNS	0.00	28.5	0	56	-6.509	0.000	Yes
		GA-VNS	25.07	22.18	21	25	-0.153	0.878	No
		GA-RVNS	0.00	28.5	0	56	-6.509	0.000	Yes
		VNS-RVNS	0.00	28.5	0	56	-6.509	0.000	Yes

R-GVNS is significantly better than other algorithms, as all p values are less than 0.05 (all p-values approximately 0.00) in terms of CPU times for all size and topologies. In addition, SA can be seen as the algorithm with the worst performance in terms of CPU time. It significantly worse than other algorithms for small, medium, and large datasets for both complete and incomplete type problems, respectively. When the SA algorithm is compared to GA in small-sized problems, GA is significantly superior to the SA algorithm, although it has p=0.20 and p=0.18 values in complete and incomplete cases, respectively. On the other hand, the p value converges to 0 in large-sized problems. For GA and VNS, on the other hand, there is no significant difference between each other in terms of solution time superiority in some problems. For instance, in small and medium size complete problems, there is no significant difference between GA and VNS since p=0.168 and p=0.293. The superiority of VNS can be seen in large data sets of Complete type. However, interestingly, the opposite is true for incomplete problems. While VNS was superior in incomplete small size problems, no significant difference was found in medium and large data sets with p=0.527 and p=0.878 values.

Algorithms are analyzed statistically in terms of obtaining the best solution or reaching the optimal solution. As can be seen from the Table 5.19, there are no statistically significant differences in small and medium size complete problems, since all approaches can obtain the optimal solution.

Table 5.19. Wilcoxon Sign Rank Test Results to Compare Algorithms' Superiorities Based on Best Solutions

Graph Topology	Network- Size	Algorithm Pairs	R+	R-	N+	N-	z-value	p-value	Sig.
Complete	Small	SA-GA	0.00	0.00	0	0	0	1.000	No
		SA-VNS	0.00	0.00	0	0	0	1.000	No
		SA-RVNS	0.00	0.00	0	0	0	1.000	No
		GA-VNS	0.00	0.00	0	0	0	1.000	No
		GA-RVNS	0.00	0.00	0	0	0	1.000	No
		VNS-RVNS	0.00	0.00	0	0	0	1.000	No
	Medium	SA-GA	0.00	0.00	0	0	0	1.000	No
		SA-VNS	0.00	0.00	0	0	0	1.000	No
		SA-RVNS	0.00	0.00	0	0	0	1.000	No
		GA-VNS	0.00	0.00	0	0	0	1.000	No
		GA-RVNS	0.00	0.00	0	0	0	1.000	No
		VNS-RVNS	0.00	0.00	0	0	0	1.000	No
	Large	SA-GA	12.60	13.10	5	20	-2.667	0.007	Yes
		SA-VNS	7.00	13.52	2	23	-3.996	0.000	Yes
		SA-RVNS	4.00	13.88	1	25	-4.356	0.000	Yes
		GA-VNS	5.00	11.47	3	17	-3.36	0.001	Yes
		GA-RVNS	2.50	11.39	2	18	-3.733	0.000	Yes
		VNS-RVNS	4.50	9.07	2	14	-3.051	0.002	Yes
Incomplete	Small	SA-GA	2.00	2.00	1	2	-0.535	0.593	No
		SA-VNS	0.00	1.50	0	2	-1.342	0.180	No
		SA-RVNS	0.00	1.50	0	2	-1.342	0.180	No
		GA-VNS	0.00	1.00	0	1	-1	0.317	No
		GA-RVNS	0.00	1.00	0	1	-1	0.317	No
		VNS-RVNS	0.00	0.00	0	0	0	1.000	No
	Medium	SA-GA	9.50	11.70	2	20	-3.49	0.000	Yes
		SA-VNS	0.00	12.00	0	23	-4.197	0.000	Yes
		SA-RVNS	1.00	12.50	1	22	-4.167	0.000	Yes
		GA-VNS	11.38	8.00	8	10	-0.24	0.811	No
		GA-RVNS	8.14	11.08	7	12	-1.529	0.126	No
		VNS-RVNS	2.00	8.93	1	15	0.001	-3.413	Yes
	Large	SA-GA	10.5	29.88	4	52	-6.167	0.000	Yes
		SA-VNS	0	28.5	0	56	-6.509	0.000	Yes
		SA-RVNS	9.00	29.22	2	54	-6.363	0.000	Yes
		GA-VNS	0	24	0	47	-5.968	0.000	Yes
		GA-RVNS	11.5	27.23	4	47	-5,783	0.000	Yes
		VNS-RVNS	23.00	23.00	5	40	-4.543	0.000	Yes

However, statistically significant differences can be seen in large size complete location problems. The superiority of GA over SA, VNS over GA, and RVNS over VNS can be seen to be significant with $p = 0.007$, $p = 0.001$, $p = 0.007$ values, respectively, and the null hypothesis is rejected.

There is no statistically significant difference between any two pairs of algorithms for small-sized incomplete type problems. In incomplete large size problems, the superiority of the RVNS algorithm over other algorithms is at a significant level. However, the performance value of GA is quite high in medium size incomplete problems. When GA is compared with both VNS and RVNS, it can be seen that there is no significant difference with $p = 0.811$ and $p = 0.126$ values in terms of superiority with these algorithms.

In a nutshell, the comparisons in all tables and demonstrate that the RVNS algorithm performs statistically better than the SA, GA, and VNS algorithms when it comes to the amount of CPU time required to solve small, medium, and large problems (in a range of hub connection topologies). In addition, RVNS has shown its superiority for large-scale problems, both complete and incomplete, based on the best-known solutions. It has been observed that the VNS algorithm and the GA method do not vary much from one another in terms of their ability to solve some types of hub location problems, such as those involving small size and medium size difficulties. On the other hand, the null hypothesis is rejected when dealing with large-scale incomplete hub location problems; to put it in other words, the VNS method statistically outperforms GA.

6. DISCUSSION AND CONCLUSION

In this thesis, incomplete hub network problems which were not previously addressed deeply in the literature were analyzed through p-hub median problems. Firstly, optimal solutions of the problem were obtained for all different scenarios (number of hubs, number of connections between hubs and coefficient of scale economies) by using CAB data set. Afterwards, in the context of these solutions, the study examined the basic characteristics influencing optimal solutions of uncapacitated p-hub median problems which was one of types of hub location problems. In this regard, it was ascertained that the designation as a hub was predicated upon node characteristics such as (i) centrality, (ii) demand-supply, (iii) distribution and (iv) proximity of unimportant nodes to important nodes (on the basis of the first three features). It was found that such characteristics had the potential to act as factors for determining locations likely to play a key role across a network, and outputs obtained as a consequence were evaluated on the basis of centrality measures and were interpreted through simple heuristic approaches. CBCA and CBCA-based heuristic approaches were presented and integrated into the original MILP model, and its solutions were analyzed. Upon the review of results, it was reported that methodologies which were developed through this study performed smoothly in terms of solution time in particular.

One of the main fundamental contribution of this thesis to hub location problems is that it places the focus on centrality measures. These measures, which were extensively used in the analysis of social networks, were not adequately studied in the context of transportation networks in the past. Initially, centrality measures such as closeness, betweenness and eigenvector, which were frequently referred to in the literature, were analyzed in terms of their ability to represent optimal hub locations, and their efficiency was measured. In this sense, it was discerned that closeness and betweenness centrality measures were effective in the framework of different scenarios on the basis of α value. Then, centrality measures were considered individually at both global level (covering the entire network structure) and local level (covering clustered network structures). This differentiation was categorized as external centrality measure and internal centrality measure, and both were weighted on the basis of scale economies. Moreover, as centrality measures were not alone sufficient for designating important nodes, demand quantities of nodes were taken into account. Besides, derivations of CBCA algorithm were not just on the basis of single centrality measure. Centrality measures were applied to the design of the network in conformity with their structures changeable in terms of different parameters. A substantive procedure was set in motion for reducing the number of candidate nodes existing in data sets.

Another contribution to the literature by this thesis on hub location problems pertains to the identification of optimal locations of incomplete hub networks. In the literature, there exist a small number of studies on network systems configured as incomplete structures. Moreover, solving this type of problems is pretty hard relative to complete hub network problems because decision-making process for connections between hubs is added to the problem. In this respect, the methodology proposed in this study simplifies the solution of such problems and facilitates the designation of optimal or near-optimal hub locations, establishment of (a specified number of) connections between designated hub locations and allocation of non-hub nodes to hub regions. For this purpose, by utilizing tools to designate sub-sets from among the entire set of nodes, this study enables the streamlining of basic hub location problem-solving approaches. In the context of this study, to the best of our knowledge, there existed no study especially in terms of incomplete hub network designs in the literature.

In this thesis, computational results are considered from two aspects. The first is the solution time, and the second is the quality of solution. In terms of the quality of solution, CBCA and CBCA-based approaches produced good results for small-scale data sets like CAB. It was also discerned that the methodology had a very satisfactory performance also in terms of solution time. This is because the candidate hub set used as input to the problem is reduced by CBCA and other algorithms derived from CBCA. In this reduction process, the outputs obtained in the general characteristics of the optimal hub positions are taken into consideration. The methodology was also applied to larger data sets such as AP100 and TR81. Although there was a slight decrease in solution quality, it was observed that the reduction rates in solution times were much higher. So for large-scale problems, CPU savings are remarkable. On the other hand, in real life, there exist several factors affecting the designation of hub locations such as geographical characteristics, transportation infrastructure, skilled labor force and technological infrastructure. In other words, hub locations are likely to be influenced by different parameters. However, most studies focuses solely on cost and time. In this respect, optima hub locations are sensitive to changes. In this context, the proposed methodology offers a practical approach that paves the way for better results under different parameters.

In the second phase of the thesis, inspired by the centrality-based solution approaches presented in the first chapter, the integration of these approaches into meta-heuristic methods is considered. Because it is clear that conventional meta-heuristic methods need modification in existing complex networks. In this context, we focused on problem reduction-based solution approaches. Just like the approaches presented in the first chapter, various graph-based and statistical metrics were defined based on the features of hub sets included in optimal solutions. Unlike the first part of the thesis, hub connections were also included in the analysis. Because hub connections also need to be optimized incomplete problems. The connections between the candidate hub sets obtained through the defined node features and the selected hub locations were included in this analysis. In the first stage, hub locations were determined, in the second stage, hub connections were defined according to edge characteristics and initial solutions were created accordingly. This approach was integrated into the GVNS meta-heuristic method, which is widely used in the literature for solving location problems. In this context, after obtaining the initial solutions, local search and diversification processes were used as in the GVNS method. However, in each iteration, the hub locations were updated on the basis of the specified features. This approach, which was integrated into the problem, was named the R-GVNS method and made significant contributions especially in terms of solution times.

Due to the rarity of studies on incomplete p-hub median problems in the literature, GA, SA and GVNS algorithms were implemented to the problem for performance comparison. Each test data and its variants were solved with these algorithms and compared with the R-GVNS algorithm. The results obtained were analyzed and it was determined that the R-GVNS algorithm was quite superior to conventional meta-heuristic methods.

One of the important contributions presented in this thesis is the proposed novel neighborhood operators. Various neighborhood operators are defined that take into account the centrality characteristics and flow sizes to the three main search operators used in current hub location problems. It has been shown that the defined neighborhood structures are quite effective in converging to the optimal solution. In addition, new neighborhood structures, which are required on the basis of hub connections, were integrated into the solution methods in search of the optimal solution of incomplete hub networks.

6.1. Limitations

The study presented in this thesis on hub location problems has some limitations in various aspects. These limits can be examined in different categories. The first of these is the economy of scale approach. Economies of scale practices often vary depending on the size of the flow. In this context, the fixed economy of scale coefficient used in transportation between o-d pairs is unrealistic. Instead, flow-dependent economies of scale approaches can be adopted. In addition, single allocation strategies may not be an adequate approach in some cases. Because it may not be realistic for a customer or warehouse to be connected to only one hub location. In this respect, suitable solution methods can be developed for logistics and telecommunication problems in which multiple-allocation strategies are adopted. In this thesis, only the single allocation strategy is focused. In addition, capacity constraints are considered in various cases. These can be hub capacity, vehicle capacity or arc capacity. Problems in which capacity constraints are taken into account should also be taken into account. Another factor not considered in this thesis is fixed costs. Hub installation and hub connection fixed costs should also be at the forefront of strategic decisions. Along with all this, this thesis focuses on the solution method. For this reason, the suggested approaches can be applied by making minor modifications in cases where the specified limits are eliminated.

6.2. Future Research Directions

This thesis presented a crucial departure point for prospective studies. Especially for large-scale problems (AP400, URAND etc.), the proposed methodology can be improved, and better results can be obtained. Besides, future studies can place the focus on different hub location problems (incomplete hub covering, incomplete p-center or incomplete p-dispersion). This study focused only on single-allocation strategy. In a similar vein, approaches concentrated on multiple-allocation (Talbi & Todosijević, 2017) or r-allocation (Peiró, Corberán, Laguna, & Martí, 2018) type of models can be developed. Moreover, the proposed approach and methodologies based on it can be integrated into heuristic methods and faster and more effective solutions can be produced. Multi-objective hub location problems studied often in recent years by researchers can be analyzed in this regard (Ghodratnama, Arbabi, & Azaron, 2018; Roni, Eksioglu, Cafferty,

& Jacobson, 2017). More efficient results can be obtained by integrating the problem reduction methodology proposed in the thesis into machine learning (ML) methods. In this context, the support of ML methods, especially for combinatorial optimization problems, is one of the most interesting topics today. In conclusion, it is essential to analyze hub location problems and to examine characteristics of hub locations based on hub network design problems intended for different purposes.

REFERENCES

Abbasi, M., Mokhtari, N., Shahvar, H., & Mahmoudi, A. (2019). Application of variable neighborhood search for solving large-scale many to many hub location routing problems. Journal of Advances in Management Research, 16(5), 683–697. https://doi.org/10.1108/JAMR-11-2018-0107

Abdinnour-Helm, S. (1998). A hybrid heuristic for the uncapacitated hub location problem. European Journal of Operational Research, 106(2–3), 489–499. https://doi.org/10.1016/S0377-2217(97)00286-5

Abdinnour-Helm, S. (2001). Using simulated annealing to solve the p-Hub Median Problem. International Journal of Physical Distribution & Logistics Management, 31(3), 203–220. https://doi.org/10.1108/09600030110389532

Abdinnour-Helm, S., & Venkataramanan, M. A. (1998). Solution approaches to hub location problems. Annals of Operations Research, 78, 31–50. https://doi.org/10.1023/a:1018954217758

Adler, N., & Hashai, N. (2005). Effect of open skies in the Middle East region. Transportation Research Part A: Policy and Practice, 39(10), 878–894. https://doi.org/10.1016/j.tra.2005.04.001

Akgün, İ., & Tansel, B. (2018). P-Hub Median Problem for Non-Complete Networks. Computers and Operations Research, 95, 56–72. https://doi.org/10.1016/j.cor.2018.02.014

Alibeyg, A., Contreras, I., & Fernández, E. (2018). Exact solution of hub network design problems with profits. European Journal of Operational Research, 266(1), 57–71. https://doi.org/10.1016/j.ejor.2017.09.024

Alkaabneh, F., Diabat, A., & Elhedhli, S. (2019). A Lagrangian heuristic and GRASP for the hub-and-spoke network system with economies-of-scale and congestion. Transportation Research Part C: Emerging Technologies, 102, 249–273. https://doi.org/10.1016/j.trc.2018.12.011

Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., & O'Kelly, M. E. (2021). Perspectives on modeling hub location problems. European Journal of Operational Research, 291(1), 1–17. https://doi.org/10.1016/j.ejor.2020.09.039

Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). The design of single allocation incomplete hub networks. Transportation Research Part B: Methodological, 43(10), 936–951. https://doi.org/10.1016/j.trb.2009.04.004

Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2012). Multimodal hub location and hub network design. Omega, 40(6), 927–939. https://doi.org/10.1016/j.omega.2012.02.005

Alumur, S. A., Nickel, S., & Saldanha-da-Gama, F. (2012). Hub location under uncertainty. Transportation Research Part B: Methodological, 46(4), 529–543. https://doi.org/10.1016/j.trb.2011.11.006

Alumur, S. A., Nickel, S., Saldanha-da-Gama, F., & Seçerdin, Y. (2016). Multi-period hub network design problems with modular capacities. Annals of Operations Research, 246(1–2), 289–312. https://doi.org/10.1007/s10479-015-1805-9

Alumur, S. A., Yaman, H., & Kara, B. Y. (2012). Hierarchical multimodal hub location problem with time-definite deliveries. Transportation Research Part E: Logistics and Transportation Review, 48(6), 1107–1120. https://doi.org/10.1016/j.tre.2012.04.001

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. European Journal of Operational Research, 190(1), 1–21. https://doi.org/10.1016/j.ejor.2007.06.008

Alumur, S., & Kara, B. Y. (2009). A hub covering network design problem for cargo applications in Turkey. Journal of the Operational Research Society, 60(10), 1349–1359. https://doi.org/10.1057/jors.2008.92

Ambrosino, D., & Sciomachen, A. (2016). A capacitated hub location problem in freight logistics multimodal networks. Optimization Letters, 10(5), 875–901. https://doi.org/10.1007/s11590-016-1022-8

An, Y., Zhang, Y., & Zeng, B. (2015). The reliable hub-and-spoke design problem: Models and algorithms. Transportation Research Part B: Methodological, 77, 103–122. https://doi.org/10.1016/j.trb.2015.02.006

Arnold, P., Peeters, D., & Thomas, I. (2004). Modelling a rail/road intermodal transportation system. Transportation Research Part E: Logistics and Transportation Review, 40(3), 255–270. https://doi.org/10.1016/j.tre.2003.08.005

Asgari, N., Farahani, R. Z., & Goh, M. (2013). Network design approach for hub ports-shipping companies competition and cooperation. Transportation Research Part A: Policy and Practice, 48, 1–18. https://doi.org/10.1016/j.tra.2012.10.020

Atta, S., & Sen, G. (2020). Multiple allocation p-hub location problem for content placement in VoD services: a differential evolution based approach. Applied Intelligence, 50(5), 1573–1589. https://doi.org/10.1007/s10489-019-01609-y

Atta, S., & Sen, G. (2021). A new variant of the p-hub location problem with a ring backbone network for content placement in VoD services. Computers and Industrial Engineering, 159. https://doi.org/10.1016/j.cie.2021.107432

Aykin, T. (1990). On "a quadratic integer program for the location of interacting hub facilities." European Journal of Operational Research, 46(3), 409–411. https://doi.org/10.1016/0377-2217(90)90018-7

Aykin, T. (1994). Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. European Journal of Operational Research, 79(3), 501–523. https://doi.org/10.1016/0377-2217(94)90062-0

Aykin, T. (1995). The hub location and routing problem. European Journal of Operational Research, 83(1), 200–219. https://doi.org/10.1016/0377-2217(93)E0173-U

Aykin, T., & Brown, G. F. (1992). Interacting New Facilities and Location-Allocation Problems. Transportation Science, 26(3), 212–222. https://doi.org/10.1287/trsc.26.3.212

Balcik, B., & Beamon, B. M. (2008). Facility location in humanitarian relief. International Journal of Logistics Research and Applications, 11(2), 101–121. https://doi.org/10.1080/13675560701561789

Bayram, V., & Yaman, H. (2018). Shelter Location and Evacuation Route Assignment Under Uncertainty: A Benders Decomposition Approach. Transportation Science, 52(2), 416–436. https://doi.org/10.1287/trsc.2017.0762

Berman, O., Drezner, Z., & Wesolowsky, G. O. (2005). The facility and transfer points location problem. International Transactions in Operational Research, 12(4), 387–402. https://doi.org/10.1111/j.1475-3995.2005.00514

Berman, O., & Wang, J. (2010). The network p-median problem with discrete probabilistic demand weights. Computers & Operations Research, 37(8), 1455–1463. https://doi.org/10.1016/j.cor.2009.10.007

Bernardes Real, L., O'Kelly, M., de Miranda, G., & Saraiva de Camargo, R. (2018). The gateway hub location problem. Journal of Air Transport Management, 73, 95–112. https://doi.org/10.1016/j.jairtraman.2018.08.006

Boccia, M., Crainic, T. G., Sforza, A., & Sterle, C. (2018). Multi-commodity location-routing: Flow intercepting formulation and branch-and-cut algorithm. Computers & Operations Research, 89, 94–112. https://doi.org/10.1016/j.cor.2017.08.013

Boland, N., Krishnamoorthy, M., Ernst, A. T., & Ebery, J. (2004). Preprocessing and cutting for multiple allocation hub location problems. European Journal of Operational Research, 155(3), 638–653. https://doi.org/10.1016/S0377-2217(03)00072-9

Bollapragada, R., Camm, J., Rao, U. S., & Junying, W. (2005). A two-phase greedy algorithm to locate and allocate hubs for fixed-wireless broadband access. Operations Research Letters, 33(2), 134–142. https://doi.org/10.1016/j.orl.2004.05.007

Bollapragada, R., Yanjun, L., & Rao, U. S. (2006). Budget-constrained, capacitated hub location to maximize expected demand coverage in fixed-wireless telecommunication networks. INFORMS Journal on Computing, 18(4), 422–432. https://doi.org/10.1287/ijoc.1050.0143y

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. The Journal of Mathematical Sociology, 2(1), 113–120. https://doi.org/10.1080/0022250X.1972.9989806

Brandes, U., & Pich, C. (2007). Centrality estimation in large networks. International Journal of Bifurcation and Chaos, 17(7), 2303–2318. https://doi.org/10.1142/S0218127407018403

Brimberg, J., Mladenović, N., Todosijević, R., & Urošević, D. (2020). A non-triangular hub location problem. Optimization Letters, 14(5), 1107–1126. https://doi.org/10.1007/s11590-019-01392-2

Bütün, C., Petrovic, S., & Muyldermans, L. (2021). The capacitated directed cycle hub location and routing problem under congestion. European Journal of Operational Research, 292(2), 714–734. https://doi.org/10.1016/j.ejor.2020.11.021

Calik, H., Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. Computers and Operations Research, 36(12), 3088–3096. https://doi.org/10.1016/j.cor.2008.11.023

Campbell, A. M., Lowe, T. J., & Zhang, L. (2007). The p-hub center allocation problem. European Journal of Operational Research, 176(2), 819–835. https://doi.org/10.1016/j.ejor.2005.09.024

Campbell, J. F. (1992). Location and allocation for distribution systems with transshipments and transportion economies of scale. Annals of Operations Research, 40(1), 77–99. https://doi.org/10.1007/BF02060471

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. European Journal of Operational Research, 72(2), 387–405. https://doi.org/10.1016/0377-2217(94)90318-2

Campbell, J. F. (1996). Hub location and the p-hub median problem. Operations Research, 44(6), 923–935. https://doi.org/10.1287/opre.44.6.923

Campbell, J. F. (2009). Hub location for time definite transportation. Computers and Operations Research, 36(12), 3107–3116. https://doi.org/10.1016/j.cor.2009.01.009

Campbell, J. F. (2013). Modeling Economies of Scale in Transportation Hub Networks. 2013 46th Hawaii International Conference on System Sciences, 1154–1163. https://doi.org/10.1109/HICSS.2013.411

Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2005a). Hub arc location problems: Part I - Introduction and results. Management Science, 51(10), 1540–1555. https://doi.org/10.1287/mnsc.1050.0406

Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2005b). Hub Arc location problems: Part II - Formulations and optimal algorithms. Management Science, 51(10), 1556–1571. https://doi.org/10.1287/mnsc.1050.0407

Campbell, J. F., & O'Kelly, M. E. (2012). Twenty-Five Years of Hub Location Research - Tags: TRANSPORTATION Science (Periodical) NETWORK hubs. Transportation Science, 46(2), 153–169. https://doi.org/10.1287/trsc.1120.0410

Campbell, J. F., Stiehr, G., Ernst, A. T., & Krishnamoorthy, M. (2003). Solving hub arc location problems on a cluster of workstations. Parallel Computing, 29(5 SPEC.), 555–574. https://doi.org/10.1016/S0167-8191(03)00042-5

Cánovas, L., García, S., & Marín, A. (2007). Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. European Journal of Operational Research, 179(3), 990–1007. https://doi.org/10.1016/j.ejor.2005.08.028

Carello, G., della Croce, F., Ghirardi, M., & Tadei, R. (2004). Solving the hub location problem in telecommunication network design: A local search approach. Networks, 44(2), 94–105. https://doi.org/10.1002/net.20020

Catanzaro, D., Gourdin, E., Labbé, M., & Özsoy, F. A. (2011). A branch-and-cut algorithm for the partitioning-hub location-routing problem. Computers and Operations Research, 38(2), 539–549. https://doi.org/10.1016/j.cor.2010.07.014

Çetiner, S., Sepil, C., & Süral, H. (2010). Hubbing and routing in postal delivery systems. Annals of Operations Research, 181(1), 109–124. https://doi.org/10.1007/s10479-010-0705-2

Chen, J. F. (2007). On solution of the capacitated single allocation hub location problem. Lecture Notes in Engineering and Computer Science, 2(2), 2353–2356.

Colbourn, C. J. (1999). Reliability Issues In Telecommunications Network Planning. In Telecommunications Network Planning (pp. 135–146). Springer US. https://doi.org/10.1007/978-1-4615-5087-7_8

Contreras, I., Cordeau, J. F., & Laporte, G. (2012). Exact solution of large-scale hub location problems with multiple capacity levels. Transportation Science, 46(4), 439–459. https://doi.org/10.1287/trsc.1110.0398

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011a). Benders decomposition for large-scale uncapacitated hub location. Operations Research, 59(6), 1477–1490. https://doi.org/10.1287/opre.1110.0965

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011b). Stochastic uncapacitated hub location. European Journal of Operational Research, 212(3), 518–528. https://doi.org/10.1016/j.ejor.2011.02.018

Contreras, I., Díaz, J. A., & Fernández, E. (2009). Lagrangean relaxation for the capacitated hub location problem with single assignment. OR Spectrum, 31(3), 483–505. https://doi.org/10.1007/s00291-008-0159-y

Contreras, I., Díaz, J. A., & Fernández, E. (2011). Branch and price for large-scale capacitated hub location problems with single assignment. INFORMS Journal on Computing, 23(1), 41–55. https://doi.org/10.1287/ijoc.1100.0391

Contreras, I., & Fernández, E. (2014). Hub location as the minimization of a supermodular set function. Operations Research, 62(3), 557–570. https://doi.org/10.1287/opre.2014.1263

Contreras, I., Fernández, E., & Marín, A. (2009). Tight bounds from a path based formulation for the tree of hub location problem. Computers and Operations Research, 36(12), 3117–3127. https://doi.org/10.1016/j.cor.2008.12.009

Contreras, I., Fernández, E., & Marín, A. (2010). The Tree of Hubs Location Problem. European Journal of Operational Research, 202(2), 390–400. https://doi.org/10.1016/j.ejor.2009.05.044

Contreras, I., Tanash, M., & Vidyarthi, N. (2017). Exact and heuristic approaches for the cycle hub location problem. Annals of Operations Research, 258(2), 655–677. https://doi.org/10.1007/s10479-015-2091-2

Correia, I., Nickel, S., & Saldanha-da-Gama, F. (2010). The capacitated single-allocation hub location problem revisited: A note on a classical formulation. European Journal of Operational Research, 207(1), 92–96. https://doi.org/10.1016/j.ejor.2010.04.015

Correia, I., Nickel, S., & Saldanha-da-Gama, F. (2011). Hub and spoke network design with single-assignment, capacity decisions and balancing requirements. Applied Mathematical Modelling, 35(10), 4841–4851. https://doi.org/10.1016/j.apm.2011.03.046

Cunha, C. B., & Silva, M. R. (2007). A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. European Journal of Operational Research, 179(3), 747–758. https://doi.org/10.1016/j.ejor.2005.03.057

Čvokić, D. D. (2020). A leader-follower single allocation hub location problem under fixed markups. Filomat, 34(8), 2463–2484. https://doi.org/10.2298/FIL2008463C

Čvokić, D. D., Kochetov, Y. A., Plyasunov, A. v, & Savić, A. (2021). A variable neighborhood search algorithm for the (r| p) hub–centroid problem under the price war. Journal of Global Optimization. https://doi.org/10.1007/s10898-021-01036-9

Čvokić, D. D., & Stanimirović, Z. (2020). A single allocation hub location and pricing problem. Computational and Applied Mathematics, 39(1). https://doi.org/10.1007/s40314-019-1025-z

Çetiner, S., Sepil, C., & Süral, H. (2010). Hubbing and routing in postal delivery systems. Annals of Operations Research, 181(1), 109–124. https://doi.org/10.1007/s10479-010-0705-2

da Graça Costa, M., Captivo, M. E., & Clímaco, J. (2008). Capacitated single allocation hub location problem-A bi-criteria approach. Computers and Operations Research, 35(11), 3671–3695. https://doi.org/10.1016/j.cor.2007.04.005

Dai, W., Zhang, J., Sun, X., & Wandelt, S. (2018). General contraction method for uncapacitated single allocation p-hub median problems. 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings, 2018-January, 1–8. https://doi.org/10.1109/SSCI.2017.8285321

Dai, W., Zhang, J., Sun, X., & Wandelt, S. (2019). HUBBI: Iterative network design for incomplete hub location problems. Computers and Operations Research, 104, 394–414. https://doi.org/10.1016/j.cor.2018.09.011

Dalal, J., & Üster, H. (2018). Combining Worst Case and Average Case Considerations in an Integrated Emergency Response Network Design Problem. Transportation Science, 52(1), 171–188. https://doi.org/10.1287/trsc.2016.0725

Danach, K., Gelareh, S., & Neamatian Monemi, R. (2019). The capacitated single-allocation p-hub location routing problem: a Lagrangian relaxation and a hyper-heuristic approach. EURO Journal on Transportation and Logistics, 8(5), 597–631. https://doi.org/10.1007/s13676-019-00141-w

Davari, S., Zarandi, M. H. F., & Turksen, I. B. (2013). The incomplete hub-covering location problem considering imprecise location of demands. Scientia Iranica, 20(3), 983–991. https://doi.org/10.1016/j.scient.2013.04.010

Degenne, A., & Forsé, M. (1999). Introducing social networks. Introducing Statistical Methods, vi, 248 s.

de Camargo, R. S., de Miranda, G., & Løkketangen, A. (2013). A new formulation and an exact approach for the many-to-many hub location-routing problem. Applied Mathematical Modelling, 37(12–13), 7465–7480. https://doi.org/10.1016/j.apm.2013.02.035

de Camargo, R. S., de Miranda, G., & Luna, H. P. L. (2009a). Benders Decomposition for Hub Location Problems with Economies of Scale. Transportation Science, 43(1), 86–97. https://doi.org/10.1287/trsc.1080.0233

de Camargo, R. S., de Miranda, G., & Luna, H. P. L. (2009b). Benders Decomposition for Hub Location Problems with Economies of Scale. Transportation Science, 43(1), 86–97. https://doi.org/10.1287/trsc.1080.0233

de Camargo, R. S., de Miranda, G., O'Kelly, M. E., & Campbell, J. F. (2017). Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. Applied Mathematical Modelling, 51, 274–301. https://doi.org/10.1016/j.apm.2017.06.035

de Camargo, R. S., de Miranda Jr., G., & Ferreira, R. P. M. (2011). A hybrid Outer-Approximation/Benders Decomposition algorithm for the single allocation hub location problem under congestion. Operations Research Letters, 39(5), 329–337. https://doi.org/10.1016/j.orl.2011.06.015

de Camargo, R. S., & Miranda, G. (2012). Single allocation hub location problem under congestion: Network owner and user perspectives. Expert Systems with Applications, 39(3), 3385–3391. https://doi.org/10.1016/j.eswa.2011.09.026

de Camargo, R. S., Miranda Jr., G., Ferreira, R. P. M., & Luna, H. P. (2009). Multiple allocation hub-and-spoke network design under hub congestion. Computers & Operations Research, 36(12), 3097–3106. https://doi.org/10.1016/j.cor.2008.10.004

de Camargo, R. S., Miranda Jr., G., & Luna, H. P. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. Computers and Operations Research, 35(4), 1047–1064. https://doi.org/10.1016/j.cor.2006.07.002

de Miranda Junior, G., de Camargo, R. S., Pinto, L. R., Conceição, S. v, & Ferreira, R. P. M. (2011). Hub location under hub congestion and demand uncertainty: The Brazilian case study. Pesquisa Operacional, 31(2), 319–349. https://doi.org/10.1590/S0101-74382011000200007

de Sá, E. M., de Camargo, R. S., & de Miranda, G. (2013a). An improved Benders decomposition algorithm for the tree of hubs location problem. European Journal of Operational Research, 226(2), 185–202. https://doi.org/10.1016/j.ejor.2012.10.051

de Sá, E., Morabito, R., & de Camargo, R. S. (2018a). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. Computers and Operations Research, 89, 31–50. https://doi.org/10.1016/j.cor.2017.08.001

de Sá, E., Morabito, R., & de Camargo, R. S. (2018b). Efficient Benders decomposition algorithms for the robust multiple allocation incomplete hub location problem with service time requirements. Expert Systems with Applications, 93, 50–61. https://doi.org/10.1016/j.eswa.2017.10.005

Dönmez, Z., Kara, B. Y., Karsu, Ö., & Saldanha-da-Gama, F. (2021). Humanitarian facility location under uncertainty: Critical review and future prospects. Omega, 102, 102393. https://doi.org/10.1016/j.omega.2021.102393

Dukkanci, O., Bektaş, T., & Kara, B. Y. (2019). Green Network Design Problems. In Sustainable Transportation and Smart Logistics (pp. 169–206). Elsevier. https://doi.org/10.1016/B978-0-12-814242-4.00007-7

Dukkanci, O., & Kara, B. Y. (2017). Routing and scheduling decisions in the hierarchical hub location problem. Computers and Operations Research, 85, 45–57. https://doi.org/10.1016/j.cor.2017.03.013

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2019). The green location-routing problem. Computers & Operations Research, 105, 187–202. https://doi.org/10.1016/j.cor.2019.01.011

Dukkanci, O., Peker, M., & Kara, B. Y. (2019). Green hub location problem. Transportation Research Part E: Logistics and Transportation Review, 125, 116–139. https://doi.org/10.1016/j.tre.2019.03.005

Ebery, J. (2001). Solving large single allocation p-hub problems with two or three hubs. European Journal of Operational Research, 128(2), 447–458. https://doi.org/10.1016/S0377-2217(99)00370-7

Ebery, J., Krishnamoorthy, M., Ernst, A., & Boland, N. (2000a). Capacitated multiple allocation hub location problem: Formulations and algorithms. European Journal of Operational Research, 120(3), 614–631. https://doi.org/10.1016/S0377-2217(98)00395-6

Ebrahimi-zade, A., Hosseini-Nasab, H., zare-mehrjerdi, Y., & Zahmatkesh, A. (2016). Multi-period hub set covering problems with flexible radius: A modified genetic solution. Applied Mathematical Modelling, 40(4), 2968–2982. https://doi.org/10.1016/j.apm.2015.09.064

Eiselt, H. A., & Marianov, V. (2009). A conditional p-hub location problem with attraction functions. Computers and Operations Research, 36(12), 3128–3135. https://doi.org/10.1016/j.cor.2008.11.014

Elhedhli, S., & Hu, F. X. (2005). Hub-and-spoke network design with congestion. Computers & Operations Research, 32(6), 1615–1632. https://doi.org/10.1016/j.cor.2003.11.016

Elhedhli, S., & Wu, H. (2010). A Lagrangean Heuristic for Hub-and-Spoke System Design with Capacity Selection and Congestion. INFORMS Journal on Computing, 22(2), 282–296. https://doi.org/10.1287/ijoc.1090.0335

Ernst, A. T., Hamacher, H., Jiang, H., Krishnamoorthy, M., & Woeginger, G. (2009). Uncapacitated single and multiple allocation p-hub center problems. Computers and Operations Research, 36(7), 2230–2241. https://doi.org/10.1016/j.cor.2008.08.021

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location Science, 4(3), 139–154. https://doi.org/10.1016/S0966-8349(96)00011-3

Ernst, A. T., & Krishnamoorthy, M. (1998a). An exact solution approach based on shortest-paths for p-hub median problems. INFORMS Journal on Computing, 10(2), 149–162. https://doi.org/10.1287/ijoc.10.2.149

Ernst, A. T., & Krishnamoorthy, M. (1998b). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. European Journal of Operational Research, 104(1), 100–112. https://doi.org/10.1016/S0377-2217(96)00340-2

Ernst, A. T., & Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research, 86, 141–159. https://doi.org/10.1023/a:1018994432663

Eskandari-Khanghahi, M., Tavakkoli-Moghaddam, R., Taleizadeh, A. A., & Amin, S. H. (2018). Designing and optimizing a sustainable supply chain network for a blood platelet bank under uncertainty. Engineering Applications of Artificial Intelligence, 71, 236–250. https://doi.org/10.1016/j.engappai.2018.03.004

Fabian Meier, J., & Clausen, U. (2013). Strategic planning in LTL logistics - increasing the capacity utilization of trucks. Electronic Notes in Discrete Mathematics, 41, 37–44. https://doi.org/10.1016/j.endm.2013.05.073

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. Computers and Industrial Engineering, 64(4), 1096–1109. https://doi.org/10.1016/j.cie.2013.01.012

Frade, I., & Ribeiro, A. (2015). Bike-sharing stations: A maximal covering location approach. Transportation Research Part A: Policy and Practice, 82, 216–227. https://doi.org/10.1016/j.tra.2015.09.014

Freeman, L. C., Roeder, D., & Mulholland, R. R. (1979). Centrality in social networks: ii. experimental results. Social Networks, 2(2), 119–141. https://doi.org/10.1016/0378-8733(79)90002-9

Friedkin 1991. (2001). Retrieved from file://localhost/Volumes/alexvanvenrooij 1/Elektronische Artikelen/Papers/Unknown/2001-7.pdf

García, S., Landete, M., & Marín, A. (2012). New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem. European Journal of Operational Research, 220(1), 48–57. https://doi.org/10.1016/j.ejor.2012.01.042

Gavish, B., & Neuman, I. (1992). Routing in a network with unreliable components. IEEE Transactions on Communications, 40(7), 1248–1258. https://doi.org/10.1109/26.153370

Gavrilouk, E. O. (2009). Aggregation in hub location problems. Computers and Operations Research, 36(12), 3136–3142. https://doi.org/10.1016/j.cor.2009.01.010

Gelareh, S., Maculan, N., Mahey, P., & Monemi, R. N. (2013). Hub-and-spoke network design and fleet deployment for string planning of liner shipping. Applied Mathematical Modelling, 37(5), 3307–3321. https://doi.org/10.1016/j.apm.2012.07.017

Gelareh, S., & Nickel, S. (2011). Hub location problems in transportation networks. Transportation Research Part E: Logistics and Transportation Review, 47(6), 1092–1111. https://doi.org/10.1016/j.tre.2011.04.009

Gelareh, S., Nickel, S., & Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. Transportation Research Part E: Logistics and Transportation Review, 46(6), 991–1004. https://doi.org/10.1016/j.tre.2010.05.005

Gelareh, S., & Pisinger, D. (2011). Fleet deployment, network design and hub location of liner shipping companies. Transportation Research Part E: Logistics and Transportation Review, 47(6), 947–964. https://doi.org/10.1016/j.tre.2011.03.002

Ghaffarinasab, N., & Kara, B. Y. (2019). Benders Decomposition Algorithms for Two Variants of the Single Allocation Hub Location Problem. Networks and Spatial Economics, 19(1), 83–108. https://doi.org/10.1007/s11067-018-9424-z

Ghaffarinasab, N., & Kara, B. Y. (2022). A conditional <math altimg="si428.svg" display="inline" id="d1e11336"> <mi>β</mi> </math> -mean approach to risk-averse stochastic multiple allocation hub location problems. Transportation Research Part E:

Logistics and Transportation Review, 158, 102602. https://doi.org/10.1016/j.tre.2021.102602

Ghaffarinasab, N., Motallebzadeh, A., Jabarzadeh, Y., & Kara, B. Y. (2018). Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems. Computers and Operations Research, 90, 173–192. https://doi.org/10.1016/j.cor.2017.09.022

Ghasemi, P., Khalili-Damghani, K., Hafezalkotob, A., & Raissi, S. (2019). Uncertain multi-objective multi-commodity multi-period multi-vehicle location-allocation model for earthquake evacuation planning. Applied Mathematics and Computation, 350, 105–132. https://doi.org/10.1016/j.amc.2018.12.061

Ghodratnama, A., Arbabi, H. R., & Azaron, A. (2018). A bi-objective hub location-allocation model considering congestion. Operational Research, 1–40. https://doi.org/10.1007/s12351-018-0404-3

Groothedde, B., Ruijgrok, C., & Tavasszy, L. (2005a). Towards collaborative, intermodal hub networks. A case study in the fast moving consumer goods market. Transportation Research Part E: Logistics and Transportation Review, 41(6 SPEC. ISS.), 567–583. https://doi.org/10.1016/j.tre.2005.06.005

Groothedde, B., Ruijgrok, C., & Tavasszy, L. (2005b). Towards collaborative, intermodal hub networks. A case study in the fast moving consumer goods market. Transportation Research Part E: Logistics and Transportation Review, 41(6 SPEC. ISS.), 567–583. https://doi.org/10.1016/j.tre.2005.06.005

Hamidi, M., Gholamian, M., & Shahanaghi, K. (2014). Developing prevention reliability in hub location models. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 228(4), 337–346. https://doi.org/10.1177/1748006X13519247

He, Y., Wu, T., Zhang, C., & Liang, Z. (2015). An improved MIP heuristic for the intermodal hub location problem. Omega, 57, 203–211. https://doi.org/10.1016/j.omega.2015.04.016

Hoff, A., Peiró, J., Corberán, Á., & Martí, R. (2017). Heuristics for the capacitated modular hub location problem. Computers and Operations Research, 86, 94–109. https://doi.org/10.1016/j.cor.2017.05.004

Horner, M. W., & O'Kelly, M. E. (2001). Embedding economies of scale concepts for hub network design. Journal of Transport Geography, 9(4), 255–265. https://doi.org/10.1016/S0966-6923(01)00019-9

Hu, W., Dong, J., Hwang, B.-G., Ren, R., & Chen, Z. (2020). Network planning of urban underground logistics system with hub-and-spoke layout: two phase cluster-based approach. Engineering, Construction and Architectural Management, 27(8), 2079–2105. https://doi.org/10.1108/ECAM-06-2019-0296

Hubbell, C. H. (1965). An Input-Output Approach to Clique Identification Author (s): Charles H . Hubbell Reviewed work (s): Published by : American Sociological Association Stable URL : http://www.jstor.org/stable/2785990 . American Sociological Association, 28(4), 377–399.

Hult, E., Jiang, H., & Ralph, D. (2014). Exact computational approaches to a stochastic uncapacitated single allocation p-hub center problem. Computational Optimization and Applications, 59(1–2), 185–200. https://doi.org/10.1007/s10589-013-9629-5

Hwang, Y. H., & Lee, Y. H. (2012). Uncapacitated single allocation p-hub maximal covering problem. Computers and Industrial Engineering, 63(2), 382–389. https://doi.org/10.1016/j.cie.2012.03.014

Ilić, A., Urošević, D., Brimberg, J., & Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research, 206(2), 289–300. https://doi.org/10.1016/j.ejor.2010.02.022

Ishfaq, R., & Sox, C. R. (2011). Hub location-allocation in intermodal logistic networks. European Journal of Operational Research, 210(2), 213–230. https://doi.org/10.1016/j.ejor.2010.09.017

Ishfaq, R., & Sox, C. R. (2012). Design of intermodal logistics networks with hub delays. European Journal of Operational Research, 220(3), 629–641. https://doi.org/10.1016/j.ejor.2012.03.010

Kansal, M. L., Kumar, A., & Sharma, P. B. (1995). Reliability analysis of water distribution systems under uncertainty. Reliability Engineering & System Safety, 50(1), 51–59. https://doi.org/10.1016/0951-8320(95)00051-3

Kara, B. Y., & Tansel, B. Ç. (2000). On the single-assignment p-hub center problem. European Journal of Operational Research, 125(3), 648–655. https://doi.org/10.1016/S0377-2217(99)00274-X

Kara, B. Y., & Tansel, B. Ç. (2001). The latest arrival hub location problem. Management Science, 47(10), 1408–1420. https://doi.org/10.1287/mnsc.47.10.1408.10258

Kara, B. Y., & Tansel, B. C. (2003). The single-assignment hub covering problem: Models and linearizations. Journal of the Operational Research Society, 54(1), 59–64. https://doi.org/10.1057/palgrave.jors.2601473

Kargar, K., & Mahmutoğulları, A. İ. (2022). Risk-averse hub location: Formulation and solution approach. Computers & Operations Research, 143, 105760. https://doi.org/10.1016/j.cor.2022.105760

Karimi, H., & Bashiri, M. (2011). Hub covering location problems with different coverage types. Scientia Iranica, 18(6), 1571–1578. https://doi.org/10.1016/j.scient.2011.09.018

Karimi, B., Bashiri, M., & Nikzad, E. (2018). Multi-commodity multimodal splittable logistics hub location problem with stochastic demands. International Journal of Engineering, Transactions B: Applications, 31(11), 1935–1942. https://doi.org/10.5829/ije.2018.31.11b.18

Karimi, H. (2018). The capacitated hub covering location-routing problem for simultaneous pickup and delivery systems. Computers and Industrial Engineering, 116, 47–58. https://doi.org/10.1016/j.cie.2017.12.020

Karimi, H., & Setak, M. (2014). Proprietor and customer costs in the incomplete hub location-routing network topology. Applied Mathematical Modelling, 38(3), 1011–1023. https://doi.org/10.1016/j.apm.2013.07.033

Karsu, Ö., Kara, B. Y., Akkaya, E., & Ozel, A. (2021). Clean Water Network Design for Refugee Camps. Networks and Spatial Economics, 21(1), 175–198. https://doi.org/10.1007/s11067-020-09514-5

Kartal, Z., Hasgul, S., & Ernst, A. T. (2017). Single allocation p-hub median location and routing problem with simultaneous pick-up and delivery. Transportation Research Part E: Logistics and Transportation Review, 108(September), 141–159. https://doi.org/10.1016/j.tre.2017.10.004

Katz, L. (1953). a New Status Indexderıved From Sociometric. Psychmetrika, 18(1), 39–43.

Kaveh, F., Tavakkoli-Moghaddam, R., Triki, C., Rahimi, Y., & Jamili, A. (2021). A new bi-objective model of the urban public transportation hub network design under uncertainty. Annals of Operations Research, 296(1–2), 131–162. https://doi.org/10.1007/s10479-019-03430-9

Kazemian, I., & Aref, S. (2017). Hub location under uncertainty: a minimax regret model for the capacitated problem with multiple allocations. International Journal of Supply Chain and Inventory Management, 2(1), 1. https://doi.org/10.1504/IJSCIM.2017.086371

Khaleghi, A., & Eydi, A. (2021). Robust sustainable multi-period hub location considering uncertain time-dependent demand. RAIRO - Operations Research, 55(6), 3541–3574. https://doi.org/10.1051/ro/2021155

Kim, H., & O'Kelly, M. E. (2009). Reliable p-hub location problems in telecommunication networks. Geographical Analysis, 41(3), 283–306. https://doi.org/10.1111/j.1538-4632.2009.00755.x

Kimms, A. (n.d.). Economies of Scale in Hub & Spoke Network Design Models: We Have It All Wrong. In Perspectives on Operations Research (pp. 293–317). DUV. https://doi.org/10.1007/978-3-8350-9064-4_17

Kınay, Ö. B., Yetis Kara, B., Saldanha-da-Gama, F., & Correia, I. (2018). Modeling the shelter site location problem using chance constraints: A case study for Istanbul. European Journal of Operational Research, 270(1), 132–145. https://doi.org/10.1016/j.ejor.2018.03.006

Klincewicz, J. G. (1991). Heuristics for the p-hub location problem. European Journal of Operational Research, 53(1), 25–37. https://doi.org/10.1016/0377-2217(91)90090-I

Klincewicz, J. G. (1992). Avoiding local optima in the p-hub location problem using tabu search and GRASP. Annals of Operations Research, 40(1), 283–302. https://doi.org/10.1007/BF02060483

Klincewicz, J. G. (1996). A dual algorithm for the uncapacitated hub location problem. Location Science, 4(3), 173–184. https://doi.org/10.1016/S0966-8349(96)00010-1

Klincewicz, J. G. (2002). Enumeration and search procedures for a hub location problem with economies of scale. Annals of Operations Research, 110(1–4), 107–122. https://doi.org/10.1023/A:1020715517162

Klincewicz, J. G., Luss, H., & Yan, D. C. K. (1998). Designing tributary networks with multiple ring families. Computers and Operations Research, 25(12), 1145–1157. https://doi.org/10.1016/S0305-0548(98)00005-7

Kratica, J., Stanimirović, Z., Tošić, D., & Filipović, V. (2007). Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research, 182(1), 15–28. https://doi.org/10.1016/j.ejor.2006.06.056

Labbé, M., & Yaman, H. (2004). Projecting the flow variables for hub location problems. Networks, 44(2), 84–93. https://doi.org/10.1002/net.20019

Labbé, M., & Yaman, H. (2008). Solving the hub location problem in a star-star network. Networks, 51(1), 19–33. https://doi.org/10.1002/net.20193

Labbé, M., Yaman, H., & Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. Mathematical Programming, 102(2), 371–405. https://doi.org/10.1007/s10107-004-0531-x

Laporte, G. (2009). Fifty Years of Vehicle Routing. Transportation Science, 43(4), 408–416. https://doi.org/10.1287/trsc.1090.0301

Lazega, E., & Burt, R. S. (1995). Structural Holes: The Social Structure of Competition. Revue Française de Sociologie (Vol. 36). https://doi.org/10.2307/3322456

Lee, H., Shi, Y., & Nazem, S. M. (1996). Supporting rural telecommunications: A compromise solutions approach. Annals of Operations Research, 68, 33–45. https://doi.org/10.1007/BF02205447

Lee, Y., Lim, B. H., & Park, J. S. (1996). A hub location problem in designing digital data service networks: Lagrangian relaxation approach. Location Science, 4(3), 185–194. https://doi.org/10.1016/S0966-8349(96)00009-5

Li, G., & Liu, S. (2013). Research on the evolution of urban logistics networkig motives, paths and model construction. Journal of Applied Sciences, 13(14), 2825–2830. https://doi.org/10.3923/jas.2013.2825.2830

Li, S., Fang, C., & Wu, Y. (2020). Robust Hub Location Problem with Flow-Based Set-Up Cost. IEEE Access, 8, 66178–66188. https://doi.org/10.1109/ACCESS.2020.2985377

Liang, H. (2013). The hardness and approximation of the star -hub center problem. Operations Research Letters, 41(2), 138–141. https://doi.org/10.1016/j.orl.2012.12.007

Limbourg, S., & Jourquin, B. (2010). Market area of intermodal rail-road container terminals embedded in a hub-and-spoke network*. Papers in Regional Science, 89(1), 135–154. https://doi.org/10.1111/j.1435-5957.2009.00255.x

Lin, C. C., Lin, J. Y., & Chen, Y. C. (2012). The capacitated p-hub median problem with integral constraints: An application to a Chinese air cargo network. Applied Mathematical Modelling, 36(6), 2777–2787. https://doi.org/10.1016/j.apm.2011.09.063

Lin, C.-C. (2010). The integrated secondary route network design model in the hierarchical hub-and-spoke network for dual express services. International Journal of Production Economics, 123(1), 20–30. https://doi.org/10.1016/j.ijpe.2009.05.030

Lin, C.-C., & Chen, S.-H. (2004). The hierarchical network design problem for time-definite express common carriers. Transportation Research Part B: Methodological, 38(3), 271–283. https://doi.org/10.1016/S0191-2615(03)00013-4

Lin, C.-C., & Lee, S.-C. (2010). The competition game on hub network design. Transportation Research Part B: Methodological, 44(4), 618–629. https://doi.org/10.1016/j.trb.2009.09.002

Lin, J.-R., Yang, T.-H., & Chang, Y.-C. (2013). A hub location inventory model for bicycle sharing system design: Formulation and solution. Computers and Industrial Engineering, 65(1), 77–86. https://doi.org/10.1016/j.cie.2011.12.006

Liu, Y., Cui, N., & Zhang, J. (2019). Integrated temporary facility location and casualty allocation planning for post-disaster humanitarian medical service. Transportation Research Part E: Logistics and Transportation Review, 128, 1–16. https://doi.org/10.1016/j.tre.2019.05.008

Lowe, T. J., & Sim, T. (2013). The hub covering flow problem. Journal of the Operational Research Society, 64(7), 973–981. https://doi.org/10.1057/jors.2012.122

Lüer-Villagra, A., Eiselt, H. A., & Marianov, V. (2019). A single allocation p-hub median problem with general piecewise-linear costs in arcs. Computers and Industrial Engineering, 128, 477–491. https://doi.org/10.1016/j.cie.2018.12.058

Lüer-Villagra, A., & Marianov, V. (2013). A competitive hub location and pricing problem. European Journal of Operational Research, 231(3), 734–744. https://doi.org/10.1016/j.ejor.2013.06.006

Mahéo, A., Kilby, P., & van Hentenryck, P. (2019). Benders Decomposition for the Design of a Hub and Shuttle Public Transit System. Transportation Science, 53(1), 77–88. https://doi.org/10.1287/trsc.2017.0756

Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., & Jamiri, A. (2020). A multi-modal competitive hub location pricing problem with customer loyalty and elastic demand. Computers and Operations Research, 123. https://doi.org/10.1016/j.cor.2020.105048

Mahmutogullari, A. I., & Kara, B. Y. (2016). Hub location under competition. European Journal of Operational Research, 250(1), 214–225. https://doi.org/10.1016/j.ejor.2015.09.008

Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as M/D/c queues. Computers and Operations Research, 30(7), 983–1003. https://doi.org/10.1016/S0305-0548(02)00052-7

Marianov, V., Serra, D., & Revelle, C. (1999). Location of hubs in a competitive environment. European Journal of Operational Research, 114(2), 363–371. https://doi.org/10.1016/S0377-2217(98)00195-7

Marianov, V., Serra, D., & ReVelle, C. (1999). Location of hubs in a competitive environment. European Journal of Operational Research, 114(2), 363–371. https://doi.org/10.1016/S0377-2217(98)00195-7

Marín, A. (2005a). Formulating and solving splittable capacitated multiple allocation hub location problems. Computers and Operations Research, 32(12), 3093–3109. https://doi.org/10.1016/j.cor.2004.04.008

Marín, A. (2005b). Uncapacitated euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. Journal of Global Optimization, 33(3), 393–422. https://doi.org/10.1007/s10898-004-6099-4

Marín, A., Cánovas, L., & Landete, M. (2006). New formulations for the uncapacitated multiple allocation hub location problem. European Journal of Operational Research, 172(1), 274–292. https://doi.org/10.1016/j.ejor.2004.09.047

Martín, J. C., & Voltes-Dorta, A. (2008). Theoretical evidence of existing pitfalls in measuring hubbing practices in airline networks. Networks and Spatial Economics, 8(2–3), 161–181. https://doi.org/10.1007/s11067-007-9051-6

Martins De Sá, E., Contreras, I., & Cordeau, J.-F. (2015). Exact and heuristic algorithms for the design of hub networks with multiple lines. European Journal of Operational Research, 246(1), 186–198. https://doi.org/10.1016/j.ejor.2015.04.017

Martins de Sá, E., Morabito, R., & de Camargo, R. S. (2018a). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. Computers and Operations Research, 89, 31–50. https://doi.org/10.1016/j.cor.2017.08.001

Martins de Sá, E., Morabito, R., & de Camargo, R. S. (2018b). Efficient Benders decomposition algorithms for the robust multiple allocation incomplete hub location problem with service time requirements. Expert Systems with Applications, 93, 50–61. https://doi.org/10.1016/j.eswa.2017.10.005

Marwah, B. R., Parti, R., & Kalra, P. K. (2005). Optimal planning of transit routes for large cities. Proceedings of the International Conference on Automated People Movers, 383–394. https://doi.org/10.1061/40766(174)33

Mayer, G., & Wagner, B. (2002). HubLocator: An exact solution method for the multiple allocation hub location problem. Computers and Operations Research, 29(6), 715–739. https://doi.org/10.1016/S0305-0548(01)00080-6

Meier, J. F. (2017). An improved mixed integer program for single allocation hub location problems with stepwise cost function. International Transactions in Operational Research, 24(5), 983–991. https://doi.org/10.1111/itor.12270

Meier, J. F., Clausen, U., Rostami, B., & Buchheim, C. (2016). A Compact Linearisation of Euclidean Single Allocation Hub Location Problems. Electronic Notes in Discrete Mathematics, 52, 37–44. https://doi.org/10.1016/j.endm.2016.03.006

Meng, Q., & Wang, X. (2011). Intermodal hub-and-spoke network design: Incorporating multiple stakeholders and multi-type containers. Transportation Research Part B: Methodological, 45(4), 724–742. https://doi.org/10.1016/j.trb.2010.11.002

Meraklı, M., & Yaman, H. (2017). A capacitated hub location problem under hose demand uncertainty. Computers and Operations Research, 88, 58–70. https://doi.org/10.1016/j.cor.2017.06.011

Meyer, T., Ernst, A. T., & Krishnamoorthy, M. (2009). A 2-phase algorithm for solving the single allocation p-hub center problem. Computers and Operations Research, 36(12), 3143–3151. https://doi.org/10.1016/j.cor.2008.07.011

Mohammadi, M., Jula, P., & Tavakkoli-Moghaddam, R. (2019). Reliable single-allocation hub location problem with disruptions. Transportation Research Part E: Logistics and Transportation Review, 123, 90–120. https://doi.org/10.1016/j.tre.2019.01.008

Mohammadi, M., Tavakkoli-Moghaddam, R., Siadat, A., & Rahimi, Y. (2016). A game-based meta-heuristic for a fuzzy bi-objective reliable hub location problem. Engineering Applications of Artificial Intelligence, 50, 1–19. https://doi.org/10.1016/j.engappai.2015.12.009

Mohammadi, M., Torabi, S. A., & Tavakkoli-Moghaddam, R. (2014). Sustainable hub location under mixed uncertainty. Transportation Research Part E: Logistics and Transportation Review, 62, 89–115. https://doi.org/10.1016/j.tre.2013.12.005

Mokhtar, H., Krishnamoorthy, M., & Ernst, A. T. (2019a). The 2-allocation p-hub median problem and a modified Benders decomposition method for solving hub location problems. Computers and Operations Research, 104, 375–393. https://doi.org/10.1016/j.cor.2018.09.006

Mokhtar, H., Krishnamoorthy, M., & Ernst, A. T. (2019b). The 2-allocation p-hub median problem and a modified Benders decomposition method for solving hub location problems. Computers & Operations Research, 104, 375–393. https://doi.org/10.1016/j.cor.2018.09.006

Mokhtar, H., Redi, A. A. N. P., Krishnamoorthy, M., & Ernst, A. T. (2019). An intermodal hub location problem for container distribution in Indonesia. Computers and Operations Research, 104, 415–432. https://doi.org/10.1016/j.cor.2018.08.012

Mokhtarzadeh, M., Tavakkoli-Moghaddam, R., Triki, C., & Rahimi, Y. (2021). A hybrid of clustering and meta-heuristic algorithms to solve a p-mobile hub location–allocation problem with the depreciation cost of hub facilities. Engineering Applications of Artificial Intelligence, 98. https://doi.org/10.1016/j.engappai.2020.104121

Momayezi, F., Chaharsooghi, S. K., Sepehri, M. M., & Kashan, A. H. (2021). The capacitated modular single-allocation hub location problem with possibilities of hubs disruptions: modeling and a solution algorithm. Operational Research, 21(1), 139–166. https://doi.org/10.1007/s12351-018-0438-6

Monemi, R. N., Gelareh, S., Nagih, A., & Jones, D. (2021). Bi-objective load balancing multiple allocation hub location: a compromise programming approach. Annals of Operations Research, 296(1–2), 363–406. https://doi.org/10.1007/s10479-019-03421-w

Musavi, M., & Bozorgi-Amiri, A. (2017). A multi-objective sustainable hub location-scheduling problem for perishable food supply chain. Computers and Industrial Engineering, 113, 766–778. https://doi.org/10.1016/j.cie.2017.07.039

Najy, W., & Diabat, A. (2020). Benders decomposition for multiple-allocation hub-and-spoke network design with economies of scale and node congestion. Transportation Research Part B: Methodological, 133, 62–84. https://doi.org/10.1016/j.trb.2019.12.003

Neamatian Monemi, R., Gelareh, S., Hanafi, S., & Maculan, N. (2017). A co-opetitive framework for the hub location problems in transportation networks. Optimization, 66(12), 2089–2106. https://doi.org/10.1080/02331934.2017.1295045

Nickel, S., Schöbel, A., & Sonneborn, T. (2001). Hub Location Problems in Urban Traffic Networks (pp. 95–107). https://doi.org/10.1007/978-1-4757-3357-0_6

O'Kelly, M. E. (1986). Location of Interacting Hub Facilities. Transportation Science, 20(2), 92–106. https://doi.org/10.1287/trsc.20.2.92

O'kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research, 32(3), 393–404. https://doi.org/10.1016/S0377-2217(87)80007-3

O'Kelly, M. E. (1992). A clustering approach to the planar hub location problem. Annals of Operations Research, 40(1), 339–353. https://doi.org/10.1007/BF02060486

O'Kelly, M. E., & Bryan, D. L. (1998). Hub location with flow economies of scale. Transportation Research Part B: Methodological, 32(8), 605–616. https://doi.org/10.1016/S0191-2615(98)00021-6

O'Kelly, M. E., & Lao, Y. (2010). Mode Choice in a Hub-and-Spoke Network: A Zero-One Linear Programming Approach. Geographical Analysis, 23(4), 283–297. https://doi.org/10.1111/j.1538-4632.1991.tb00240.x

O'Kelly, M., Skorin-Kapov, D., & Skorin-Kapov, J. (1995). Lower Bounds for the Hub Location Problem. Management Science, 41(4), 713–721. https://doi.org/10.1287/mnsc.41.4.713

Oliveira, F. A., de Sá, E. M., & de Souza, S. R. (2022a). Benders decomposition applied to profit maximizing hub location problem with incomplete hub network. Computers & Operations Research, 105715. https://doi.org/10.1016/j.cor.2022.105715

Oliveira, F. A., de Sá, E. M., & de Souza, S. R. (2022b). Benders decomposition applied to profit maximizing hub location problem with incomplete hub network. Computers & Operations Research, 142, 105715. https://doi.org/10.1016/j.cor.2022.105715

Osorio-Mora, A., Núñez-Cerda, F., Gatica, G., & Linfati, R. (2020). Multimodal Capacitated Hub Location Problems with Multi-Commodities: An Application in Freight Transport. Journal of Advanced Transportation, 2020. https://doi.org/10.1155/2020/2431763

Owsiński, J., Stańczak, J., Barski, A., Sęp, K., & Sapiecha, P. (2015). Graph based approach to the minimum hub problem in transportation network. 1641–1648. https://doi.org/10.15439/2015F218

Ozbay, E., Çavuş, Ö., & Kara, B. Y. (2019). Shelter site location under multi-hazard scenarios. Computers & Operations Research, 106, 102–118. https://doi.org/10.1016/j.cor.2019.02.008

Öztürk, C., Tuzkaya, G., & Bulkan, S. (2021). Centrality based solution approaches for median-type incomplete hub location problems. Computers and Industrial Engineering, 156. https://doi.org/10.1016/j.cie.2021.107275

Pan, J.-S., Song, P.-C., Chu, S.-C., & Peng, Y.-J. (2020). Improved Compact Cuckoo Search Algorithm Applied to Location of Drone Logistics Hub. Mathematics, 8(3), 333. https://doi.org/10.3390/math8030333

Pearce, R. H., & Forbes, M. (2018). Disaggregated Benders decomposition and branch-and-cut for solving the budget-constrained dynamic uncapacitated facility location and network design problem. European Journal of Operational Research, 270(1), 78–88. https://doi.org/10.1016/j.ejor.2018.03.021

Peiró, J., Corberán, Á., Laguna, M., & Martí, R. (2018). Models and solution methods for the uncapacitated r-allocation p-hub equitable center problem. International Transactions in Operational Research, 25(4), 1241–1267. https://doi.org/10.1111/itor.12441

Peker, M., & Kara, B. Y. (2015). The P-Hub maximal covering problem and extensions for gradual decay functions. Omega (United Kingdom), 54, 158–172. https://doi.org/10.1016/j.omega.2015.01.009

Peker, M., Kara, B. Y., Campbell, J. F., & Alumur, S. A. (2016). Spatial Analysis of Single Allocation Hub Location Problems. Networks and Spatial Economics, 16(4), 1075–1101. https://doi.org/10.1007/s11067-015-9311-9

Pirkul, H., & Schilling, D. A. (1998). An efficient procedure for designing single allocation hub and spoke systems. Management Science, 44(12 PART 2), S235–S242. https://doi.org/10.1287/mnsc.44.12.s235

Pozo, M. A., Puerto, J., & Rodríguez Chía, A. M. (2021). The ordered median tree of hubs location problem. TOP, 29(1), 78–105. https://doi.org/10.1007/s11750-020-00572-z

Puerto, J., Ramos, A. B., & Rodríguez-Chía, A. M. (2011). Single-allocation ordered median hub location problems. Computers and Operations Research, 38(2), 559–570. https://doi.org/10.1016/j.cor.2010.07.018

Puerto, J., Ramos, A. B., & Rodríguez-Chía, A. M. (2013). A specialized branch & bound & cut for Single-Allocation Ordered Median Hub Location problems. Discrete Applied Mathematics, 161(16–17), 2624–2646. https://doi.org/10.1016/j.dam.2013.05.035

Puerto, J., Ramos, A. B., Rodríguez-Chía, A. M., & Sánchez-Gil, M. C. (2016). Ordered median hub location problems with capacity constraints. Transportation Research Part C: Emerging Technologies, 70, 142–156. https://doi.org/10.1016/j.trc.2015.05.012

Qin, Z., & Gao, Y. (2017). Uncapacitated p -hub location problem with fixed costs and uncertain flows. Journal of Intelligent Manufacturing, 28(3), 705–716. https://doi.org/10.1007/s10845-014-0990-8

Quadros, H., Costa Roboredo, M., & Alves Pessoa, A. (2018). A branch-and-cut algorithm for the multiple allocation r-hub interdiction median problem with fortification. Expert Systems with Applications, 110, 311–322. https://doi.org/10.1016/j.eswa.2018.05.036

Racunica, I., & Wynter, L. (2005). Optimal location of intermodal freight hubs. Transportation Research Part B: Methodological, 39(5), 453–477. https://doi.org/10.1016/j.trb.2004.07.001

Rahimi, Y., Torabi, S. A., & Tavakkoli-Moghaddam, R. (2019). A new robust-possibilistic reliable hub protection model with elastic demands and backup hubs under risk. Engineering Applications of Artificial Intelligence, 86, 68–82. https://doi.org/10.1016/j.engappai.2019.08.019

Rahmati, R., Bashiri, M., Nikzad, E., & Siadat, A. (2021). A two-stage robust hub location problem with accelerated Benders decomposition algorithm. International Journal of Production Research, 1–23. https://doi.org/10.1080/00207543.2021.1953179

Randall, M. (2008). Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation. Computational Optimization and Applications, 39(2), 239–261. https://doi.org/10.1007/s10589-007-9069-1

Ratli, M., Urošević, D., el Cadi, A. A., Brimberg, J., Mladenović, N., & Todosijević, R. (2020). An efficient heuristic for a hub location routing problem. Optimization Letters. https://doi.org/10.1007/s11590-020-01675-z

Redondi, R., Malighetti, P., & Paleari, S. (2011). Hub competition and travel times in the world-wide airport network. Journal of Transport Geography, 19(6), 1260–1271. https://doi.org/10.1016/j.jtrangeo.2010.11.010

Rieck, J., Ehrenberg, C., & Zimmermann, J. (2014). Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. European Journal of Operational Research, 236(3), 863–878. https://doi.org/10.1016/j.ejor.2013.12.021

Rodríguez-Déniz, H., Suau-Sanchez, P., & Voltes-Dorta, A. (2013). Classifying airports according to their hub dimensions: An application to the US domestic network. Journal of Transport Geography, 33, 188–195. https://doi.org/10.1016/j.jtrangeo.2013.10.011

Rodríguez-Martín, I., & Salazar-González, J. J. (2008). Solving a capacitated hub location problem. European Journal of Operational Research, 184(2), 468–479. https://doi.org/10.1016/j.ejor.2006.11.026

Rodríguez-Martín, I., Salazar-González, J.-J., & Yaman, H. (2014). A branch-and-cut algorithm for the hub location and routing problem. Computers and Operations Research, 50, 161–174. https://doi.org/10.1016/j.cor.2014.04.014

Roni, M. S., Eksioglu, S. D., Cafferty, K. G., & Jacobson, J. J. (2017). A multi-objective, hub-and-spoke model to design and manage biofuel supply chains. Annals of Operations Research, 249(1–2), 351–380. https://doi.org/10.1007/s10479-015-2102-3

Rostami, B., Buchheim, C., Meier, J. F., & Clausen, U. (2016). Lower Bounding Procedures for the Single Allocation Hub Location Problem. Electronic Notes in Discrete Mathematics, 52, 69–76. https://doi.org/10.1016/j.endm.2016.03.010

Rostami, B., Kämmerling, N., Buchheim, C., & Clausen, U. (2018). Reliable single allocation hub location problem under hub breakdowns. Computers and Operations Research, 96, 15–29. https://doi.org/10.1016/j.cor.2018.04.002

Rothenbächer, A.-K., Drexl, M., & Irnich, S. (2016). Branch-and-price-and-cut for a service network design and hub location problem. European Journal of Operational Research, 255(3), 935–947. https://doi.org/10.1016/j.ejor.2016.05.058

Sasaki, M. (2005). Hub network design model in a competitive environment with flow threshold. Journal of the Operations Research Society of Japan, 48(2), 158–171. https://doi.org/10.15807/jorsj.48.158

Sasaki, M., Campbell, J. F., Krishnamoorthy, M., & Ernst, A. T. (2014). A Stackelberg hub arc location model for a competitive environment. Computers and Operations Research, 47, 27–41. https://doi.org/10.1016/j.cor.2014.01.009

Sasaki, M., & Fukushima, M. (2001). STACKELBERG HUB LOCATION PROBLEM. Journal of the Operations Research Society of Japan, 44(4), 390–402. https://doi.org/10.15807/jorsj.44.390

Sasaki, M., & Fukushima, M. (2003). On the hub-and-spoke model with arc capacity conatraints. Journal of the Operations Research Society of Japan, 46(4), 409–428. https://doi.org/10.15807/jorsj.46.409

Sasaki, M., Suzuki, A., & Drezner, Z. (1999a). On the selection of hub airports for an airline hub-and-spoke system. Computers and Operations Research, 26(14), 1411–1422. https://doi.org/10.1016/S0305-0548(99)00043-X

Sasaki, M., Suzuki, A., & Drezner, Z. (1999b). On the selection of hub airports for an airline hub-and-spoke system. Computers & Operations Research, 26(14), 1411–1422. https://doi.org/10.1016/S0305-0548(99)00043-X

Sayah, D., & Irnich, S. (2017). A new compact formulation for the discrete p-dispersion problem. European Journal of Operational Research, 256(1), 62–67. https://doi.org/10.1016/j.ejor.2016.06.036

Sayyady, F., & Fathi, Y. (2016). An integer programming approach for solving the p-dispersion problem. European Journal of Operational Research, 253(1), 216–225. https://doi.org/10.1016/j.ejor.2016.02.026

Sen, G., & Krishnamoorthy, M. (2018). Discrete particle swarm optimization algorithms for two variants of the static data segment location problem. Applied Intelligence, 48(3), 771–790. https://doi.org/10.1007/s10489-017-0995-z

Sha, Y., & Huang, J. (2012). The Multi-period Location-allocation Problem of Engineering Emergency Blood Supply Systems. Systems Engineering Procedia, 5, 21–28. https://doi.org/10.1016/j.sepro.2012.04.004

Shahabi, M., & Unnikrishnan, A. (2014). Robust hub network design problem. Transportation Research Part E: Logistics and Transportation Review, 70, 356–373. https://doi.org/10.1016/j.tre.2014.08.003

Shen, H., Liang, Y., & Shen, Z.-J. M. (2021). Reliable hub location model for air transportation networks under random disruptions. Manufacturing and Service Operations Management, 23(2), 388–406. https://doi.org/10.1287/msom.2019.0845

Sheu, J.-B., Lin, A. Y. S., & Chen, Y.-J. (2008). A prototype of a hierarchical global logistics network planning model. International Journal of Risk Assessment and Management, 10(3), 206–223. https://doi.org/10.1504/IJRAM.2008.021374

Silva, M. R., & Cunha, C. B. (2017). A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem. European Journal of Operational Research, 262(3), 954–965. https://doi.org/10.1016/j.ejor.2017.03.066

Skorin-Kapov, D., & Skorin-Kapov, J. (1994). On tabu search for the location of interacting hub facilities. European Journal of Operational Research, 73(3), 502–509. https://doi.org/10.1016/0377-2217(94)90245-3

Skorin-Kapov, D., Skorin-Kapov, J., & O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. European Journal of Operational Research, 94(3), 582–593. https://doi.org/10.1016/0377-2217(95)00100-X

Smith, K., Krishnamoorthy, M., & Palaniswami, M. (1996). Neural versus traditional approaches to the location of interacting hub facilities. Location Science, 4(3), 155–171. https://doi.org/10.1016/S0966-8349(96)00017-4

Sohn, J., & Park, S. (1998). Efficient solution procedure and reduced size formulations for p-hub location problems. European Journal of Operational Research, 108(1), 118–126. https://doi.org/10.1016/S0377-2217(97)00201-4

Sohn, J., & Park, S. (2000). The single allocation problem in the interacting three-hub network. Networks, 35(1), 17–25. https://doi.org/10.1002/(SICI)1097-0037(200001)35:1

Society, A. (2018). Data and Decision Sciences in Action, 133–148. https://doi.org/10.1007/978-3-319-55914-8

Stanojević, P., Marić, M., & Stanimirović, Z. (2015). A hybridization of an evolutionary algorithm and a parallel branch and bound for solving the capacitated single allocation hub location problem. Applied Soft Computing Journal, 33, 24–36. https://doi.org/10.1016/j.asoc.2015.04.018

SteadieSeifi, M., Dellaert, N. P., Nuijten, W., van Woensel, T., & Raoufi, R. (2014). Multimodal freight transportation planning: A literature review. European Journal of Operational Research, 233(1), 1–15. https://doi.org/10.1016/j.ejor.2013.06.055

Sun, X., Dai, W., Zhang, Y., & Wandelt, S. (2017). Finding p -Hub Median Locations: An Empirical Study on Problems and Solution Techniques. Journal of Advanced Transportation, 2017. https://doi.org/10.1155/2017/9387302

Sung, C. S., & Jin, H. W. (2001). Dual-based approach for a hub network design problem under non-restrictive policy. European Journal of Operational Research, 132(1), 88–105. https://doi.org/10.1016/S0377-2217(00)00114-4

Taherkhani, G., & Alumur, S. A. (2019). Profit maximizing hub location problems. Omega (United Kingdom), 86, 1–15. https://doi.org/10.1016/j.omega.2018.05.016

Taherkhani, G., Alumur, S. A., & Hosseini, M. (2020). Benders Decomposition for the Profit Maximizing Capacitated Hub Location Problem with Multiple Demand Classes. Transportation Science, 54(6), 1446–1470. https://doi.org/10.1287/trsc.2020.1003

Talbi, E. G., & Todosijević, R. (2017). The robust uncapacitated multiple allocation p-hub median problem. Computers and Industrial Engineering, 110, 322–332. https://doi.org/10.1016/j.cie.2017.06.017

Tan, P. Z., & Kara, B. Y. (2007). A hub covering model for cargo delivery systems. Networks, 49(1), 28–39. https://doi.org/10.1002/net.20139

Tanash, M., Contreras, I., & Vidyarthi, N. (2017). An exact algorithm for the modular hub location problem with single assignments. Computers and Operations Research, 85, 32–44. https://doi.org/10.1016/j.cor.2017.03.006

Teymourian, E., Sadeghi, A., & Taghipourian, F. (2011). A dynamic virtual hub location problem in airline networks - formulation and metaheuristic solution approaches. First International Technology Management Conference, 1061–1068. https://doi.org/10.1109/ITMC.2011.5996004

Thomadsen, T., & Larsen, J. (2007). A hub location problem with fully interconnected backbone and access networks. Computers and Operations Research, 34(8), 2520–2531. https://doi.org/10.1016/j.cor.2005.09.018

Tiwari, R., Jayaswal, S., & Sinha, A. (2021a). Alternate solution approaches for competitive hub location problems. European Journal of Operational Research, 290(1), 68–80. https://doi.org/10.1016/j.ejor.2020.07.018

Tiwari, R., Jayaswal, S., & Sinha, A. (2021b). Alternate solution approaches for competitive hub location problems. European Journal of Operational Research, 290(1), 68–80. https://doi.org/10.1016/j.ejor.2020.07.018

Topcuoglu, H., Corut, F., Ermis, M., & Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. Computers and Operations Research, 32(4), 967–984. https://doi.org/10.1016/j.cor.2003.09.008

Torkestani, S. S., Seyedhosseini, S. M., Makui, A., & Shahanaghi, K. (2018). The reliable design of a hierarchical multi-modes transportation hub location problems (HMMTHLP) under dynamic network disruption (DND). Computers & Industrial Engineering, 122, 39–86. https://doi.org/10.1016/j.cie.2018.05.027

Tran, T. H., O'Hanley, J. R., & Scaparra, M. P. (2017). Reliable Hub Network Design: Formulation and Solution Techniques. Transportation Science, 51(1), 358–375. https://doi.org/10.1287/trsc.2016.0679

Vasconcelos, A. D., Nassi, C. D., & Lopes, L. A. S. (2011). The uncapacitated hub location problem in networks under decentralized management. Computers and Operations Research, 38(12), 1656–1666. https://doi.org/10.1016/j.cor.2011.03.004

Wagner, B. (2004). A note on "the latest arrival hub location problem." Management Science, 50(12), 1751–1752. https://doi.org/10.1287/mnsc.1040.0312

Wagner, B. (2008a). A note on "Location of hubs in a competitive environment." European Journal of Operational Research, 184(1), 57–62. https://doi.org/10.1016/j.ejor.2006.10.057

Wagner, B. (2008b). Model formulations for hub covering problems. Journal of the Operational Research Society, 59(7), 932–938. https://doi.org/10.1057/palgrave.jors.2602424

Wagner, B. (2007). An exact solution procedure for a cluster hub location problem. European Journal of Operational Research, 178(2), 391–401. https://doi.org/10.1016/j.ejor.2006.02.011

Wang, J. J., & Cheng, M. C. (2010). From a hub port city to a global supply chain management center: a case study of Hong Kong. Journal of Transport Geography, 18(1), 104–115. https://doi.org/10.1016/j.jtrangeo.2009.02.009

Wang, S., Chen, Z., & Liu, T. (2020). Distributionally robust hub location. Transportation Science, 54(5), 1189–1210. https://doi.org/10.1287/TRSC.2019.0948

Wasner, M., & Zäpfel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. International Journal of Production Economics, 90(3), 403–419. https://doi.org/10.1016/j.ijpe.2003.12.002

Wolf, S. (2007). On the Complexity of the Uncapacitated Single Allocation p-Hub Median Problem with Equal Weights 1 Problem Description. Group, 1–5.

Wu, T., Shi, Z., & Zhang, C. (2021). The hub location problem with market selection. Computers & Operations Research, 127, 105136. https://doi.org/10.1016/j.cor.2020.105136

Xu, Y., Dai, W., Sun, X., & Wandelt, S. (2018). Improved benders decomposition for capacitated hub location problem with incomplete hub networks. 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings, 2018-January(July 2018), 1–8. https://doi.org/10.1109/SSCI.2017.8285341

Yaman, H. (2005). Polyhedral analysis for the uncapacitated hub location problem with modular arc capacities. SIAM Journal on Discrete Mathematics, 19(2), 501–522. https://doi.org/10.1137/S0895480103439157

Yaman, H. (2008). Star p-hub median problem with modular arc capacities. Computers and Operations Research, 35(9), 3009–3019. https://doi.org/10.1016/j.cor.2007.01.014

Yaman, H. (2009). The hierarchical hub median problem with single assignment. Transportation Research Part B: Methodological, 43(6), 643–658. https://doi.org/10.1016/j.trb.2009.01.005

Yaman, H. (2011). Allocation strategies in hub networks. European Journal of Operational Research, 211(3), 442–451. https://doi.org/10.1016/j.ejor.2011.01.014

Yaman, H., & Carello, G. (2005). Solving the hub location problem with modular link capacities. Computers and Operations Research, 32(12), 3227–3245. https://doi.org/10.1016/j.cor.2004.05.009

Yaman, H., Kara, B. Y., & Tansel, B. Ç. (2007). The latest arrival hub location problem for cargo delivery systems with stopovers. Transportation Research Part B: Methodological, 41(8), 906–919. https://doi.org/10.1016/j.trb.2007.03.003

Yang, T.-H. (2009). Stochastic air freight hub location and flight routes planning. Applied Mathematical Modelling, 33(12), 4424–4430. https://doi.org/10.1016/j.apm.2009.03.018

Yang, X., Bostel, N., & Dejax, P. (2019). A MILP model and memetic algorithm for the Hub Location and Routing problem with distinct collection and delivery tours. Computers and Industrial Engineering, 135, 105–119. https://doi.org/10.1016/j.cie.2019.05.038

Yıldız, B., & Karaşan, O. E. (2015). Regenerator Location Problem and survivable extensions: A hub covering location perspective. Transportation Research Part B: Methodological, 71, 32–55. https://doi.org/10.1016/j.trb.2014.10.004

Yıldız, B., Yaman, H., & Karaşan, O. E. (2021). Hub Location, Routing, and Route Dimensioning: Strategic and Tactical Intermodal Transportation Hub Network Design. Transportation Science, 55(6), 1351–1369. https://doi.org/10.1287/trsc.2021.1070

Yu, J., Liu, Y., Chang, G. L., Ma, W., & Yang, X. (2009). Cluster-based hierarchical model for urban transit hub location planning: Formulation, solution, and case study. Transportation Research Record, (2112), 8–16. https://doi.org/10.3141/2112-02

Yu, V. F., Kuo, C. W., & Dat, L. Q. (2014). Selection of key component vendor from the aspects of capability, productivity, and reliability. Mathematical Problems in Engineering, 2014. https://doi.org/10.1155/2014/124652

Yuan, Y., & Yu, J. (2018). Locating transit hubs in a multi-modal transportation network: A cluster-based optimization approach. Transportation Research Part E: Logistics and Transportation Review, 114(April), 85–103. https://doi.org/10.1016/j.tre.2018.03.00

Zarrinpoor, N., Fallahnezhad, M. S., & Pishvaee, M. S. (2017). Design of a reliable hierarchical location-allocation model under disruptions for health service networks: A two-stage robust approach. Computers & Industrial Engineering, 109, 130–150. https://doi.org/10.1016/j.cie.2017.04.036

Zetina, C. A., Contreras, I., Cordeau, J.-F., & Nikbakhsh, E. (2017). Robust uncapacitated hub location. Transportation Research Part B: Methodological, 106, 393–410. https://doi.org/10.1016/j.trb.2017.06.008

Zhalechian, M., Tavakkoli-Moghaddam, R., & Rahimi, Y. (2017). A self-adaptive evolutionary algorithm for a fuzzy multi-objective hub location problem: An integration of responsiveness and social responsibility. Engineering Applications of Artificial Intelligence, 62, 1–16. https://doi.org/10.1016/j.engappai.2017.03.006

Zhao, L., Zhou, J., Li, H., Yang, P., & Zhou, L. (2021). Optimizing the design of an intra-city metro logistics system based on a hub-and-spoke network model. Tunnelling and Underground Space Technology, 116. https://doi.org/10.1016/j.tust.2021.104086

Zhao, Y., Chen, N., Zhou, J., & Kuang, H. (2019). Mathematical Formulations and Solution Algorithm for Reliable Hub-And-Spoke Network Design for Container Shipping. Journal of Coastal Research, 94(sp1), 540–546. https://doi.org/10.2112/SI94-107.1

CURRICULUM VITAE

İsim-Soyisim

CİHAT ÖZTÜRK

Eğitim

- Ph.D. in Industrial Engineering / *Marmara University*
  Starting/End Date: 2014/ 2022
- M.Sc. in Industrial Engineering/ *Kocaeli University*
  Starting/End Date: 2011/ 2013
- B.S. in Industrial Engineering/ *Yıldız Technical University*
  Starting/End Date: 2008/ 2010

Araştırma Alanları

Methodology and Tools

Heuristic Optimization, Machine Learning, Multi-Criteria and Multi-Objective Decision Making

Application Areas

Blockchain and Applications, Sustainable Supply Chain, Network and Location Theory, OR Applications on Telecommunication, Scheduling

A. Uluslararası SCI-SCI Expanded indexli dergilerde yayımlanan makaleler :

- **A1.** Öztürk, C., Tuzkaya, G., & Bulkan, S. (2021). Centrality based solution approaches for median-type incomplete hub location problems. Computers & Industrial Engineering, 156, 107275.

- **A2.** Öztürk, C., & Yildizbaşi, A. (2020). Barriers to implementation of blockchain into supply chain management using an integrated multi-criteria decision-making method: a numerical example. Soft Computing, 24(19), 14771-14789.

- **A3.** Mangla, S. K., Kazançoğlu, Y., Yıldızbaşı, A., Öztürk, C., & Çalık, A. (2022). A conceptual framework for blockchain-based sustainable supply chain and evaluating implementation barriers: A case of the tea supply chain. Business Strategy and the Environment.

- **A4.** Yıldızbaşı, A., Öztürk, C., Efendioğlu, D., & Bulkan, S. (2021). Assessing the social sustainable supply chain indicators using an integrated fuzzy multi-criteria decision-making methods: a case study of Turkey. Environment, Development and Sustainability, 23(3), 4285-4320.

- **A5.** Abubakar, A. I., Ozturk, C., Ozturk, M., Mollel, M. S., Asad, S. M., Hassan, N. U., ... & Imran, M. A. (2022). Revenue Maximization through Cell Switching and Spectrum Leasing in 5G HetNets. IEEE Access, 10, 48301-48317.

B. <u>Uluslararası Dergilerde yayımlanan makaleler:</u>

- **B1**. Öztürk, C., Efendioğlu, D., & Yıldızbaşı, A. (2018). A systematic approach for health workforce management in turkey. Iioab journal, 9(6), 12-20.

- **B2.** Abubakar, A. I., Ahmad, I., Omeke, K. G., Ozturk, M., Ozturk, C., Abdel-Salam, A. M., ... & Imran, M. A. (2022). A Survey on Energy Optimization Techniques in UAV-Based Cellular Networks: From Conventional to Machine Learning Approaches. arXiv preprint arXiv:2204.07967.

C. <u>Kitap Bölümleri</u>

- **C1.** Uluslararası Ticarette Blok Zincir Entegrasyonu Ve Olası Engel Er ÇALIK A., YILDIZBAŞI A., ÖZTÜRK C.
  İn: Dijital Temelli Uluslararası Ticaret, Murat Canıtez; Bilge Afşar, Editor, Ekin Yayınevi, Bursa, Pp.55-86, 2021.

- **C2.** Elektronik Ürün Üreten Firmaların Döngüsel Ekonomi Adaptasyon Düzeylerinin Aralık Değerli Sezgisel Bulanık VIKOR Metodu İle Değerlendirilmesi Öztürk C., Eraslan E.
  in: Bulanık Çokkriterli Kararverme Yöntemleri, Prof. Dr. Mehmet Kabak, Doç. Dr. Babek Erdebilli, Editor, Nobel Yayınevi, Ankara, pp.141-163,202.

- **C3.** MACBETH Yöntemi ile Sürdürülebilir Enerji Alternatifi Seçimi Yıldızbaşı A., Öztürk C.,Çalık A., Eraslan E.
  in: Çok Kriterli Karar Verme Yöntemleri MS Excel Çözümlü Uygulamalar, Kabak Mehmet, Çınar Yetkin, Editor, Nobel Yayınevi, Ankara, pp.333-347,2020.

- **C4.** Güleç, N., Öztürk, C., & Efendioğlu, D. (2018). The Status of Foreign Trades Under Globalization in Developing and Developed Countries With Turkey. In Globalization and Trade Integration in Developing Countries (pp. 167-201). IGI Global.

D. <u>Uluslararası ve Ulusal Bildiriler</u>

- **D1.** Big Data-Driven in COVID-19 Pandemic Management System: Evaluation of Barriers with Spherical Fuzzy AHP Approach Ariöz Y., Yilmaz İ., Yildizbaşi A., Öztürk C. International Conference on Intelligent and Fuzzy Systems, INFUS2021, İstanbul, Turkey, 24-26 August 2021, vol.308, pp.811-818.

- **D2.** Key Challenges of Lithium-Ion Battery Recycling Process in Circular Economy Environment: Pythagorean Fuzzy AHP Approach Yildizbaşi A., Öztürk C., Yilmaz İ., Ariöz Y. International Conference on Intelligent and Fuzzy Systems, INFUS2021, İstanbul, Turkey,24-26August2021, vol.308, pp.561-568.

- **D3.** Hospital Type Location Decisions by Using Pythagorean Fuzzy Sets Composition: A Case Study of COVID-19 Yilmaz İ., Ariöz Y., Öztürk C., Yildizbaşi A. International Conference on Intelligent and Fuzzy Systems, INFUS2021, İstanbul, Turkey, 24-26 August2021, vol.308, pp.589-597 IV.

- **D4.** Vaccine Selection Using Interval-Valued Intuitionistic Fuzzy VIKOR: A Case Study of Covid-19 Pandemic Öztürk C., Yildizbaşi A., Yilmaz İ., Ariöz Y. International Conference on Intelligent and Fuzzy Systems, INFUS2021, İstanbul, Turkey,24 -26August2021, vol.308, pp.101-108.

- **D5.** The Granger Causality Analysis of Construction Industry: Case Study in Turkey Efendioğlu D., Güleç N., Öztürk C. 11th International Symposium on Intelligent Manufacturing & Service Systems, Sakarya, Turkey, 27 May 2021.

- **D6.** Optimal Siting for A New Bioenergy Power Plant in Turkey: An Intuitionistic Fuzzy Multi-Criteria Decision Analysis under Circular Economy Perspective Polat L., Yildizbaşi A., Öztürk C. 2nd Online Symposium on Circular Economy and Sustainability, Alexandroupoli, Greece, 14 July 2021.

- **D7.** Determining the circular economy implementation barriers in the agriculture supply chain: Evidence from the food supply chain Barough S.A., Yildizbaşi A., Öztürk C. 2nd Online Symposium on Circular Economy and Sustainability, Greece, 14 July 2021.

- **D8.** Evaluation of Key Technological Tools in terms of Supply Chain Sustainability in the Digitalization Era with Different Analytic Hierarchy Process Methods Yildizbaşi A., Öztürk C., Kumar S.M., Kazançoğlu Y. ISAHP2020, United States of America, 03 December 2020.

- **D9.** Closed-Loop Recycling of Electronic Cards Contributing to Sustainability: A Case Study on a Circular Economy Perspective Ariöz Y., Yildizbaşi A., Öztürk C. Online Symposium on Circular Economy and Sustainability, ATİNA, Greece ,01 July 2020.

- **D10.** Blockchain and Sustainability Öztürk C., Yildizbaşi A., Ariöz Y. Online Symposium on Circular Economy and Sustainability, ATİNA, Greece,01 July 2020.

- **D11.** Determining the Position of Countries According to the Quality-of-Life Index Criteria Güleç N., Efendioğlu D., Öztürk C. NCM Conferences, Ankara, Turkey, 11 -12 September 2018, vol.1.

- **D12.** BIST 100 Index Estimation Using Bayesian Regression Modeling ÖZTÜRK C., EFENDİOĞLU D., GÜLEÇ N. International Web Conference onForecasting,16 -18October 2017.

- **D13.** Performance Assessment of Multi-objective Optimization Methods Öztürk C., Özkale C.,8 Fiğlali A. International Symposium on Intel igent & Manufacturing Systems,27 -28 September 2012.

- **D14.** Bulanık Tabanlı Analitik Ağ Süreci Yaklaşımı ile Orta Ölçekli İşletmeler için Kurumsal Kaynak Planlama Yazılımı Seçimi Öztürk C., Erdem A. R., Aladağ Z., Yıldız Kumru P. YAEM, Turkey, 5 -07 July2011.

E. Projeler

- **E1.** Covid-19 tipi salgınların kontrolüne yönelik Blokzinciri ve yapay zekâ tabanlı sistem yaklaşımı (AYBU BAP destekli proje-2021)