

A CONGESTED CAPACITATED LOCATION PROBLEM WITH CONTINUOUS NETWORK DEMAND

MAHMOUD GOLABI^{1,*}, SEYED MAHDI SHAVARANI² AND LHASSANE IDOUMGHAR³

Abstract. This paper presents a multi-objective mixed-integer non-linear programming model for a congested multiple-server discrete facility location problem with uniformly distributed demands along the network edges. Regarding the capacity of each facility and the maximum waiting time threshold, the developed model aims to determine the number and locations of established facilities along with their corresponding number of assigned servers such that the traveling distance, the waiting time, the total cost, and the number of lost sales (uncovered customers) are minimized simultaneously. Also, this paper proposes modified versions of some of the existing heuristics and metaheuristic algorithms currently used to solve NP-hard location problems. Here, the memetic algorithm along with its modified version called the stochastic memetic algorithm, as well as the modified add and modified drop heuristics are used as the solution methods. Computational results and comparisons demonstrate that although the results obtained from the developed stochastic memetic algorithm are slightly better, the applied memetic algorithm could be considered as the most efficient approach in finding reasonable solutions with less required CPU times.

Mathematics Subject Classification. 90B80, 90C27, 90C11.

Received September 27, 2021. Accepted September 24, 2022.

1. INTRODUCTION

As one of the elements of paramount importance in the field of logistics, the facility location problem (FLP) has been scrutinized by a myriad number of researchers, especially in the context of operations research applications. According to the existing literature reviews [1–3], FLP could be categorized into various classes based on the nature of applied parameters, mobility of servers, type of variables, objective function(s), and applications.

Based on the uncertainty of the model parameters such as traveling times, demands, and costs, FLP could be classified into stochastic or deterministic problems [4,5]. As instance, [6] studied a warehouse location problem subject to data contamination to address uncertain environment and human errors in disaster response. Li *et al.* [7] proposed a bi-level stochastic optimization model for selecting the shelter locations in the case of disaster evacuation based on dynamic traffic assignment. An *et al.* [8] studied a stochastic location problem

Keywords. Capacitated facility location problem, queuing theory, continuous network demand, heuristics and metaheuristic algorithms.

¹ University of Haute-Alsace, IRIMAS UR 7499, 68100 Mulhouse, France.

² Alliance Manchester Business School, University of Manchester, Manchester M15 6PB, UK.

³ University of Haute-Alsace, IRIMAS UR 7499, 68100 Mulhouse, France.

*Corresponding author: mahmoud.golabi@uha.fr

integrated with en-route traffic congestion, in-facility queuing delay, and facility disruptions for disaster response planning.

According to 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 [9–11]. In contrast, customers are supposed to visit the servers in the case of immobile servers [12, 13].

FLPs could be studied on discrete or continuous space problems. In discrete space problems, the facilities are to be located on a finite set of pre-defined candidate locations [14, 15], but in the continuous space problems, any point in the plane could be opted for establishing facilities [16, 17].

Based on the objective function, FLPs are categorized into center, covering, and median problems. Center problems minimize the maximum traveling distance or service times [18]. Median problems minimize the total traveling or waiting times [19]. Covering problems maximize the number of covered demands [20]. These problems are classified into set covering and maximal covering location problems [21]. Set covering aims to satisfy a specified level of coverage, while maximal covering tries to increase the possible coverage rate.

In contrast with the incapacitated problem with no bound for the services, capacitated location problems refer to those FLPs in which the required resources are limited and all the assigned demands may not be satisfied [22, 23]. The unsatisfied demands, also known as outliers, are defined as those portions of the demand that their satisfaction is not economically justified [24]. The outliers will be referred to as “lost sales” in the rest of this paper. On the premise that the number of assigned customers cannot exceed the capacity of each facility, Torrent *et al.* [12] studied a capacitated location-allocation problem with lost sales to maximize the number of covered demands. The idea of lost sales has been rarely considered in the capacitated median problems. While respecting the capacity restrictions, the majority of related studies have assumed that there would be the possibility of covering all the demands using the available resources [7, 10, 25]. Harris *et al.* [26] developed a bi-objective capacitated inventory-location model to minimize the environmental effects of established facilities and the financial costs simultaneously. The capacity restrictions reflect the maximum number of items that can be stored and the maximum number of customers that can be served. Referring to the extended enterprise as a set of entities with different capacity levels, such as internal units and external partners, Abbal *et al.* [27] studied a bi-level facility location problem considering customers-depots and depots-plants relationships at different levels.

The idea of congestion has been used rampantly in FLPs. In congested location problems, customers may wait in line to receive the service. As it is reviewed by Boffey *et al.* [28], congested location problems can be classified according to criteria such as the number of servers in each facility, rules for assigning customers to the facilities, and the probabilistic distribution of arrival and service times. Aboolian *et al.* [29] presented one of the first multiple server congested center location problems using the M/M/C queuing system to minimize the maximum traveling time plus the expected waiting time. Considering the same queuing framework, Sayarshad and Chow [30] proposed a non-linear programming model for the relocation problem of on-demand mobility services. Their model minimizes the travel time between customers and servers, the relocation costs to get the vehicle to the new locations, and the delay costs.

Incorporating capacity restrictions into the node-based congested location problems has attracted great interest in recent years. Applying the queue length and the number of servers restrictions, Tavakkoli *et al.* [31] proposed a multi-objective capacitated location model without lost sales. Shavarani *et al.* [32] studied a multi-level capacitated fuzzy facility location problem to determine the locations of launching and refueling stations in an aerial delivery system in which the Unmanned Aerial Vehicles (UAVs) are used as the delivery means. While considering a general distribution for the service time, they assumed that there are enough resources to cover all the demands. The idea of lost sales has also been considered in some of the studied capacitated congested location problems. Using an exponential service time in a single-server congested system with queue length restrictions (M/M/1/K), Berman *et al.* [33] studied a location problem to determine the minimum number and locations of required facilities such that the number of demands lost due to coverage radius and congestion restriction does not exceed a pre-defined threshold. Using the same queuing framework, Rahmati *et al.* [34]

TABLE 1. A comparison of the present study with the most related works.

Ref.	Type	Queuing	Capacity	Lost*	Servers	Solution Method
Li <i>et al.</i> [7]	node		space-resource		immobile	Lagrangian relaxation
Aboolian <i>et al.</i> [29]	node	M/M/C			immobile	GA, Descent
Sinha [25]	node		service capacity		immobile	Add-Drop-Interchange
An <i>et al.</i> [8]	node	M/M/1	service capacity		immobile	Lagrangian relaxation
Pasandideh <i>et al.</i> [36]	node	M/M/1	service capacity		immobile	SA, GA, RS
Abbal <i>et al.</i> [27]	node		service capacity		immobile	Branch and Cut
Ilkhanizadeh <i>et al.</i> [10]	node		flying distance		mobile	Exact (CPLEX)
Torrent <i>et al.</i> [12]	node		service capacity	✓	immobile	Clustering-based SA
Harris <i>et al.</i> [26]	node		service capacity		immobile	SEAMO2
Rahmati <i>et al.</i> [34]	node	M/M/1/K	queue length	✓	immobile	NSGAI, NRGA
Etebari [35]	node	M/M/1/K	service & queue	✓	immobile	GA, PSO
Berman <i>et al.</i> [33]	node	M/M/1/K	queue length	✓	immobile	Add-Drop, Tabu Search
Tavakkoli <i>et al.</i> [31]	node	M/M/C/K	queue length		immobile	MOVDO, NSGAI, NRGA
Sayarshad and Chow [30]	node	M/M/C			immobile	Lagrangian decomposition
Zamani <i>et al.</i> [13]	node	M/M/1	service capacity	✓	immobile	Ant Lion, GA
Shavarani <i>et al.</i> [32]	node	M/G/M	endurance		mobile	MA
Shavarani <i>et al.</i> [11]	edge		flying distance		mobile	GA, MA
Jafari & Arkat [37]	edge	M/M/1			immobile	SA, GA, MA
Golabi <i>et al.</i> [38]	edge		flying distance		combined	GA, MA, SA
Shavarani <i>et al.</i> [39]	edge		endurance	✓	mobile	NSGAI, NSGAI
Golabi <i>et al.</i> [40]	edge	M/M/C	total servers		immobile	GA, MA, SA
This study	edge	M/M/C	servers at each facility	✓	immobile	MA, SMA, ADD, DROP

considered a bi-objective FLP to minimize the establishing costs along with the summation of traveling and waiting times, simultaneously. The lost demand was only applied in their calculation of effective waiting time. The same queuing system was applied by Etebari [35] to maximize the total profit gained from covered demands. On the premise that customers leave the system in the case of facing a disrupted facility, Zamani *et al.* [13] applied an M/M/1 queuing framework with service capacity to maximize the profit gained from fully covered demands. Table 1 illustrates a summary of the aforementioned congested and/or capacitated location problems, as the main focus of the current study.

Adverse to the conventional location problems in which a network of nodes is studied, flow capturing location problems (FCLP) and flow refueling location problems (FRLP) consider a network of edges each directed toward a specific flow [41]. Inspired by FCLPs, the edge-based location problem has been recently introduced as an alternative to conventional node-based FLPs. It is believed that the idea of distributed demands along the network edges reflects a more realistic image of the problem. The edge-based location studies are considered a new branch of facility location problems. Although only a few studies focus on this variant, all the definitions, and categorizations used for node-based studies also work for edge-based facility location problems. One of the first works focusing on edge-based FLPs was studied by Jafari and Arkat [37]. They proposed a novel model for a single-server congested location problem using an M/M/1 queuing system. Golabi *et al.* [38] developed a mixed-integer non-linear programming model for a stochastic edge-based median problem considering both mobile and immobile servers depending on the nature of the demand. Their objective function minimizes the traveling time of both customers and immobile servers over a set of feasible scenarios. The idea of distributed demands was also used in a study by Shavarani *et al.* [11] wherein regarding the flying range restrictions, a bi-level edge-based mobile location problem is modeled to determine the location of launching and refueling stations in a drone-based delivery system. Later, Shavarani *et al.* [39] considered a capacitated bi-objective mobile facility location problem with uniformly distributed demands in UAV-supported delivery operations to minimize the facility costs and lost sales simultaneously.

To the best of the authors' knowledge, the only study considering the congested capacitated edge-based location problem is done by Golabi *et al.* [40], where the authors introduced an expanded version of [37] by using a multiple-server environment and considering the maximum available servers restriction. Assuming a pre-known number of opened centers, the location of facilities and their corresponding number of servers were the decision variables. Despite using a restriction on the total number of assigned servers, the authors did not consider any limitation for the maximum number of servers at each location. They also assumed that the available servers have the potential of covering all the customers and thus no lost demand was considered. Using the M/M/C queuing framework, the objective function was to minimize the summation of customers' traveling and waiting times.

To reflect a more realistic image of the problem and account for the available space limitation, the current study extends [40] by applying a restriction on the number of servers assigned to each facility and incorporating the idea of lost sales. The classical constraints for the assignment of customers and the same formulation introduced in [40] for decomposing the network edges are applied in this study. Besides, new constraints are introduced to account for lost sales. It is also assumed that although all the customers travel to the location of facilities, due to capacity restriction, just a proportion of them would wait in the queue to be served and the rest are dismissed. The current study makes two contributions to the edge-based facility location literature. First, it proposes a new problem definition by incorporating the idea of lost sales into the congested location problems in a multi-objective context. Second, several modified solution approaches are developed to solve the problem. Besides, it is tried to present a comprehensive review of the related literature. For a fuller view, the main contributions of the this study could be listed as follows:

- The number of opened facilities is not known and fixed in advance, and each facility has a threshold for the number of assigned servers to account for the available space restrictions.
- Lost sales and waiting time for satisfied customers are optimized.
- A new mathematical formulation is developed to tackle the complications of the problem.
- A multi-objective version of the study with conflicting functions is introduced.
- New modified solution algorithms are introduced.

The objective functions that are considered include the aggregate customers' traveling distance, the aggregate covered customers waiting time, the total cost of the system, and the total number of lost sales. Here, a method based on the weighted distance metric is used for transforming the developed multi-objective problem into a single-objective one.

The complexity of traditional facility location problems has been thoroughly analysed in the literature. It has been shown that center problems, maximal covering, and median problem are NP-hard [42–44]. Set covering location problems are also known to be NP-complete [42]. Therefore, the application of metaheuristics in solving location problems is prevalent [45]. It is clear that the contributions mentioned in our work heightens the complexity of the studied problem and the concomitant solution methods. As the solution methods, this study proposes modified versions of some of the existing heuristics and metaheuristic algorithms widely used to solve NP-hard location problems. Memetic algorithm which is known as an effective metaheuristic to solve the edge-based location problems [11,37,38,40] is applied in this study. Add and Drop algorithms that are popular greedy heuristics able to find high-quality solutions efficiently [46–48] are also used as solution methods. Due to the complexity of the developed mathematical model and the need for optimizing the number of servers with the number and location of facilities, the performance of the original Add and Drop heuristics were not promising. Thus, this study develops modified versions of applied algorithms to improve the quality of obtained results.

The remainder of this study is organized as follows. Section 2 describes the problem and the mathematical model. Solution methods are described in Section 3. Numerical results are reported in Section 4 and the conclusion is drawn in Section 5.

2. PROBLEM DEFINITION

The edge-based facility location problem considers the idea of distributed demands along the network edges. Due to the uncertain nature of demand generation, the location and the rate of demands on each edge follow a probabilistic distribution. Any point on network edges could be a potential point for demand occurrence. Thus, the uniform distribution could be used to model the position of demand generation points along the network edges. On the premise that the demands are uniformly distributed along the network edges, this paper studies a multiple-server multi-objective congested immobile facility location problem on the discrete plane. As a continuous distribution commonly used to measure the expected time for an event to occur [49], it is assumed that the time interval between the arrival of any two consecutive demands and the service time of each server follow exponential distributions. Therefore, the demand occurrence rate on each network edge follows a Poisson distribution with a rate dependent on the population density of the area and the length of the edge. These arrival and service time distributions are widely used in queuing systems⁴. Since the customers located on the same network edge (homological demands) have different distances to their closest open facility, the time interval between demand generations differs from the time interval between the arrivals of customers and as a result, the demands generated earlier may wait longer. The number of the opened facilities and their corresponding number of assigned servers are the decision variables limited by a restriction on the maximum number of servers at each facility. Since the customers travel to the location of the closest open facility, their movement along the network edges could be considered as M/G/ ∞ queuing systems with Poisson outputs [50]. Therefore, as the demand enters the open facilities, the queuing system at each facility would be according to M/M/C. It is assumed that customers are served based on the First-in-First-out policy. The capacity of each facility is restricted by its total number of assigned servers and their associated service quality coefficient. The total number of demands that can be satisfied through each facility is limited and the excess demands that arrive at the facility but fail to receive services are considered as lost sales. It should be noted that the lost demand does not affect the calculations of the waiting time. Minimizing the aggregate customers' traveling distance, the aggregate customers' waiting time, the total cost of the system, and the total number of lost sales are the four objective functions addressed in this study.

2.1. Model formulation

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

Sets:

- I : the set of network nodes ($i, i' \in I$)
- E : the set of network edges ($E \subseteq I, (i, i') \in E$)
- J : the set of candidate locations ($j, j' \in J$)

Parameters & Scalars:

- $l_{ii'}$: the length of edge ($i, i' \in I$)
- $\lambda_{ii'}$: the demand rate of edge ($i, i' \in I$)
- D_{ij} : the minimum distance between node i and facility j
- P_j^f : the cost of establishing a facility at candidate location j
- P^c : the unit cost of servers
- c_j^{\max} : the maximum number of servers that can be assigned to location j
- μ : the common service rate of servers
- w_{\max} : the maximum time that each customer can wait in the system
- α : the coefficient of service quality level
- ϵ : an infinitesimal positive number
- M : a large positive value

⁴The same assumption used in all the congested studies mentioned in Table 1, except for [32].

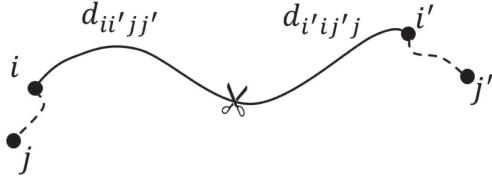


FIGURE 1. A decomposed network edge.

Decision Variables:

- x_{ij} : 1 if node i is assigned to facility j ; 0 otherwise
- y_j : 1 if location j is selected for opening a facility; 0 otherwise
- c_j : the number of servers assigned to facility j

Output Variables:

- $d_{ii'jj'}$: the distance between node i and decomposing point of the edge (i, i') , where i and i' are assigned to opened facilities j and j' , respectively
- w_j : the expected waiting time at opened facility j
- π_j^0 : the probability that opened facility j is idle
- γ_j : the demand entrance rate of opened facility j
- S_j : the number of customers served at opened facility j

According to Figure 1, if the closest opened facilities to the endpoints of a network edge are not identical, the edge would be decomposed into two different segments, each assigned to its closest opened facility.

As it is illustrated in Figure 1, on the premise that j and j' are respectively the closest opened facilities to the endpoints i and i' , the edge (i, i') is decomposed into two segments. Customers located on the first segment with the length of $d_{ii'jj'}$ are assigned to facility j , whereas the rest are assigned to the facility j' . Since the decomposing point has the same distance to the closest opened facilities of the endpoints ($d_{ij} + d_{ii'jj'} = d_{i'j'} + d_{i'ij'j}$) and having the length of the edge ($d_{ii'jj'} + d_{i'ij'j} = l_{ii'}$), the length of each segment is calculated using equation 1:

$$d_{ii'jj'} = \frac{d_{i'j'} + l_{ii'} - d_{ij}}{2}. \quad (1)$$

It is incontrovertible that $d_{ii'jj'}$ has a non-negative value less than or equal to $l_{ii'}$, since on the contrary case, facilities j and j' would not be the closest open facilities to nodes i and i' respectively, and it is in contradiction with the considered assumptions. It is clear that if the closest facility to nodes i and i' is identical, the entire edge is assigned to the same facility.

Using the general formula for the probability density function of the uniform distribution, the expected traveling distance of customers located on the edge (i, i') for arriving at the corresponding endpoint is calculated as:

$$\text{Traveling Distance} = \lambda_{ii'} \int_0^{l_{ii'}} \frac{T}{l_{ii'}} dT = \lambda_{ii'} \frac{l_{ii'}}{2}. \quad (2)$$

Based on the splitting property of Poisson distribution, in the case of decomposing edge (i, i') , the average traveling distance for the demands located on the segment $d_{ii'jj'}$ to arrive at the location of facility j is calculated using equation 3:

$$\text{Traveling Distance} = \frac{\lambda_{ii'} d_{ii'jj'}}{l_{ii'}} \left(D_{ij} + \frac{d_{ii'jj'}}{2} \right). \quad (3)$$

As the result, the developed multi-objective mixed-integer non-linear programming (MINLP) model can be formulated as follows:

$$\text{Min } Z_1 = \sum_{j \in J} \sum_{j' \in J} \sum_{(i, i') \in I} \frac{d_{ii'jj'}}{l_{ii'}} \left(D_{ij} + \frac{d_{ii'jj'}}{2} \right) x_{ij} x_{i'j'} \quad (4)$$

$$\text{Min } Z_2 = \sum_{j \in J} S_j w_j \quad (5)$$

$$\text{Min } Z_3 = \sum_{j \in J} p_j^f y_j + p^c c_j \quad (6)$$

$$\text{Min } Z_4 = \sum_{j \in J} L_j. \quad (7)$$

Subject to:

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (8)$$

$$x_{ij} \leq y_j \quad \forall i \in I, \forall j \in J \quad (9)$$

$$\sum_{j' \in J} x_{ij'} D_{ij'} \leq D_{ij} M(1 - y_j) \quad \forall i \in I, \forall j \in J \quad (10)$$

$$2 d_{ii'jj'} = D_{i'j'} + l_{ii'} - D_{ij} \quad \forall i, i' \in I, \forall j, j' \in J \quad (11)$$

$$\gamma_j = \sum_{j' \in J} \sum_{(i, i') \in I} \frac{\lambda_{ii'} d_{ii'jj'}}{l_{ii'}} x_{ij} x_{i'j'} \quad \forall j \in J \quad (12)$$

$$c_j \leq M y_j \quad \forall j \in J \quad (13)$$

$$S_j = \min(\alpha c_j \mu, \gamma_j) \quad \forall j \in J \quad (14)$$

$$S_j + L_j = \gamma_j \quad \forall j \in J \quad (15)$$

$$\pi_j^0 = \left(\sum_{n=0}^{c_j-1} \left(\frac{S_j^n}{\mu^n n!} \right) + \frac{S_j^{c_j}}{\mu^{c_j} c_j! \left(1 - \frac{S_j}{c_j \mu} \right)} \right)^{-1} \quad \forall j \in J \quad (16)$$

$$w_j = \frac{1}{\mu} + \left(\frac{S_j^{c_j}}{\mu^{c_j} (c_j!) (c_j \mu) \left(1 - \frac{S_j}{c_j \mu} \right)^2} \right) \pi_j^0 \quad \forall j \in J \quad (17)$$

$$w_j \leq w_{\max} \quad \forall j \in J \quad (18)$$

$$c_j \leq c_j^{\max} \quad \forall j \in J \quad (19)$$

$$c_j \geq \frac{S_j}{\mu} + \epsilon \quad \forall j \in J \quad (20)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (21)$$

$$\gamma_j, c_j, d_{ii'jj'}, w_j, S_j, L_j \geq 0 \quad \forall i, i' \in I, \forall j, j' \in J \quad (22)$$

$$0 \leq \pi_j^0 \leq 1 \quad \forall j \in J. \quad (23)$$

Objective functions (4), (5), (6), and (7) minimize the customers' aggregate expected traveling distance, the customers' aggregate expected waiting time, the total cost, and the total number of lost sales, respectively. Constraint (8) guarantees that each customer is assigned to only one opened facility. Assigning customers to closed facilities is refrained by constraint (9). Constraint (10) assures that each customer is assigned to its closest opened facility. Constraints (11) and (12) calculate the length of partitioned segments of each network edge and the demand entrance rate at each facility, respectively. Constraint (13) ensures that the servers are assigned just to the opened facilities. While assuring that the queuing system will reach a steady-state mode, constraint (14) reflects the capacity of each opened facility based on the number of assigned servers and their service quality level. Constraint (15) categorizes the customers to covered and lost sales. Constraints (16) and (17) measure

the idle probability and expected waiting time at each opened facility, respectively. The maximum waiting time threshold is considered by constraint (18). Constraint (19) sets an upper bound for the number of servers assigned to each facility. Constraint (20) assures reaching the steady state mode. Constraints (21), (22), and (23) reflect the binary and non-negative decision variables.

3. SOLUTION METHODS

Since the developed mathematical model is categorized as a constrained mixed-integer nonlinear programming model, it is considered as an NP-hard problem, so achieving the exact solution for such a problem is difficult, if not say impossible, specifically when the size of the problem is big [36]. However, the first attempt to solve the problem was to code the model in General Algebraic Modeling System (GAMS) and solve it using exact MINLP solvers such as BONMIN, DICOPT, and KNITRO. In order to transform the four objective functions given in equation (4)–(7) into a single one, the L_p -metric method is used as a weighting scheme in this study [51]. Using equation (24), the method minimizes the differences between objective functions and their pertinent desired values ($z_i^* \in \mathbb{R}^Q$):

$$\text{Min } Z(x) = \left[\sum_{i=1}^Q \left[w_i \left| \frac{z_i^* - z_i(x)}{z_i^*} \right| \right]^p \right]^{\frac{1}{p}}. \quad (24)$$

Subject to:

$$x \in X \subseteq \mathbb{R}^Q. \quad (25)$$

Where w_i , which is defined by the decision-maker, is the weight of the objective function $z_i(x)$, p is a positive integer, and z_i^* is the optimum value of objective function i , if considered alone. If $p = 1$, which is the case of this study, the method is called the Manhattan metric.

It is noteworthy to mention that it is not possible to solve the model in its current form using GAMS. It is due to having a *min* function over the decision variables in constraint (14), a decision variable ($c_j - 1$) as the upper-bound of the first summation operator in constraint (16), and the factorial of a decision variable ($c_j!$) in constraints (16) and (17). To get rid of these problems, a constraint manipulation was done by introducing a new decision variable called z_{kj} where z_{kj} : 1 if node there are k servers assigned to facility j ; 0 otherwise. Now, constraints (13) is replaced with:

$$\sum_{k=1}^{c_j^{\max}} k z_{kj} \leq M y_j \quad \forall j \in J. \quad (26)$$

Introducing a new set of binary decision variables called q_1 and q_2 , such that $q_1 + q_2 = 1$, constraint (14) could be replaced with the following set of constraints:

$$S_j \leq \sum_{k=1}^{c_j^{\max}} \alpha k z_{kj} \mu \quad \forall j \in J \quad (27)$$

$$S_j \leq \gamma_j \quad \forall j \in J \quad (28)$$

$$S_j \geq \sum_{k=1}^{c_j^{\max}} \alpha k z_{kj} \mu - M q_1 \quad \forall j \in J \quad (29)$$

$$S_j \geq \gamma_j - M q_2 \quad \forall j \in J. \quad (30)$$

In a similar way, constraints (16), (17), (20), and (21) are replaced with the following set of constraints, respectively. It is noteworthy to mention that constraint (19) is not required anymore, as the total number of

servers is calculated using the summation of kz_{kj} over k , where k runs from 1 to c_j^{\max} .

$$\pi_j^0 = \left(\sum_{k=1}^{c_j^{\max}} z_{kj} \left(\sum_{n=0}^{k-1} \left(\frac{S_j^n}{\mu^n n!} \right) + \frac{S_j^k}{\mu^k k! \left(1 - \frac{S_j}{k\mu} \right)} \right) \right)^{-1} \quad \forall j \in J \quad (31)$$

$$w_j = \frac{1}{\mu} + \left(\sum_{k=1}^{c_j^{\max}} z_{kj} \frac{S_j^k}{\mu^k (k!) (k\mu) \left(1 - \frac{S_j}{k\mu} \right)^2} \right) \pi_j^0 \quad \forall j \in J \quad (32)$$

$$\sum_{k=1}^{c_j^{\max}} k z_{kj} \geq \frac{S_j}{\mu} + \epsilon \quad \forall j \in J \quad (33)$$

$$x_{ij}, y_j, z_{kj}, q_1, q_2 \in \{0, 1\} \quad \forall i \in I, \forall j \in J. \quad (34)$$

The edge-based facility location literature suffers from the lack of benchmark instances required to check the validity of developed models, or to analyze the performance of the proposed solution algorithms. Except for the real-life case studies, all the previously mentioned works in this domain have generated random instances to discuss the performance of their solution methods. This study also uses some randomly generated instances to compare the results obtained from different methods. To generate semi-realistic instances, a desired number of network nodes are randomly positioned in a network. Using a random number of connecting links passing through each network node, the generated nodes are linked to form a connected graph. To simulate more realistic transportation networks, it is assumed that the number of connecting links varies from 1 to 8 while considering a higher probability for numbers between 2 and 4. A random number in range [1, 3] is multiplied by the euclidean distance of all pairs of connected network nodes to account for the length of the corresponding network edge. The candidate locations are randomly selected among the existing network nodes. The cost of establishing facilities at each candidate location is a random integer in range [100K, 200K]. The demand rate on each network edge is randomly chosen in range (0, 4]. The cost of each server is fixed to 5000. The common service rate and maximum allowed waiting time are fixed at 25 and 0.5, respectively. Using this procedure, several small instances are randomly generated to check the performance of developed algorithms and compare the results with optimal solutions obtained from exact MINLP solvers of GAMS compiler.

The results obtained from solving the model in the GAMS environment using different solvers and a time limit of 7200 seconds are shown in Table 2. The reported values are obtained from the Manhattan-metric method by considering equal weights for normalized objective functions. The normalization is done by dividing each objective function by its corresponding maximum value. Assigning each customer to its farthest facility would yield the maximum value of the first objective function. Considering the maximum allowed waiting time for each customer returns the maximum value of the second objective function. Opening all the candidate sites and assigning the maximum number of allowed servers yields the maximum value of the third objective function. The maximum value of the fourth objective is estimated by considering all demands as lost sales. According to Table 2, the obtained results revealed that the maximum solvable size of the problem is limited to 70 network nodes (endpoints of the edges). Comparing this size with the real city networks having thousands of network nodes is proof of the incapability of exact methods in solving such a complex problem for large instances. Since all the MINLP solvers of GAMS failed to solve the model for realistic instances of the problem, the genetic algorithm hybridized by a local search technique, called Memetic Algorithm (MA), along with its modified stochastic version called Stochastic Memetic Algorithm (SMA), as well as modified versions of two well-known heuristics in solving location problems, called modified Add and modified Drop algorithms are used to solve the proposed model. It is noteworthy to mention that the same results of solving small instances using GAMS solvers were also obtained by all of the developed solution algorithms (see Tab. 2).

TABLE 2. Computational results for small instances.

Tests	Nodes	GAMS (BONMIN)	SMA	MA	ADD	DROP
1	30	0.6125	0.6125	0.6125	0.6125	0.6125
2	40	0.5241	0.5241	0.5241	0.5241	0.5241
3	50	0.6672	0.6672	0.6672	0.6672	0.6672
4	60	0.6893	0.6893	0.6893	0.6893	0.6893
5	70	0.5132	0.5132	0.5132	0.5132	0.5132
6	80	—	0.5617	0.5617	0.5617	0.5617
7	90	—	0.7151	0.7151	0.7151	0.7151

3.1. Memetic Algorithm (MA)

Evolutionary algorithms combined with local search techniques are generally called memetic algorithms [52]. This combination is used to merge the biological evolution with the individual learning procedure to mimic the cultural evolution [53]. The individual learning procedures are exerted *via* local search techniques [54]. In this paper, MA equips the Genetic Algorithm (GA) with a local search technique. The algorithm starts with a population of N_{pop} number of randomly generated solutions called chromosomes. Each chromosome (S) is represented as a matrix of size $2 \times n$, where n is the total number of the candidate locations. The first n genes of the encoding matrix form a binary vector representing open facilities by 1 and closed facilities by 0. The second part determines the number of assigned servers by a randomly generated integer number between 1 and the maximum number of servers for the corresponding facility. It is noteworthy to mention that the second part alleles corresponding to closed facilities is set to be equal to zero. Since the satisfaction of constraints (18), (19), and (20) could not be guaranteed using the mentioned structure, a repairing operation that may be followed by applying a penalty function is used to refrain from dealing with infeasible solutions in the initial population. The repairing operation tries to satisfy constraints (18) and (20) by iteratively adding one server to the facility for which the the current number of assigned servers cannot satisfy the steady state and waiting time threshold restrictions. If satisfaction of constraints (18) and (20) results in violating constraint (19), a penalty is added to the objective function. After selecting parents using the roulette wheel procedure, the two-point crossover operation depicted in Figure 2a is applied to merge the parents and generate two offspring.

In this study, one of the three different types of mutation operators is randomly selected for being applied to each generated offspring (Fig. 2b). The mutate operator selects a random number of genes in the first part of the chromosome and changes the status of the corresponding candidate location. The corresponding genes in the second part are filled in a manner described in solution representation. After selecting two random points along the chromosome, the reverse operator inverts the order of genes trapped between them. Finally, the swap operator considers two randomly selected candidate locations and exchanges their corresponding genes. The crossover and mutation operations are controlled by the probabilities called p_c and p_m , respectively. The repairing operator and the penalty function are considered for infeasible solutions generated.

Through successive iterations, the local search technique generates a predetermined number of neighborhood solutions ($LocIt$) for each offspring generated *via* mutation operation. Neighborhood solutions are generated using the mutate technique described for the mutation operator. It means that not only the number of assigned servers of the selected genes may be changed, but also the status of the candidate location is reconsidered. It is noteworthy to mention that the repairing operator and penalty function are considered for infeasible solutions. In each iteration of developed MA, the generated solutions through the crossover, mutation, and local search operators are added to the population. After sorting the population based on the fitness value of the solutions, the best solutions are kept to resize the population to the initial N_{pop} . The algorithm stops after reaching a predetermined number of iterations ($MaxIt$). The pseudo-code of the applied MA is illustrated in Figure 3a.

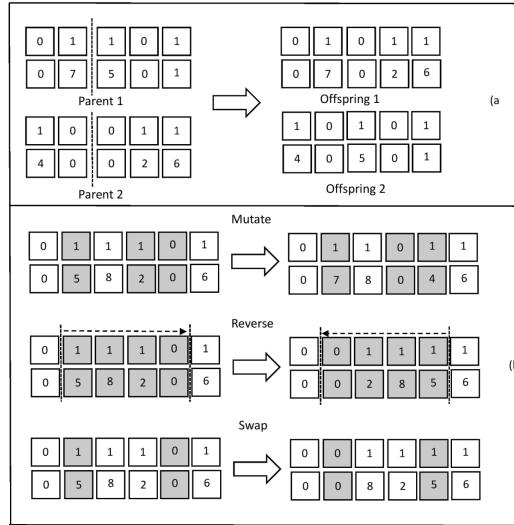


FIGURE 2. a) The crossover operation; b) The mutation operation.

(a) MA	(b) Stochastic Local Search
<i>Generate the initial population (S)</i>	$i \leftarrow 1$
<i>For each chromosome $s \in S$</i>	$BS_l \leftarrow (M)$
<i>Apply repairing function if necessary</i>	$LocSol \leftarrow (M)$
<i>End for</i>	<i>While { $i < locIt$ } do</i>
<i>While { the stopping criteria is not reached} do</i>	<i>Apply local search on $LocSol$ to generate Li</i>
<i>Select parents (P)</i>	<i>if $obj(Li) < obj (BS_l)$</i>
<i>Crossover (P) to generate (C)</i>	<i>$BS_l \leftarrow Li$</i>
<i>Apply repairing function on (C), if necessary</i>	<i>$LocSol \leftarrow Li$</i>
<i>Mutate (C) to generate (M)</i>	<i>Else if $rand(0,1) < p_l$</i>
<i>Apply repairing function on (M), if necessary</i>	<i>$LocSol \leftarrow Li$</i>
<i>Local Search (M) to generate (L)</i>	<i>Else</i>
<i>Apply repairing function on (L), if necessary</i>	<i>$LocSol \leftarrow BS_l$</i>
<i>$(S) \leftarrow [(S)+(M)+(L)]$</i>	<i>End if</i>
<i>Evaluate & Sort (S)</i>	<i>$i \leftarrow i+1$</i>
<i>$(S) \leftarrow S(1:Npop)$</i>	<i>End while</i>
<i>End while</i>	

FIGURE 3. The pseudo-code of a) MA; b) stochastic local search technique.

3.2. Stochastic Memetic Algorithm (SMA)

The difference between the applied SMA and MA is in the structure of the local search technique. All the neighborhood solutions of MA are generated by applying the local search method to the solution produced by the mutation operator. On the contrary, each generated neighborhood solution in SMA has the chance of being the basis for applying the local search technique and generating other solutions. In other words, the local search in MA is applied on a fixed solution while inspired by the Simulated Annealing algorithm, the solution that is undergoing the local search technique is stochastically updated in SMA. To do so, first, the offspring generated *via* mutation operator is set to be the best solution of the local search technique (BS_l).

The generated neighborhood solution is compared to the best solution in terms of the objective function. If the generated neighborhood solution has a better objective value, BS_l is updated and the next local search operator is applied on the updated BS_l . Otherwise, based on a predetermined probability (p_l) used for refraining from trapping in local optima, a decision is made to apply the next round of local search technique on the current solution or the BS_l . The pseudo-code of the applied stochastic local search technique is illustrated in Figure 3b. Considering the applied mutation, crossover, and roulette wheel selection operations, the complexity of GA is $\mathcal{O}(MaxIt(Npop \times |J| + Npop \times |J| + Npop))$, which is equivalent to $\mathcal{O}(MaxIt \times Npop \times |J|)$ [55]. The overall time complexity of the applied memetic algorithm as a hybridization of GA with the local search operation is $\mathcal{O}(MaxIt \times Npop \times |J| \times LocIt)$ in the worst case.

3.3. Modified Add & Drop Heuristics

In the local search-based Add [56] and Drop [57] heuristics, facilities are successively placed in or deleted from locations, dependent on some measures of suitability. In this way, different combinations of selected locations are checked in the search procedure [25].

Drop heuristic initiates by opening all the candidate locations with the maximum number of servers. The objective function is calculated and then the algorithm determines if closing any of the facilities leads to cost savings or improvement of the objective function. If so, the selected facility will be closed and the algorithm continues to search for the next facility to close. In case there are several facilities that their drop from the solution leads to the improvement of the objective function, the one with the highest impact is selected. If no cost saving is possible, the algorithm terminates.

Add heuristic starts with all the facilities closed. Then the algorithm tries to open facilities one at a time. The facility that is selected is the one with the highest gains in the objective function. The candidate location which yields the best objective function is opened and the algorithm proceeds to the next step where it would look for the second candidate location to open beside the first one. The algorithm continues to increase the size of the selected candidate locations while the objective function is improved compared to the previous step. Thus the algorithm terminates in a finite number of iterations.

In this study, the maximum number of servers is assigned to the facilities. In the next step with the set of opened facilities at hand, a greedy algorithm is utilized to justify the total number of assigned servers for each facility. In this respect, facilities are selected one by one and the total number of servers is decreased continuously while both constraints (18) and (20) are still satisfied and the quality of objective function improves, otherwise, the greedy algorithm moves to the next facility. The complete solution comprised of opened facilities and their respective number of servers is returned by the algorithm. The pseudo-codes of the modified Add and Drop heuristics are illustrated in Figure 4.

The add heuristic starts with the determination of the cheapest facility. First, each facility is considered independently and as the only open facility. Having $|J|$ potential facilities, the cost evaluations are repeated $|J|$ times in the first step to select the cheapest facility (lets say F_{c1}). In the next step, we will consider the combination of F_{c1} with the remaining $|J| - 1$ facilities. The process goes on as described in pseudocode 4. Thus, the total number of cost evaluations for selecting open facilities is $|J|(|J| - 1)/2$. For each facility we consider different number of servers $\in \{1, \dots, c_j^{\max}\}$, c_j^{\max} being the maximum number of servers for j^{th} facility. Considering $c^{\max} = \max_{j \in J} c_j^{\max}$, the maximum number of evaluations for the Add heuristic will be $c^{\max}|J|(|J| - 1)/2$. Accordingly, the complexity of this algorithm is equivalent to $\mathcal{O}(c^{\max} \times |J|^2)$. The drop heuristic complexity is equivalent to the ADD heuristic.

3.4. Parameter Tuning

The applied algorithm parameters affect the performance of developed metaheuristic solution algorithms. Therefore, tuning the parameters before the implementation will improve the quality of obtained solutions. This study applies the Taguchi method to tune the algorithm parameters. Taguchi uses orthogonal arrays to investigate a large number of controllable factors through a small number of experiments [32]. Taguchi determines

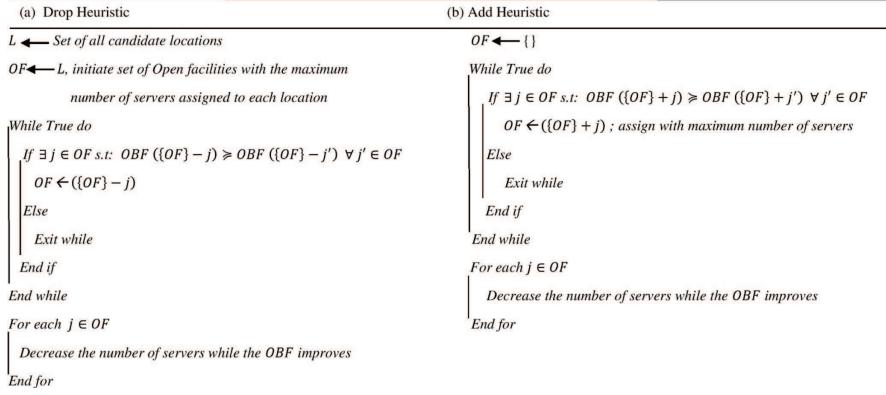


FIGURE 4. The pseudo-code of a) Drop Heuristic; b) Add Heuristic.

TABLE 3. Parameter levels.

Algorithms	Parameters	Range	Low (1)	Medium (2)	High (3)
SMA	MaxIt (A)	200–400	200	300	400
	Npop (B)	100–200	100	150	200
	P _c (C)	0.6–0.8	0.6	0.7	0.8
	P _m (D)	0.2–0.4	0.2	0.3	0.4
	LocIt (E)	5–10	5	8	10
	P _l (F)	0.1–0.2	0.1	0.15	0.2
MA	MaxIt (A)	200–400	200	300	400
	Npop (B)	100–200	100	150	200
	P _c (C)	0.6–0.8	0.6	0.7	0.8
	P _m (D)	0.2–0.4	0.2	0.3	0.4
	LocIt (E)	5–10	5	8	10

the best level of parameters *via* minimizing the noise effect. The variation of response is evaluated by a signal to noise ratio *S/N* that is calculated by using equation (35), in which Y denotes the response value and n represents the number of orthogonal arrays.

$$S/N = -10 \times \log(S(Y^2)/n). \quad (35)$$

The applied MA and SMA parameters with their corresponding ranges are shown in Table 3. Using each parameter combination shown in Table 4, five randomly generated instances with different specifications are solved five times and the average of obtained objective values are considered as the response value of the applied algorithms. The algorithms are coded in Python 3.8 and implemented on an Intel Core i7-6500U CPU @ 2.50 GHz laptop with 8 GB RAM, 6 MB L3 Cache, and 1 MB L2 Cache. The Minitab 19 is used to perform the Taguchi method. The obtained *S/N* ratios for applied MA and SMA are shown in Figure 5. As it is illustrated in Figure 5a, the SMA parameters should be set at their third, third, second, third, second, and third levels respectively.

4. EXPERIMENTAL RESULTS

All of the existing FLP benchmark datasets are designed for node-based location problems. To translate such benchmarks to edge-based instances, for each node one should consider a random number of edges to select

TABLE 4. Computational results for tuning MA and SMA.

Run	SMA Parameters						Response SMA	MA Parameters					Response MA
	A	B	C	D	E	F		A	B	C	D	E	
1	1	1	1	1	1	1	0.7829	1	1	1	1	1	0.7755
2	1	1	1	1	2	2	0.7778	1	1	1	1	2	0.7846
3	1	1	1	1	3	3	0.7718	1	1	1	1	3	0.7887
4	1	2	2	2	1	1	0.7701	1	2	2	2	1	0.7763
5	1	2	2	2	2	2	0.7589	1	2	2	2	2	0.7701
6	1	2	2	2	3	3	0.7694	1	2	2	2	3	0.7684
7	1	3	3	3	1	1	0.7634	1	3	3	3	1	0.7633
8	1	3	3	3	2	2	0.7589	1	3	3	3	2	0.7719
9	1	3	3	3	3	3	0.7589	1	3	3	3	3	0.7589
10	2	1	2	3	1	2	0.7589	2	1	2	3	1	0.7729
11	2	1	2	3	2	3	0.7694	2	1	2	3	2	0.7589
12	2	1	2	3	3	1	0.7589	2	1	2	3	3	0.7822
13	2	2	3	1	1	2	0.7793	2	2	3	1	1	0.7589
14	2	2	3	1	2	3	0.7589	2	2	3	1	2	0.7691
15	2	2	3	1	3	1	0.7701	2	2	3	1	3	0.7729
16	2	3	1	2	1	2	0.7653	2	3	1	2	1	0.7589
17	2	3	1	2	2	3	0.7652	2	3	1	2	2	0.7633
18	2	3	1	2	3	1	0.7633	2	3	1	2	3	0.7589
19	3	1	3	2	1	3	0.7589	3	1	3	2	1	0.7589
20	3	1	3	2	2	1	0.7684	3	1	3	2	2	0.7701
21	3	1	3	2	3	2	0.7822	3	1	3	2	3	0.7589
22	3	2	1	3	1	3	0.7589	3	2	1	3	1	0.7820
23	3	2	1	3	2	1	0.7589	3	2	1	3	2	0.7639
24	3	2	1	3	3	2	0.7589	3	2	1	3	3	0.7589
25	3	3	2	1	1	3	0.7589	3	3	2	1	1	0.7633
26	3	3	2	1	2	1	0.7633	3	3	2	1	2	0.7774
27	3	3	2	1	3	2	0.7653	3	3	2	1	3	0.7822

directly connected nodes, random edge lengths, and random demands on each generated edge. There is no doubt that this would completely change the nature of the considered benchmark dataset. Thus, generating appropriate instances to evaluate the performance of applied algorithms would be inevitable in edge-based location problems. Using the random instance generation procedure and the Manhattan-metric method for normalized objective functions of equal weights explained in Section 3, the results of implementing the proposed algorithms on 16 randomly generated test cases of different sizes and specifications are shown in Table 5. The average objective value and required CPU times of 10 runs of each proposed algorithm on each test case are shown in this table. In order to discuss the potentials of implementing the solution algorithms on real-life large-scale networks, all the test cases are generated in large size networks with a plethora of nodes and edges. To simulate a more realistic image, the number of edges passing through each network node varies between two and five. Generally, the downtowns are the main crowded areas of real-life case studies. To simulate this behavior, it has been tried to consider higher demand rates for the central edges, in comparison with those that are closer to the network barriers.

As the GAMS optimization compiler failed to solve the developed MINLP model even on small instances, the performances of the proposed algorithms are compared together. Using the same number of evaluations, Figures 6 and 7 provide the graphical comparison of applied algorithms in terms of obtained mean objective value and required CPU times, respectively. Additionally, the one-way analysis of variance (ANOVA) for the

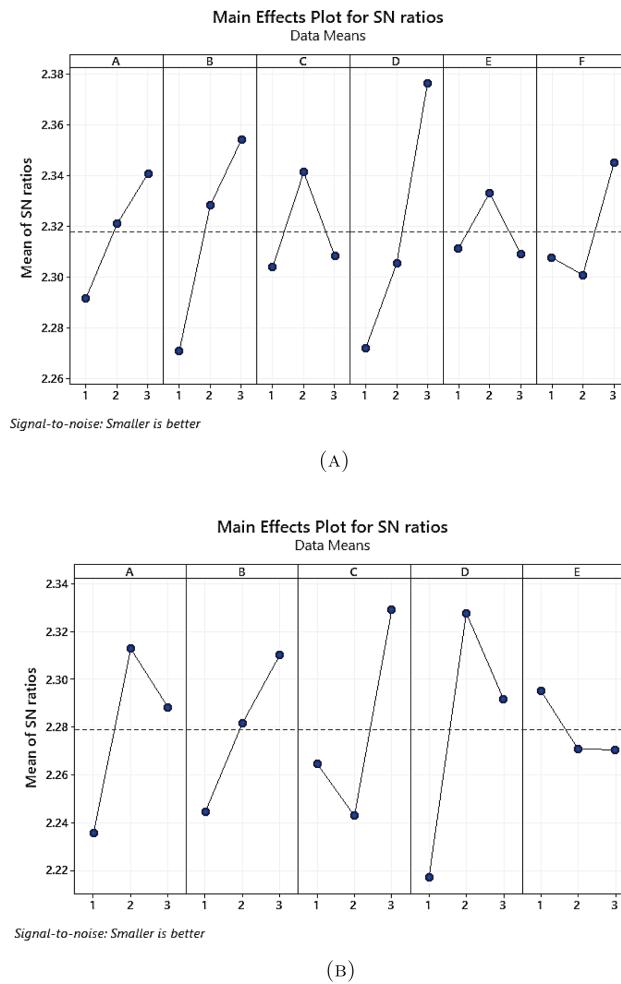


FIGURE 5. S/N plot. (A) SMA. (B) MA.

mean of objective values and CPU times is performed by Minitab software. Figure 8 illustrates obtained *p*-values and interval plots of the ANOVA test.

As it is shown in Figure 8, the worst results are obtained by the Add heuristic. Although there is no significant difference between the results obtained by MA and Drop algorithms, MA could be considered as a more efficient algorithm due to requiring fewer CPU times. The developed SMA slightly outperforms MA and Drop algorithms. It could be said that for the case of emergency applications such as determining shelter or relief distribution centers, MA could be considered as the most efficient algorithm to determine reasonable solutions. However, in strategic decision-making procedures where the required CPU time is not among the main criteria to evaluate the performance of applied algorithms, SMA could be considered as the best solution approach to find high-quality solutions.

5. CONCLUSION

Although the literature is replete with studies focusing on different types of node-based facility location problems, there are just a few studies concerning the idea of distributed demands along the network edges.

TABLE 5. Computational results.

#	I	J	SMA		MA		ADD		DROP	
			Obj	Time	Obj	Time	Obj	Time	Obj	Time
1	1000	150	0.4819	358	0.4829	264	0.5336	81	0.4952	342
2	1300	165	0.4782	446	0.5311	322	0.5985	115	0.5194	406
3	1700	180	0.5535	519	0.6842	381	0.7116	181	0.6333	529
4	2000	200	0.4811	649	0.4857	439	0.5613	271	0.4879	621
5	2250	225	0.4622	797	0.4995	491	0.5526	315	0.5038	840
6	2500	250	0.5024	943	0.5266	563	0.6019	392	0.5481	1169
7	2750	275	0.5125	1267	0.5465	670	0.6048	523	0.5721	1672
8	3000	300	0.4490	1567	0.5155	801	0.5585	676	0.4782	2294
9	3250	325	0.5148	1809	0.5309	1056	0.5911	778	0.5374	2970
10	3500	350	0.6281	2153	0.7164	1261	0.7475	814	0.6931	3642
11	3750	375	0.5512	2421	0.6071	1549	0.6623	892	0.6219	4307
12	4000	400	0.4924	2904	0.5047	1811	0.5359	946	0.5162	5461
13	4250	425	0.5491	3251	0.5701	2200	0.6219	1203	0.5823	6347
14	4500	450	0.4834	3518	0.6241	2509	0.6757	1471	0.5672	7151
15	4750	475	0.5134	3783	0.5272	2733	0.5373	1623	0.5334	7862
16	5000	500	0.4658	4034	0.5074	2936	0.5290	1846	0.4924	8663

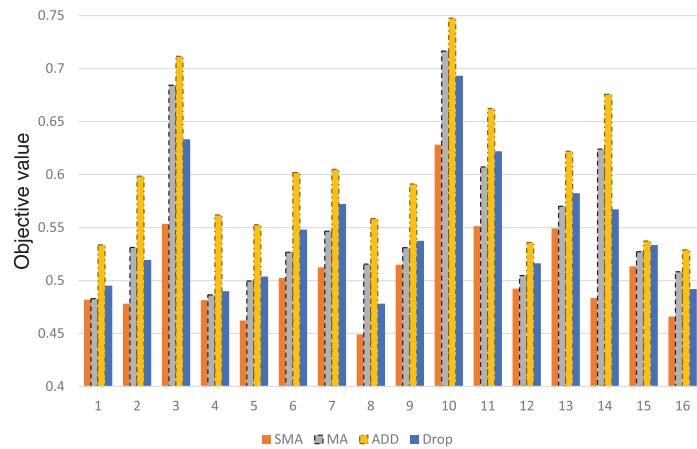


FIGURE 6. Obtained mean objective values.

Source	DF	SS	MS	F	P
Algorithm	3	0.07105	0.02368	6.33	0.001
Error	636	0.22458	0.00374		
Total	639	0.29563			
Individual 95% CIs For Mean Based on Pooled StDev					
Level	N	Mean	StDev		
ADD	160	0.60151	0.06697	(-----*-----)	
DROP	160	0.54896	0.06011	(-----*-----)	
MA	160	0.55384	0.06952	(-----*-----)	
SMA	160	0.50749	0.04517	(-----*-----)	
				-----+-----+-----+-----	
				0.480 0.520 0.560 0.600	

FIGURE 7. Required CPU times.

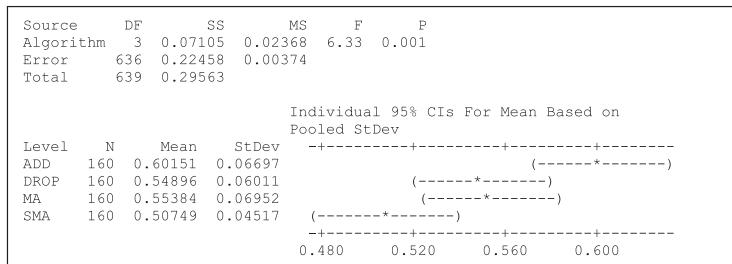


FIGURE 8. ANOVA and related interval plots.

Since the edge-based location problems could present a better image of the real-life problems, the necessity of enriching the literature by scrutinizing this specific type of location problem is more highlighted. Thus, this study presents a novel multi-objective mixed-integer non-linear programming model for a congested capacitated facility location problem with uniformly distributed demands along the network edges. The demand occurrence rate on each network edge follows a Poisson distribution, while the service time of each server follows an exponential distribution. It is assumed that the number of servers at each location is limited by a predetermined threshold. Therefore, just a fraction of demand could be satisfied and the remaining is considered as the lost demand. This paper considers an M/M/C queuing system with a maximum waiting time threshold for the satisfied demands. This study aims to determine the number of facilities, their location, and the number of servers assigned to each facility such that the traveling distance, the waiting time, the total cost, and the number of lost sales are minimized simultaneously.

Due to the NP-hardness of the proposed mathematical model, a stochastic memetic algorithm, a memetic algorithm, and modified add and drop algorithms are developed as the solution methods. After tuning the parameters, algorithms are implemented to solve four randomly generated large-scale test cases and the obtained results are compared using the ANOVA test. According to the obtained results, SMA outperforms the developed MA, Add, and Drop algorithms.

Future studies may check the possibility of obtaining LP-relaxed optimal solutions as a bound to compare the solution algorithms. They also may use different queuing systems by considering different distributions for the demand generation and service time, or by restricting the length of the queue to account for the capacity of servers based on the number of customers arrive at each center at the same time. Furthermore, using fuzzy inputs such as the cost of establishing the facilities or implementing the servers and the rate of demand or service time could be of great interest. Also, one may use other metaheuristic or heuristic solution algorithms to obtain better results.

REFERENCES

- [1] A.B. Arabani and R.Z. Farahani, Facility location dynamics: An overview of classifications and applications. *Comput. Ind. Eng.* **62** (2012) 408-420.
- [2] A. Klose and A. Drexl, Facility location models for distribution system design. *Eur. J. Oper. Res.* **162** (2005) 4-29.
- [3] Y. Liu, Y. Yuan, J. Shen and W. Gao, Emergency response facility location in transportation networks: a literature review. *J. Traffic Transp. Eng.* (2021).
- [4] M. Bieniek, A note on the facility location problem with stochastic demands. *Omega* **55** (2015) 53-60.
- [5] L.V. Snyder, Facility location under uncertainty: a review. *IIE Trans.* **38** (2006) 547-564.
- [6] X. Gao and C. Cui, A note on the warehouse location problem with data contamination. *RAIRO: OR* **55** (2021) 1113.
- [7] A.C. Li, L. Nozick, N. Xu and R. Davidson, Shelter location and transportation planning under hurricane conditions. *Transp. Res. Part E Logist. Transp. Rev.* **48** (2012) 715-729.
- [8] S. An, N. Cui, Y. Bai, W. Xie, M. Chen and Y. Ouyang, Reliable emergency service facility location under facility disruption, en-route congestion and in-facility queuing. *Transp. Res. Part E Logist. Transp. Rev.* **82** (2015) 199-216.
- [9] R. Halper, S. Raghavan and M. Sahin, Local search heuristics for the mobile facility location problem. *Comput. Oper. Res.* **62** (2015) 210-223.

- [10] S. Ilkhanizadeh, M. Golabi, S. Hesami and H. Rjoub, The Potential Use of Drones for Tourism in Crises: A Facility Location Analysis Perspective. *J. Risk Financ. Manag.* **13** (2020) 246.
- [11] S.M. Shavarani, M.G. Nejad, F. Rismanchian and G. Izbirak, Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco. *J. Adv. Manuf. Technol.* **95** (2018) 3141-3153.
- [12] F. Torrent-Fontbona, V. Munoz and B. López, Solving large immobile locationallocation by affinity propagation and simulated annealing. Application to select which sporting event to watch. *Expert Syst. Appl.* **40** (2013) 4593-4599.
- [13] S. Zamani, J. Arkat, S.T.A. Niaki and F. Ahmadizar, Locations of congested facilities with interruptible immobile servers. *Comput. Ind. Eng.* **156** (2021) 107220.
- [14] S. Basu, M. Sharma and P. S. Ghosh, Metaheuristic applications on discrete facility location problems: a survey. *Opsearch* **52** (2015) 530-561.
- [15] Z. Ulukan and E. Demircioğlu, A survey of discrete facility location problems. *Int. J. Ind. Manuf. Eng.* **9** (2015) 2487-2492.
- [16] D. Dinler, M.K. Tural and C. Iyigun, Heuristics for a continuous multifacility location problem with demand regions. *Comput. Oper. Res.* **62** (2015) 237-256.
- [17] I.M. Al-Mudahka, M.S. Al-Jeraiwi and R. M'Hallah, Lower and upper bounds for the continuous single facility location problem in the presence of a forbidden region and travel barrier. *RAIRO: OR* **55** (2021) 141-165.
- [18] M. Hasanzadeh, B. Alizadeh and F. Baroughi, The cardinality constrained inverse center location problems on tree networks with edge length augmentation. *Theor. Comput. Sci.* **865** (2021) 12-33.
- [19] S. Beheshtifar and A. Alimoahmmadi, A multiobjective optimization approach for location-allocation of clinics. *Int. Trans. Oper. Res.* **22** (2015) 313-328.
- [20] A. Eydi and J. Mohebi, Modeling and solution of maximal covering problem considering gradual coverage with variable radius over multi-periods. *RAIRO: OR* **52** (2018) 1245-1260.
- [21] R.Z. Farahani, N. Asgari, N. Heidari, M. Hosseiniinia and M. Goh, Covering problems in facility location: A review. *Comput. Ind. Eng.* **62** (2012) 368-407.
- [22] K. Aardal, P.L. van den Berg, D. Gijswijt and S. Li, Approximation algorithms for hard capacitated k-facility location problems. *Eur. J. Oper. Res.* **242** (2015) 358-368.
- [23] Y. Yan, *An Improved Ant Colony Optimization Algorithm with Local Search for the Capacitated Facility Location Problem*. Ph.D. thesis, New Mexico State University (2021).
- [24] M. Charikar, S. Khuller, D.M. Mount and G. Narasimhan, Algorithms for facility location problems with outliers. *SODA* **1** (2001) 642-651.
- [25] P. Sinha, Performance of an add-drop-interchange heuristic for the capacitated facility location problem. *Int. J. Appl. Manag. Sci.* **1** (2009) 388-400.
- [26] I. Harris, C.L. Mumford and M.M. Naim, A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transp. Res. Part E Logist. Transp. Rev.* **66** (2014) 1-22.
- [27] K. Abbal and Y. Benadada, Bi-level Multi-capacitated Facility location Problem, in *2020 5th International Conference on Logistics Operations Management (GOL)*. IEEE (2020) 1-5.
- [28] B. Boffey, R. Galvao and L. Espejo, A review of congestion models in the location of facilities with immobile servers. *Eur. J. Oper. Res.* **178** (2007) 643-662.
- [29] R. Aboolian, O. Berman and Z. Drezner, The multiple server center location problem. *Ann. Oper. Res.* **167** (2009) 337-352.
- [30] H.R. Sayarshad and J.Y. Chow, Non-myopic relocation of idle mobilityon- demand vehicles as a dynamic location-allocation-queueing problem. *Transp. Res. Part E Logist. Transp. Rev.* **106** (2017) 60-77.
- [31] R. Tavakkoli-Moghaddam, S. Vazifeh-Noshafagh, A.A. Taleizadeh, V. Hajipour and A. Mahmoudi, Pricing and location decisions in multi-objective facility location problem with M/M/m/k queuing systems. *Eng. Optim.* **49** (2017) 136-160.
- [32] S.M. Shavarani, S. Mosallaeipour, M. Golabi and G. Izbirak, A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system. *Comput. Oper. Res.* **108** (2019) 57-68.
- [33] O. Berman, D. Krass and J. Wang, Locating service facilities to reduce lost demand. *IIE Trans.* **38** (2006) 933-946.
- [34] S.H.A. Rahmati, A. Ahmadi, M. Sharifi and A. Chambari, A multiobjective model for facility locationallocation problem with immobile servers within queuing framework. *Comput. Ind. Eng.* **74** (2014) 1-10.
- [35] F. Etebari, A column generation algorithm for the choice-based congested location-pricing problem. *Comput. Ind. Eng.* **130** (2019) 687-698.
- [36] 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.* **24** (2013) 331-348.
- [37] R. Jafari and J. Arkat, Network location problem with stochastic and uniformly distributed demands. *Int. J. Eng.* **29** (2016) 654-662.
- [38] M. Golabi, S.M. Shavarani and G. Izbirak, An edge-based stochastic facility location problem in UAV-supported humanitarian relief logistics: a case study of Tehran earthquake. *Nat. Hazards* **87** (2017) 1545-1565.
- [39] S.M. Shavarani, M. Golabi and G. Izbirak, A capacitated biobjective location problem with uniformly distributed demands in the UAV-supported delivery operation. *Int. Trans. Oper. Res.* **28** (2021) 3220-3243.
- [40] M. Golabi, G. Izbirak and J. Arkat, Multiple-server facility location problem with stochastic demands along the network edges. *J. Eng. Res.* **6** (2018).
- [41] M.J. Hodgson, A flow-capturing location-allocation model. *Geogr. Anal.* **22** (1990) 270-279.

- [42] M.R. Garey and D.S. Johnson, Computers and Intractability. Freeman San Francisco, 174 (1979).
- [43] M.S. Daskin and K.L. Maass, The p-median problem, in *Location Science*. Springer (2015) 21-45.
- [44] N. Megiddo, E. Zemel and S.L. Hakimi, The maximum coverage location problem. *SIAM J. Alg. Discrete Meth.* **4** (1983) 253-261.
- [45] S. Dantrakul, C. Likasiri and R. Pongvuthithum, Applied p-median and p-center algorithms for facility location problems. *Expert Syst. Appl.* **41** (2014) 3596-3604.
- [46] Z.-J.M. Shen, R.L. Zhan and J. Zhang, The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS J. Comput.* **23** (2011) 470-482.
- [47] P. Sinha, Observations on some heuristic methods for the capacitated facility location problem. *Opsearch* **49** (2012) 86-93.
- [48] J. Wollenweber, A multi-stage facility location problem with staircase costs and splitting of commodities: model, heuristic approach and application. *OR Spectr.* **30** (2008) 655-673.
- [49] R. Kissell and J. Poserina, Chapter 4 - Advanced math and statistics, in *Optimal Sports Math, Statistics, and Fantasy*. Academic Press (2017) 103-135.
- [50] J.F. Shortle, J.M. Thompson, D. Gross and C.M. Harris, Fundamentals of Queueing Theory. John Wiley & Sons, 399 (2018).
- [51] M. Ehrgott and X. Gandibleux, Multiobjective combinatorial optimization theory, methodology, and applications, in *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Springer (2003) 369-444.
- [52] N. Krasnogor and J. Smith, A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans. Evol. Comput.* **9** (2005) 474-488.
- [53] R. Tavakkoli-Moghaddam, N. Safaei and F. Sassani, A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Comput. Oper. Res.* **36** (2009) 402-414.
- [54] P. Moscato and M.G. Norman, A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. *Parallel Comput. Transp. Appl.* **1** (1992) 177-186.
- [55] S. Samanta, A. Choudhury, N. Dey, A. Ashour and V. Balas, Quantuminspired evolutionary algorithm for scaling factor optimization during manifold medical information embedding, in *Quantum Inspired Computational Intelligence*. Elsevier (2017) 285-326.
- [56] A.A. Kuehn and M.J. Hamburger, A heuristic program for locating warehouses. *Manag. Sci.* **9** (1963) 643-666.
- [57] E. Feldman, F. Lehrer and T. Ray, Warehouse location under continuous economies of scale. *Manag. Sci.* **12** (1966) 670-684.

Subscribe to Open (S2O)

A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/math-s2o-programme>