Stochastic Facility Location Problem with Distributed Demands along the Network Edges

Mahmoud Golabi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Industrial Engineering

Eastern Mediterranean University
August 2017
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Mustafa Tümer
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy in Industrial Engineering.

Assoc. Prof. Dr. Gökhan İzbırak
Chair, Department of Industrial Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Industrial Engineering.

Assoc. Prof. Dr. Jamal Arkat
Co-Supervisor

Assoc. Prof. Dr. Gökhan İzbırak
Supervisor

Examining Committee

1. Prof. Dr. Serpil Erol
2. Prof. Dr. Zoltán Lakner
3. Assoc. Prof. Dr. Gökhan İzbırak
4. Prof. Dr. Béla Vizvári
5. Asst. Prof. Dr. Sahand Daneshvar

ABSTRACT

Since 1960s, facility location problem (FLP) has been studied by a myriad number of researchers. Nowadays, it is one of the most prominent branches of operations research which is applied in different fields such as determining the location of warehouses, hazardous materials sites, automated teller machines (ATMs), coastal search and rescue stations, etc. Also, the application of FLP in emergency logistics for choosing the best location of service centers has become rampant recently.

On the premise that demands are uniformly distributed along the network edges, two network location problems are investigated in this study. For both problems, some of the candidate locations will be selected to establish the facilities. The first problem is a multiple-server congested facility location problem. It is assumed that demands are generated according to the Poisson process. Furthermore, the number of servers in each established facility is considered as a decision variable and the service time for each server follows an exponential distribution. Using queuing system analysis, a mathematical model is developed to minimize the customers' aggregate expected traveling times and the aggregate expected waiting times.

The second problem is a combined mobile and immobile pre-earthquake facility location problem. Each facility is used in the relief distribution operation. It's incontrovertible that due to earthquakes, some network edges collapse and corresponding areas may lose their accessibility. Thus, it's assumed that people on intact and accessible edges travel to the location of the distribution centers to receive the relief. For those who are located on collapsed or inaccessible network edges, the medium-scale Unmanned Aerial Vehicle (UAV) helicopters are utilized in the relief distribution operation. The mathematical model developed for this problem minimizes the aggregate traveling time for both people and UAVs over a set of feasible scenarios. In order to demonstrate the applicability of the model developed, a case study based on Tehran earthquake scenarios is presented.

Since network location problems are NP-hard, three metaheuristic algorithms including genetic algorithm, memetic algorithm, and simulated annealing are investigated and developed to solve the proposed problems.

Keywords: Facility location problem, Distributed demand, Queuing theory, Humanitarian logistics, UAV, Metaheuristics

ÖZ

1960'lardan itibaren tesis yer seçimi problemi çok sayıda araştırmacı tarafından incelenmiştir. Tesis yer seçimi günümüzde de yöneylem araştırmasının en önemli dallarından olup, farklı uygulamalarda örneğin, depo, tehlikeli madde sahaları, bankamatikler (ATM), kıyı arama ve kurtarma istasyonlarının yer seçimi gibi alanlarda kullanılır. Ayrıca, tesis yer seçimi uygulamalarından olan acil durum lojistiğinde en uygun servis merkezi yerinin belirlenmesi problemi de son zamanlarda çok yaygınlaşmıştır.

Bu çalışmada taleplerin şebeke (ağ) boyunca birbiçimli (uniform) dağılımlı olduğu varsayımı ile iki şebeke yer seçimi problemi üzerinde çalışılmıştır. Her iki problemde de, birkaç aday yer arasından birkaç tesis seçilecektir. Birinci problem, çoklu sunucu tıkanık bir tesis konum sorunudur. Taleplerin Poisson sürecine göre oluşturulduğu varsayılmıştır. Ayrıca, kurulu tesislerin her birinde bulunan sunucu sayısı bir karar değişkeni olarak kabul edilmiş ve her sunucu için servis süresi üssel (exponential) bir dağılım izlemektedir. Kuyruk sistemi analizi kullanılarak, müşterilerin beklenen toplam seyahat süreleri ve beklenen toplam bekleme sürelerini en aza indirgemek için bir matematiksel model geliştirilmiştir.

İkinci problem ise, hareketli ve sabit olmak üzere birleşik deprem öncesi tesis konum problemidir. Her tesis yardım dağıtım işleminde kullanılır. Açıktır ki depremlerden dolayı bazı şebeke arklarının (arc) çökmesi ve bundan dolayı bunlara karşılık gelen alanların erişilebilir olmaları sözkonusu olamaz. Böyle durumlarda, sağlam ve erişilebilir arklardaki insanların yardım alabilmek için dağıtım merkezlerine kendilerinin gittiği varsayılmıştır. Şebekenin çökmüş veya erişimin mümkün olmadığı durumlarda, orta ölçekli İnsansız Hava Aracı (İHA) helikopterleri yardım dağıtım operasyonunda kullanılacaktır. Bu problem için geliştirilen matematiksel model, bir dizi uygun senaryo çerçevesinde, hem insanların hem de İHA'larının toplam seyahat süresini en aza indirir. Geliştirilen modelin uygulanabilirliğini göstermek için Tahran'daki olası deprem senaryolarına dayanan bir vaka çalışması da sunulmuştur.

Şebeke üzerinde tesis yer seçimi problemleri NP-zor olduğundan, önerilen problemleri çözecek genetik algoritma, memetik algoritma ve benzetimli tavlama gibi üç sezgi ötesi (metaheuristic) algoritma araştırılmış ve geliştirilmiştir.

Anahtar Kelimeler: Tesis yeri sorunu, Dağıtılmış talep, Kuyruk teorisi, İnsancıl lojistik, İHA, Sezgi ötesi

DEDICATION

To my beloved daughter, the meaning of my life, Vian

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Gökhan İzbırak for his unwavering support, patience, motivation, and mentorship throughout this study. Despite his busy schedule, he has always made himself available to clarify my doubts and answer my questions.

I would like to extend my thanks to my co-supervisor Assoc. Prof. Dr. Jamal Arkat for his steady supports and guidance. Without his brilliant ideas and vast knowledge it would not be possible to conduct this study.

A very special "thanks" go to Prof. Dr. Béla Vizvári for his invaluable insights and concise on this study. I would like to appreciate him for allowing me to grow as a research scientist.

My gratitude is also extended to Asst. Prof. Dr. Sahand Daneshvar, Assoc. Prof. Dr. Adham Mackieh, Asst. Prof. Dr. Emine Atasoylu, Asst. Prof. Dr. Hüseyin Güden, and Assoc. Prof. Dr. Orhan Korhan for their help and supports through the past years.

I would also like to thank my committee members, Prof. Dr. Serpil Erol, Prof. Dr. Zoltán Lakner, Prof. Dr. Béla Vizvári, Assoc. Prof. Dr. Gökhan Izbırak, and Asst. Prof. Dr. Sahand Daneshvar for serving as my committee members. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

I also wish to acknowledge my invaluable friend, Seyed Mahdi Shavarani for his contribution in one of the research papers extracted from this study.

I owe a lot to my parents for their constant and unconditional support, both emotionally and financially, at every stage of my personal and academic life. Words cannot express how grateful I am to my mother and my father for all of the sacrifices they have made on my behalf. Thank you for all of the advice, love, care, and compassion you have provided.

I would also like to thank my sisters for their reassurance and support. Your prayers for me were what sustained me thus far. Thank you for believing in me and pushing me to strive for the best.

I would like to acknowledge my wife. She has been a constant source of strength and inspiration. There were times during the past four years when everything seemed hopeless and I didn't have any hope. I can honestly say that it was only her determination and constant encouragement that ultimately made it possible for me to see this study through to the end.

Finally I would also like to thank my father-in-law and my mother-in-law for supporting every decision I have made.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Determining the best location for establishing the facilities is a matter of paramount importance in service and production management. The Alfred Weber's classic problem which was formulated in 1909 to determine the location of a warehouse could be considered as one of the first facility location problems (FLP) [1]. In 1964, Hakimi categorized this problem into min-sum and min-max problems and studied the network location problem [2], [3]. Since the seventieth decade, facility location problem has been studied by a myriad number of researchers. Nowadays, it is one of the most prominent branches of operations research which could be applied to a wide variety of cases such as:

- Determining the warehouse location problem in supply chain management to minimize the mean travelling times to the market [4].
- Determining the location of hazardous materials sites to minimize the public exposure risk [5].
- Determining the location of Automated Teller Machines (ATMs) for maximizing the number of covered customers [6].
- Determining the location of coastal search and rescue stations to minimize the maximum rescue time [7].
- Determining the location of distribution centers in humanitarian logistics after a large-scale disaster [8], [9].

As it has been reviewed by Klose and Drexl [10], Hale and Moberg [11], and Boloori Arabani and Farahani [12], FLP could be studied in various subsections. Based on the essence of servers, FLP could be subdivided into two categories: mobile and immobile servers. The mobile servers are predominantly used in emergency location problems. In this case, the servers travel to the location of customers. Baptista and Oliviera [13] presented an ambulance location problem to minimize the total operating costs and maximize the service quality simultaneously. Halper et al. [14] scrutinized the formulations and developed heuristic methods concomitant to mobile facility location problem. Adverse to the mobile case, in the case of immobile facility location problem customers are supposed to visit the servers. Torrent-Fontbona et al. [15] presented a combined heuristic-clustering method which in comparison with the existing solution methods, solves the large immobile facility location problems in a shorter time. Considering both customer and provider points of view, Wang et al. [16] proposed several mathematical formulations for immobile facility location problem.

FLPs could be categorized as discrete or continuous space problems [17]. In discrete space problems, the facilities are located on pre-defined candidate locations. Revelle at al. [18] reviewed different studies concomitant to discrete location problem. In continuous space problems any point in the plane could be opted for establishing the facilities. Brimberg et al. [19] reviewed different mathematical models proposed for continuous space location problem along with the exact and approximate methods developed to solve them.

FLP can be deterministic or stochastic. In stochastic problems, as opposed to deterministic ones, some parameters like demand or cost are uncertain. The uncertain parameters can be described by identifying scenarios or by using probabilistic distributions [20]. Considering different possible disaster scenarios, Mete and Zabinsky [21] studied a stochastic facility location problem to find the location of medical supplies and related inventory levels. Considering capacity restrictions, Bieniek [22] studied a single source facility location problem with stochastic demands with arbitrary distribution. On the premise that the demand is not known, Zare Mehrjerdi and Nabizadeh [23] studied a stochastic location-routing problem with fuzzy demands. Assuming that demands and transportation costs are uncertain parameters, Rahmanian et al. [24] proposed a stochastic location problem with capacity restrictions to minimize the cost of establishing the facilities, the total transportation cost, and the costs concomitant to lost demands. Snyder [25] reviewed the developed mathematical models for stochastic facility location problem.

The most prominent distinguishing aspect of facility location problems is their objective functions. The objective function classifies these problems into three categories: center, covering and median problems.

Center problems minimize the maximum distance or service times [26]. This type of problems was introduced by Hakimi [2] for the first time. Generally the center problems are applied in emergency location problems such as locating ambulance stations, fire stations, etc. Since center problems are NP-hard [27], [28] many metaheuristic algorithms have been applied in order to solve these problems. Mladenović et al. [29] developed Tabu search and Variable neighborhood search algorithms to solve the center problem. Elloumi et al. [30] proposed a new relaxed integer linear programming model for the center problem. Tansel [31] studied different developed exact methods for solving the center problem. Calik [32] expanded the previous work by adding the developed heuristic algorithms.

The objective function of covering models is to maximize the number of covered clients [33]. Regarding the maximum coverage distance or service time restrictions, Church and Revelle [34] presented a maximal covering location problem. They developed some heuristic algorithms to solve the proposed model. Considering the partial coverage of clients, Berman and Krass [35] proposed a generalized maximal covering problem. Berman et al. [36] proposed a gradual covering decay problem. According to two different coverage distances, a small one along with a large one, they assumed that the full coverage is gradually decreased based on the distance between clients and the facility they are assigned in. Villegas et al. [37] presented a bi-objective facility location problem in which the first objective is to minimize the operational costs and the second objective maximizes the coverage of purchasing centers. Different types of covering location problem have been reviewed by Li et al. [38].

Median problems refer to those problems wherein the objective function is to minimize the total travelling or waiting times [39]. Baldacci et al. [40] applied lagrangian relaxation to find the lower bound to the capacitated median problem. Kariv and Hakimi [28] proved that the median problem is NP-hard on a general graph. Since exact methods may not be able to solve the median problems, the application of metaheuristic algorithms becomes unavoidable. Different metaheuristic algorithms such as genetic algorithm [41]–[43], simulated annealing [44], and variable neighborhood search [45], [46] have been developed for solving the median problems. Brimberg and Drezner [47] developed a new heuristic algorithm to solve the continuous space median problem. An et al. [48] proposed two-stage robust mathematical models to account for the change in demand and the change in capacity as a result of disruption in median problems. They developed two exact algorithms to solve the problem. Shen et al. [49] studied a median problem in which due to the facility failure, the assignment of customers should be updated. Mladenović et al. [50] reviewed many metaheuristic and exact methods to solve this problem.

Flow capturing location problems (FCLP) are another type of problems which are similar to FLPs. Adverse to studies mentioned before in which it is considered that the customers are located at the network nodes, a network of paths each directed towards a specific flow is considered in FCLPs. Hodgson [51] and Berman et al. [52] were the firsts to introduce these models. Hodgson and Berman [53] applied an FCLP model for locating a billboard in order to maximize the number of customers who can see it. Flow refueling location problems (FRLP) could be considered as one of the branches of FCLP. Kuby and Lim [54] developed a FRLP model for finding the optimal location of p stations in the network in order to maximize the refueling availability of those vehicles which traverse origin-destination routes. They used three metaheuristic methods for solving the problem. Kuby et al. [55] applied the previous model for locating Hydrogen fuel stations in order to maximize the number of vehicles which could utilize them. The inputs of their model are the set of network edges, flow volume between origin and destination, the maximum driving interval needed between fuel stations and the number of stations that should be opened. For dealing with real data, they used Geography Positioning System (GPS) and applied the proposed model in the case of Florida for setting up Hydrogen fuel stations. Arslan and Ekin Karaşan [56] proposed a FRLP model to maximize the vehicle's traveling miles using the electricity. They presented Benders decomposition algorithm accelerated through Pareto-optimal cuts to find the exact solution.

Inspired by FCLPs, Arkat and Jafari [57] proposed a more realistic congested p-median facility location problem in which for receiving the service, customers have to wait in a queue. They assumed that the demands are uniformly distributed along the network edges. Using this idea, two different facility location problems with uniformly distributed demands along the network edges are investigated in this study. On the premise that demands are generated according to Poisson process, a congested median location problem is presented in the first part. This part aims to find the best locations for establishing the facilities and the number of assigned servers in each facility. Considering the probability of link failure, second part scrutinizes a combined mobile and immobile pre-earthquake facility location problem in humanitarian relief logistics. It is assumed that people on intact and accessible edges travel to the location of the distribution centers to receive the relief. For those who are located on collapsed or inaccessible network edges, the medium-scale Unmanned Aerial Vehicle (UAV) helicopters are utilized in the relief distribution operation. Since the discussed problems are NP-hard, some metaheuristic algorithms are proposed as the solution methods.

The remainder of this study is organized as follows. Chapter 2 investigates a multi-server facility location problem with stochastic demands along the network edges. An edge-based stochastic facility location problem in UAV-supported humanitarian relief logistics is described in chapter 3. In order to check the applicability of the developed model, a case study based on feasible earthquake scenarios in Tehran is presented in this chapter.

Chapter 2

MULTIPLE-SERVER FACILITY LOCATION PROBLEM WITH STOCHASTIC DEMANDS ALONG THE NETWORK EDGES

2.1 Introduction

For the first time, Larson [58], [59] challenged the deterministic essence of real life FLPs by presenting the idea of congestion. In congested facility location problem (CFLP), service times are considerable in comparison with customer’s arrival intervals. So customers need to wait in a queue in order to receive the service. CFLPs can be classified according to different criteria like the number of servers in each facility, rules for assigning customers to the facilities, and the probabilistic distribution of arrival and service times [60].

Although CFLP could be considered as a new branch in location science, it has been studied by a myriad number of researchers. Li et al. [61] utilized a dynamic programming approach to minimize the total latency of searching the target web server by finding the optimal location of web proxies. Considering the demand and delay restrictions, Gautam [62] proposed a model to determine the optimal number and location of proxy servers in a manner that the total operational cost is minimized. Aboolian et al. [63] presented a congested center location problem in which more than one server could be assigned to each open facility. On the premise that each customer is assigned to the closest open facility, they developed a mathematical model to minimize the maximum elapsed time for each customer, including the travelling times and waiting times to receive the service.

Marianov and Serra [64] presented several stochastic congested maximal covering location-allocation models by considering the waiting time restriction. Marianov and Serra [65] studied the congested hierarchical facility location problem in which some of the served customers require the secondary type of services. They presented two mathematical models. Subject to the queue length restriction, the first one which is a hierarchical queuing set covering location model determines the minimum number of required servers along with their locations. The second model is a hierarchical queuing maximal covering location formulation which maximizes the number of covered customers. Marianov and Serra [66] extended the previous study by considering multiple servers assigned to each facility. According to some waiting time restrictions they determined the best locations for establishing the facilities and then, they used a set covering model to find the minimum number of required servers for each facility. Considering the waiting time restrictions, Shavandi and Mahlooji [67] presented a single server maximal covering location-allocation model in a fuzzy environment. Using the M/G/1 queuing systems, Hamaguchi and Nakadeh [68] formulated a model in order to maximize the total number of covered demands. Moghadas and Taghizadeh Kakhki [20] studied a congested maximal covering location-allocation problem according to M/M/C queuing framework in which the total budget concomitant to establishing the facilities and the average customers' waiting time are restricted. By considering the budget constraint, Hu et al. [69] presented a model according to M/M/C queuing system with the objective function of maximizing the covered demands. Farahani et al. [70] compiled different versions of covering problems.

Wang et al. [71] scrutinized a single-server congested median location problem according to M/M/1 queuing system. On the premise that each customer is assigned to its closest open facility, they proposed a mathematical model to minimize the aggregate customers’ waiting and travelling times by restricting the maximum allowed waiting time in the system. Berman and Drezner [44] expanded the previous model by considering multiple-servers assigned to each open facility. On the premise that the number of servers assigned to each facility is not restricted, they used several heuristic methods including genetic and simulated annealing algorithms to solve the proposed model. Marianov et al. [72] developed a multiple-server capacitated queue model for a congested median problem in which service times follow the Erlang distribution. Their developed model follows the M/Er/C/N queuing system. The application of bi-objective models is a sought approach in median CFLPs. Marianov and Serra [73] proposed a bi-objective multiple-server congested median model in which the first objective function minimizes the the travelling costs and the second one, minimizes the waiting costs in the system. The bi-objective mathematical model for the congested median location problem proposed by Chambari et al. [74] considers the both customer and server aspects simultaneously. The first objective function minimizes the customers’ travelling and waiting times, and the second one minimizes the servers’ expected idle times. They used the single-server M/M/1/N queuing framework by considering the lower-bound and the upper-bound for the capacity of facilities. Some studies have expanded these bi-objective models by adding one extra objective function to account for pecuniary aspects. Pasandideh et al. [42] studied a single-server congested facility location problem using the M/M/1 queuing framework. They proposed a multi-objective mathematical model to minimize the weighted sum of customers' travelling and waiting times, the maximum servers' idle time, and the costs related to establishing the facilities.

In all of the mentioned studies, the customers are assumed to be located at the network nodes. Inspired by FCLPs, Arkat and Jafari [57] proposed a more realistic p-median facility location problem according to M/M/1 queuing system in which customers are uniformly distributed along the network edges. Since in the majority of real-life problems more than one server is required to satisfy customer demands at each facility, this study expands the model proposed by Arkat and Jafari [57] by incorporating the M/M/C queuing framework in order to reflect a more realistic image of the problem. This expansion transforms the developed mixed integer linear model to a mixed integer nonlinear model which intensely heightens the complexity of the problem and solution methods. After determining the location of open facilities, a predefined number of servers are distributed among the open facilities in order to satisfy customer demands. The objective function minimizes the aggregate expected transportation time between the customers and the open facilities, and the aggregate waiting times for the customers in the system. Applications of such a model include the location of bank branches, post offices, healthcare centers etc. where the number of staff or tellers at each location should be determined.

This chapter is organized as follows. In Section 2.2, the problem is described and a mathematical model is presented. Further, in order to check the model validation, a small problem is solved. In Section 2.3, the characteristics of three applied metaheuristic algorithms including GA, MA, and combined SA are described. In order to examine the applicability of solution algorithms, some numerical examples are generated in Section 2.4. In Section 2.5, the numerical results are reported and the conclusion is drawn in Section 2.6.

2.2 Problem Definition and Assumptions

The problem studied in this study is a facility location problem for immobile facilities where several servers are settled to serve the customers. In this study, it is assumed that demands are generated along the network edges with pre-defined geographical positions. The demands are uniformly distributed along the edges and the time interval between any two consecutive demands follows an exponential distribution with a known parameter. Customers who arrive at a busy server will wait in a queue with innings system. The service time of each server follows an exponential distribution with a defined parameter. In this problem, homological[1]demands have different distances to their closest open facility. Therefore in the queue, based on traveling distance, the demands that are generated sooner may stand after the demands that are generated later. As a result, the time interval between demand generations follows a different distribution with the service time interval, and this difference makes the related model more complex. This problem is a discrete location problem, which means, there are a set of potential locations to set up facilities and from them, a certain number is selected to cover customers' demands and minimize the aggregate customers' expected traveling and waiting times. The total number of available servers which are distributed among the open facilities is known in advance. Due to the fact that the distance between any demand point and the end points of the corresponding edge follows the uniform

[1] Customers who are settled on the same edge are called homological customers.

distribution, customer's movement along each network edge could be assumed as M/U/∞ queuing system. Since the output process of M/G/∞ is Poisson [75], and these outputs enter the facilities in order to receive the service, the queuing system for each facility will be according to M/M/C. The remaining assumptions are summarized as follows:

- The customers' moving speed along all the network edges is identical.
- Potential locations for setting up the facilities are known.
- The servicing system of queue follows FIFO.
- In order to receive the service, each customer goes to the closest open facility.
- Based on the budget constraint, the number of open facilities is known in advance.
- A customer, who arrives at a facility which is too crowded, could not cancel receiving the service or skip the facility to go to another one.

For more clarification, Figure 2.1 presents the problem schematically.

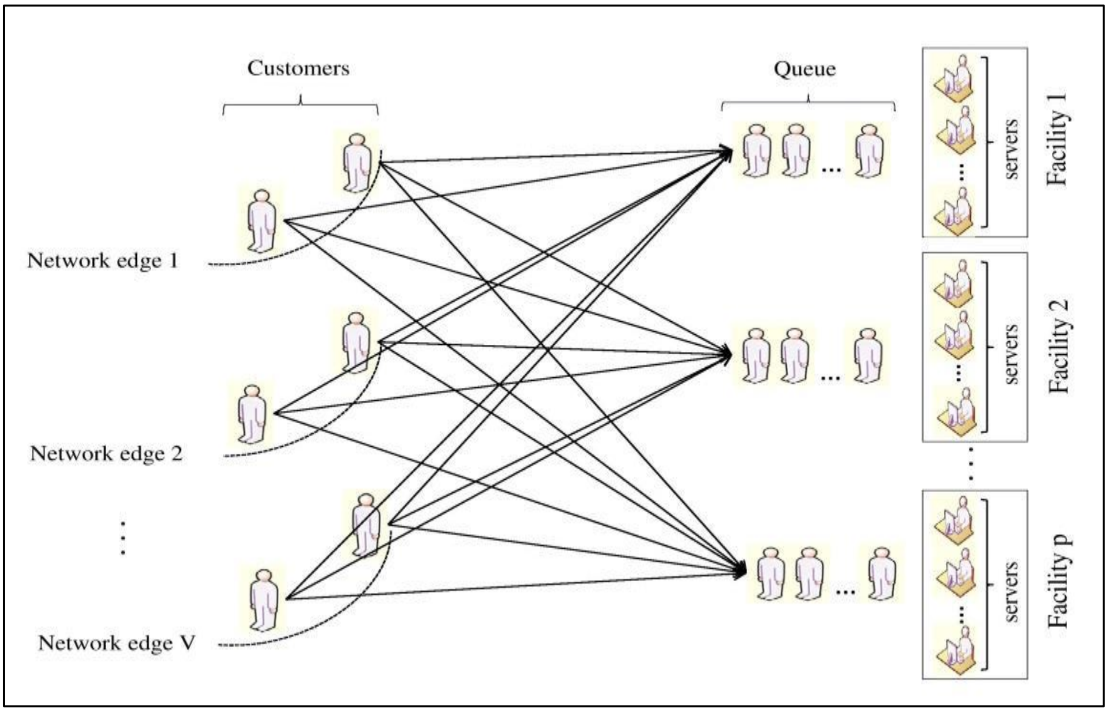

Figure 2.1: Congested facility location problem scheme

2.2.1 Model Formulation

In order to develop the mathematical model, the sets, parameters, and decision variables are defined as follows:

Sets:

- $G\ (V, A)$: A network comprised of the set of nodes (V) and the set of edges (A)
- V: The set of network nodes ($v, v' \in V$)
- A: The set of network edges ($A \subseteq V \times V$, $(v, v') \in A$)
- J: The set of candidate locations for establishing facilities ($j, j' \in J$)

Parameters:

- $l_{vv'}$: The length of edge (v, v')
- $\lambda_{vv'}$: The rate of demand occurrence for edge (v, v')

- t_{vj}: The minimum distance between node v and facility j obtained from the Dijkstra algorithm

Decision variables:

- c_j: Number of servers at facility j
- w_j: The expected waiting time at facility j
- y_j: $\begin{cases} 1 & if\ facility\ j\ is\ open \\ 0 & otherwise \end{cases}$
- x_{vj}: $\begin{cases} 1 & if\ the\ closest\ open\ facility\ to\ node\ v\ is\ facility\ j \\ 0 & otherwise \end{cases}$
- γ_j: The demand entrance rate at facility j
- π_{0_j}: The probability that opened facility j contains no customers (idle probability)
- $b_{vv'jj'}$: The distance between node v and decomposing point of edge (v, v'), if nodes v and v' are respectively assigned to open facilities j and j'

Scalars:

- p: Number of open facilities
- M: A large positive value whose lower bound is the biggest amount of t_{vj}
- M' : A large positive value whose lower bound is c_{max}
- c_{total} : The total number of available servers
- w_{max}: The maximum time that customer could wait in the system
- μ: Common service rate of each server
- ε: An infinitesimal positive number

As it is shown in Figure 2.2, suppose that j and j' are the closest opened facilities to the nodes v and v' respectively ($x_{vj} = x_{v'j'} = 1$). The customers who are located between node v and decomposing point of the edge (v, v') are assigned to facility j. The remaining customers of edge (v, v') are assigned to facility j'.

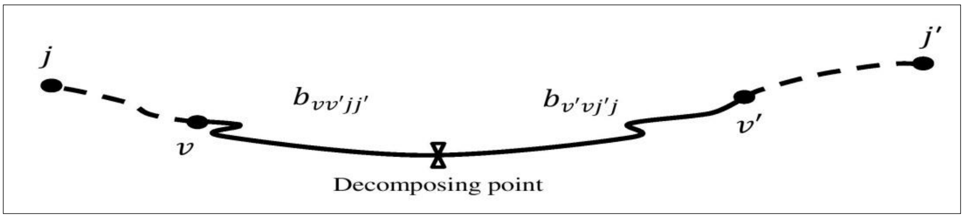

Figure 2.2: A network edge

According to the assumptions of Figure 2.2, the edge (v, v') is decomposed as:

$$t_{vj} + b_{vv'jj'} = t_{v'j'} + b_{v'vj'j} \tag{2.1}$$

Since $b_{vv'jj'} + b_{v'vj'j} = l_{vv'}$, so $b_{vv'jj'} = \frac{t_{v'j'}+l_{vv'}-t_{vj}}{2}$ (2.2)

Lemma 1) if facilities j and j' are the closest opened facilities to the nodes v and v' respectively then: $0 \leq b_{vv'jj'} \leq l_{vv'}$

Proof: suppose that $b_{vv'jj'} < 0$, so:

$$t_{v'j'} + l_{v'v} - t_{vj} < 0 \qquad \rightarrow \qquad t_{vj} > (t_{v'j'} + l_{v'v} = t_{vj'}) \tag{2.3}$$

It means that instead of facility j , now j' is the closest facility to the node v and it is in contradiction with our assumptions. The same result could be obtained for the case in which $b_{vv'jj'} > l_{v'v}$.

Since customers' moving speed along all of the network edges is identical, instead of time needed to traverse the edges, the length of edges is used. The number of generated demands along each network edge is related to the length of the edge, so for each edge, the aggregate expected customers' traveling time for reaching one of the end points of the edges could be calculated as:

$$T = \lambda \int_0^L \frac{x}{L} dx = \frac{\lambda}{L} \int_0^L x dx = \lambda \frac{L}{2} \tag{2.4}$$

By applying these assumptions, the proposed mathematical model is as follows:

$$Min \ \sum_{j \in J} \sum_{j' \in J} \sum_{(v,v') \in V} \frac{\lambda_{vv'} b_{vv'jj'}}{l_{vv'}} \left(t_{vj} + \frac{b_{vv'jj'}}{2} \right) x_{vj} x_{v'j'} + \sum_{j \in J} \gamma_j w_j \tag{2.5}$$

subject to:

$$\sum_{j \in J} y_j = p \tag{2.6}$$

$$\sum_{j \in J} x_{vj} = 1 \qquad \forall \, v \in V \tag{2.7}$$

$$x_{vj} \leq y_j \qquad \forall \, v \in V, \forall j \in J \tag{2.8}$$

$$\sum_{j' \in J} x_{vj'} t_{vj'} \leq t_{vj} + M(1 - y_j) \qquad \forall \, v \in V, \forall j \in J \tag{2.9}$$

$$2b_{vv'jj'} = (t_{v'j'} + l_{vv'} - t_{vj}) x_{vj} x_{v'j'} \qquad \forall (v, v') \in A, \forall j \in J, \forall \, j' \in J \tag{2.10}$$

$$\gamma_j = \sum_{j' \in J} \sum_{(v,v') \in V} \frac{\lambda_{vv'} b_{vv'jj'}}{l_{vv'}} x_{vj} x_{v'j'} \qquad \forall j \in J \tag{2.11}$$

$$\sum_{j \in J} c_j = c_{total} \qquad \forall j \in J \tag{2.12}$$

$$c_j \leq M' y_j \qquad \forall j \in J \tag{2.13}$$

$$c_j \geq \frac{\gamma_j}{\mu} + \varepsilon \qquad \forall j \in J \tag{2.14}$$

$$\pi_{0_j} = \left(\left(\sum_{n=0}^{c_j-1} \frac{\gamma_j^n}{\mu^n n!} \right) + \frac{\gamma_j^{c_j}}{\mu^{c_j} c_j! \left(1 - \frac{\gamma_j}{\mu c_j} \right)} \right)^{-1} \qquad \forall j \in J \qquad (2.15)$$

$$w_j = \frac{1}{\mu} + \left(\frac{\gamma_j^{c_j}}{\mu^{c_j} (c_j!)(c_j \mu)(1 - \frac{\gamma_j}{\mu c_j})^2} \right) \pi_{0_j} \qquad \forall j \in J \qquad (2.16)$$

$$w_j \leq w_{max} \qquad \forall j \in J \qquad (2.17)$$

$$x_{vj}, y_j \in \{0,1\} \qquad \forall v \in V, \forall j \in J \qquad (2.18)$$

$$\gamma_j \geq 0, \ c_j \geq 0, \ b_{vv'jj'} \geq 0 \qquad \forall (v, v') \in A, \forall j \in J, \forall j' \in J \qquad (2.19)$$

At Eq. (5), the customers' aggregate expected traveling times and waiting times are minimized. Constraint (2.6) ensures that among all candidates, p locations are selected to establish the facilities. Constraint (2.7) guarantees that a unique facility is assigned to each node. Constraint (2.8) shows that no customer is assigned to closed facilities. Constraint (2.9) assures that each customer will be assigned to the closest open facility. Constraint (2.10) calculates the decomposing point for each network edge. Constraint (2.11) calculates the demand entrance rate for each open facility. Constraint (2.12) ensures that the total number of assigned servers is c_{Total}. Constraint (2.13) guarantees that servers are not assigned to closed facilities. Constraint (2.14) assures that the queuing system will achieve a steady state mode. According to the characteristics of M/M/C queuing systems [75], constraints (2.15) and (2.16) calculate the idle probabilities and expected waiting times at the open facilities. Constraint (2.17) considers an upper bound for customers' expected waiting time at each facility. Constraints (2.18) and (2.19) preserve the binary and nonnegative restrictions on decision variables.

2.2.2 An Illustrative Example

In order to examine the model, according to Figure 2.3, a small network consisting of 7 nodes, 11 edges and 4 candidate locations is presented. The length of each edge is written above it, and the numbers written in parentheses are related to the corresponding rate of demand generation. It is assumed that the common service rate is 8 customers per hour, totally 15 servers are available, maximum allowed waiting time is 25 minutes, and two facilities could be opened. This problem has been coded in general algebraic modeling system (GAMS) and solved by the Bonmin solver. According to the obtained results, 9 and 6 servers should be assigned to facilities established at nodes 3 and 4 respectively.

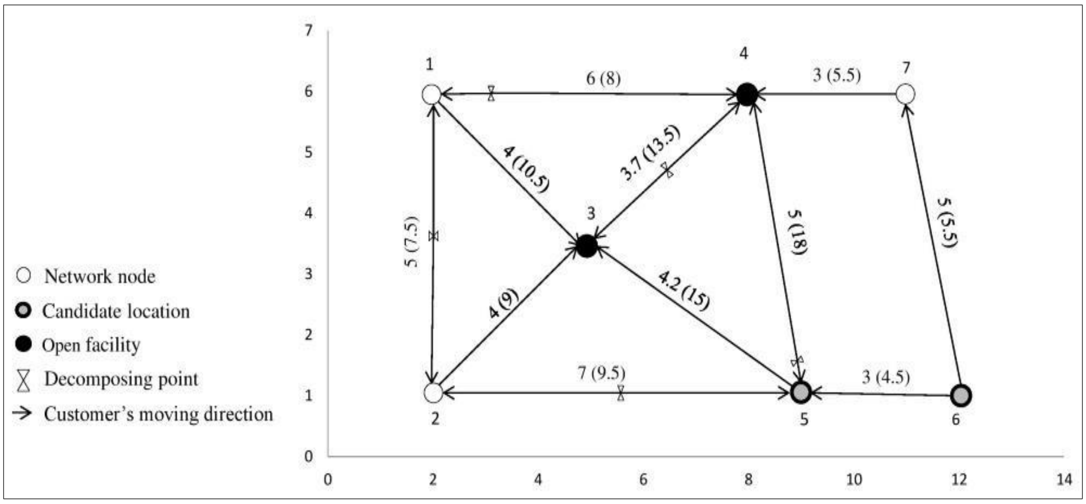

Figure 2.3: A small network

2.3 Solution Methods

As mentioned earlier, the median problems on a general graph are NP-hard problems. Further, a constrained non-linear mixed integer programming model, such as the proposed herein, is considered to be NP-hard [42], [76]. Due to these reasons, finding the exact solution for the proposed model is hard (if not possible). Therefore, in order to solve the problem, in addition to coding the model in GAMS for finding the best solution, three metaheuristic algorithms are developed. The developed algorithms are the genetic algorithm (GA), the memetic algorithm (MA), and the combined simulated annealing algorithm (SA). While describing the applied algorithms, it should be noted that due to some technical restrictions, the proposed model could not be solved by GAMS in its basic form. Coding summation operator with a variable upper bound as used in constraint (2.15) is not allowed in GAMS. Also, the factorial function with variable inputs (constraints (2.15) and (2.16)) could not be coded in GAMS. In order to get rid of these restrictions, after defining a new binary variable (z_{kj}), the following constraint manipulation is used.

$$z_{kj}\colon \begin{cases} 1 & if\ k\ servers\ are\ assigned\ to\ facility\ j \\ 0 & otherwise \end{cases}$$

Constraints (2.12), (2.13), (2.14), (2.15) and (2.16) are replaced with the following constraints respectively.

$$\sum_j \sum_{k=1}^{c_{total}} k z_{kj} = c_{total} \qquad \forall j \in J \qquad (2.20)$$

$$\sum_{k=1}^{c_{total}} z_{kj} = y_j \qquad \forall j \in J \qquad (2.21)$$

$$\sum_{k=1}^{c_{total}} k z_{kj} \geq \frac{\gamma_j}{\mu} + \varepsilon \qquad \forall j \in J \qquad (2.22)$$

$$\pi_{0_j} = \left(\sum_{k=1}^{c_{total}} z_{kj} \left(\sum_{n=0}^{k-1} \frac{\gamma_j^n}{\mu^n n!} + \frac{\gamma_j^k}{\mu^k k! \left(1 - \frac{\gamma_j}{\mu k}\right)} \right) \right)^{-1} \qquad \forall j \in J \qquad (2.23)$$

$$w_j = \frac{1}{\mu} + \left(\sum_{k=1}^{c_{total}} z_{kj} \frac{\gamma_j^k}{\mu^k (k!)(k\mu)(1 - \frac{\gamma_j}{\mu k})^2} \right) \pi_{0_j} \qquad \forall j \in J \qquad (2.24)$$

2.3.1 Genetic Algorithm (GA)

Genetic algorithms, which are derived from observed processes in natural evolution, were first introduced by John Holland in the 1970s. GA is a search technique which starts with a population of random solutions [77]. Each solution is called a chromosome. Through successive iterations, called generations, the chromosomes are evolved and the algorithm is converged to the best chromosome. (see Eiben and Smith [78] for more information). During each generation, the genetic operators such as selection, crossover, and mutation are implemented in order to generate new chromosomes. The main steps of GA applied in this chapter are explained in next subsections.

2.3.1.1 Initialization

In this step, the parameters of GA are initialized. The parameters applied here are population size (Npop), maximum number of iterations (MaxIt), crossover probability (Pc), and mutation probability (Pm).

2.3.1.2 Representation

Encoding is one of the most important steps in designing GA algorithms to create an appropriate definition of solutions. In this study, each solution (chromosome) is shown by a matrix with two rows. The length of each row is equal to the number of open facilities (p), which means that each row has exactly p genes. The allele of each gene in the first row represents the index of locations chosen for establishing the facilities. In the second row, the allele of each gene determines the number of servers assigned to the corresponding facility. Figure 2.4 is an illustration of a chromosome for a problem with 5 facilities and 10 servers. According to the first row of Figure 2.4, facilities are located in locations 3, 9, 7, 4 and 12. Furthermore, according to the second row, there are 2, 1, 3, 2, and 2 servers in the aforementioned facilities, respectively.

First row: Open facilities	3	9	7	4	12
Second row: Assigned servers	2	1	3	2	2

Figure 2.4: Chromosome encoding

2.3.1.3 Initial Population

The initial population is randomly generated. In order to generate each random solution, the first row of the chromosome is randomly filled by non-repetitive indices of candidate locations. In order to fill the second row, firstly one server is assigned to each facility and then, the remaining servers are randomly distributed among the facilities.

2.3.1.4 Fitness Evaluation

The fitness value of each chromosome is computed by Eq. (2.5). Since the considered structure for chromosomes does not guarantee the satisfaction of constraints (2.14) and (2.17), some generated chromosomes could be infeasible. One of the most prominent ways to handle the infeasible solutions is to apply penalty functions [79]. In the case of infeasibility, the penalty function is added to the fitness value of the solution. If the constraint (2.14) is violated, the applied penalty function is [76]:

$$p(x) = \alpha \times max\left\{\left(\frac{\gamma_j}{\mu c_j} - 1\right), 0\right\} \tag{2.25}$$

The penalty function which is considered for violation of constraint (17) is defined as:

$$p(x) = \alpha \times w_j \times Y_j \tag{2.26}$$

In the above equations, α is a big positive number.

2.3.1.5 Parent Selection

The total number of parental chromosomes for carrying out the crossover operator is calculated by $(Pc \times Npop)$. The selection process among the parental chromosomes is based on the roulette wheel procedure. In this method, the parents with better fitness values have a greater chance of being selected. In other words, according to the fitness value, a cumulative probability which shows the chance of each parent for being selected is calculated. (See Kumar [80] for more information)

2.3.1.6 The Crossover Operator

In this section according to one-point-cut crossover operator, two offspring chromosomes are reproduced by mating two parental chromosomes. At the beginning, a crossover point is selected randomly along the length of the mating chromosomes. This point breaks each parent chromosome into two segments. Up to the crossover point, the first segment genes of the first parent chromosome are copied to the first offspring. The remaining genes of the first offspring are taken from the second segment of the second parent chromosome. If the first row alleles of the second parent are present in the offspring chromosome, the first segment genes are copied. In a similar process, the second offspring is produced by exchanging the role of first and second parents. Figure 2.5 shows a graphical representation of the crossover operation.

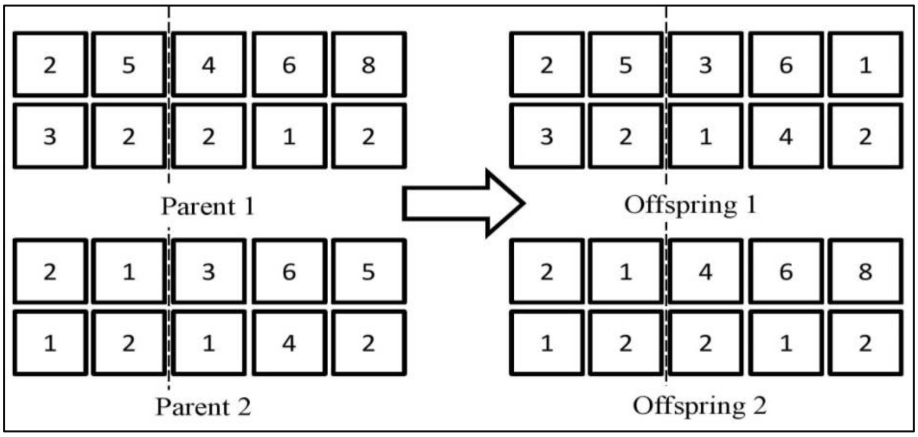

Figure 2.5: An example of crossover operation

2.3.1.7 Repairing Operator

Since in the crossover operation, each offspring inherits the servers directly from its parents, occasionally constraint (2.12) can be violated. While constraint (2.12) is satisfied, the following process is repeated. If the total number of assigned servers (summation of the second-row alleles) is greater than the total number of available servers (c_{total}), in each repetition, a second-row gene containing more than one server is randomly selected and its allele is decreased by one. Otherwise, in each repetition, the number of assigned servers of a randomly selected second-row gene is increased by one.

2.3.1.8 Mutation Operator

The mutation operator is applied to the second row of each chromosome. According to the mutation probability (Pm), two randomly selected second-row alleles are swapped with each other. Figure 2.6 represents the mutation operator.

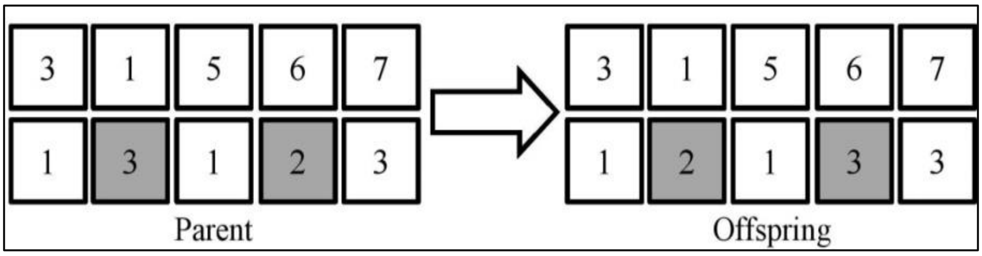

Figure 2.6: An example of mutation operation

2.3.1.9 Replacement and Stopping Criteria

In each iteration, according to the steady state strategy [81], the best offspring generated through crossover and mutation operations, is compared with the worst individual of the current population. If the fitness value of the offspring is better, it replaces. When the algorithm reaches a predetermined number of iterations, the GA is stopped. Algorithm 2.1 shows the pseudo code of the proposed genetic algorithm.

```
Initialize the GA parameters (Npop, Pc, Pm, Maxit)
Generate the initial population
Calculate the fitness value of each individual
S′ ← worst individual
S̄ ← best individual
iterations ← 0
while (iterations < Maxit) do
    use Rollete Wheel to select two parents (P1, P2)
    child1, child2 ← Crossover (P1, P2)
    Repair (child1, child2)
    child3 ← Mutate (child1)
    child4 ← Mutate (child2)
    Calculate the fitness value of each child
    C* ← best child
    if fitness (C*) < fitness (S′)
        C* replaces S′
    end if
    order the new population and update S′ and S̄
    iterations ← iterations + 1
end while
return S̄
```

Algorithm 2.1: The pseudo code of GA

2.3.2 Memetic Algorithm (MA)

Similar to GA, MA is also a population-based metaheuristic search method. MA combines the biological evolution of GA with the individual learning procedures in order to mimic the cultural evolution [82]. These individual learning procedures could be implemented by local search techniques [83]. Therefore, a genetic local search algorithm could be considered as an MA [84].

The MA proposed in this study differs from the applied GA in the application of a local search technique. In order to improve the quality of the generated offspring, after applying the mutation operator to each generation, a local search method is implemented in the MA. In this method, through successive iterations, a predetermined number of neighborhood solutions (localit) are generated for each chromosome. At each iteration of the local search algorithm, the current solution is replaced with a generated neighborhood solution, which has a better fitness value. In order to generate neighborhood solutions, at first, a random integer number (R) in the $[1, \left\lceil \frac{p}{4} \right\rceil]$ interval is generated. Then, in the first row of the current solution, the alleles of R randomly selected genes are replaced with the candidate location indices which are not present in this solution. The second-row alleles corresponding to exchanged genes are replaced with randomly generated integer numbers in the $[\left\lceil \frac{c_{total}}{p} \right\rceil, \left\lceil \frac{c_{total}}{0.5 \times p} \right\rceil]$ interval. At the end, the generated neighborhood solution is repaired in a manner explained in step 2.3.1.7 of the developed GA. Figure 2.7 shows the manner of generating neighborhood solutions.

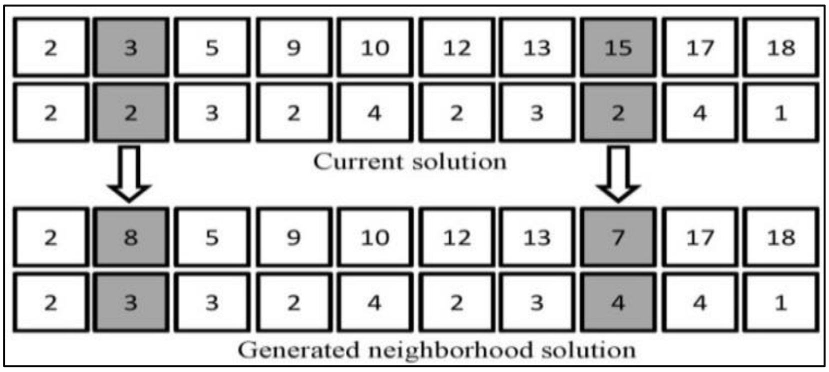

Figure 2.7: An example of local search operation

Algorithm 2.2 shows the pseudo code of local search procedure. In order to construct the MA, this procedure should be added between steps 11 and 12 of Algorithm 2.1.

```
j ← 3
while (j < 5) do
    child(j+2) ← child(j)
    local iteration ← 0
    while (local iteration < Localit) do
        childlocal ← local Search (child(j+2))
        repair (childlocal)
        if fitness(childlocal) < fitness (child(j+2))
            child(j+2) ← childlocal
        end if
        local iteration ← local iteration + 1
    end while
    return child(j+2)
    j ← j + 1
end while
```

Algorithm 2.2: The pseudo code of local search procedure

2.3.3 Simulated Annealing (SA)

Simulated annealing is a metaheuristic method based on the local search techniques in order to approximate the global optimum solution. SA refrains from being trapped in local minima by accepting worse solutions according to a certain probability [85]. Due to the good quality of the solutions found by SA, this algorithm is applied to solve complicated combinatorial optimization problems in a wide variety of areas. SA was independently introduced by Kirkpatrick et al. [86] and Černý [87].

In this chapter, a combined SA algorithm with an inner layout algorithm (ILA) and an outer layout algorithm (OLA) is developed [88]. OLA optimizes the location of open facilities and ILA optimizes the number of assigned servers. The temperature (T) is set to be in the initial level (T_0) in the first step of the proposed SA. The algorithm starts with an initial solution (S). The representation of solution, generating the initial solution, and computing the fitness values are carried out according to steps 2, 3, and 4 of the developed GA respectively. The global optimum solution ($\bar{S}$) is set to be the initial solution. At each temperature level, through N1 successive iterations, the OLA generates neighborhood solutions (S') from the current solution (S). At each iteration of OLA, in order to generate neighborhood solutions (S'), a random integer number (R1) in $[1,\left\lceil\frac{p}{4}\right\rceil]$ interval is generated. Then, R1 number of randomly selected facilities in the first row of current solution matrix is replaced with the candidate location indices, which are not present in the current solution. If the objective value of S' is less than $\bar{S}$, S' replaces $\bar{S}$. In the next step, S' is compared with S. Let Δ be the difference between the objective values of S' and S, i.e., $\Delta = \mathrm{obj}(S') - obj(S)$. If $\Delta \leq 0$, S' replaces S; otherwise S' replaces S according to a probability ($p = \exp(\frac{-\Delta}{T})$). At each iteration of OLA, ILA is repeated N2 times. At each repetition of ILA, S' is set to be the current solution and S'' is the generated neighborhood solution from the current solution. In order to generate S'', a random integer number (R2) in the $[1,\left\lceil\frac{p}{4}\right\rceil]$ interval is generated. Then the number of servers assigned to R2 randomly selected facilities in the second row of current solution matrix is altered to random integer numbers in the $[\left\lceil\frac{c_{total}}{p}\right\rceil, \left\lceil\frac{c_{total}}{0.5\times p}\right\rceil]$ interval. Since the number of servers has been changed, occasionally constraint (12) can be violated. Thus, as described in step 7 of the developed GA, the repairing operator is applied. The same replacing procedure described in OLA is applied for S' and S'' in the next step. After reaching N1, the temperature is reduced and this process is stopped when the final temperature (T_f) is reached. Algorithm 2.3 shows the pseudo code of proposed SA.

```
Initialize the SA parameters (t_0, α, t_f, N1, N2)
Generate the initial individual solution (S)
Best individual solution (S̄) ← S
T ← t_0
while (T < t_f) do
    C1 ← 1
    while (C1 < N1) do
        S' ← local Search S
        C2 ← 1
        while (C2 < N2) do
            S'' ← local Search S'
            repair (S'')
            if obj (S'')< obj (S')
                S' ← S''
            else if rand (0,1) < e^(-((obj(S'') - obj(S'))/T))
                S' ← S''
            end if
            C2 ← C2 + 1
        end while
        if obj (S') < obj (S̄)
            S̄ ← S'
        end if
        if obj (S') < obj (S)
            S ← S'
        else if rand (0,1) < e^(-((obj(S') - obj(S))/T))
            S ← S'
         end if
         C1 ← C1 + 1
    end while
    T = t_0 × α
end while
return (S̄)
```

Algorithm 2.3: The pseudo code of SA

2.3.4 Parameter Tuning

Since the parameters influence the efficiency and effectiveness of metaheuristic algorithms, it is necessary to adjust them in advance to implement the algorithms. Different parameters used for proposed algorithms with their relative ranges are given in Table 2.1. In order to calibrate the parameters, the Taguchi method is utilized in this study. Since Taguchi proposes fractional factorial experiments, it has been known as an efficient technique for tuning the parameters [89]. This method uses orthogonal arrays in order to investigate a large number of controllable factors with a small number of experiments [90]. This method finds the optimal level of controllable factors by minimizing the effect of noise. In order to evaluate the variation of the response, the signal to noise ratio (S/N) is calculated according to Eq. (2.27) in which, Y denotes the response value and n shows the number of orthogonal arrays.

$$S/_N = -10 \times \log(S(Y^2)/n) \tag{2.27}$$

In this study, the variation is modeled by applying the smaller-is-better response. According to proposed parameter combinations for each algorithm shown in Tables 2.2-2.4, ten test problems of different sizes and specifications are solved five times in order to find the average objective values. Figure 2.8 shows the S/N ratios obtained for GA, MA, and SA, respectively. According to these results, the best parameter combination for each algorithm is found. For example, Figure 2.8(A) shows that for GA, parameters A (Npop), B (Pc), C (Pm) and D (Maxit) are better to be at second, third, third and second levels, respectively.

Table 2.1: Parameter levels

Algorithms	Algorithm parameters	Parameter range	Low(1)	Medium(2)	High(3)
GA	Npop (A)	100 - 300	100	200	300
	Pc (B)	0.7 - 0.9	0.7	0.8	0.9
	Pm (C)	0.3 - 0.5	0.3	0.4	0.5
	Maxit (D)	100 - 300	100	200	300
MA	Npop (A)	50 - 150	50	100	150
	Pc (B)	0.7 - 0.9	0.7	0.8	0.9
	Pm (C)	0.1 - 0.3	0.1	0.2	0.3
	Maxit (D)	50 - 150	50	100	150
	Localit (E)	10 - 30	10	20	30
SA	T_0 (A)	50 - 70	50	60	70
	T_f (B)	0.05 - 1	0.05	0.1	1
	A (C)	0.85 – 0.95	0.85	0.9	0.95
	N1 (D)	10 - 30	10	20	30
	N2 (E)	10 - 30	10	20	30

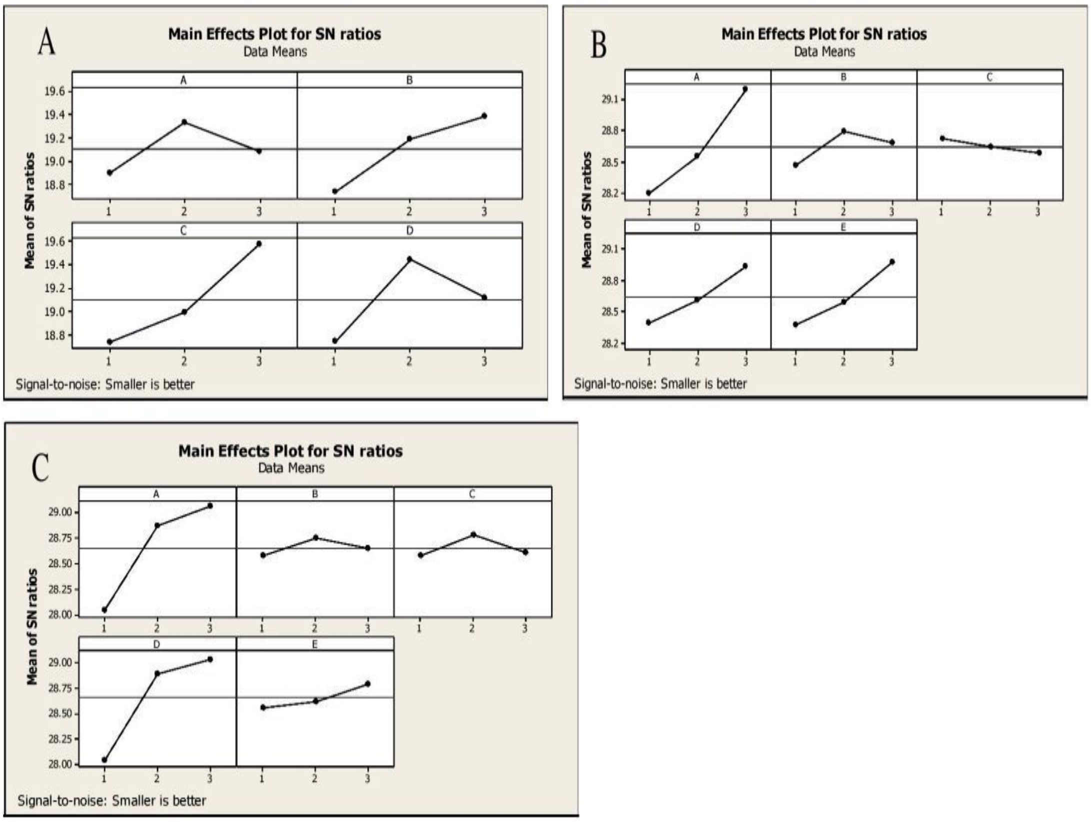

Figure 2.8: S/N ratio plot for: (A) GA parameters; (B) MA parameters; (C) SA parameters

Table 2.2: Computational results for tuning GA

	Algorithm parameters				Obtained responses
Run order	A	B	C	D	Objective
1	1	1	1	1	2721.53
2	1	2	2	2	2311.90
3	1	3	3	3	2197.74
4	2	1	2	3	2407.95
5	2	2	3	1	2229.35
6	2	3	1	2	2217.82
7	3	1	3	2	2229.42
8	3	2	1	3	2423.62
9	3	3	2	1	2401.99

Table 2.3: Computational results for tuning MA

	Algorithm parameters					Obtained responses
Run order	A	B	C	D	E	Objective
1	1	1	1	1	1	2261.27
2	1	1	1	1	2	2176.18
3	1	1	1	1	3	2036.01
4	1	2	2	2	1	2186.32
5	1	2	2	2	2	2055.95
6	1	2	2	2	3	1903.39
7	1	3	3	3	1	2069.74
8	1	3	3	3	2	2035.24
9	1	3	3	3	3	1921.65
10	2	1	2	3	1	1940.55
11	2	1	2	3	2	1998.20
12	2	1	2	3	3	1946.91
13	2	2	3	1	1	2178.23
14	2	2	3	1	2	2014.53
15	2	2	3	1	3	1896.36
16	2	3	1	2	1	1994.45
17	2	3	1	2	2	1961.22
18	2	3	1	2	3	1949.61
19	3	1	3	2	1	1914.17
20	3	1	3	2	2	1980.70
21	3	1	3	2	3	1829.30
22	3	2	1	3	1	1820.30
23	3	2	1	3	2	1731.87
24	3	2	1	3	3	1667.94
25	3	3	2	1	1	1919.07
26	3	3	2	1	2	1877.49
27	3	3	2	1	3	1882.12

Table 2.4: Computational results for tuning SA

	Algorithm parameters					Obtained responses
Run order	A	B	C	D	E	Objective
1	1	1	1	1	1	2537.88
2	1	1	1	1	2	2594.22
3	1	1	1	1	3	2547.36
4	1	2	2	2	1	2265.63
5	1	2	2	2	2	2233.44
6	1	2	2	2	3	2174.11
7	1	3	3	3	1	2378.25
8	1	3	3	3	2	2217.35
9	1	3	3	3	3	2182.28
10	2	1	2	3	1	2005.78
11	2	1	2	3	2	2065.47
12	2	1	2	3	3	2009.64
13	2	2	3	1	1	2230.65
14	2	2	3	1	2	2403.78
15	2	2	3	1	3	2191.38
16	2	3	1	2	1	2253.48
17	2	3	1	2	2	1993.21
18	2	3	1	2	3	2045.04
19	3	1	3	2	1	2136.25
20	3	1	3	2	2	1992.48
21	3	1	3	2	3	2048.07
22	3	2	1	3	1	1969.01
23	3	2	1	3	2	1998.91
24	3	2	1	3	3	2006.97
25	3	3	2	1	1	2138.01
26	3	3	2	1	2	2303.22
27	3	3	2	1	3	2179.23

The Taguchi method is performed by Minitab 16 and the parameter-tuned algorithms are coded in C# programming language and implemented on Intel Xeon E5-2660v2@2.5 GHz computers with 8 GB RAM and 25 MB Cache.

2.4 Instance Generation

In order to generate random networks, which are close to the reality, network edges are classified into three levels based on the crowdedness criteria: crowded, semi-crowded and less crowded edges. For each level, a specific rate is considered. For each network edge, multiplying the length by corresponding crowdedness rate determines the demand generation rate. According to the crowdedness criteria, the majority of candidate locations are selected among those network nodes which are located in crowded areas. For entire generated instances, it is assumed that the length of edges follows a uniform distribution in [1, 10], crowdedness rates are 0.4, 0.8 and 1.2 for less crowded, semi-crowded and crowded edges respectively, common service rate is 20, and maximum allowed waiting time is 0.35. Figure 2.9 represents a network of 550 nodes and 65 candidate locations generated by the mentioned process.

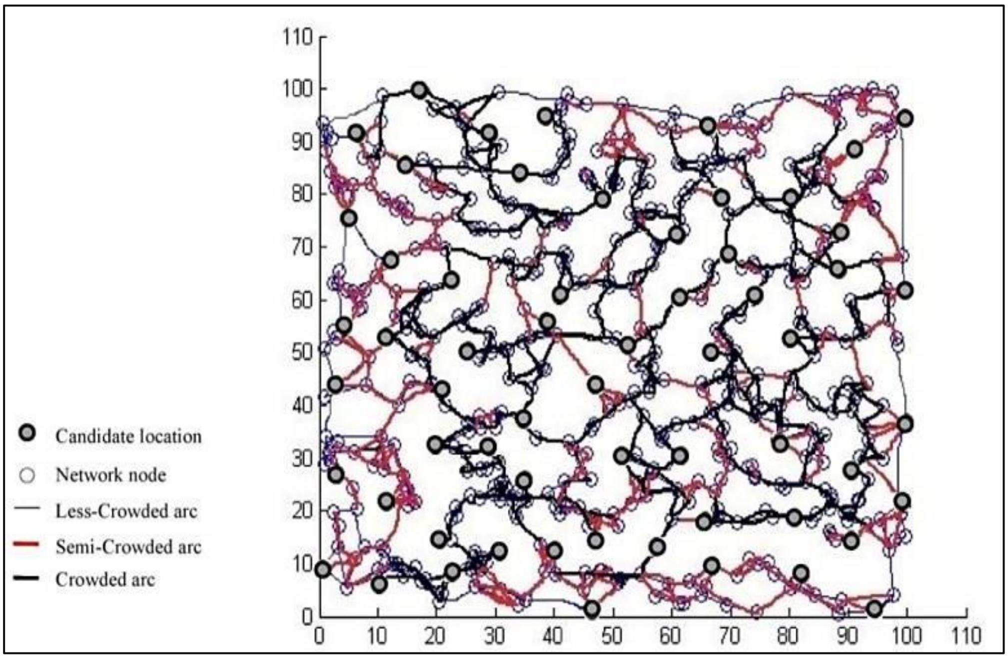

Figure 2.9: A sample random network

2.5 Results and Discussion

The computational result of implementing the proposed solution methods on 24 problems of different sizes and specifications is shown in Table 2.5. In order to mitigate the effects of uncertainty, each algorithm is implemented ten times on each problem and the average and best objective values along with required CPU times of these runs are shown in this table.

As it is shown in Table 2.5, GAMS mathematical programming package failed to solve the majority of developed instances and even for very small size instances which have been solved by GAMS, the required CPU time was not reasonable. Since it is not possible to obtain the global optimum solution for the developed MINLP model, the performances of proposed algorithms are compared together. Figure 2.10 compares the proposed metaheuristic algorithms according to the average, best, and worst objective values along with required CPU times. As illustrated in this figure, MA outperforms GA and SA on the basis of objective function values, while according to required CPU times, GA outperforms the other two algorithms.

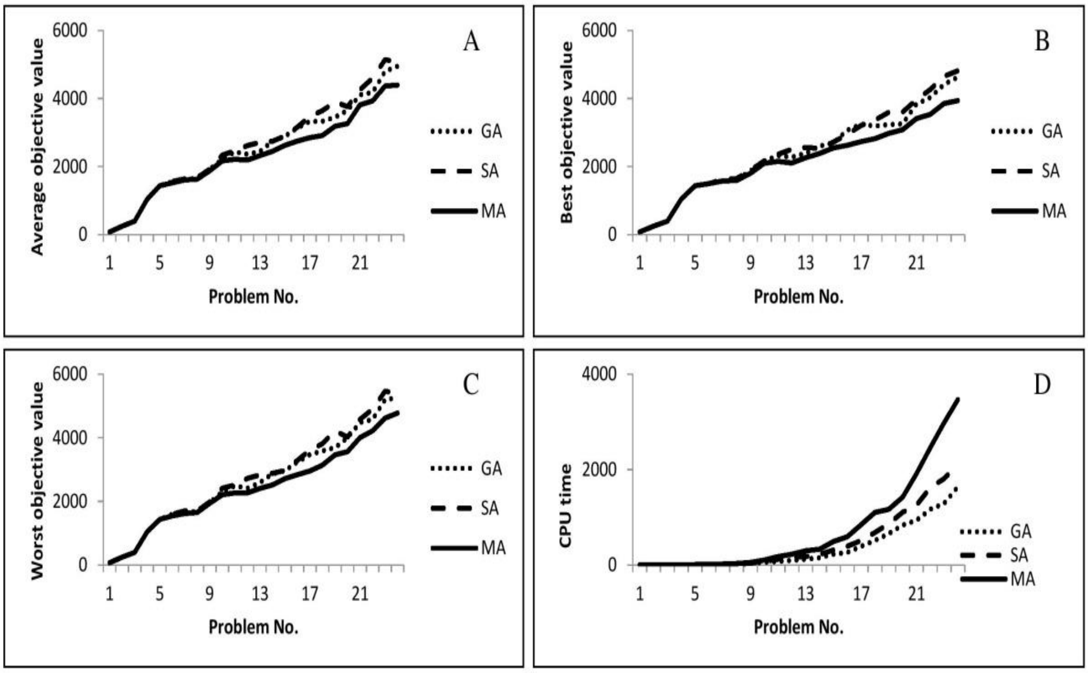

Figure 2.10: (A) Average objective values; (B) Best objective values; (C) Worst objective values; (D) Required CPU time of algorithms for different test problems

Table 2.5: Computational results of solving methodologies

#	V	c_{total}	J	P	Proposed GA			Proposed SA			Proposed MA			GAMS	
					Average	Best	Time	Average	Best	Time	Average	Best	Time	Objective	Time
1	6	10	4	2	71.6	71.6	1	71.6	71.6	1	71.6	71.6	1	71.6	2
2	10	12	5	2	243.6	243.6	1	243.6	243.6	2	243.6	243.6	3	243.6	32
3	15	15	10	3	392.6	392.6	1	392.6	392.6	3	392.6	392.6	4	392.6	252
4	20	30	12	3	1041	1041	2	1041	1041	5	1041	1041	6	1041.0	683
5	30	35	15	4	1436.7	1436.7	3	1436.7	1436.7	7	1436.7	1436.7	8	1436.7	1467
6	40	45	20	5	1548.4	1498.4	7	1559.2	1498.4	11	1498.4	1498.4	12	1498.4	2535
7	50	50	25	6	1621.1	1574.2	10	1637.6	1614.7	13	1605.8	1574.2	18	1605.8	3360
8	60	55	27	7	1655.5	1621.3	16	1663.7	1642.3	22	1626.9	1592.7	29	***	***
9	80	65	30	8	1912.6	1873.8	29	1922.2	1882.6	38	1874.3	1803.3	51	***	***
10	100	70	35	10	2217.7	2162.1	58	2338.1	2165.6	74	2163.8	2091.8	105	***	***
11	140	75	40	12	2429.9	2318.4	74	2462.1	2357.3	118	2211.5	2147.3	176	***	***
12	180	85	50	14	2358.3	2275.1	95	2612.9	2506.5	152	2187.6	2102.5	227	***	***
13	220	95	60	16	2447.1	2391.6	119	2712.2	2562.2	182	2320.4	2261.9	294	***	***
14	260	100	65	18	2739.3	2602.7	155	2741.5	2533.3	218	2443.6	2384.4	331	***	***
15	300	110	70	20	2889.4	2695.5	226	2894.2	2714.3	313	2622.8	2549.2	490	***	***
16	330	120	75	21	3139.8	3068.8	266	3162.3	2943.7	393	2748.7	2627.1	592	***	***
17	370	130	80	22	3319.7	3244.1	392	3473.9	3203.9	510	2846.6	2734.2	843	***	***
18	400	140	85	24	3328.2	3192.3	515	3638.8	3361.9	678	2912.4	2821.1	1099	***	***
19	430	150	90	24	3450.6	3227.1	656	3901.7	3592.4	863	3185.3	2793.6	1164	***	***
20	470	160	95	26	3651.7	3254.2	836	3759.2	3602.6	1104	3263.1	3084.7	1423	***	***
21	500	170	100	28	4107.3	3836.8	934	4256.4	3963.9	1252	3814.2	3411.4	1906	***	***
22	550	180	105	30	4176.8	4031.8	1164	4594.1	4264.5	1620	3932.1	3529.6	2455	***	***
23	600	190	110	30	4796.2	4408.9	1288	5138.2	4650.1	1808	4365.5	3848.7	2976	***	***
24	650	200	115	30	4944.7	4629.1	1638	5072.6	4813.5	2100	4392.2	3935.9	3464	***	***

Sign (***) denotes that GAMS (Bonmin solver) can not find integer solution

In order to compare the performance of the proposed algorithms statistically, one-way analysis of variance (ANOVA) for the average, best, and worst obtained objective values along with the required CPU times was performed by using SPSS software. According to Table 2.6 obtained from ANOVA, algorithms significantly differ in the objective values and the required CPU times. Table 2.7 shows the results of Tukey's test for multiple comparisons of the proposed algorithms. As it can be seen from this table, while GA outperforms SA, and SA outperforms MA in terms of required CPU times, MA outperforms GA, and GA outperforms SA in terms of objective values.

Table 2.6: ANOVA for performance comparisons

		Sum of Squares	df	Mean Square	F	Sig.
Average objective value	Between Groups	0.012	2	0.006	50.724	0.000
	Within Groups	0.008	69	0.000		
	Total	0.021	71			
Best objective value	Between Groups	0.011	2	0.006	43.070	0.000
	Within Groups	0.009	69	0.000		
	Total	0.020	71			
Worst objective value	Between Groups	0.013	2	0.006	55.623	0.000
	Within Groups	0.008	69	0.000		
	Total	0.021	71			
CPU Time	Between Groups	0.69	2	0.345	247.010	0.000
	Within Groups	0.087	69	0.001		
	Total	0.777	71			

Table 2.7: Tukey's test for multiple comparisons

Tukey's Test	Algorithm Method	N	Subset for alpha = 0.05		
			1	2	3
Average objective value	MA	24	0.3161		
	GA	24		0.3360	
	SA	24			0.3478
Best objective value	MA	24	0.3163		
	GA	24		0.3375	
	SA	24			0.3461
Worst objective value	MA	24	0.3154		
	GA	24		0.3368	
	SA	24			0.3476
CPU time	GA	24	0.2182		
	SA	24		0.3241	
	MA	24			0.4579

2.6 Conclusion and Future Work

In this chapter, an MINLP model is developed for multiple-server facility location problem with stochastic and uniformly distributed demands within M/M/C queuing framework. It has been assumed that each customer is assigned to the closest open facility. Considering distributed demands along the network edges increases the complexity of the developed mathematical model. Since the proposed model is NP-hard, in addition to utilization of GAMS optimization compiler, three metaheuristic algorithms including GA, SA, and MA were proposed to solve the model. In order to tune the parameters of these algorithms, the Taguchi method was used and then, the parameter-tuned algorithms were implemented on 24 test problems of different sizes and characteristics. Finally, in order to compare the performance of the proposed algorithms, one-way ANOVA method and Tukey's test were applied. The obtained results demonstrate that although GA requires less CPU times, MA finds better solutions according to the objective function values.

For the future research, the problem can be modeled as a multi-objective problem in order to consider both customer and server aspects simultaneously. Furthermore, one can utilize other queuing frameworks by considering different distributions for demand generation, different service time distributions, or the capacity restrictions on facilities. In this case, the application of simulation optimization algorithms would be of great interest. Developing other heuristic or metaheuristic algorithms may also result in better solutions in shorter times.

Chapter 3

AN EDGE-BASED STOCHASTIC FACILITY LOCATION PROBLEM IN UAV-SUPPORTED HUMANITARIAN RELIEF LOGISTICS: A CASE STUDY OF TEHRAN EARTHQUAKE

3.1 Introduction

According to United States Geological Survey (USGS), there are several million earthquakes happening each year in all over the planet earth, and around 15 earthquakes with the magnitude of higher than 7. There have been more than 800 thousand deaths due to earthquakes from 2000 to 2015 [91]. Some catastrophic instances during recent years include the earthquake of Great Sichuan with 70 thousand casualties [92], Haiti with 230 thousand kills [93], and Japan (Great East) with about 16 thousand lost lives [94]. The significance of this issue is further highlighted by Nemiroff and Bonnel [95] who declare that most populated cities are located on risky faults.

Following a major earthquake, in order to decrease the fatality rate, supplying survivors with relief in a timely manner is integral. Emergency relief may contain medicines, food, water and shelter [96]. Major objectives of post-earthquake relief are defined as the distribution of relief crews and resources, locating the relief center, and optimization of transportation routes in many studies [97], [98]. Khalil et al. [99]

consider the well-planned provident disaster response as the only solution to minimize economic and fatality losses. In this regard, Eguchi [100] claims that prompt disaster response should be provided for affected zones using innovative technologies and developing required service centers.

As a salient branch of operation research, facility location problem (FLP) is predominantly utilized in devising long-term and strategic decisions for governments or private-sector companies. The majority of facility location models in emergency logistics combine the selection process of locations with pre-positioning, evacuation or relief distribution [101]. Since facility location decisions determine the number and location of distribution centers and the amount of relief supply stocks held therein, they obviously affect the response time and costs of relief distribution operations [102].

As some examples of the application of facility location problem in emergency logistics, Balcik and Beamon [102] developed a maximal covering location problem by incorporating location and inventory decisions. Moghadas et al. [103] proposed a multiple-server congested covering model for the emergency location problem with the waiting time restriction.

Mirchandani [104] was one of the pioneers in proposing the median model for emergency problems. Paul and Batta [105] developed a capacitated model in order to minimize the average traveling time of a casualty to the closest hospital. The model proposed by Duran et al. [106] minimizes the expected average response time by restricting the maximum allowed number of opened sites. Görmez et al. [107] presented a bi-objective location model in order to minimize both average distances between facilities and their closest open facilities and the number of opened facilities simultaneously. Verma and Gaukler [108] suggested both deterministic and stochastic models in order to minimize the expected transportation cost overall disaster scenarios.

The facility location model developed by Jia et al. [109] generalizes covering, median and center models, each propitious for different needs in a large scale disaster. Lu and Sheu [110] proposed a discrete center model in order to minimize the worst-case deviation in maximum traveling time from the optimal solution.

Link failure is one of the consequences of large-scale disasters, which affect the effectiveness of emergency operations. Selçuk and Yücemen [111] developed probabilistic models in order to evaluate the seismic reliability of lifeline networks of water distribution system in Bursa, Turkey. VanVactor [112] considered relief network as a matter of crucial importance in relief distribution and claimed that any damage to this network can seriously compromise the operations. Earthquakes would damage access routes and make areas inaccessible; furthermore, power and water flow would be disconnected, and buildings would be destroyed or damaged [113]. Kamp et al. [114] reported that in an earthquake in Kashmir even with the availability of supplies, the condition of roads made the relief distribution impossible for blocked areas. Eiselt et al. [115] proposed a location model of a network wherein one link or one node could fail. The objective function of their model was to minimize the expected disconnected demands. Melachrinoudis and Helander [116] developed a single facility location model with unreliable links in order to maximize the reachable nodes. Salman and Yücel [117] presented a covering location model for Istanbul in order to maximize the expected demand coverage by considering dependent link failures.

As a solution to link failure, helicopters and unmanned aerial vehicles (UAV) have been widely used for post-disaster operations and pre-disaster assessments. On the premise that the helicopters could be used to deliver the medical care items and evacuate the injured people, Ozdamar [118] proposed a mathematical model to minimize the total flight time along with the required load/unload times. Qi et al. [119] employed the search and rescue rotary-wing UAVs (SR-RUAV) in order to support rescue teams in tracking down damaged buildings rapidly to increase survival rates. Nedjati et al. [120] addressed ground transportation difficulties following a disaster due to road's blockage and time limits in the disaster response phase using autonomous small UAV helicopter as relief distributors. Chen et al. [121] suggested application of a UAV system and the aerial image analysis method to evaluate the damage degree of earthquake affected area. In a similar study, Shaodan Li et al. [122] used UAVs to detect Earthquake-Triggered roof holes automatically. By exploiting GSM network and GPS systems, Li et al. [123] provided a UAV tracking method in order to monitor the location of all the UAVs and assign emergency surveying tasks so as cost and time are minimized.

In this chapter, on the premise that the service recipients are uniformly distributed along each network edge, a combined mobile and immobile facility location problem is studied. The problem is a discrete median location problem wherein a pre-defined number of locations are selected among a set of candidate locations in order to establish the facilities. Each facility is used in the relief distribution operation responding to a major earthquake. It’s assumed that people on intact and accessible edges travel to the location of the distribution centers to receive the relief. For those who are located on collapsed or inaccessible network edges, UAV helicopters are utilized in the relief distribution system. The objective function of the proposed mathematical model minimizes the aggregate traveling time for both people and UAVs over a set of feasible scenarios. Since the median problem is NP-hard on a general graph [28] and due to the complexity of developed mixed integer nonlinear model, three metaheuristic algorithms comprising genetic, memetic and simulated annealing are developed to solve the problem and then applied to the case study of Tehran metropolis.

The remainder of the study is organized as follows. Section 3.2 describes the problem and presents a mathematical model. Section 3.3 provides effective algorithms in order to solve the model. Section 3.4 provides the application of the developed model to the case study of Tehran’s earthquake preparedness. Section 3.5 provides conclusion and future work directions.

3.2 Problem Definition and Assumptions

This study studies a combined mobile and immobile facility location problem in order to determine the location of relief distribution centers responding to a major earthquake. Since the studied problem is a discrete facility location problem, it is assumed that the set of candidate locations is pre-known. The total number of open facilities is determined beforehand based on budget limitations. It is supposed that the people are uniformly distributed along each network edge. Due to the paucity of transportation resources after large-scale disasters like the earthquakes, it is presumed that relief recipients on intact and accessible edges travel to the location of deployed facilities, while the demand on collapsed or inaccessible edges is met by UAVs. The demand at each point is satisfied by the closest open facility. If the closest facilities to the endpoints of an edge are not identical, the edge is partitioned into two different segments, each assigned to its closest facility. As the UAVs' flying speed, the traveling speed of people along the network edges is identical and constant. Consequently, instead of traveling time, the traveling distance could be used in proposed calculations. The maximum flying duration of each UAV mission is restricted. Based on the weight of each relief kit and the payload of the UAV, the maximum number of kits which could be delivered in each mission is determined. Since the demand is uniformly distributed, the kits are dropped one by one at equal distances along the network edges. For each edge, UAV flies to the closest node and begins to drop the kits along the edge. After dropping the last kit, if the designated segment of the edge is not covered, UAV returns to the facility in order to be reloaded and then by flying back to the last dropping point, the mission is continued. In this chapter, curvy edges are decomposed to straight segments to enable further calculations. For more clarification, Figure 3.1 presents the problem schematically.

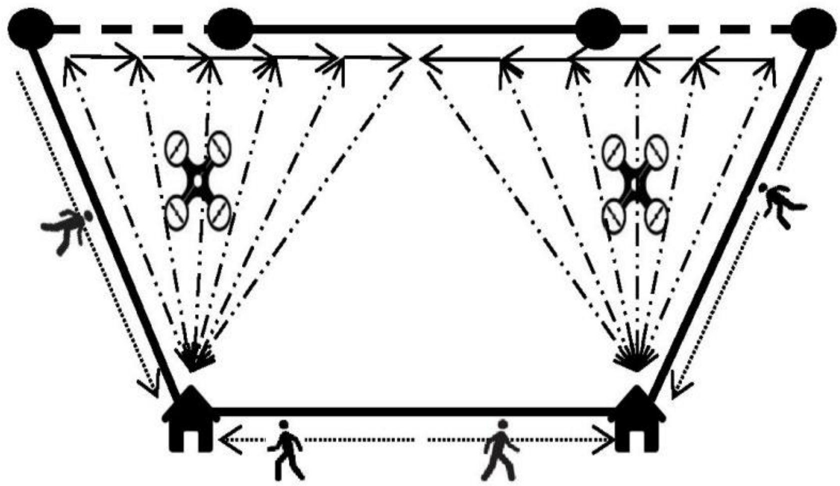

Figure 3.1: The schematic representation of the problem

3.2.1 Model Formulation

In order to develop the mathematical model, the sets, parameters and variables are defined as follows:

Sets:

- $G\ (V, A)$: a network of nodes (V) and edges (A)
- S: the set of scenarios ($s \in S$)
- V: the set of network nodes ($v, v' \in V$)
- A: the set of network edges ($A \subseteq V \times V$, $(v, v') \in A$)
- $A1^s$: the set of intact and accessible edges in scenario s ($A1^s \subseteq A$)
- $A2^s$: the set of collapsed or inaccessible edges in scenario s ($A2^s \subseteq A$)
- J: the set of potential facility locations ($j, j' \in J$)

Parameters and Scalars:

- k: number of open facilities
- M: a large positive value
- C: the maximum number of relief kits which could be delivered in each UAV flight
- E: the endurance of each UAV (time)
- $l_{vv'}$: length of edge (v, v')
- p_s: probability of occurrence of scenario s
- D_v^s:

  $\begin{cases} 1 & if\ node\ v\ has\ overland\ access\ to\ at\ least\ one\ facility\ in\ scenario\ s \\ 0 & otherwise \end{cases}$
- $a_{vv'}^s$: $\begin{cases} 1 & if\ edge\ (v, v')\ is\ still\ intact\ after\ the\ disaster\ in\ scenario\ s \\ 0 & otherwise \end{cases}$

- t_{vj}^{s}: the length of shortest path between node v and facility j in scenario s obtained from Dijkstra algorithm
- q_{jv}: the Euclidian distance between facility j and node v
- $\lambda_{vv'}^{s}$: the demand of edge (v, v') in scenario s
- $\mathrm{r}_{vv'}^{s}$: the segment of edge (v, v') which is covered in one UAV flight in scenario s $\left(\mathrm{r}_{vv'}^{s} = \frac{Cl_{vv'}}{\lambda_{vv'}^{s}}\right)$

Variables:

In the following variables, the subscript $vv'jj'$ indicates that facilities j and j' are the closest open facilities to nodes v and v' respectively. Also, the superscript s represents the corresponding scenario.

- y_j: $\begin{cases} 1 & if\ facility\ j\ is\ open \\ 0 & otherwise \end{cases}$
- x_{vj}^{s}: $\begin{cases} 1 & if\ \ j\ is\ the\ closest\ accessible\ facility\ to\ node\ v\ in\ scenario\ s \\ 0 & otherwise \end{cases}$
- f_{jv}^{s}:

  $\begin{cases} 1 & if\ \ j\ has\ the\ minimum\ Euclidian\ distance\ to\ node\ v\ in\ scenario\ s \\ 0 & otherwise \end{cases}$
- $b_{vv'jj'}^{s}$: the distance between node v and partitioning point of intact and accessible edge (v, v')
- $d_{vv'jj'}^{s}$: the distance between node v and partitioning point of collapsed or inaccessible edge (v, v')
- $n_{vv'jj'}^{s}$: required number of reloads to satisfy the demand of assigned segment of edge (v, v')
- $\gamma_{ivv'jj'}^{s}$: the length of ith reloading trip on collapsed or inaccessible edge (v, v')

- $U_{vv'jj'}^{s}$: a variable to adjust the UAV's flight time regarding the ultimate flight on assigned segment of edge (v, v')
- $g_{vv'jj'}^{s}$: the flight time between partitioning point of edge (v, v') to facility j

In each scenario, on the premise that nodes v and v' are assigned to facilities j and j' respectively, the intact and accessible edge $(v, v') \in A1$, where $A1 = \{(v, v') \in A | D_v + a_{vv'} = 2\}$, is partitioned according to the manner explained in Section 2.2.1. By assuming that nodes v and v' are assigned to facilities j and j' respectively, the collapsed and inaccessible edge $(v, v') \in A2$, where $A2 = \{(v, v') \in A | D_v + a_{vv'} < 2\}$, is partitioned in each scenario according to Figure 3.2.

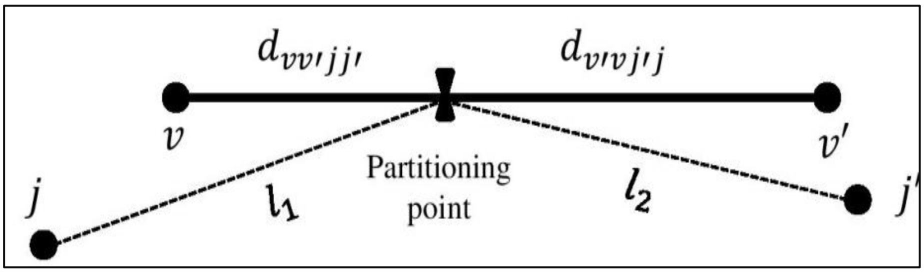

Figure 3.2: Partitioning a collapsed or inaccessible edge

In the case of collapsed or inaccessible edges, the partitioning point is a point along the edge with the same aerial distance to facilities j and j'. It follows that for the partitioning point $l_1 = l_2$. Considering Figure 3.3, l_1 could be calculated as follows:

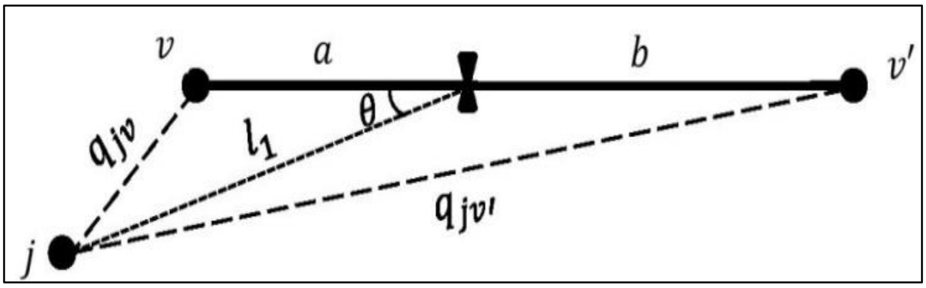

Figure 3.3: Triangular representation of partitioning point

$$q_{jv}{}^2 = a^2 + l_1^2 - 2al_1 \cos\theta \tag{3.1}$$

$$q_{jv'}{}^2 = b^2 + l_1^2 + 2bl_1 \cos\theta \tag{3.2}$$

Substituting $\cos\theta$ from the first equation, it follows:

$$q_{jv'}{}^2 = b^2 + l_1^2 + 2bl_1 \left(\frac{a^2 + l_1^2 - q_{jv}{}^2}{2al_1}\right) \tag{3.3}$$

Solving Eq. 6 for l_1 yields:

$$l_1 = \sqrt{\frac{bq_{jv}{}^2 + aq_{jv'}{}^2 - abl_{vv'}}{l_{vv'}}} \tag{3.4}$$

Replacing actual values for a and b yields:

$$l_1 = \sqrt{\frac{d_{v'vj'j}q_{jv}{}^2 + d_{vv'jj'}q_{jv'}{}^2 - d_{vv'jj'}d_{v'vj'j}l_{vv'}}{l_{vv'}}} \tag{3.5}$$

Using the same logic l_2 is calculated as follows:

$$l_2 = \sqrt{\frac{d_{v'vj'j}q_{j'v}{}^2 + d_{vv'jj'}q_{j'v'}{}^2 - d_{vv'jj'}d_{v'vj'j}l_{vv'}}{l_{vv'}}} \tag{3.6}$$

By considering $l_1 = l_2$, $d_{vv'jj'}$ is calculated as:

$$d_{vv'jj'} = \frac{l_{vv'}\left(q_{j'v}{}^2 - q_{jv}{}^2\right)}{q_{jv'}{}^2 - q_{j'v'}{}^2 + q_{j'v}{}^2 - q_{jv}{}^2} \tag{3.7}$$

In the case where nodes v and v' are assigned to the same facility ($j = j'$), UAV flies from that facility and starts covering the entire edge by passing through the closest endpoint of edge (v, v'). In other words, the edge is not partitioned. For this case, based on the proximity of v or v' to that facility, $d_{vv'jj'}$ is either equal to zero or $l_{vv'}$ and is calculated according to Equation 3.8.

$$d_{vv'jj} = max\left(\frac{q_{jv'} - q_{jv}}{\left|q_{jv'} - q_{jv}\right|}, 0\right) l_{vv'} \tag{3.8}$$

In each mission, if the UAV's capacity (C) is greater than or equal to the demand of assigned segment of edge (v, v') which due to uniformity is calculated by Equation 3.9, the UAV can cover the segment in one flight without any need to reload.

$$\lambda_{d_{vv'jj'}} = \frac{\lambda_{vv'} d_{vv'jj'}}{l_{vv'}} \tag{3.9}$$

Otherwise, it is reloaded $\mathrm{n}_{vv'jj'}$ times:

$$\mathrm{n}_{vv'jj'} = \left\lfloor \frac{d_{vv'jj'}}{r_{vv'}} \right\rfloor \tag{3.10}$$

Each time by dropping the last kit, the UAV covers the length $r_{vv'}$, then heads back to the facility to reload and from the same route, it returns to the last dropping point to continue the mission. Thus, according to Equation 3.4 and Figure 3.4, the length of the return path for i^{th} reloading trip is calculated by Equation 3.11.

$$\gamma_{ivv'jj'} = \sqrt{\frac{ir_{vv'} {q_{jv'}}^2 + (d_{vv'jj'} - ir_{vv'}) {q_{jv}}^2 - ir_{vv'}(d_{vv'jj'} - ir_{vv'}) d_{vv'jj'}}{d_{vv'jj'}}} \tag{3.11}$$

Similarly, the aerial distance between partitioning point of edge (v, v') and facility j is calculated according to Equation 3.12.

$$g_{vv'jj'} = \sqrt{\frac{d_{vv'jj'} {q_{jv'}}^2 + (l_{vv'} - d_{vv'jj'}) {q_{jv}}^2 - d_{vv'jj'}(l_{vv'} - d_{vv'jj'}) l_{vv'}}{l_{vv'}}} \tag{3.12}$$

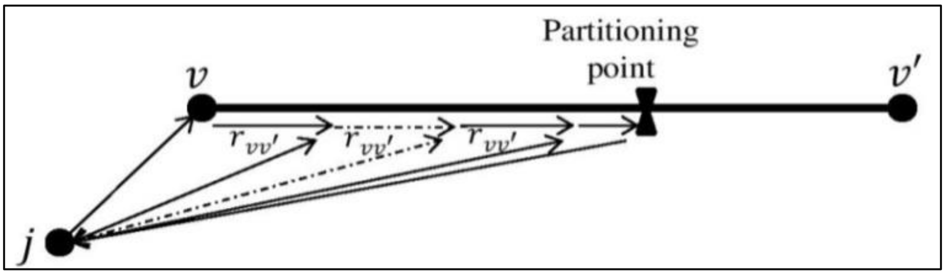

Figure 3.4: Reloading flights on segment

As it is discussed, overall traveled aerial distance over the assigned segment of edge (v, v') is formulated as:

$$T_f = q_{jv} + d_{vv'jj'} + U_{vv'jj'} + 2\sum_{i=1}^{n_{vv'jj'}} \gamma_{ivv'jj'} \tag{3.13}$$

The summation is multiplied by 2 to account for the round trips. It's noticeable that when $\frac{d_{vv'jj'}}{r_{vv'}}$ is an integer, the last term under the summation is equal to $g_{vv'jj'}$ and considering the coefficient 2, the distance $g_{vv'jj'}$ is added twice and must be subtracted. However, when this ratio is decimal, the distance $g_{vv'jj'}$ is not yielded from the summation and therefore it should be added. Hence, variable $U_{vv'jj'}$ is added to the above equation to exert this adjustment and is defined as follows:

$$U_{vv'jj'} = \left(-2\left\lceil\left\lfloor\frac{d_{vv'jj'}}{r_{vv'}}\right\rfloor - \frac{d_{vv'jj'}}{r_{vv'}}\right\rceil - 1\right) g_{vv'jj'} \tag{3.14}$$

According to Equation 3.14, if $\frac{d_{vv'jj'}}{r_{vv'}}$ is integer, $U_{vv'jj'}$ is equal to $(-g_{vv'jj'})$, otherwise it is equal to $(+g_{vv'jj'})$. Using the discussed calculations, the proposed mathematical model is as follows:

$$\begin{aligned} Min\ \ & \sum_{s\epsilon S}\sum_{j\epsilon J}\sum_{j'\epsilon J} p_s \left(\sum_{(v,v')\in A1^s} \frac{\lambda_{vv'}^s b_{vv'jj'}^s}{l_{vv'}} \left(t_{vj}^s + \frac{b_{vv'jj'}^s}{2}\right) x_{vj}^s x_{v'j'}^s + \right. \\ & \left. \sum_{(v,v')\in A2^s} \left(d_{vv'jj'}^s + q_{jv} + U_{vv'jj'}^s + 2\sum_{i=1}^{n_{vv'jj'}^s} \gamma_{ivv'jj'}^s \right) f_{jv}^s f_{j'v'}^s \right) \end{aligned} \tag{3.15}$$

subject to:

$\sum_{j\in J} y_j = k$ (3.16)

$x_{vj}^s \leq y_j \qquad \forall\ v \in V, j \in J, s \in S$ (3.17)

$f_{jv}^s \leq y_j \qquad \forall\ v \in V, j \in J, s \in S$ (3.18)

$\sum_{j\in J} x_{vj}^s = D_v^s \qquad \forall\ v \in V, s \in S$ (3.19)

$\sum_{j\in J} f_{jv}^s = max\left(min\left(1, \sum_{(v'\in V|(v,v')\in A)}(1 - a_{vv'}^s)\right), (1 - D_v^s)\right) \qquad \forall\ v \in V, s \in S$ (3.20)

$\sum_{j'\in J} x_{vj'}^s\, t_{vj'}^s \leq t_{vj}^s + M(1 - y_j) \qquad \forall\ v \in V,\ j \in J, s \in S$ (3.21)

$\sum_{j'\in J} f_{j'v}^s\, q_{j'v} \leq q_{jv} + M(1 - y_j) \qquad \forall\ v \in V,\ j \in J, s \in S$ (3.22)

$2max\left(q_{jv}, g_{vv'jj'}^s\right) + \mathrm{r}_{vv'}^s \leq E \qquad \forall\ (v, v') \in A2^s, (j, j' \in J), s \in S$ (3.23)

$x_{vj}^s, f_{jv}^s, y_j \in \{0,1\} \qquad \forall\ v \in V, j \in J, s \in S$ (3.24)

$n_{vv'jj'}^s \geq 0,\ g_{vv'jj'}^s \geq 0,\ \gamma_{ivv'jj'}^s \geq 0 \qquad \forall\ (v, v') \in A, (j, j' \in J), s \in S$ (3.25)

$b_{vv'jj'}^s,\ U_{vv'jj'}^s,\ d_{vv'jj'}^s\ \ \textit{free in sign} \qquad \forall\ (v, v') \in A, (j, j' \in J), s \in S$ (3.26)

The objective function minimizes the aggregate traveling time of both people and the UAVs. Constraint (3.16) limits the number of opened facilities. Constraints (3.17) and (3.18) ensure that no demand is assigned to close facilities. Constraint (3.19) guarantees that each accessible node is assigned to a unique facility. Similarly, constraint (3.20) does the same for each node that belongs to at least one collapsed edge or each inaccessible node. Constraints (3.21) and (3.22) assign each node to its closest opened facility. Constraint (3.23) reflects the endurance restriction of the UAVs. Constraints (3.24), (3.25), and (3.26) determine the binary, positive, and free in sign variables respectively. As it was described before, $b^{s}_{vv'jj'}$ and $d^{s}_{vv'jj'}$ are positive if $x^{s}_{vj} = x^{s}_{v'j'} = 1$ and $f^{s}_{jv} = f^{s}_{j'v'} = 1$ respectively.

3.3 Solution Methods

As discussed earlier, the median problems on a general graph are NP-hard problems. Further, constrained non-linear mixed integer programming models (MINLP), analogous to the current model, are known to be NP-hard [42], [76]. Since NP-hard problems may not be solved by exact methods, the utilization of metaheuristic algorithms becomes inevitable. Here, in addition to coding the problem in General algebraic modeling system (GAMS), three metaheuristic algorithms consisting of genetic, memetic, and simulated annealing are developed in order to solve the proposed model.

3.3.1 Genetic Algorithm (GA)

The main steps of GA applied in this study are explained in next subsections.

3.3.1.1 Initialization

The parameters of applied GA which are population size ($Npop$), maximum number of iterations ($MaxIt$), crossover probability (P_c), and mutation probability (P_m) are determined in this step.

3.3.1.2 Encoding

As described before, encoding is a step with paramount importance in developing GA in which the solutions are defined appropriately. In this chapter, each solution (chromosome) is shown by a matrix that has one row and k columns. Each column (gene) represents the index of one candidate location, which has been chosen for establishing the facility. Figure 3.5 illustrates a chromosome for a problem with 5 facilities, which are established in candidate locations 5, 2, 7, 4, and 9.

5	2	7	4	9

Figure 3.5: Chromosome encoding

3.3.1.3 Initial Population

The initial population is a set of randomly generated chromosomes. Each chromosome consists of non-repetitive indices of candidate locations.

3.3.1.4 Fitness Evaluation

The fitness value of each chromosome is calculated according to Eq. (3.15). Since the proposed structure of chromosomes cannot stymie the violation of constraint (3.23), some generated chromosomes could be infeasible. The penalty function is a rampant method which is applied to handle the infeasible solutions [79]. If the constraint (3.23) is violated, the applied penalty function which is added to the fitness value of the infeasible solution is [76]:

$$\mathrm{p}_{(x)} = M \times \max\left\{\left(\frac{2max\left(q_{jv}, g_{vv'jj'}\right)+r_{vv'}}{E} - 1\right), 0\right\} \tag{3.27}$$

3.3.1.5 Parent Selection

Here, Roulette wheel procedure is used to increase the selection chance of parents with better fitness values among all $(P_c \times Npop)$ number of parental chromosomes.

3.3.1.6 The Crossover Operator

Here, using the one-point-crossover operator, two parental chromosomes are mated in order to reproduce two offspring chromosomes. The crossover point, which decomposes each chromosome into two segments, is selected randomly. The genes from the first segment of first parent and the second segment of the second parent are used to produce the first offspring. If the second segment genes of the second parent already exist in the first offspring chromosome, they will be replaced by the first segment genes of the second parent. By switching the role of first and second parent, the second offspring is produced in an analogous manner. A graphical representation of the crossover operation is shown in Figure 3.6.

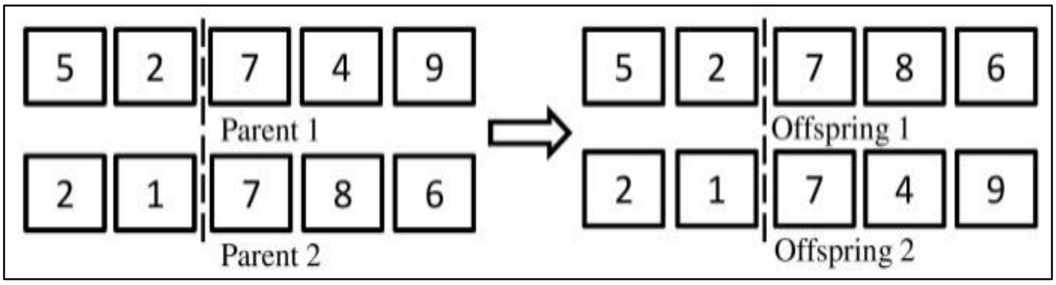

Figure 3.6: An example of the crossover operation

3.3.1.7 Mutation Operator

In this section, according to the mutation probability (P_m), the allele of a randomly selected gene of offspring chromosome is replaced with the index of a randomly selected candidate location which does not exist in this chromosome.

3.3.1.8 Replacement and Stopping Criteria

In each iteration, the replacement process follows the steady-state strategy which compares the best generated offspring with the worst individual of the current population. Algorithm 3.1 depicts the pseudo code of the applied GA.

Initialize the GA parameters ($Npop$, P_c , P_m , $MaxIt$)
Generate the initial population
Calculate the *fitness* value of each individual
$S_{worst} \leftarrow$ worst individual
$S_{best} \leftarrow$ best individual
$iterations \leftarrow 0$
while ($iterations < MaxIt$) **do**
    use *Rollete Wheel* to select two parents (P_1 , P_2)
    $child_1$, $child_2$ $\leftarrow$ *Crossover* (P_1 , P_2)
    $child_3 \leftarrow$ *Mutate* ($child_1$)
    $child_4 \leftarrow$ *Mutate* ($child_2$)
    Calculate the *fitness* value of each child
    $C^* \leftarrow$ best child
    if *fitness* (C^*) < *fitness* (S_{worst})
        C^* replaces S_{worst}
    end if
    order the new population and update S_{worst} and S_{best}
    $iterations \leftarrow iterations + 1$
end while
return S_{best}

Algorithm 3.1: The pseudo code of the proposed GA

3.3.2 Memetic Algorithm (MA)

As described in section 2.3.2, the developed MA implements a local search method to the offspring produced by mutation operation at each generation of the proposed GA. For each chromosome, the applied local search method generates a predefined number of neighborhood solutions ($localit$) through successive iterations. In order to produce the neighborhood solutions, an integer number (r) is randomly generated ($r \epsilon (1, \left\lfloor \frac{k}{3} \right\rfloor)$), then the alleles of r randomly selected genes are replaced with the candidate location indices which do not exist in the current solution. At each iteration of the local search algorithm, if the fitness value of generated neighborhood solution is better than the current solution, it replaces. The pseudo code of applied local search procedure is shown in Algorithm 3.2. The proposed MA is constructed by adding this algorithm to step 10 of Algorithm 3.1.

```
j ← 3
while (j < 5) do
    child_(j+2) ← child_(j)
    local iteration ← 0
    while (local iteration < localit) do
        childlocal ← local Search (child_(j+2))
        if fitness (childlocal) < fitness (child_(j+2))
            child_(j+2) ← childlocal
        end if
        local iteration ← local iteration + 1
    end while
    return child_(j+2)
    j ← j + 1
end while
```

Algorithm 3.2: The pseudo code of local search procedure

3.3.3 Simulated Annealing (SA)

The developed SA in this chapter is analogous to the SA described in section 2.3.3. In the first step of proposed SA, the temperature (T) is set to the initial level (T_0). Using an initial individual solution (S) the algorithm is commenced. The representation of solutions, generation of the initial solution, and objective value calculation are carried out in accordance with Sections 3.1.2, 3.1.3, and 3.1.4 respectively. At the beginning, the global optimum solution $(\bar{S})$ is considered to be identical to the initial solution. At each temperature level, through N consecutive iterations, the algorithm generates neighborhood solutions (S') from the current solution (S) by using a manner explained in Section 3.2. At each iteration, if S' is better than $\bar{S}$, S' replaces $\bar{S}$. In the next step, S' is compared with S. Let Δ be the difference between the objective values of S' and S, i.e., $\Delta = obj(S') - obj(\mathrm{S})$. If $\Delta \leq 0$, S' replaces S; otherwise S' replaces S with probability p ($p = \exp(\frac{-\Delta}{T})$). After N iterations, the temperature is decreased. While the temperature is above T_f, this process continues. Algorithm 3.3 depicts the pseudo code of proposed SA.

```
Initialize the SA parameters (T_0, α, T_f, N)
Generate the initial individual solution (S)
Best individual solution (S̄) ← S
T ← T_0
while (T > T_f) do
    C ← 1
    while (C < N) do
        S' ← local Search S
        if obj (S') < obj (S̄)
            S̄ ← S'
        end if
        if obj (S') < obj (S)
            S ← S'
        else if rand (0,1) < e^(-((obj(S') - obj(S)) / T))
            S ← S'
        end if
        C ← C + 1
    end while
    T = T_0 × α
end while
return (S̄)
```

Algorithm 3.3: The pseudo code of SA

3.3.4 Parameter Tuning

Here the Taguchi method is used to calibrate the parameters of developed metaheuristic algorithms (see section 2.3.4). Table 3.1 shows the parameters applied in proposed algorithms and their relative ranges.

Table 3.1: Parameter levels

Algorithms	Parameters	Parameter range	Low(1)	Medium(2)	High(3)
GA	$Npop$ (A)	100 - 300	100	200	300
	P_c (B)	0.7 - 0.9	0.7	0.8	0.9
	P_m (C)	0.3 - 0.5	0.3	0.4	0.5
	$Maxit$ (D)	600 - 800	600	700	800
MA	$Npop$ (A)	100 - 200	100	150	200
	P_c (B)	0.6 - 0.8	0.6	0.7	0.8
	P_m (C)	0.1 - 0.3	0.1	0.2	0.3
	$Maxit$ (D)	100 - 300	100	200	300
	$localit$ (E)	10 - 20	10	15	20
SA	T_0 (A)	90 - 99	90	95	99
	T_f (B)	0.5 - 1	0.5	0.75	1
	α (C)	0.85 – 0.95	0.85	0.9	0.95
	N (D)	10 - 30	10	20	30

According to proposed parameter combinations for each algorithm shown in Tables 3.2-3.4, for the case of Tehran, which will be described in section 3.4, five problems with different numbers of open facilities are solved twice in order to find the average objective value. These values are normalized through division by their summation and are considered as the response. The Taguchi method is performed by Minitab 16, and the algorithms are coded in Matlab (R2016a) software environment and implemented on an Intel Core i7-6500U@3.1 GHz laptop with 8 GB RAM. Figure 3.7 depicts the S/N ratios acquired from GA, MA, and SA respectively. Based on these results, the best parameter combination for each algorithm is determined. For instance, Figure 3.7 demonstrates that for GA, parameters A, B, C, and D are better to be at first, first, third, and third levels respectively.

Table 3.2: Computational results for tuning GA

Run order	GA parameters				Response
	A	B	C	D	Objective (GA)
1	1	1	1	1	4,415,887.00
2	1	2	2	2	4,403,631.44
3	1	3	3	3	4,395,492.11
4	2	1	2	3	4,399,178.90
5	2	2	3	1	4,434,711.20
6	2	3	1	2	4,443,657.27
7	3	1	3	2	4,458,840.80
8	3	2	1	3	4,447,018.30
9	3	3	2	1	4,489,714.54

Table 3.3: Computational results for tuning SA

Run order	SA parameters				Response
	A	B	C	D	Objective (SA)
1	1	1	1	1	5,289,918.12
2	1	2	2	2	5,285,800.06
3	1	3	3	3	5,136,253.37
4	2	1	2	3	5,601,885.06
5	2	2	3	1	5,410,215.95
6	2	3	1	2	5,451,598.51
7	3	1	3	2	5,108,647.55
8	3	2	1	3	5,175,765.98
9	3	3	2	1	5,343,194.09

Table 3.4: Computational results for tuning MA

Run order	MA parameters					Response
	A	B	C	D	E	Objective (MA)
1	1	1	1	1	1	4,329,345.68
2	1	1	1	1	2	4,315,849.60
3	1	1	1	1	3	4,281,127.24
4	1	2	2	2	1	4,299,257.78
5	1	2	2	2	2	4,280,738.09
6	1	2	2	2	3	4,287,468.15
7	1	3	3	3	1	4,298,650.44
8	1	3	3	3	2	4,280,738.09
9	1	3	3	3	3	4,280,581.64
10	2	1	2	3	1	4,309,138.94
11	2	1	2	3	2	4,294,374.20
12	2	1	2	3	3	4,288,356.35
13	2	2	3	1	1	4,355,182.27
14	2	2	3	1	2	4,311,463.32
15	2	2	3	1	3	4,289,421.97
16	2	3	1	2	1	4,315,413.88
17	2	3	1	2	2	4,286,339.84
18	2	3	1	2	3	4,289,543.77
19	3	1	3	2	1	4,345,355.12
20	3	1	3	2	2	4,296,057.84
21	3	1	3	2	3	4,297,906.32
22	3	2	1	3	1	4,327,163.72
23	3	2	1	3	2	4,281,374.57
24	3	2	1	3	3	4,286,298.89
25	3	3	2	1	1	4,368,969.48
26	3	3	2	1	2	4,310,465.56
27	3	3	2	1	3	4,298,139.12

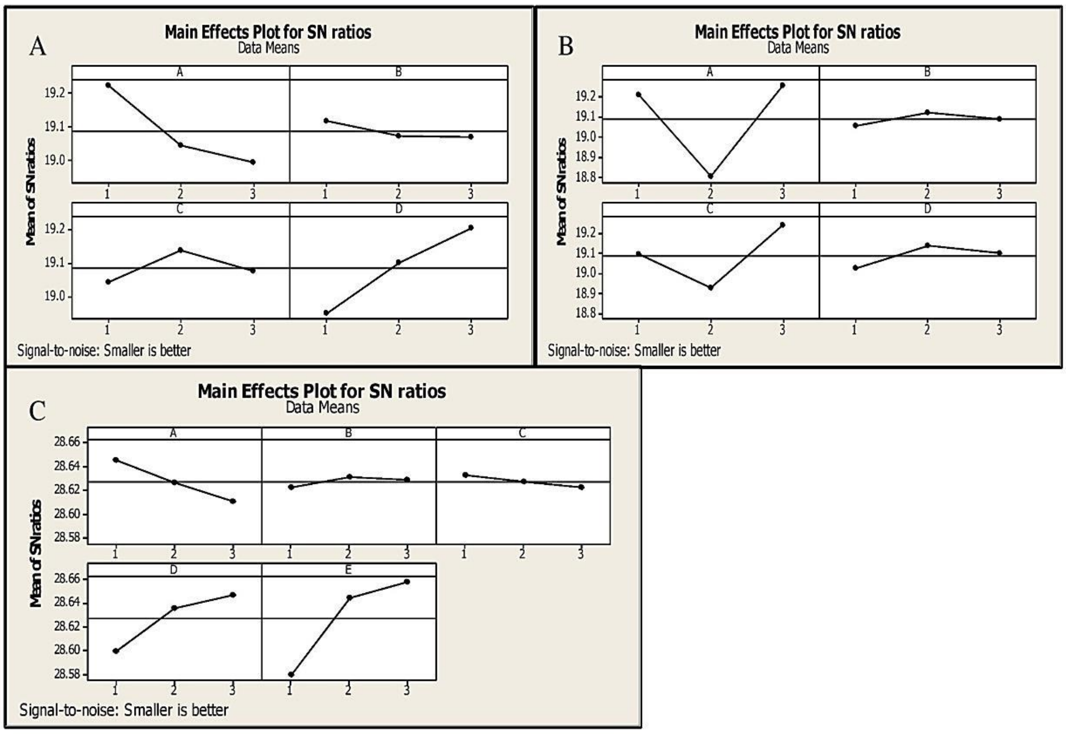

Figure 3.7: S/N ratio plot for: (A) GA parameters; (B) SA parameters; (C) MA parameters

3.4 The Case of Tehran, Iran

Tehran is the capital of Iran in an area of 686.3 km^2 with an estimated population of 9 million inhabitants in 2016 [124]. Tehran is subdivided into 22 municipal districts. The population density of each district is illustrated in Figure 3.8. Since the return period for Tehran earthquake is estimated to be around 173 years [125], and the city has not experienced a major earthquake in the last 186 years [126], the probable lurid catastrophic consequences of a massive earthquake have obsessed the inhabitants.

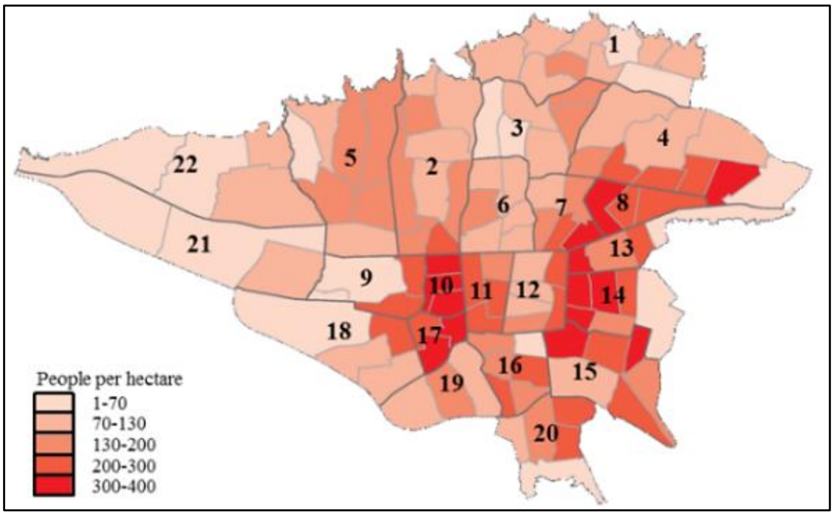

Figure 3.8: The population density of Tehran's municipal districts [127]

Tehran is besieged by 3 main active faults which are shown in Figure 3.9. The Mosha Fault with almost 200 km length is one of the main faults of Central Alborz area, which is located on the northeast of Tehran. The North Tehran Fault with 90 km length starts from Kan and joins the Mosha Fault in Lashgarak. The South and North Ray Faults with 20 km length are potentially the most perilous faults located in the southern Tehran.

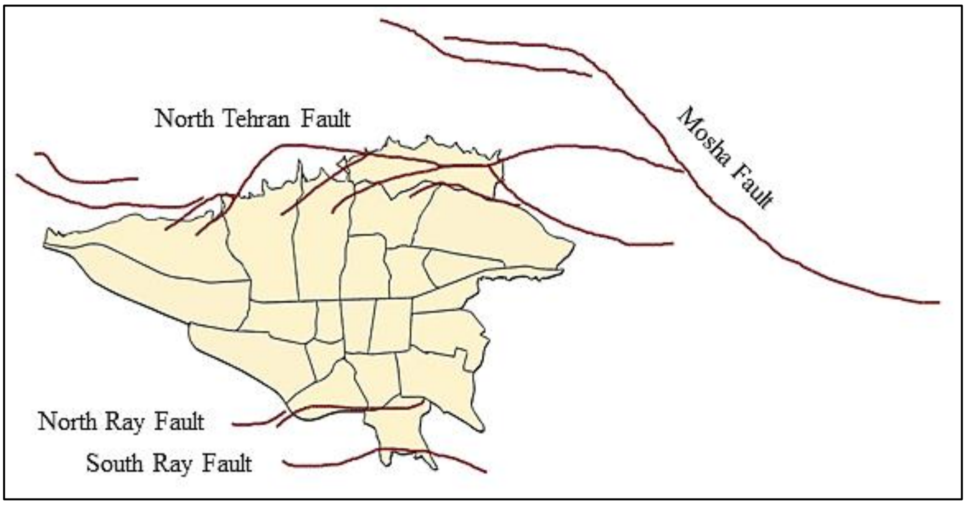

Figure 3.9: : Main active faults of Tehran adapted from Berberian et al. [128]

As it has been described by Japan International Cooperation Agency (JICA), 4 different seismic models can be considered for Tehran [129]; Mosha Fault model has the minimum effect on Tehran and causes at most 13% detriment to the city in the worst case. North Tehran Fault model which has a potential to make 36% detriment to the city, mainly targets districts 1 to 5 with 50% damage rate. North and South Ray Faults model causes 55% detriment to the city. Districts 10, 11, 12, 15, and 17 to 20 are mostly affected by this model with an estimated damage rate of 80%. Finally, Floating model is predicted to have most damages in districts 3, 10, 11, 12, 14, 17, and 20 with 51% estimated detriment rate. Considering these models, districts 10, 11, 12, 17, and 20 are potentially the most affected areas of Tehran. Regarding unpropitious conditions of the roads located in these districts and the high density of population residing in them, there is a high probability of road collapse which eventuates in inaccessibility of myriads of people in the case of a major earthquake. Thus for these areas, emergency relief distribution which enhances the surviving rate can be accomplished using UAVs.

In this chapter, it is premised that each relief kit weighs 2 kg consisting of 1.2 kg water, 0.3 kg high energy foods, 0.03 kg brown sugar, 0.15 kg blanket, 0.2 kg first-aid kit and antiseptics, 0.02 kg medicine like antibiotics, 0.05 kg plastics and toilet papers, and 0.05 kg led flashlight. This amount of relief could provide minimum survival requirements for a period of 10 hours. Any UAV drone with a minimum payload of 20 kg and 60 minutes endurance could be considered in the case of Tehran. The UAV drone which is considered in this chapter is Rotomotion SR-200 with a maximum payload of 22.7 kg and more than five hours endurance. The gasoline engine of this UAV affords a maximum speed of 60 km per hour, which makes it appropriate for the case of Tehran [130]. According to the weight of relief kits, it is assumed that each UAV can deliver 11 kits in each flight.

In order to apply the discussed problem to the case of Tehran, corresponding to the main streets of the city, a transportation network consisting of 4127 nodes and 9483 edges is considered. The complexity of this network with the high density of connection links is illustrated in Fig. 10 which is a map extracted from ArcGIS software. The length of edges and coordinates of each node is derived from this software and the shortest path is calculated by Dijkstra algorithm. As it is shown in Fig. 11, 33 potential locations are specified among the parks and open space areas in order to establish the relief distribution centers.

Figure 3.10: Edges representing the main streets of Tehran

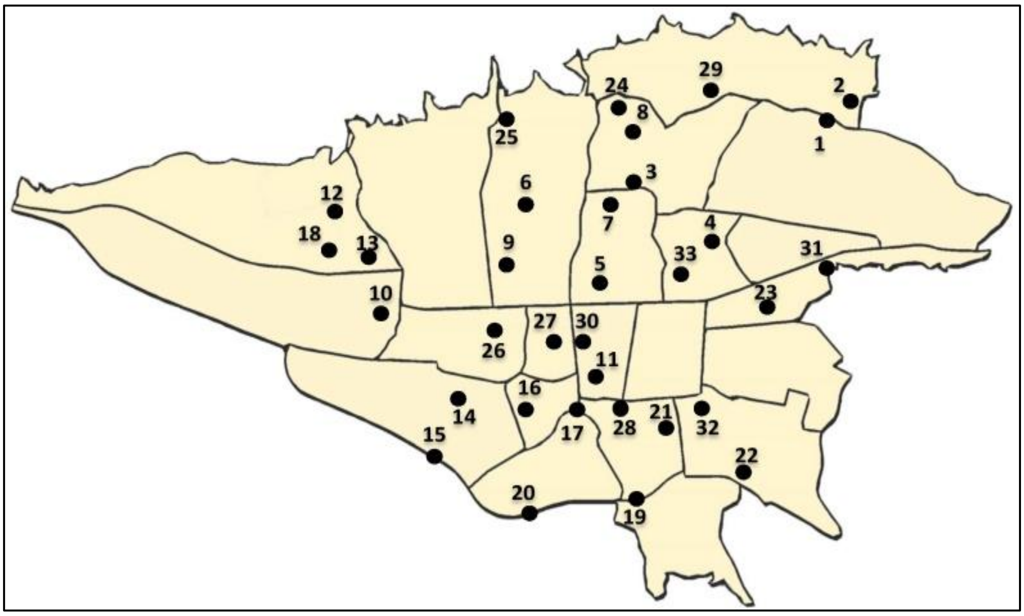

Figure 3.11: Potential relief distribution centers

3.4.1 Scenario Generation

In this chapter, totally 35 scenarios have been generated for Tehran by considering Mosha Fault model, North Tehran Fault model, North and South Ray Fault model, Floating model of northern Tehran, and Floating model of southern Tehran in seven different magnitudes. The occurrence probabilities of generated scenarios are considered to be identical; it means that occurrence probability for each scenario s is calculated as $p_s = 1/N$, where N is the total number of scenarios. In order to calculate the survival probability of each network edge (p_{ij}) in each scenario, the formula developed by Salman and Yücel [117] is used:

$$\mathrm{p}_{ij} = 1 - \beta_{ij} PGA_{ij} \tag{3.28}$$

PGA_{ij} is the peak ground acceleration at edge (i, j) which is calculated using the formula developed by Panoussis [131] as:

$$\mathrm{PGA}_{ij} = \alpha \frac{e^{0.8\mu}}{\left(r_{ij}+40\right)^2} \tag{3.29}$$

Where μ represents the magnitude of earthquake and r_{ij} is the distance to earthquake's hypocenter in km. β_{ij} as the seismic zone factor represents the damage risk of the region in which edge (i, j) is located. According to the JICA report, four seismic zone factors consisting of 0.9, 0.8, 0.7, and 0.6 are considered for Tehran. Since survival probabilities for Tehran streets are prognosticated to be between 0.6 and 0.9 by experts, accordingly parameter α which determines the gamut of survival probabilities is set equal to 2. In each scenario, the number of relief recipients on each street is estimated proportional to its peak ground acceleration and population.

3.5 Results and Discussions

The computational results of implementing the proposed metaheuristic algorithms to the case of Tehran for different numbers of open facilities are shown in Table 3.5. For each number of open facilities, each algorithm has been run five times to acquire the best concomitant solution. Figures 3.12 and 3.13 illustrate two main metrics, including the mean objective value and required CPU times of developed metaheuristic algorithms for different numbers of open facilities. In order to evaluate the performance of proposed GA, MA, and SA algorithms, the problem was coded in general algebraic modeling system (GAMS). However, since even for small instances, GAMS mathematical programming package failed to obtain the global (or even the local) optimum solution of the developed MINLP model, one-way analysis of variance (ANOVA) for the mean of objective values is performed by Minitab software to compare the performance of devised algorithms. According to the Figure 3.14, although there is not a significant difference between the results of proposed GA and MA, they significantly outperform the proposed SA. The same conclusion can also be made based on Figure 3.12. According to Figure 3.13, since the required CPU times for GA are flagrantly less than MA, the proposed GA is the most efficient algorithm for this problem.

Table 3.5: Computational results of solving methodologies

# Open facilities		Proposed GA		Proposed SA		Proposed MA	
		Objective	Time	Objective	Time	Objective	Time
8	*Min:*	4,578,794.48	860.37	4,803,543.09	1,189.59	4,578,794.34⁎	1,973.01
	Ave:	4,587,772.61	860.88	5,021,901.68	1,215.21	4,578,794.40	1,989.87
	(⁎) Best result for open facilities: {15,1,32,28,6,3,14,12}						
10	*Min:*	4,151,165.88⁎	861.56	4,605,920.36	1,230.26	4,151,165.88⁎	1,986.92
	Ave:	4,170,580.91	871.76	4,745,241.37	1,232.70	4,159,060.77	2,012.01
	(⁎) Best result for open facilities: {1,12,14,3,15,2,32,23,6,28}						
12	*Min:*	3,824,482.94⁎	859.86	4,274,460.65	1,203.46	3,824,482.94⁎	1,951.17
	Ave:	3,831,527.04	877.40	4,365,756.01	1,205.23	3,840,267.62	1,946.64
	(⁎) Best result for open facilities: {12,31,15,2,23,14,1,3,28,22,6,32}						
14	*Min:*	3,555,503.52⁎	853.57	4,026,391.75	1,234.34	3,555,503.52⁎	1,901.12
	Ave:	3,563,989.16	872.33	4,201,603.37	1,212.19	3,565,150.58	1,907.23
	(⁎) Best result for open facilities: {22,5,11,3,6,23,17,12,32,1,31,14,15,2}						
16	*Min:*	3,341,429.53⁎	876.01	3,555,646.74	1,224.54	3,341,429.53⁎	1,899.69
	Ave:	3,355,579.66	876.89	3,743,241.89	1,225.55	3,348,759.72	1,915.36
	(⁎) Best result for open facilities: {15,14,10,24,2,1,23,6,31,12,22,11,17,5,3,32}						
18	*Min:*	3,167,649.53⁎	869.72	3,352,683.67	1,241.27	3,167,649.53⁎	1,899.34
	Ave:	3,176,107.37	876.35	3,474,078.78	1,228.95	3,176,951.24	1,931.51
	(⁎) Best result for open facilities: {30,3,1,31,5,6,22,12,32,17,10,4,2,23,15,14,26, 24}						
20	*Min:*	3,018,863.15⁎	884.00	3,214,050.07	1,232.02	3,018,863.15⁎	1,909.83
	Ave:	3,025,158.02	881.38	3,254,406.51	1,240.75	3,023,050.07	1,942.90
	(⁎) Best result for open facilities: {6,3,25,32,26,1,10,31,15,14,5,2,21,30 ,4,24,17,23,12,22}						

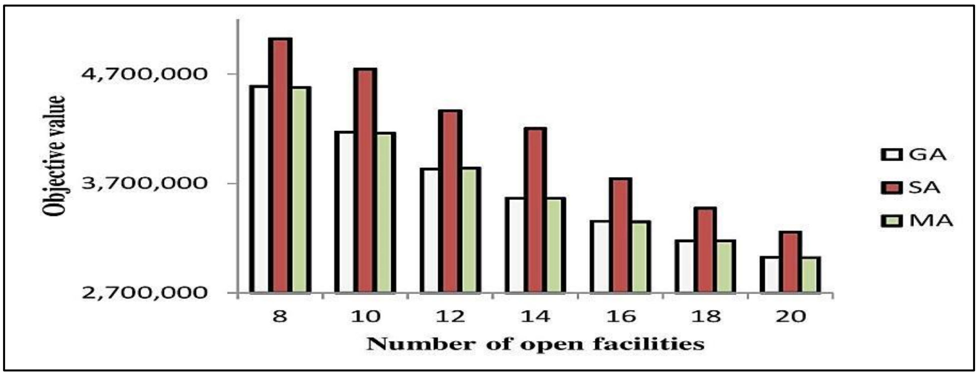

Figure 3.12: Mean objective values for different numbers of open facilities

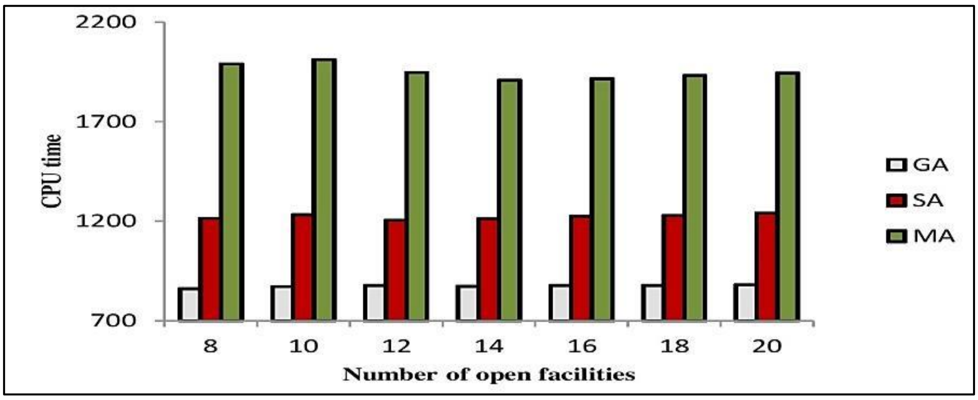

Figure 3.13: Mean required CPU times for different numbers of open facilities

```
Source        DF          SS             MS          F        P
Algorithms     2      4.59066E+12    2.29533E+12    7.26     0.001
Error        102      3.22575E+13    3.16250E+11
Total        104      3.68482E+13

        Individual 95% CIs For Mean Based on Pooled StDev
Level   N   Mean      StDev        -+---------+---------+---------+--
GA      35  3672959   527340        (-------*------)
MA      35  3670291   524595        (-------*------)
SA      35  4115176   628858                          (-------*------)
                                   -+---------+---------+---------+--
                                 3500000   3750000   4000000   4250000
```

Figure 3.14: ANOVA and related interval plots for objective value

According to Figure 3.12 and Table 3.5, it is incontrovertible that increasing the number of open facilities will decrease the objective value. Although bigger numbers of open facilities will increase the total costs of establishing relief distribution centers, smaller numbers of facilities will increase the traveling distance of people and the flying distance of UAVs, which in return increases the traveling time and waiting time and escalates the fatality rate due to delay in relief distribution. It is also obvious that in order to satisfy the demand in a reasonable time, more UAVs will be required for smaller numbers of open facilities and therefore, the cost of purchasing UAVs will be increased. Table 3.6 shows the percent of people who have to travel a maximum distance of 2, 4, 6, 8, and 10 km to get to their nearest facility and the required number of UAVs for covering the total demand in a period of 4, 6, and 8 hours in the worst-case scenario for different numbers of open facilities. This table provides an appropriate framework for decision makers to determine the number of open facilities by comparing the establishment costs, required number of UAVs and the prognosticated increase in fatality rates as the result of the distance between people and the relief distribution centers.

Table 3.6: Categorized percentage of people and required No. of UAVs

# open facilities	Categorized percentage of people					Required No. Of UAVs		
	2 km	4 km	6 km	8 km	10 km	4 h	6 h	8 h
8	28.84	40.08	25.27	5.13	0.68	2274	1516	1137
10	32.57	44.99	18.24	3.52	0.68	2022	1348	1011
12	35.85	47.44	14.11	2.00	0.6	1878	1252	939
14	40.00	47.59	10.99	0.88	0.54	1780	1187	890
16	43.46	48.33	6.79	0.88	0.54	1649	1099	824
18	49.45	42.87	6.26	0.88	0.54	1602	1068	801
20	53.18	39.82	5.59	0.88	0.53	1492	994	746

3.6 Conclusion

In this chapter, on the premise that people are uniformly distributed along the network edges, a stochastic MINLP model is developed for a combined mobile and immobile pre-earthquake facility location problem in order to find the best location of relief distribution centers among a set of candidate locations. It is assumed that the people who are located on intact and accessible edges travel to their closest open facility. Those demands that are located on collapsed or inaccessible edges are satisfied by UAVs through their closest open facility. The objective function of the proposed mathematical model minimizes the aggregate traveling time for both people and UAVs over a set of feasible scenarios. The developed mathematical model was applied to Tehran's network. A large-scale network of Tehran with 4127 nodes and 9483 edges was constructed by ArcGIS software to represent the main streets of the city. In this chapter, totally 35 scenarios have been generated for Tehran by considering seven different magnitudes of five possible seismic models of the city. For each scenario, the survival probability of each street was calculated based on its distance to the earthquake's hypocenter and the risk level of the corresponding region. Since the proposed model is NP-hard, in addition to utilization of GAMS optimization compiler, three metaheuristic algorithms consisting of GA, SA, and MA were proposed to solve the model. The parameters of the proposed algorithms were tuned using Taguchi method. The parameter-tuned algorithms were implemented for the case of Tehran by considering different numbers of allowed open facilities. Since the GAMS optimization compiler was not able to solve the model, one-way ANOVA method was used to compare the performance of proposed algorithms. According to the obtained results, although there was not a significant difference between developed GA and MA, GA is the most efficient solution method based on the required run times.

REFERENCES

[1] G. A. Marcoulides, Modern methods for business research. Psychology Press, 1998.

[2] S. L. Hakimi, “Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph,” Oper. Res., vol. 12, no. 3, pp. 450–459, 1964.

[3] J. Reese, “Solution methods for the p-median problem: An annotated bibliography,” Networks, vol. 48, no. 3, pp. 125–142, Oct. 2006.

[4] W. J. Baumol and P. Wolfe, “A Warehouse-Location Problem,” Oper. Res., vol. 6, no. 2, pp. 252–263, Apr. 1958.

[5] P. G. Berglund and C. Kwon, “Robust Facility Location Problem for Hazardous Waste Transportation,” Networks Spat. Econ., vol. 14, no. 1, pp. 91–116, Mar. 2014.

[6] A. Alhaffa, O. Al Jadaan, W. Abdulal, and A. Jabas, “Rank Based Genetic Algorithm for solving the Banking ATM’s Location Problem using convolution,” in 2011 IEEE Symposium on Computers & Informatics, 2011, pp. 6–11.

[7] D. Afshartous, Y. Guan, and A. Mehrotra, “US Coast Guard air station location with respect to distress calls: A spatial statistics and optimization based methodology,” Eur. J. Oper. Res., vol. 196, no. 3, pp. 1086–1096, 2009.

[8] C. Boonmee, M. Arimura, and T. Asada, “Facility location optimization model for emergency humanitarian logistics,” Int. J. Disaster Risk Reduct., 2017.

[9] M. Moeini, Z. Jemai, and E. Sahin, “Location and relocation problems in the context of the emergency medical service systems: a case study,” Cent. Eur. J. Oper. Res., vol. 23, no. 3, pp. 641–658, Sep. 2015.

[10] A. Klose and A. Drexl, “Facility location models for distribution system design,” Eur. J. Oper. Res., vol. 162, no. 1, pp. 4–29, 2005.

[11] T. S. Hale and C. R. Moberg, “Location Science Research: A Review,” Ann. Oper. Res., vol. 123, no. 1, pp. 21–35, 2003.

[12] A. Boloori Arabani and R. Z. Farahani, “Facility location dynamics: An overview of classifications and applications,” Comput. Ind. Eng., vol. 62, no. 1, pp. 408–420, 2012.

[13] S. Baptista and R. C. Oliveira, “A case study on the application of an approximated hypercube model to emergency medical systems management,” Cent. Eur. J. Oper. Res., vol. 20, no. 4, pp. 559–581, Dec. 2012.

[14] R. Halper, S. Raghavan, and M. Sahin, “Local search heuristics for the mobile facility location problem,” Comput. Oper. Res., vol. 62, pp. 210–223, 2015.

[15] F. Torrent-Fontbona, V. Muñoz, and B. López, “Solving large immobile location–allocation by affinity propagation and simulated annealing.

Application to select which sporting event to watch," Expert Syst. Appl., vol. 40, no. 11, pp. 4593–4599, 2013.

[16] Q. Wang, R. Batta, and C. M. Rump, "Facility location models for immobile servers with stochastic demand," Nav. Res. Logist., vol. 51, no. 1, pp. 137–152, Feb. 2004.

[17] S. Dantrakul, C. Likasiri, and R. Pongvuthithum, "Applied p-median and p-center algorithms for facility location problems," Expert Syst. Appl., vol. 41, no. 8, pp. 3596–3604, 2014.

[18] C. S. ReVelle, H. A. Eiselt, and M. S. Daskin, "A bibliography for some fundamental problem categories in discrete location science," Eur. J. Oper. Res., vol. 184, no. 3, pp. 817–848, 2008.

[19] J. Brimberg, P. Hansen, N. Mladenović, and S. Salhi, "A Survey of Solution Methods for the Continuous Location-Allocation Problem," Oper. Res., vol. 5, no. 1, pp. 1–12, 2008.

[20] Moeen Moghadas and Taghizadeh Kakhki, "Maximal covering location-allocation problem with M/M/k queuing system and side constraints," Iran. J. Oper. Res., vol. 2, no. 2, pp. 1–16, 2011.

[21] H. O. Mete and Z. B. Zabinsky, "Stochastic optimization of medical supply location and distribution in disaster management," Int. J. Prod. Econ., vol. 126, no. 1, pp. 76–84, 2010.

[22] M. Bieniek, "A note on the facility location problem with stochastic demands," Omega, vol. 55, pp. 53–60, 2015.

[23] Y. Zare Mehrjerdi and A. Nadizadeh, "Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands," Eur. J. Oper. Res., vol. 229, no. 1, pp. 75–84, 2013.

[24] R. Rahmaniani, M. Saidi-Mehrabad, and H. Ashouri, "Robust capacitated facility location problem: Optimization model and solution algorithms," J. Uncertain Syst., vol. 7, no. 1, pp. 22–35, 2013.

[25] L. V. Snyder, "Facility location under uncertainty: a review," IIE Trans., vol. 38, no. 7, pp. 547–564, 2006.

[26] G. Laporte, S. Nickel, and F. S. Da Gama, Location science. Springer, 2015.

[27] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co., 1979.

[28] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. I: The p-Centers," SIAM J. Appl. Math., vol. 37, no. 3, pp. 513–538, 1979.

[29] N. Mladenović, M. Labbé, and P. Hansen, “Solving the p -Center problem with Tabu Search and Variable Neighborhood Search,” Networks, vol. 42, no. 1, pp. 48–64, Aug. 2003.

[30] S. Elloumi, M. Labbé, and Y. Pochet, “A New Formulation and Resolution Method for the p-Center Problem,” INFORMS J. Comput., vol. 16, no. 1, pp. 84–94, Feb. 2004.

[31] B. Ç. Tansel, “Discrete Center Problems,” in Foundations of Location Analysis, H. A. Eiselt and V. Marianov, Eds. Boston, MA: Springer US, 2011, pp. 79–106.

[32] H. Calik, “Exact solution methodologies for the p-center problem under single and multiple allocation strategies,” Bilkent University, 2013.

[33] V. Marianov and C. ReVelle, “The Queueing Maximal availability location problem: A model for the siting of emergency vehicles,” Eur. J. Oper. Res., vol. 93, no. 1, pp. 110–120, Aug. 1996.

[34] R. Church and C. ReVelle, “The maximal covering location problem,” Pap. Reg. Sci. Assoc., vol. 32, no. 1, pp. 101–118, 1974.

[35] O. Berman and D. Krass, “The generalized maximal covering location problem,” Comput. Oper. Res., vol. 29, no. 6, pp. 563–581, 2002.

[36] O. Berman, D. Krass, and Z. Drezner, "The gradual covering decay location problem on a network," Eur. J. Oper. Res., vol. 151, no. 3, pp. 474–480, Dec. 2003.

[37] J. G. Villegas, F. Palacios, and A. L. Medaglia, "Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example," Ann. Oper. Res., vol. 147, no. 1, pp. 109–141, Oct. 2006.

[38] X. Li, Z. Zhao, X. Zhu, and T. Wyatt, "Covering models and optimization techniques for emergency response facility location and planning: a review," Math. Methods Oper. Res., vol. 74, no. 3, pp. 281–310, Dec. 2011.

[39] P. Murali, F. Ordóñez, and M. M. Dessouky, "Facility location under demand uncertainty: Response to a large-scale bio-terror attack," Socioecon. Plann. Sci., vol. 46, no. 1, pp. 78–87, 2012.

[40] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, and A. Mingozzi, "A new method for solving capacitated location problems based on a set partitioning approach," Comput. Oper. Res., vol. 29, no. 4, pp. 365–386, 2002.

[41] O. Alp, E. Erkut, and Z. Drezner, "An Efficient Genetic Algorithm for the p-Median Problem," Ann. Oper. Res., vol. 122, no. 1, pp. 21–42, 2003.

[42] S. H. R. Pasandideh, S. T. A. Niaki, and V. Hajipour, "A multi-objective facility location model with batch arrivals: two parameter-tuned meta-heuristic algorithms," J. Intell. Manuf., vol. 24, no. 2, pp. 331–348, 2013.

[43] S. H. R. Pasandideh and S. T. A. Niaki, “Genetic application in a facility location problem with random demand within queuing framework,” J. Intell. Manuf., vol. 23, no. 3, pp. 651–659, 2012.

[44] O. Berman and Z. Drezner, “The Multiple Server Location Problem,” J. Oper. Res. Soc., vol. 58, no. 1, pp. 91–99, 2007.

[45] R. Rahmaniani and A. Ghaderi, “An algorithm with different exploration mechanisms: Experimental results to capacitated facility location/network design problem,” Expert Syst. Appl., vol. 42, no. 7, pp. 3790–3800, 2015.

[46] P. Domínguez-Marín, S. Nickel, P. Hansen, and N. Mladenović, “Heuristic Procedures for Solving the Discrete Ordered Median Problem,” Ann. Oper. Res., vol. 136, no. 1, pp. 145–173, Apr. 2005.

[47] J. Brimberg and Z. Drezner, “A new heuristic for solving the p-median problem in the plane,” Comput. Oper. Res., vol. 40, no. 1, pp. 427–437, 2013.

[48] Y. An, B. Zeng, Y. Zhang, and L. Zhao, “Reliable p-median facility location problem: two-stage robust models and algorithms,” Transp. Res. Part B Methodol., vol. 64, pp. 54–72, 2014.

[49] Z.-J. M. Shen, R. L. Zhan, and J. Zhang, “The Reliable Facility Location Problem: Formulations, Heuristics, and Approximation Algorithms,” INFORMS J. Comput., vol. 23, no. 3, pp. 470–482, Aug. 2011.

[50] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, “The p-median problem: A survey of metaheuristic approaches,” Eur. J. Oper. Res., vol. 179, no. 3, pp. 927–939, 2007.

[51] M. J. Hodgson, “A Flow-Capturing Location-Allocation Model,” Geogr. Anal., vol. 22, no. 3, pp. 270–279, 1990.

[52] O. Berman, R. C. Larson, and N. Fouska, “Optimal Location of Discretionary Service Facilities,” Transp. Sci., vol. 26, no. 3, pp. 201–211, 1992.

[53] M. J. Hodgson and O. Berman, “A billboard location model,” Geogr. Environ. Model., vol. 1, no. 1, pp. 25–43, 1997.

[54] M. Kuby and S. Lim, “Location of Alternative-Fuel Stations Using the Flow-Refueling Location Model and Dispersion of Candidate Sites on Arcs,” Networks Spat. Econ., vol. 7, no. 2, pp. 129–152, 2007.

[55] M. Kuby, L. Lines, R. Schultz, Z. Xie, J.-G. Kim, and S. Lim, “Optimization of hydrogen stations in Florida using the Flow-Refueling Location Model,” Int. J. Hydrogen Energy, vol. 34, no. 15, pp. 6045–6064, 2009.

[56] O. Arslan and O. E. Karaşan, “A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles,” Transp. Res. Part B Methodol., vol. 93, pp. 670–695, Nov. 2016.

[57] J. Arkat and R. Jafari, "Network Location Problem with Stochastic and Uniformly Distributed Demands," Int. J. Eng. (IJE), Trans. B Appl., vol. 29, no. 5, pp. 654–662, 2016.

[58] R. C. Larson, "A hypercube queuing model for facility location and redistricting in urban emergency services," Comput. Oper. Res., vol. 1, no. 1, pp. 67–95, Mar. 1974.

[59] R. C. Larson, "Approximating the Performance of Urban Emergency Service Systems," Oper. Res., vol. 23, no. 5, pp. 845–868, 1975.

[60] B. Boffey, R. Galvão, and L. Espejo, "A review of congestion models in the location of facilities with immobile servers," Eur. J. Oper. Res., vol. 178, no. 3, pp. 643–662, 2007.

[61] B. Li, X. Deng, M. J. Golin, and K. Sohraby, "On the Optimal Placement of Web Proxies in the Internet: The Linear Topology," in High Performance Networking, Boston, MA: Springer US, 1998, pp. 485–495.

[62] N. Gautam, "Performance analysis and optimization of web proxy servers and mirror sites," Eur. J. Oper. Res., vol. 142, no. 2, pp. 396–418, 2002.

[63] R. Aboolian, O. Berman, and Z. Drezner, "The multiple server center location problem," Ann. Oper. Res., vol. 167, no. 1, pp. 337–352, 2009.

[64] V. Marianov and D. Serra, “Probabilistic, Maximal Covering Location—Allocation Models forCongested Systems,” J. Reg. Sci., vol. 38, no. 3, pp. 401–424, Aug. 1998.

[65] V. Marianov and D. Serra, “Hierarchical location–allocation models for congested systems,” Eur. J. Oper. Res., vol. 135, no. 1, pp. 195–208, 2001.

[66] V. Marianov and D. Serra, “Location–Allocation of Multiple-Server Service Centers with Constrained Queues or Waiting Times,” Ann. Oper. Res., vol. 111, no. 1/4, pp. 35–50, 2002.

[67] H. Shavandi and H. Mahlooji, “A fuzzy queuing location model with a genetic algorithm for congested systems,” Appl. Math. Comput., vol. 181, no. 1, pp. 440–456, 2006.

[68] T. Hamaguchi and K. Nakadeh, “Optimal Location of Facilities on a Network in Which Each Facility is Operating as an M/G/1 Queue,” J. Serv. Sci. Manag., vol. 3, no. 3, pp. 287–297, 2010.

[69] D. Hu, Z. W. Liu, and W. Hu, “Congestion service facilities location problem with promise of response time,” Math. Probl. Eng., vol. 2013, 2013.

[70] R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh, “Covering problems in facility location: A review,” Comput. Ind. Eng., vol. 62, no. 1, pp. 368–407, 2012.

[71] Q. Wang, R. Batta, and C. M. Rump, “Algorithms for a Facility Location Problem with Stochastic Customer Demand and Immobile Servers,” Ann. Oper. Res., vol. 111, no. 1, pp. 17–34, 2002.

[72] V. Marianov, T. B. Boffey, and R. D. Galvão, “Optimal location of multi-server congestible facilities operating as M/E r /m/N queues,” J. Oper. Res. Soc., vol. 60, no. 5, pp. 674–684, May 2009.

[73] V. Marianov and D. Serra, “Location of Multiple-Server Common Service Centers or Facilities, for Minimizing General Congestion and Travel Cost Functions,” Int. Reg. Sci. Rev., vol. 34, no. 3, pp. 323–338, 2011.

[74] A. Chambari, S. H. Rahmaty, V. Hajipour, and A. Karimi, “A bi-objective model for location-allocation problem within queuing framework,” World Acad. Sci. Eng. Technol., vol. 54, pp. 138–145, 2011.

[75] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, Fundamentals of Queueing Theory, 4th ed. New York, NY, USA: Wiley-Interscience, 2008.

[76] V. Hajipour, S. H. A. Rahmati, S. H. R. Pasandideh, and S. T. A. Niaki, “A multi-objective harmony search algorithm to optimize multi-server location–allocation problem in congested systems,” Comput. Ind. Eng., vol. 72, pp. 187–197, 2014.

[77] M. Khodaparasti, M. Ganji, S. Amirgholipour, and A. M. Sharifi, “A new multi-objective cluster ensemble based on modularity maximization,” J. Eng. Res., vol. 4, no. 2, pp. 1–16, 2016.

[78] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[79] Ö. Yeniay, “Penalty function methods for constrained optimization with genetic algorithms,” Math. Comput. Appl., vol. 10, no. 1, pp. 45–56, 2005.

[80] R. Kumar, “Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms,” Int. J. Mach. Learn. Comput., vol. 2, no. 4, pp. 365–370, 2012.

[81] M. Lozano, F. Herrera, and J. R. Cano, “Replacement strategies to preserve useful diversity in steady-state genetic algorithms,” Inf. Sci. (Ny)., vol. 178, no. 23, pp. 4421–4433, 2008.

[82] R. Tavakkoli-Moghaddam, N. Safaei, and F. Sassani, “A memetic algorithm for the flexible flow line scheduling problem with processor blocking,” Comput. Oper. Res., vol. 36, no. 2, pp. 402–414, 2009.

[83] P. Moscato and M. Norman, “A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems,” Parallel Comput. Transput. Appl., vol. 1, no. 1, pp. 177–186, 1992.

[84] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software," ACM Trans. Math. Softw., vol. 7, no. 1, pp. 17–41, 1981.

[85] A. H. Falah and E. A. Khorshid, "Optimum modeling of a flexible multi-bearing rotor system," J. Eng. Res., vol. 2, no. 2, pp. 155–181, 2014.

[86] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," Science (80-.)., vol. 220, no. 4598, pp. 671–680, 1983.

[87] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," J. Optim. Theory Appl., vol. 45, no. 1, pp. 41–51, 1985.

[88] J. Qin, H. Xiang, Y. Ye, and L. Ni, "A Simulated Annealing Methodology to Multiproduct Capacitated Facility Location with Stochastic Demand," Sci. World J., vol. 2015, 2015.

[89] G. S. Peace, Taguchi Methods: A Hands-On Approach. Addison-Wesley Publishing Company, 1993.

[90] S. M. Mousavi, S. T. A. Niaki, E. Mehdizadeh, and M. R. Tavarroth, "The capacitated multi-facility location–allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms," Int. J. Syst. Sci., vol. 44, no. 10, pp. 1897–1912, 2013.

[91] USGS, “United States Geological Survey,” 2012. [Online]. Available: http://earthquake.usgs.gov/earthquakes/browse/stats.php.

[92] F. Wang, Q. Cheng, L. Highland, M. Miyajima, H. Wang, and C. Yan, “Preliminary investigation of some large landslides triggered by the 2008 Wenchuan earthquake, Sichuan Province, China,” Landslides, vol. 6, no. 1, pp. 47–54, Mar. 2009.

[93] P. Qiang and A. Nagurney, “A bi-criteria indicator to assess supply chain network performance for critical needs under capacity and demand disruptions,” Transp. Res. Part A Policy Pract., vol. 46, no. 5, pp. 801–812, 2012.

[94] H. Wang, L. Du, and S. Ma, “Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake,” Transp. Res. Part E Logist. Transp. Rev., vol. 69, pp. 160–179, 2014.

[95] R. Nemiroff and J. Bonnel, “Where People Live on Planet Earth,” 2003. [Online]. Available: http://apod.nasa.gov/apod/ap030305.html.

[96] M. Safeer, S. P. Anbuudayasankar, K. Balkumar, and K. Ganesh, “Analyzing Transportation and Distribution in Emergency Humanitarian Logistics,” Procedia Eng., vol. 97, pp. 2248–2258, 2014.

[97] C. G. Rawls and M. A. Turnquist, “Pre-positioning of emergency supplies for disaster response,” Transp. Res. Part B Methodol., vol. 44, no. 4, pp. 521–534, 2010.

[98] F.-S. Chang, J.-S. Wu, C.-N. Lee, and H.-C. Shen, “Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling,” Expert Syst. Appl., vol. 41, no. 6, pp. 2947–2956, 2014.

[99] I. M. Khalil, A. Khreishah, F. Ahmed, and K. Shuaib, “Dependable wireless sensor networks for reliable and secure humanitarian relief applications,” Ad Hoc Networks, vol. 13, pp. 94–106, 2014.

[100] R. T. Eguchi, “We have come a long way, yet we still have far to go,” Nat. Hazards, vol. 68, no. 1, pp. 201–202, Aug. 2013.

[101] A. M. Caunhye, X. Nie, and S. Pokharel, “Optimization models in emergency logistics: A literature review,” Socioecon. Plann. Sci., vol. 46, no. 1, pp. 4–13, 2012.

[102] B. Balcik and B. M. Beamon, “Facility location in humanitarian relief,” Int. J. Logist. Res. Appl., vol. 11, no. 2, pp. 101–121, 2008.

[103] F. M. Moghadas, E. Monabbati, and H. T. Kakhki, “Emergency Location Problems with an M/G/k Queueing System,” Iran. J. Oper. Res., vol. 4, no. 1, pp. 1–13, 2013.

[104] P. B. Mirchandani, “Locational Decisions on Stochastic Networks,” Geogr. Anal., vol. 12, no. 2, pp. 172–183, 1980.

[105] J. A. Paul and R. Batta, “Models for hospital location and capacity allocation for an area prone to natural disasters,” Int. J. Oper. Res., vol. 3, no. 5, p. 473, 2008.

[106] S. Duran, M. A. Gutierrez, and P. Keskinocak, “Pre-Positioning of Emergency Items for CARE International,” Interfaces (Providence)., vol. 41, no. 3, pp. 223–237, 2011.

[107] N. Görmez, M. Köksalan, and F. S. Salman, “Locating disaster response facilities in Istanbul,” J. Oper. Res. Soc., vol. 62, no. 7, pp. 1239–1252, Jul. 2011.

[108] A. Verma and G. M. Gaukler, “Pre-positioning disaster response facilities at safe locations: An evaluation of deterministic and stochastic modeling approaches,” Comput. Oper. Res., vol. 62, pp. 197–209, 2015.

[109] H. Jia, F. Ordóñez, and M. Dessouky, “A modeling framework for facility location of medical services for large-scale emergencies,” IIE Trans., vol. 39, no. 1, pp. 41–55, Jan. 2007.

[110] C.-C. Lu and J.-B. Sheu, “Robust vertex p-center model for locating urgent relief distribution centers,” Comput. Oper. Res., vol. 40, no. 8, pp. 2128–2137, 2013.

[111] A. S. Selçuk and M. S. Yücemen, “Reliability of Lifeline Networks with Multiple Sources under Seismic Hazard,” Nat. Hazards, vol. 21, no. 1, pp. 1–18, 2000.

[112] J. D. VanVactor, “Strategic health care logistics planning in emergency management,” Disaster Prev. Manag. An Int. J., vol. 21, no. 3, pp. 299–309, Jun. 2012.

[113] E. Shieh, K. Habibi, K. Torabi, and H. E. Masoumi, “Earthquake risk in urban street network: an example from region 6 of Tehran, Iran,” Int. J. Disaster Resil. Built Environ., vol. 5, no. 4, pp. 413–426, Nov. 2014.

[114] U. Kamp, L. A. Owen, B. J. Growley, and G. A. Khattak, “Back analysis of landslide susceptibility zonation mapping for the 2005 Kashmir earthquake: an assessment of the reliability of susceptibility zoning maps,” Nat. Hazards, vol. 54, no. 1, pp. 1–25, Jul. 2010.

[115] H. A. Eiselt, M. Gendreau, and G. Laporte, “Location of facilities on a network subject to a single-edge failure,” Networks, vol. 22, no. 3, pp. 231–246, May 1992.

[116] E. Melachrinoudis and M. E. Helander, “A single facility location problem on a tree with unreliable edges,” Networks, vol. 27, no. 3, pp. 219–237, May 1996.

[117] F. S. Salman and E. Yücel, “Emergency facility location under random network damage: Insights from the Istanbul case,” Comput. Oper. Res., vol. 62, pp. 266–281, 2015.

[118] L. Ozdamar, “Planning helicopter logistics in disaster relief,” OR Spectr., vol. 33, no. 3, pp. 655–672, Jun. 2011.

[119] J. Qi et al., “Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake,” J. F. Robot., vol. 33, no. 3, pp. 290–321, May 2016.

[120] A. Nedjati, B. Vizvari, and G. Izbirak, “Post-earthquake response by small UAV helicopters,” Nat. Hazards, vol. 80, no. 3, pp. 1669–1688, Feb. 2016.

[121] J. Chen et al., “Damage Degree Evaluation of Earthquake Area Using UAV Aerial Image,” Int. J. Aerosp. Eng., vol. 2016, pp. 1–10, 2016.

[122] Shaodan Li et al., “Unsupervised Detection of Earthquake-Triggered Roof-Holes From UAV Images Using Joint Color and Shape Features,” IEEE Geosci. Remote Sens. Lett., vol. 12, no. 9, pp. 1823–1827, Sep. 2015.

[123] G. Q. Li, X. G. Zhou, J. Yin, and Q. Y. Xiao, “An UAV scheduling and planning method for post-disaster survey,” ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XL-2, pp. 169–172, Nov. 2014.

[124] Statistical Center of Iran, “Statistical Center of Iran,” 2014. [Online]. Available: https://www.amar.org.ir/.

[125] K. Amini Hosseini, M. Hosseini, S. Hosseinioon, Y. O. Izadkhah, T. Shaw, and R. Takahashi, “A survey on evacuation planning and its challenges for potential earthquake in Tehran,” Int. J. Disaster Resil. Built Environ., vol. 5, no. 1, pp. 38–52, Mar. 2014.

[126] S. Yaghmaei-Sabegh and N. T. K. Lam, “Ground motion modelling in Tehran based on the stochastic method,” Soil Dyn. Earthq. Eng., vol. 30, no. 7, pp. 525–535, 2010.

[127] Tehran Municipalty, “Atlas of Tehran metropolice,” 2006. [Online]. Available: http://atlas.tehran.ir/.

[128] M. Berberian, M. Qorashi, B. Arzhangravesh, and A. Mohajer-Ashjai, “Recent tectonics, seismotectonics and earthquake-fault hazard study in the Greater Tehran region,” Geol Surv Iran, vol. 57, p. p 130 (in Persian), 1983.

[129] JICA and CEST, The study on seismic microzoning of the Greater Tehran Area in the Islamic Republic of Iran, no. November. Japan International Cooperation Agency (JICA), Center of Earthquake and Environmental Studies of Tehran (CEST), Tehran Municipality, Pacific Consultants International Report, 2000.

[130] Airforce Technology, "SR200 Rotary Unmanned Aerial Vehicle (UAV), United States of America," 2016. [Online]. Available: http://www.airforce-technology.com/projects/sr200rotaryuav/.

[131] G. Panoussis, "Seismic reliability of lifeline networks," Cambridge, USA, 1974.