Industrial Engineering & Management

Jahedsani and Azimi, Ind Eng Manage 2018, 7:3
DOI: 10.4172/2169-0316.1000258

Research Article
Open Access

Optimizing a Hierarchical Location: Allocation Problem Using the M/M/M Queue Model and Solving It Employing a Genetic Algorithm

Jahedsani H[1]* and Azimi P[2]

[1]Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Abstract

The main objective in location-allocation problems is to locate a series of service centers and optimally allocate the population to the located facilities. In many service systems, there is a hierarchical structure in nature, i.e. the demand and its corresponding services are provided at different levels. This paper presents a novel bi-objective hierarchical location-allocation model based on an M/M/m queuing model which is the main contribution of this research. The objective functions minimize the average waiting time for customers in the queue at all levels and also minimize the maximum facility idle-time which is confronted with a maximum idle-time among the available facilities. Since the proposed model is NP-Hard, it is solved employing a multi-objective meta-heuristic algorithm called non-dominated sorting genetic algorithm (NSGA-II). Given the significant correlation between output of the meta-heuristic algorithm and the input parameters, the Taguchi's experimental design method is applied to adjust the parameters. The model is solved for some experimentally generated problems in different sizes, and a detailed analysis has been provided for the results. Moreover, a comparison between the current model and a single level location- allocation model with an M/M/m queue structure in the literature is made to show the efficiency of the proposed solving algorithm. The results show that the proposed model and the corresponding solving algorithm are efficient enough to be applied in real world applications.

Keywords: Location allocation problems; Hierarchical models; M/M/M queue model; Multi-objective programming; Taguchi method; NSGA-II algorithm

Introduction

Nowadays, the discussion of facility location and the ways to optimally allocate population to the located facilities are the important problems in location problems [1]. In the majority of the previous models, just one level of facilities has been considered where demand points refer to a facility and the customers leave the system after receiving a service; but in the real world and in many service systems (such as health care, education, transport, postal services, collection of damaged and waste goods and etc.), whether in the private or public sector, there are some hierarchical cases, i.e., the demands and services are located at different levels. In these systems, the problem cannot be analysed separately at each level of services. Because the relationship between different levels of services and service centres providing different levels of services, require an optimal location assignment for the servers and suitable assignment of demand points to the servers, simultaneously. In this context, because of the assignment of suitable locations to the servers and the necessity of high quality services at the right time, it is feasible to provide all services at a specific level. In the real world, it is very important to combine the location problems with a queue structure to be able to analyse the system performance in terms of the average system waiting time or the average system length. Development of queuing theory by Johansson and Erlang [2] dates back to the early 1900s for the phone systems that were described with fixed facilities and random demand. Larson combined the concept of queuing theory with the facility location, which includes problems related to vehicle location and the regional response to emergency service systems [3,4]. Narula proposed a bi-level hierarchical location-allocation model in health care, in which level One and level Two were considered first care and specialized services, respectively [4]. Moore and Revelle presented a bi-level hierarchical location model for maximal covering problem, assuming a low-level server only covers basic services, and a high level server covers the low-level services, in addition to the high-level services [5]. In his research, Hodgson introduced a hierarchical location-allocation model in exponential form by taking in to account the benefits of different levels of the facilities and the reduced distances between facilities and customers [6]. In their study, Weaver and Church were considered the nested hierarchical problem for median models [7]. A hierarchical location model was introduced by Serra et al. for competitive environments [8]. Such a covering location model based systems coherent, presented by Serra [9].

Daskin also formulated a hierarchical problem with the aim of maximal covering [10]. Ardal et al. formulated a hierarchical location-allocation model on the basis of flow pattern [11]. Then Sahin and Sural presented a model with mean objective in their paper [12]. Marianov and Serra proposed two hierarchical location-allocation models using the approach to M/M/1 queueing model for dense systems [13]. Wang et al. presented several models for facility location as an M/M/1 queueing model by taking into account the capacity constraints and proposed them using an objective function to minimize the waiting times in queue and also travel times [14]. In this research, customers can directly enter into the higher levels without having to refer the lower levels. Sahin and Sural conducted a review on the problems and models of the hierarchical facility location [12]. The research helped to partly fill the gap in the literature related to these problems. Berman developed a model proposed by Wang using the M/M/m queue model, with the assumption that customers can choose the nearest-facility

***Corresponding author:** Jahedsani H, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Tel: 982833665275; E-mail: Jahedsani.hamed@gmail.com

Received July 23, 2018; **Accepted** August 20, 2018; **Published** August 25, 2018

Citation: Jahedsani H, Azimi P (2018) Optimizing a Hierarchical Location: Allocation Problem Using the M/M/M Queue Model and Solving It Employing a Genetic Algorithm. Ind Eng Manage 7: 258. doi:10.4172/2169-0316.1000258

(service center) [15,16]. Syam proposed a nonlinear location-allocation model by considering several servers to design the service system, by taking into account the M/M/m queuing model and the associated costs [17].

Aboolian et al. presented a location-allocation model by considering several servers and constraints of referring the nearest demand, with the aim of minimizing maximum travel and waiting times [18]. Teixeira and Antunes proposed a hierarchical location model for public facilities [19]. Pasandideh and Niaki presented the bi-objective model for the location-allocation problem with M/M/1 queue model framework to minimize the average total of travel and waiting times of the customers, and minimize the average percentage of the equipment idleness [20]. They considered the customer demand, randomly and assigned a fixed location, servers and service rate. Bhattacharya and Bandyopadhya proposed a bi-objective facility location model which was solved using an evolutionary algorithm of solving multi-objective problems, called NSGA-II [21]. Drezner and ZviDrezner presented a model for the location of multiple-server service centers with the M/M/m queue system, in which the allocation of customers to the service centers is based on the attraction of service centers and/ or the interest of customers, are not assigned to the nearest service center [22].

The objective function of this study is to minimize the travel time of the customers to service centers and their waiting time in queue. Seifbarghy et al. presented a model for the location of service centers from the customers' perspective with the M/M/m queuing system and showed the assignment of customers to service centers using the criteria of distance from center and the number of providers in each service center, with the aim of minimizing the average queue length [23]. Pasandideh et al. presented a three-objective model by entering a group of customers into service centers and converted it to a single-objective using a LP-metric method, and solved it by simulated annealing and genetic algorithm [24]. Chambari et al. suggested a bi-objective location model with queue, and used the Pareto-based Algorithms NSGA-II and NRGA to solve it [25]. Lakshmi and Sivakumar conducted a review on the use of queuing theory for locating service centers and presented it as a classification system [26].

Pasandideh proposed a multi-objective location problem in a queuing framework with three objective functions:

a. To minimize the overall travel time to the facilities and the waiting time of the customers in the queue, for receiving service;

b. To minimize the probability of idleness for a facility which is confronted with the highest probability of idleness; and

c. To minimize total costs which are associated in the open facilities [27].

In his research, MalekiMoghaddam examined a multi-objective hierarchical location-allocation model with an approach to M/M/1 queue model, and used two approaches to solve the model [28]. In the first approach, he turned his model into a single-objective model, using LP-metric method, and solved it using genetic algorithm. In the second approach, he solved the problem using the algorithm NSGA-II, directly. Rahmati et al. presented a multi-objective location model with the structure of M/M/m and considered two constraints to choose the nearest center and service level constraint [29].

To solve the problem, they used the Pareto-based multi-objective harmony search algorithm, the non-dominated sorting genetic algorithm (NSGA-II) and dominated ranking genetic algorithm (NRGA). Farahani et al. have considered the possibility of failure for facilities due to real world disasters and has developed a model in which a hierarchy of facilities are located in an order to maximize the total covered demand [30]. They have also proposed a hybrid artificial bee colony (HABC) algorithm to solve the model, efficiently. In another work, Hajipour et al. have developed a novel bi-objective multi-server location-allocation model, in which the facilities are modeled as an M/M/m system [31]. Moreover, in this study, capacity and budget limitations are provided to make the LA problem more realistic. Two objective functions in this work include minimizing aggregate waiting times and minimizing the maximum idle time of all facilities. Authors have developed a multi-objective harmony search algorithm (MOHA) in which a new presentation scheme that satisfies most of the model constraints is proposed.

Furthermore, Rahmati et al. investigated a practical bi-objective model for the facility location–allocation problem with immobile servers and stochastic demand within the M/M/1/K queue system [32]. The authors sought to develop a mathematical model in which customers and service providers were considered as perspectives. In this research, the objectives of the developed model were minimization of the total cost of server provider and of the total time of customers. To solve the model, two popular multi-objective evolutionary algorithms, namely NSGA-II and NRGA, were implemented. In another recent study, Brimberg and Drezner have considered a continuous location problem for p-concentric circles serving a given set of demand points [33]. Employing heuristic and exact algorithms, they have analyzed the problem of minimizing the sum of weighted distances between demand points and their closest circle when demand is uniformly and continuously distributed in a disk. Farahani et al. in their review paper have presented a comprehensive review of over 40 years of hierarchical facility location modeling efforts [34]. In their comprehensive survey, they have also identified the gaps in the current literature and therefore have suggested directions for future modeling efforts. In one most recent study, Zhu et al. have proposed a hierarchical location-allocation model for trauma centers in Shenzhen [35]. In their work, the Location Set Covering Model is used to calculate the smallest number of low-level trauma centers that could meet the demands of the covered area in the optimal time. They have used a multi-objective model that included response, coverage, treatment, and cost capacities to solve the location-allocation problem of high-level trauma centers. The objectives of this study were to optimize trauma center locations, improve the allocation of medical trauma resources and reduce the rate of deaths and disabilities due to trauma. In another recent study, Paul have analyzed the effectiveness of the current and optimal locations of a set of existing regional assets maintained by In the current paper, a hierarchical location-allocation model using an M/M/m queue structure is presented. Structure of the paper is as follows: Firstly, the proposed model is described in Section 3 and then details of the method for solving the problem are discussed in Section 4. Numerical results and finally, conclusions along with some suggestions for future work are presented in Sections 5 and 6, respectively.

The Proposed Model

Since it could be very expensive to construct a new facility and determine a suitable location for it, therefore some comprehensive studies should be carried out to decide, optimally. We propose a hierarchical location-allocation problem with the structure of the M/M/m queue model so that the output of the model, in fact, encompasses the following criteria:

a. Finding the optimal location of each facility at different levels;
b. Creating the closest connection between the locations of demand and the facilities;
c. Reducing costs to create facilitate and communication paths; and
d. Determining the optimal number of servers deployed in each facility at different levels to deliver services.

Assumptions

There are m servers, the system is considered to be a hierarchy type and has two levels of services. No constraint is considered in terms of the queue capacity and the potential customer population. The inter-arrival rate of customers is constant and independent of the system population, and is according to a Poisson process. The service time for each server has an exponential distribution. In each service center, several types of services are provided, which are independent from each other. Each service center acts as an M/M/m queue system for each type of service. Diverse needs of a demand node can be supplied from multi-service centers, and server-sites are assumed to be constant.

Indices

a. *i:* an index for customer i=1,…,*I*.
b. *j:* an index of the potential *j-th* low-level facility, j=1,…,*J*.
c. *n:* an index of the potential *n-th* high level facility, n=1,…,*N*.

The input parameters

- P^L: The maximum number of low-level facilities required to be located.
- P^H: The maximum number of high-level facilities required to be located.
- λ_i: The demand rate of service requested by customer *i*.
- τ_j^L: Total demand rate service for the *j*th low-level facility.
- μ_j^L: Service rate at *j*-th low-level sever.
- τ_n^H: Total demand rate service at nth high-level facility.
- μ_n^H: Service rate at the nth high- level sever.
- k_j^L: Demand capacity at the *j*th low-level facility.
- k_n^H: Demand capacity at the *n*th high-level facility.
- T^L: Waiting time in the queue at low-level facility.
- T^H: Waiting time in the queue at high-level facility.
- φ^L: Service level at the low-level facility.
- φ^H: Service level at the high-level facility.
- U^L: The maximum number of servers that can be placed in the low-level facility.
- U^H: The maximum number of servers that can be placed in the high-level facility.
- $\pi_{0,j}^L$: Idle probability of the *j*th low-level facility.
- $\pi_{0,n}^H$: Idle probability of the *n*th high-level facility.
- m_j^L: The number of servers located in the *j*th low-level facility.
- m_n^H: The number of servers located in the nth high-level facility.
- *B*: Total available budget.
- $C_{j,1}^L$: Fixed cost of establishing a low-level facility at potential node *j*.
- $C_{j,2}^L$: Staffing cost for a low-level server at potential node *j*.
- $C_{n,1}^H$: Fixed cost of establishing a high-level facility at potential node *n*.
- $C_{n,2}^H$: Staffing cost for a high-level server at potential node *n*.
- β_j: Percentage of requests to low-level serve, that request high-level service.
- t_{ijn}: Traveling time for customer *i* to go from low-level facility node *j* to the high-level facility node *n*.

Decision variables

- x_{ijn}: Allocation variable which take the value of 1 if customer *i* is allocated to a low-level server on a candidate node *j* will go to a high- level server located at a candidate node *n*, and zero otherwise.
- y_j: Location variable that takes the value of 1 if a low-level server is located at node *j*, and zero otherwise.
- y_n: Location variable that takes the value of 1 if a high-level server is located at node *n*, and zero otherwise.

Model

Waiting time at the low-level facility j and the high-level facility n in an M/M/m queuing system are obtained as Gross and Harris [36]:

$$W_j^L = \left[\frac{\pi_{0,j}^L}{m_j^L!} \left(\frac{\tau_j^L}{\mu_j^L} \right)^{m_j^L} \frac{m_j^L \mu_j^L}{\left(m_j^L \mu_j^L - \tau_j^L \right)^2} + \frac{1}{\mu_j^L} \right] \quad (1)$$

$$W_n^H = \left[\frac{\pi_{0,n}^H}{m_n^H!} \left(\frac{\tau_n^H}{\mu_n^H} \right)^{m_n^H} \frac{m_n^L \mu_n^L}{\left(m_n^H \mu_n^H - \tau_n^H \right)^2} + \frac{1}{\mu_n^H} \right] \quad (2)$$

According to Figure 1, when all customers go to low-level facilities, a percentage of them leave the system, after receiving the desired service, and the rest go to higher level facilities [37]. Total demand rate at the low level facility *j* and the high-level facility n have the following relations:

Mathematical model

The proposed hierarchical model is formulated as a constrained non-linear mixed integer programming model as follows:

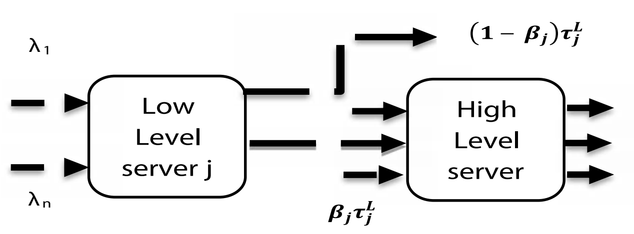

Figure 1: Quenching diagram of the hierarchical system.

$$Min\ f_1 = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N}\lambda_i t_{ijn} x_{ijn} + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N}\lambda_i W_j^L x_{ijn} + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N}\lambda_i W_n^H x_{ijn} \quad (3)$$

$$Min\ f_2 = Max\left\{\left(\pi_{0,j}^L \cdot y_j\right),\left(\pi_{0,n}^H \cdot y_n^{'}\right)\right\}; \quad \forall j,n \quad (4)$$

$$\sum_{j=1}^{J} y_j \le P^L; \quad (5)$$

$$\sum_{n=1}^{N} y_n^{'} \le P^H; \quad (6)$$

$$\sum_{j=1}^{J}\sum_{n=1}^{N} X_{ijn} \le 1; \quad \forall\ i \quad (7)$$

$$x_{ijn} \le y_j; \quad \forall\ i; j\ ; n \quad (8)$$

$$x_{ijn} \le y_n^{'}; \quad \forall\ i; j\ ; n \quad (9)$$

$$\tau_j^L \le m_j^L \mu_j^L y_j\ ; \quad \forall\ j \quad (10)$$

$$\tau_n^H \le m_n^H \mu_n^H y_n^{'}\ ; \quad \forall\ n \quad (11)$$

$$1 \le m_j^L \le U^L y_j; \quad \forall\ j \quad (12)$$

$$1 \le m_n^H \le U^H y_n^{'}; \quad \forall\ n \quad (13)$$

$$\frac{\left(\tau_j^L / \mu_j^L\right)^{m_j^L} \pi_{0,j}^L e^{-\left(m_j^L \mu_j^L - k_j^L\right)T^L}}{1-\left[\left(\tau_j^L / \mu_j^L\right)^{m_j^L} m_j^L!\right]} \le \varphi^L; \quad \forall\ j \quad (14)$$

$$\frac{\left(\tau_n^H / \mu_n^H\right)^{m_n^H} \pi_{0,n}^H e^{-\left(m_n^H \mu_n^H - k_n^H\right)T^H}}{1-\left[\left(\tau_n^H / \mu_n^H\right)^{m_n^H} m_n^H!\right]} \le \varphi^H; \quad \forall n \quad (15)$$

$$\sum_{j=1}^{J}\sum_{n=1}^{N}[(C_{j,1}^L y_j + C_{j,2}^L m_j^L) + (C_{n,1}^H y_n^{'} + C_{n,2}^H m_n^H)] \le B; \quad \forall\ j,n \quad (16)$$

$$\pi_{0,j}^L = \frac{1}{\sum_{j=1}^{m_j^L-1}\left(\frac{\tau_j^L}{\mu_j^L}\right)^j\left(\frac{1}{j!}\right) + \frac{\left(\tau_j^L / \mu_j^L\right)^{m_j^L}}{\left(\left[1-\left(\tau_j^L / \mu_j^L\right)^{m_j^L}\right] m_j^L!\right)}}; \quad \forall\ j \quad (17)$$

$$\pi_{0,n}^H = \frac{1}{\sum_{n=1}^{m_n^H-1}\left(\frac{\tau_n^H}{\mu_n^H}\right)^n\left(\frac{1}{n!}\right) + \frac{\left(\tau_n^H / \mu_n^H\right)^{m_n^H}}{\left(\left[1-\left(\tau_n^H / \mu_n^H\right)^{m_n^H}\right] m_n^H!\right)}}; \quad \forall n \quad (18)$$

$$y_j, y_n^{'}, x_{ijn} \in (0,1); \quad \forall\ i; j\ ; n \quad (19)$$

$$m_j^L, m_n^H\ integer; \quad \forall\ j; \quad (20)$$

- The first objective function minimizes the total travel time to each high and low-level facility plus the average waiting time for customers in the queue.
- The second objective function minimizes the maximum idle-time for each facility at high and low-level facilities. The constraint sets in eqns. (5) and (6) indicate that the maximum total number of high and low-level facilities must be less than or equal to P^L and P^H.

The constraint set in eqn. (7) assures that each customer could be assigned to a high and/or a low-level facility. The constraints in eqns. (8) and (9) assure that if a high or low-level facility is not selected, no facility will be assigned to any customer. The constraints in eqns. (10) and (11) indicate that the service capacity must be greater than or equal to the total demand rate in any of the high and low-level facilities. The constraints in eqns. (12) and (13) indicate the minimum and maximum number of servers that can be placed in any of the high and low-level facilities. The constraints in eqns. (14) and (16) show that the probability of waiting time in the facility located at j-th node and nth node, at times T^L and T^H is less than or equal to its level of service. The constraint in eqn. (16) implies the budget constraint for the model; i.e. the total cost to build a new facility and the cost to hire human resources at any of different level should be less than the given budget. Using the constraints in eqns. (17) and (18), the probability of a server to be idle is calculated for the j-th low-level facility and n-th high-level facility. The constraints in eqns. (19) and (20) state that the variables of problem decision, i.e., y_j,$y_n^{'}$,x_{ijn} and m_j^L ,m_n^H are equal to zero and one. The model has many constraints, several 0-1 decision variables, two objectives and including non-linear relations. So it is an integer nonlinear programming (INLP) type, which has been proven to be NP-Hard [22,24,29]. Consequently, given that it is time-consuming to solve the problem using classical exact methods for large scale problems, developing a metaheuristic algorithm is inevitable to solve the real world problems.

The proposed solving method

Metaheuristic methods which are also known as evolutionary search methods are very more efficient in solving multi-objective optimization problems, compared to the traditional tools. Evolutionary algorithms can, without the need for a linear combination of multiple attributes in a composite scalar objective function, improve a family of solutions along the equilibrium level through the concept of Pareto optimality. To solve the proposed model, we used the non-dominated sorting genetic algorithm (NSGA-II) that, as one of the most well-known multi-objective optimization algorithm which was introduced by Deb et al. [38]. In this method, two answers are randomly selected from the population. Then, a comparison is made between the two answers, and whichever is better is ultimately chosen. The most important criteria of selection in the NSGA-II are the solutions rank and the crowding distance related to the answer, respectively. Whatever the solution rank is less and its crowding distance is higher, it is more desirable. If a binary choice operator is repeated on the population of each generation, a series of people in that generation are selected to participate in the crossover and mutation processes. The crossover operation is done on the part of the selected individuals, and mutation operation is performed on the others, and thus a population of Offspring and mutants is created. Furthermore, the population is merged with the main population, and members of the newly formed population are initially sorted in an ascending order according to their ranks. Members of the population who are of the same rank are sorted in a descending order according to the crowding distance. Now, the members who are equal to the original population number are selected from the top items of the sorted list, and rest of the population is discarded. The selected members form the population of the next generation; and the cycle in this section is repeated until the termination conditions are realized. The non-dominated solutions obtained from the solving multi-objective optimization problem are often known as the Pareto front.

Solution representation

The first and the most important component to increase the efficiency of optimization algorithms is selecting how to display the answer. Converting a solution from the solution space into a chromosome is called encoding, and converting the chromosomes into a solution of the problem-solving space is called decoding. Figure 2 shows the two-part structure of the problem chromosomes. The first part is a vector whose length is equal to the number of potential low-level facility nodes (j) that each of their arrays is a random number between zero and one. The second part is a vector whose length is equal to the number of potential high-level facility nodes (n) that each of their arrays is a random number between zero and one.

Chromosome evaluation

The other important point in a meta heuristic algorithm is the process of decoding. After producing chromosome, we must assign a fitness value to it. In the optimization problems, the fitness assignment is the same value as the objective function. To decode the proposed solution, we continue the decoding process with a numerical example. Consider a problem with 10 and 5 candidate locations to locate high and low-level facilities so that up to 7 and 3 places can be selected for low-level and high-level facilities, respectively. Locations 1, 3, 4, 7, 9 and 10 (y_j) are selected among the low-level facilities, and locations 2, 3 and 4 (y_n) are selected among the high-level facilities. Among a maximum of 8 servers that can be used for each facility, assume 5, 6, 8, 6, 4 and 2 servers are assigned to each located low-level facility, respectively. For the high-level facilities, among a maximum of 6 servers that can be assigned to each located facility, 5, 4 and 3 servers are assigned, respectively. The main chromosome, in fact, has a structural model similar to Figure 3, on which crossover and mutation operators are performed. From the chromosome, first the binary variables related to

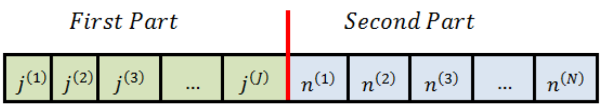

Figure 2: Solution representation.

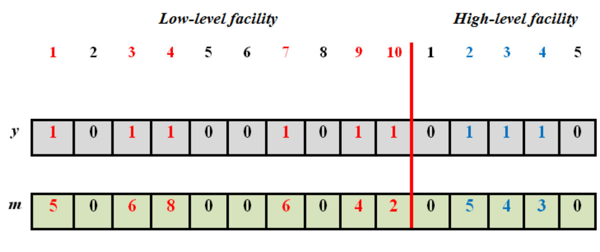

Figure 3: Decoding problem.

the assignment of each customer are randomly initialized to a high- and low-level facility (x) in a way that for each customer, a location is randomly selected from the located high and low-level facilities, and correspondingly the relevant variables will be equal to one, and the rest gets zero. For example, the first customer is assigned to the low-level facility No. 4 and the high-level facility No. 2, the related matrix for x_{ijn} is shown in Figure 4.

After each customer is assigned to a high and low-level facility, variables τ and π_0 are calculated according to the eqns. (3, 4, 19, and 20). The constraints related to these variables are checked, and if the solutions are not feasible, their objective functions are fined. A penalty function is one of the effective ways of dealing with infeasibilities in decision problems with constraints [39]. Fine functions can reduce the unjustified generated solutions, according to the proportion of constraint violations. In fact, a fine function converts the problems with constraints into problems without constraint.

$$F(x)=\begin{cases} f(x) & ; x\in \textit{feasible Region} \\ f(x)+P(x) & ; x\notin \textit{feasible Region} \end{cases} \tag{21}$$

Where, P(x) represents the fine value. If a constraint is satisfied or has no violation, the P(x) will be equal to zero, otherwise, it takes an amount higher than zero. Since different constraints can have different degrees of magnitude, it is necessary to normalize all the constraints, before the relationship described above is used. As a result, the deviation of normalized constraints is converted to the same value, and all these deviations can be easily added together, and finally, an overall fine parameter can be added to the objective function for all the constraints of the problem.

Genetic operators

Creating a population of offspring from parents, which is carried out to produce solutions with a quality better than previous generation and is actually an evolutionary process that is conducted by the crossover and mutation operators in genetic algorithms.

Crossover operator

They are some operators that choose one or more points of two or more solutions and replace their values. In this study, a single cut-off point crossover operator is used, and a point is considered as the cut-off point on the length of the two chromosomes that have been selected as the parent. Since the produced offspring may be impossible, the corrective strategy is used to justify the solutions. If a chromosome is infeasible, a gene is randomly selected and re-initialized. This procedure is repeated until the chromosome is justified (Figure 5).

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 4: *Xijn* variable matrix.

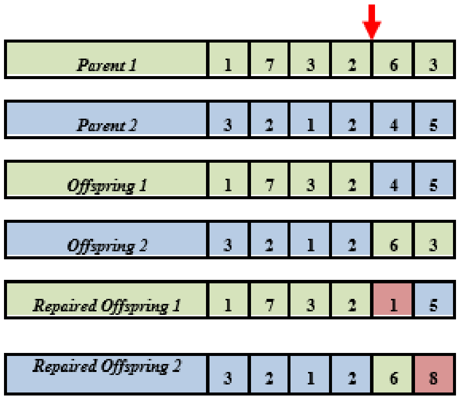

Figure 5: Crossover operator.

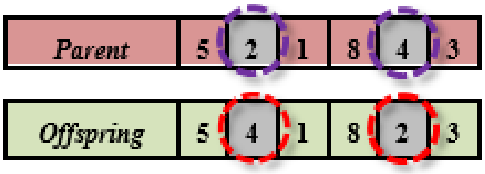

Figure 6: Mutation operator.

Mutation operator

Mutation operator searches the space of solutions that has been found by the crossover operator. They are operators that can select one or more genes from a chromosome and change its values. In this study, the swap mutation operator is used (Figure 6).

Stopping condition

The last step in the genetic algorithm is to examine the stopping condition. In fact, it is a measure which specifies the extent that the loop of algorithm can continue and can be different on the view of the designer. In this research, to achieve better results, we intended to consider the time as the stop condition so that algorithm is stopped after a certain time.

Numerical Example

To assess the model and the performance of the algorithm developed in this research, 22 test problems are generating randomly. These problems are categorized based on the number of costumers (*I*), the number of low-level facilities (*J*), and the number of high-level facilities (*N*).

Parameter setting

Table 1 contains different values of these parameters. Moreover, the following information is also given.

- The demand rate of the service requests from a customer node i follow a uniform distribution, i.e., λ_i ~ Uniform 2, 10.
- The service rate of the jth low-level server has a uniform distribution, i.e., μ_j^L~ Uniform (30, 50).
- The service rate of the nth high-level server has a uniform distribution, i.e., μ_n^H~ Uniform (30, 50).
- Demand rate at a low-level facility node j has a uniform distribution, i.e., k_j^L~ Uniform (55, 90).

Size	Test Problem Number	I	J	N
Small	1	5	4	2
	2	10	4	2
	3	15	6	3
	4	20	6	3
	5	25	8	4
Average	6	30	8	4
	7	35	10	5
	8	40	10	5
	9	45	12	6
	10	50	12	6
	11	55	14	7
	12	60	14	7
	13	65	16	8
Large	14	70	16	8
	15	75	18	9
	16	80	18	9
	17	85	20	10
	18	90	20	10
	19	95	22	11
	20	100	22	11
	21	150	30	20
	22	200	40	30

Table 1: Input of the model.

- Demand rate at a high-level facility node n has a uniform distribution, i.e., k_n^H~ Uniform (55, 90).
- Queue waiting time at each low-level facility follows a uniform distribution, i.e., T^L~ Uniform (10, 20).
- Queue waiting time at each high-level facility follows a uniform distribution, i.e., T^H~ Uniform (10, 20).
- The service level at each low-level facility is considered 0.3, i.e., (φ^L=0.3).
- The service level at each high-level facility is considered 0.3, i.e., (φ^H=0.9).
- Maximum number of servers that can be established at each low-level facility has a uniform distribution, i.e., U^L~ Uniform (5, 15).
- Maximum number of servers that can be established at each high-level facility has a uniform distribution, i.e., U^H~ Uniform (5, 15).
- Fixed cost of establishing the potential *j*-th low-level facility follows a uniform distribution, i.e., C_1^L~ Uniform (100,200).
- Fixed cost of establishing the potential *n*-th high-level facility follows a uniform distribution, i.e. C_1^H~ Uniform (100,400).
- Unit staffing cost at the *j*-th low-level facility has a uniform distribution, i.e., C_2^L~ Uniform (10, 55).
- Unit staffing cost at the *n*-th high-level facility has a uniform distribution, i.e., C_2^H~ Uniform (20, 60).
- Traveling time from customer i to the low-level facility node *j* (for low-level services) and to the high-level facility node *n* (for high level of service) has a uniform distribution, i.e., t_{ijn}~ Uniform (50,100).
- Percentage of requests of a low-level server, that request for

high-level services has a uniform distribution, i.e., beta~ Uniform (0.2, 0.7).

Parameters tuning

The efficiency and quality of any meta heuristic algorithms depend on the correct choice of parameter values. Different combinations of parameters involved in the implementation of an algorithm can help to achieve solutions with different quality. Taguchi [40] presented an efficient method called the robust parameter design. He assumed that there are two types of factors that act on a process: The objective is to minimize the effects of disturbance factors and to find the best level of controllable factors. Taguchi determined the relative importance of each factor in relation to its main impact on the performance of the algorithm, and by changing the value of objective function to another value is the change scale, which was developed by Taguchi. This change is in the signal to noise ratio (the word signal is good value, and the word noise is undesirable value), which should be maximized as a goal.

$$S/N = -10 \times \log_{10}(Objective\ Function) \tag{22}$$

Taguchi's experimental design method uses an orthogonal array to organize the results of experiment, and is as follows: a series of different levels of the parameters affecting the algorithm is examined based on the input indices that usually use the value of the objective. Then according to the results of tests conducted on the basis of selected orthogonal arrays, the best combination is suggested as the optimal values to adjust the parameters. Table 2 shows the parameters which are affecting the proposed algorithm efficiency. In this research, the potential factors which can influence the quality of the response obtained by the proposed non-dominated sorting genetic algorithm are pop size (the GA population size), ρ_C (the crossover probability), ρ_M (the mutation probability), and maxgen (Number of iteration).

To select a proportional array for the algorithm, the total degrees of freedom for the intended algorithm should be calculated, and the number of rows in the selected arrays should be at least equal to the total degrees of freedom of the algorithm. The total degrees of freedom for an algorithm are equal to one plus the number of levels of each parameter minus one. Given that the algorithm has four three-level parameters and 9 degrees of freedom 1+(3-1)+ (3-1)+ (3-1)+ (3-1)=9], then array L_9 (3^4) is selected for testing. Table 3 shows the modified array L_9 for the proposed algorithm.

Taguchi's method is implemented by the selected arrays for the algorithm via some tests and how to combine the parameters in the proposed algorithm. To analyse the results of the Taguchi's experiment, one must introduce an index that is able to compare the algorithm for different combinations of parameters. Therefore, since the value of the objective function in each problem i different for single-objective problems and cannot be used directly, relative deviation index (RDI) is used for each problem.

Parameter	Symbol	Levels
Pop size	A	A(1) – 30
		A(2) – 50
		A(3) – 80
ρ_C	B	B(1) – 0.7
		B(2) – 0.8
		B(3) – 0.9
ρ_M	C	C(1) – 0.01
		C(2) – 0.03
		C(3) – 0.05
Max.gen	D	D(1) – 100
		D(2) – 200
		D(3) – 300

Table 2: Algorithm Parameters.

Trial	A	B	C	
1	1	1	1	1
2	1	2	3	2
3	1	3	2	3
4	2	1	3	3
5	2	2	2	1
6	2	3	1	2
7	3	1	2	2
8	3	2	1	3
9	3	3	3	1

Table 3: The modified array L_9.

	Level 1	Level 2	Level 3
Pop size (A)	6.7091	12.8120	16.6965
ρ_C (B)	10.5075	14.3395	11.3706
ρ_M (C)	15.8415	10.6373	9.7388
max.gen (D)	10.2282	12.7621	13.2272

Table 4: The average quality obtained for each level of parameter.

$$RDI = \frac{Alg_{sol} - Best_{sol}}{Worst_{sol} - Best_{sol}} \times 100 \tag{23}$$

The objective function value for each replication of the experiment is to find the best and the worst solutions obtained. After converting the value of objective function into RDI, the S/N ratio is calculated based on RDI, according to the structure of Taguchi parameter design. Then, the average S/N ratio of the experiments is calculated for each level of parameter. The best value for each parameter has the maximum value of the average S/N. But given that a multi-objective model is presented in this study, a set of non-dominated Pareto solutions (rather than one solution) is obtained, so the quality of Pareto solutions is used instead of the RDI index. The index of quality considers all solutions calculated for each parameter combination for each problem, and we simultaneously perform the non-dominated operations for all the points. Base on contribution of each parameter combination of new Pareto solutions, the quality obtained for the intended combination is taken into account. In this study, 22 sample problems were run, each problem was replicated ten times, on each parameter combination; and the search time for different sizes of the problem is considered to be equal to 5×(I+J+N) seconds, in order to have the same experimental conditions. Thus, the average quality obtained for each level of parameter is listed in Table 4 and also in Figure 7 as percentages. Finally, the parameters set for the algorithm NSGA-II in Table 5 can be explained as follows.

Analysis of taguchi results

Based on the results reported in Table 5, the best parameter values are as follows: the initial population size (pop size) and the maximum number of repetitions (maxgen) at the third level, Crossover operator in the second level and mutation operator at the first level for desired algorithm 3.4.

Comparison indicators for multi-objective problems

Because the suggested model (Multi-level with M/M/m structure) is a new one, we cannot find a similar model in the literature to compare the results and adjust our algorithm. However, to show the performance of the algorithm, we decided to compare the results to the ones reported by Rahmati et al. [29]. Although their model is a single level location- allocation model with an M/M/m queue structure, we set our model to a single level to provide a comparison base.

Therefore, to adjust the proposed algorithm, four indicators have been used including Number of Pareto Solution index (NPS), Spacing Metric index (dispersion) between Pareto Front solutions (DS), Diversification Metric index (DM) and CPU Time metric index (Time).

Number of Pareto solution index: This indicator measures the number of Pareto optimal solutions. It is clear that whatever the number of solutions produced is more, the algorithm is more efficient.

Spacing metric index: Using this index, we measure the distribution uniformity of Pareto Points in Pareto solutions set obtained. Distance index can be calculated using eqn. (24).

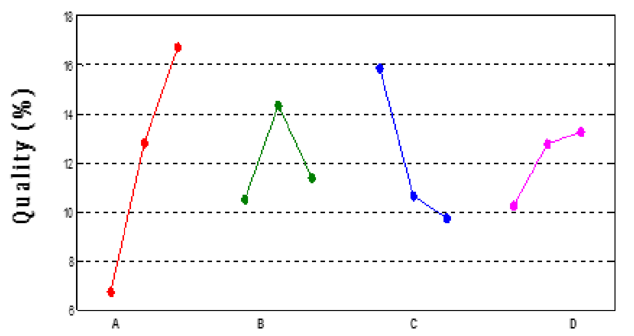

Figure 7: Average percentage of quality parameters.

Popsize	ρ_C	ρ_M	Max.gen
80	0.8	0.01	300

Table 5: The parameters set for the algorithm NSGA-II.

$$S = \left[\frac{1}{N-1} \sum_{i=1}^{N-1} \left(1 - \frac{d_i}{\bar{d}}\right)^2 \right]^{1/2} \quad (24)$$

In this equation, d_i is the distance between i^{th} solution and the closest solution in Pareto set, $\bar{d}$ is average Euclidean distances di and N is the number of Partum set solutions. If this index is used to compare two algorithms, whatever the distance index is smaller for an algorithm, that algorithm is better.

Diversification metric index: Diversity index is usually used to measure solutions diversity on the Pareto solution set produced, as follows:

$$D = \sqrt{\sum_{i=1}^{N} \max_j \left(x_i - y_j\right)} \quad (25)$$

In this formula, $\| x_i - y_i \|$ is Euclidean distance between two Pareto solutions x_i and y_i, N is the number of Pareto set solutions. To calculate the index, after calculating distance between two solutions, the farthest solution is determined for each point which is square root of the sum of these distances. If two algorithms are compared with each other, whatever diversity between solutions are more for an algorithm, it will be more efficient.

CPU time metric index: This indicator also shows the time (seconds) needed for the implementation of the proposed algorithm, the results of which are shown in Table 6. Both algorithms were coded in Matlab software (R2016B) and the all computations were run on a Peintium IV Core i7 PC with 64 Gbyte Ram and the average results of all four criteria were summarized in Table 4. For each test problem, we replicated each algorithm 10 times. As the results show, the proposed algorithm dominates the competitor in terms of NPS, DS, and DM criteria over all test problems. However, both algorithms have nearly the same computational times (Time).

Also the Pareto Front diagram for sample problems has been shown in different sizes in Figures 8 and 9. In total average, the

Size	Test Problem Number	I	J	N	Proposed NSGA- II				Compared NSGA-II			
					NPS	DS	DM	Time	NPS	DS	DM	Time
Small	1	4	2	1	1	-	0	76.406	1	-	0	77.1701
	2	4	2	10	10	1.136236	54.83811	66.97375	9	1.170323	52.72895	66.9738
	3	6	3	8	8	0.823974	86.946	88.71175	14	0.848693	85.24118	89.5989
	4	6	3	9	9	0.79852	83.84413	79.53225	10	0.830461	80.61936	79.5323
	5	8	4	7	7	0.888644	68.80133	82.24975	6	0.915303	66.15513	83.0722
Average	6	8	4	19	19	1.038589	149.2594	83.634	22	1.069747	142.1518	84.4703
	7	10	5	13	13	0.512647	126.1674	82.30275	11	0.528026	121.3148	83.1258
	8	10	5	12	12	0.704319	137.9351	89.99075	14	0.718405	133.9176	89.9908
	9	12	6	13	13	0.686662	136.4045	99.7815	15	0.707262	131.1582	100.7793
	10	12	6	12	12	1.148511	166.1479	100.3588	16	1.194451	162.8901	101.8538
	11	14	7	14	14	1.475638	171.2611	111.8538	12	1.519907	164.6741	111.8538
	12	14	7	16	16	1.212191	145.1623	117.0708	15	1.260679	139.5791	118.2415
	13	16	8	11	11	0.903896	128.1186	121.437	13	0.9310113	124.387	122.6514
Large	14	16	8	13	13	0.971052	156.8853	127.2668	15	1.000184	150.8513	127.2668
	15	18	9	15	15	0.773346	150.9321	154.676	17	0.796546	145.127	156.2228
	16	18	9	15	15	0.737828	139.4652	147.0355	13	0.759963	135.4031	147.0355
	17	20	10	11	11	0.812722	151.3509	175.394	14	0.853358	145.5297	177.1479
	18	20	10	5	5	0.762984	75.99954	174.145	5	0.785874	73.07648	175.8865
	19	22	11	6	6	1.258503	88.07259	221.425	7	1.283673	83.87866	223.6393
	20	22	11	7	7	0.673953	101.983	209.9233	5	0.694172	98.06058	209.9233
	21	30	20	4	4	0.431267	95.40003	617.3753	6	0.448518	93.52944	623.5491
	22	40	30	14	14	0.715099	212.94448	1399.133	12	0.743703	204.7546	1413.1243
Total Average	-	-	-	-	10.68182	0.879361	119.4509	201.2126	11.45455	0.907631	115.2286	202.8463

Table 6: The average results of these indicators for ten times.

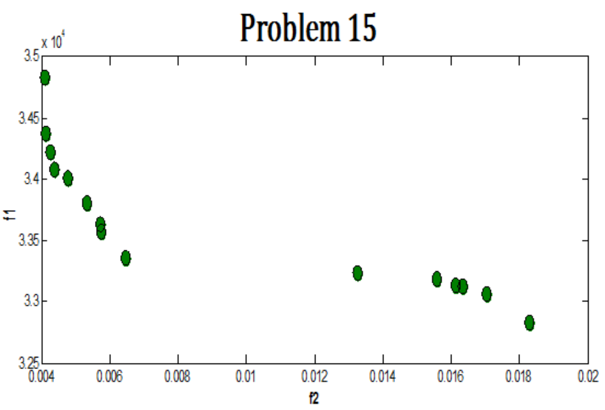

Figure 8: Obtained Pareto-front of the algorithms on test problem 15.

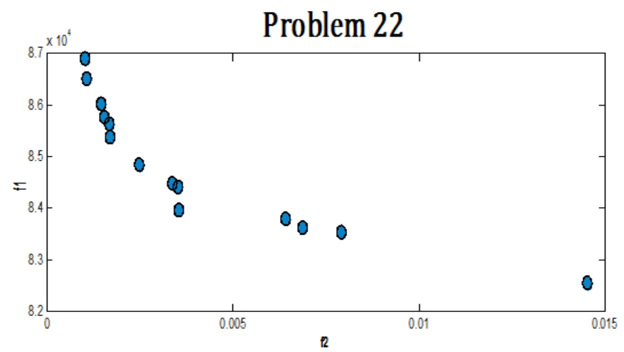

Figure 9: Obtained Pareto-front of the algorithms on test problem 22.

proposed algorithm NPS is 10.68 while the competitor value is 11.45 (6.7% better). Regarding the DS criterion, the values are 0.879 and 0.908 which shows and improve of (3.1%). About the DM, the values are 119.45 and 115.23 (3.7% better).

Discussion and Conclusion

In this study, a novel multi-objective hierarchical location-allocation problem was developed using an M/M/m queue structure, for the first time in order to (1) minimize the average waiting times for customers in the queue at all levels and (2) minimize the maximum facility idle-times which is confronted with a maximum idle-time among the available facilities. According to the best knowledge of the authors, this is the first research which develops an M/M/m queue model for the problem which is more close to the real world applications. Given that the hierarchical location-allocation problems are classified in the NP-hard group, a non-dominated sorting genetic algorithm (NSGA-II) was developed to solve the model. Since the correct choice of values for algorithm parameters can affect the performance of a set of Pareto-optimal solutions, all parameters were tuned using the Taguchi's experimental design method. Finally, to show the efficiency and the quality of the algorithm, we compared the results to a previous research in the literature.

For future researches, the conditions of uncertainty can be modelled as fuzzy sets. Also, extension of the model to more than two levels, the use of other queue models such as M/M/1/k or M/M/m/k and the use of simulation technique to optimize the system when the non-exponential inter-arrival and service times could be considered.

References

1. Francis RL, McGinnis FJ, White JA (1992) Facility Layout and Location: An Analytical Approach.
2. Saaty TL (1961) Elements of queuing theory with Application.
3. Larson RC (1974) A hypercube Queuing model for facility location and redistricting in urban emergency services. Comput Oper Res 1: 67-95.
4. Narula SC (1986) Minisum hierarchical location-allocation problems in a network: a survey. Ann Oper Res 6: 257-72.
5. Moore GC, ReVelle CS (1982) The hierarchical service location problem. Manage Sci 28: 775-80.
6. Hodgson MJ (1988) An hierarchical location-allocation model for primary health care delivery in a developingarea. Soc Sci Med 26: 153-161.
7. Weaver JR, Church RL (1987) The formal and computational relationship of the supporting median problem to the p-median problem. Transport Res Part B: Methodological 21: 323-329.
8. Serra D, Marianov V, ReVelle CS (1992) The maximum-capture hierarchical location problem. Eur J Oper Res 62: 363-71.
9. Serra D (1996) The coherent covering location problem. Pap Reg Sci 75: 79-101.
10. Daskin MS (1995) Network and Discrete Location: Models, Algorithms, and Applications.
11. Aardal K, Chudak FA, Shmoys DB (1999) A 3-approximation algorithm for the k-level uncapacitated facility location problem. Inf Process Lett 72: 161-7.
12. Sahin GH (2007) A review of hierarchical facility location models. Comput Oper Res 34: 2310-2331.
13. Marianov V, Sera D (2001) Location models for airline Hubs behaving as M/D/c queues. Comput Oper Res 30: 983-1003.
14. Wang Q, Batta R, Rump C (2002) Algorithms for a facility location problem with stochastic customer demand and immobile servers. Ann Oper Res 111: 17-34.
15. Berman, Huang R (2007) The minsum Multipurpose Trip Location Problem on Network. Transport Sci 4: 500-515.
16. Wang Q, Batta R, Rump RC (2004) Facility Location Models for Immobile Servers with Stochastic Demand. NEV RAS LOG 51: 137-15.
17. Syam SS (2008) A multiple server location-allocation model for service system design. Comput Oper Res 35: 2248-2265.
18. Aboolian R, Berman O, Drezner Z (2009) The multiple server center location problem. Ann Oper Res 167: 337-52.
19. Teixeira JC, Antunes AP (2008) A hierarchical location model for public facility planning. Eur J Oper Res 185:92-104.
20. Pasandideh SHR, Niaki SDR (2010) Genetic application in a facility location problem with random demand withen queuing framework. J Intell Manuf 21: 234-546.
21. Bhattacharya R, Bandyopadhyay S (2010) Solving conflicting bi-objective facility location problem by NSGA-II evolutionary algorithm. Int J Adv Manuf Technol 51: 397-414.
22. Drezner T, Drezner Z (2011) The gravity multiple server location problem. Computers & Operational Research 38: 694-701.
23. Seifbarghy M, Rabieian R, Karimi F (2011) Locating service centers optimizing customer's perspective criteria. Int J Adv Manuf Technol 54: 811-819.
24. Psandideh SHR, Niaki SDR, Tokhmehchi N (2011) A parameter-tuned genetic algorithm to optimize two-echelon continuous review inventory systems. Expert Syst Appl 38:11708-11714.
25. Chambari AH, Rahmaty SH, Hajipour V, Karimi A (2011) A bi-objective modelforlocation-allocation problem within queuing framework. World Acad Sci Eng Technol 78: 138-145.
26. Lakshmi C, Sivakumar AI (2013) Application of queueing theory in health care: A literature review. Oper Res Health Care 2: 25-39.
27. Pasandideh SHR, Niaki SDR, Hajipour V (2013) A multi-objective facility location model with batch arrivals: Two parameter-tuned meta-heuristic algorithms. J Intell Manuf 24: 331-348.

28. MalekiMoghaddam M (2013) Modeling a Multi objective Hierarchical location problem with queuing approach and solvethat by metaheuristic algorithms.

29. Rahmati SH, Hajipour V, Niaki STA (2013) A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. Appl Soft Comput 13: 1728-1740.

30. Farahani RZ, Hassani A, Mousavi SM, Baygi MB (2014) A hybrid artificial bee colony for disruption in a hierarchical maximal covering location problem. Comput Ind Eng 75: 129-141.

31. Hajipour V, Rahmati SH, Pasandideh SHR, Niaki STA (2014) A Multi-Objective Harmony SearchAlgorithm to Optimize Multi-Server Location-Allocation Problem in Congested Systems. Comput Ind Eng 72: 187-197.

32. Rahmati SH, Ahmadi A, Sharifi M , Chambari A (2014) A multi-objective model for facility location-allocation problem with immobile servers within queuing framework. Comput Ind Eng 74: 1-10.

33. Brimberg J, Drezner Z (2015) A location-allocation problem with concentric circles. IIE Trans 47: 15-20.

34. Farahani RZ, Hekmatfar M, Fahimnia B, Kazemzadeh N (2014) Hierarchical facility location problem: Models, classifications, techniques, and applications. Comput Ind Eng 68: 104-117.

35. Zhu Y, Du Q, Liang S, Shi G, Lin D (2016) Location Optimization Using a Hierarchical Location-Allocation Model for Trauma Centers in Shenzhen, China.

36. Paul N, Lunday B, Nurre S (2017) A multi objective, maximal conditional covering location problem applied to the relocation of hierarchical emergency response facilities. Omega 66: 147-158.

37. Gross D, Harris CM (1988) Fundamental of Queuing Theory.

38. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast elitist multi-objective genetic algorithm: NSGA-II. EEE Trans Evol Comput 6: 182-197.

39. Yeniay, Ankare B (2005) Penalty function methods for constrained optimization with genetic algorithms. Math Comput Appl 10: 45-56.

40. Taguchi G (1986) Introduction to quality engineering, White Plains: Asian Productivity Organization/UNIPUB.