

Designing a hub covering location problem under uncertainty conditions

Mir Reza Zahedi^a and Javid Ghahremani Nahr^{a*}

^aFaculty Member of Faculty member of Development and Planning Institute of ACECR, Tabriz, Iran

CHRONICLE

Article history:

Received November 28, 2019

Received in revised format:

December 28, 2019

Accepted December 23, 2019

Available online

February 17, 2020

Keywords:

Hub Covering Location Problem

Network Design

MOALO Algorithm

Robust-Box Optimization Method

ABSTRACT

Increased demand for transportation is inevitable along with the growth of social welfare and economic development. Furthermore, providing cheap transportation services has brought about a bulk of benefits in the ever-expanding development of countries. Therefore, the optimization of fuel consumption in the transportation sector is essential for the continued low cost of transportation and reduced amount of pollution caused by the traffic. A major strategy for optimizing fuel consumption and transportation costs in the transportation of goods and passengers is the proper design of transportation networks. Accordingly, in this study, hub covering location problem is modeled taking into account the queuing system in order to minimize the cost of the whole network as well as to minimize emissions of greenhouse gases. The model presented in this study includes uncertain parameters of demand and transportation costs. To control the parameters, a robust-box optimization method is used. As the hub covering network model is multi-objective and NP-hard, three meta-heuristic algorithms; namely MOPSO, NSGA II and MOALO are presented. Computational results show the high efficiency of the MOALO algorithm for obtaining efficient problem-solving in large sizes.

© 2020 by the authors; licensee Growing Science, Canada.

1. Introduction

The important and prevalent application of hub location problem is in the field of transportation and telecommunications. In the last few decades, there have been a bulk of studies in these areas. In the field of transportation, such as the public transportation network, air transport, postal networks, etc., demand is a physical flow of goods or humans from the source to the destination. Also, different vehicles such as airplanes, trains, trucks, ships, etc. are considered at different levels of the hub network. In public transportation networks, regional airports, central railway stations or bus terminals are referred to as hubs. People are transported to the hub from different geo-diagramical areas and the displacement in the hub happens among centers. Hub connection lines include large planes capable of long distance flights, special trains or special buses with specific performances (Hassanzadeh et al., 2018). In the cargo network, loading terminals are considered to be hubs, in which small trucks deliver their load to larger vehicles for transportation in the hub line connections. According to what was mentioned above, there are various types of hub location problems one of which is hub covering location problem. In this category of hub covering location problems, customers usually receive services and goods based on distance to facility. Most studies in hub covering problems consider the coverage radius as zero and one, according to which a demand center is covered when covered within the coverage radius and if it is not covered outside. Such a hypothesis is not applicable in the real world. Because a) the quality of the coverage is not necessarily constant and the closer it is to the facility, the

* Corresponding author.

E-mail address: javid.ghahremani@yahoo.com (J. Ghahremani Nahr)

higher it will be; and b) there is no clear boundary around the facility in the real world around which it can reach zero by passing through that level of service. On the other hand, hub covering location problems focus on serving the full or maximum service to demand points. Maximum covering problem locates a constrained number of facilities in order to maximize the covered demand points. The probable type of maximum covering location problems is the maximum access covering location problems, which seek to maximize the population covered by an ease. In hierarchical systems, services provided by facilities, such as the facilities themselves, have α levels of varying degrees, hence, the hub covering models of these systems have a particular complexity (Merakli & Kara, 2015). The most important application of the hub covering problem is in the transportation of goods that are to be delivered at a predetermined time to the client, such as the use of it in postal companies or airlines. Another point is that the following two issues are important in transportation networks (Peker & Kara, 2015):

- A. Reliability of the network, which refers to the ability to serve in critical situations, when part of the network is defective, and
- B. The response time, which is meant to be the time needed to transport the shipment through the network in order to meet customer demand.

Another consideration in hub covering location problem is the attention to the reduction in the demand queue length in the hubs. Therefore, increasing demand queuing in the hubs is an increase in vehicle traffic, which will simultaneously increase costs and greenhouse gas emissions. Also, uncertainty about the demand for products among nodes along with the queue system is another issue that should be considered in the modeling of the hub network. In order to control the uncertainty of demand and transportation costs, a robust-box planning method will be used. A robust-box optimization method is a special type of optimization methods for controlling non-deterministic parameters that, by changing the uncertainty rate, ensures the robustness of the model based on cost increases. Therefore, if parameters such as demand increase suddenly, the model will be able to meet that demand by imposing additional costs. The main idea of this paper is to model a hub covering location problem o considering the queuing system in an uncertainty mode. The main variables include strategic decision-making variables (determining the number and optimal location of hubs) and tactical decision variables (optimal allocation of nodes to hubs) with the goal of simultaneously reducing network costs and greenhouse gas emissions. Since uncertainty has been considered in cost and demand parameters, a robust-box optimization method has been used to control these parameters. In this problem, the reduction of demand queuing distance between nodes has also been taken into account.

2. Literature review

The idea of the Hub Network was raised in 1969 by Goldman (1969). Then, O'Kelly (1986) conducted the first study of the hub network in the context of the air network. Also, O'Kelly (1987) presented the first second order mathematical model for hub networks. Since then, numerous scholars have worked on various areas of hub location problems. This focus is largely due to the need to create a new infrastructure for transportation and communication systems because, because of the relevance of these topics and the lapse of time and design of complex systems, they do not follow traditional network models. In traditional service networks, each location receives demand directly from the source of demand. However, in modern networks, a number of points are selected in the network and constitute an intermediate structure. Therefore, the flow of demand from the origin to the destination is transmitted through these midpoints, which are called hub in the network (Campbell et al., 2002). For the first time O'Kelly, in 1992 used the fixed cost as the cost of creating a hub in the objective function model. In this model, the number of hubs is considered as the decision variable instead of being assumed to be fixed beforehand (O'Kelly, 1992). Campbell presented a model in 1994 in which for each network arc, a minimum flow was considered as the threshold of flow. Also, the fixed cost of creating demand moods is also considered in the median problem model with p number of hubs (Campbell, 1994). In 2001, Nickel et al. (2001) presented a mathematical model for the hub location problems, in which the fixed

cost, in addition to demand centers, was also considered for the connecting arcs of demand nodes and the connecting arcs of demand centers. Hub location plays a fundamental role in the design of the network of demand centers and the hub because the total shipping cost affects the capacity of the middle centers and hence the service time and the amount of congestion in the system. There are a number of general reviews about hub covering location problems, the most recent of which is the study by Alumur et al. (2008). They have compiled all of the models for finding hubs under the network until 2007. Farahani et al. (2013) collected papers and research on hub location problem in a variety of discrete, continuous network issues, and so on since 2007. To select the locations and dimensions of urban transit hubs, Yu et al. (2009), examined a hierarchical location model based on clustering aimed at minimizing total travel time with regard to demand. Calık et al. (2009) worked on incomplete single-assignment hub covering problems hubs and provided an integer programming formulation for it. Their problem was the location of the hubs and how the links were created between the hubs and the allocation of non-hub nodes to the locations of the hub, so that the travel time between each source-destination within a time boundary was defined. Gavrilouk (2009) in an article examines single-assignment central hub problem of unmatched hubs, and introduces a method for solving large-scale problems. He also used a heuristic method with a solution to solve his model. Huang et al. (2009) introduced a robust hub location in 2009. In his model, he reduced the total shipping costs with the uncertainty of demand. Another model presented in this paper was a model that Makui et al. (2011) presented as "a robust optimization model for hub location problem of with capacity". The objective function of their model was composed of three parts. The first part is to minimize the cost of establishing a hub and the cost of shipping goods. The second part minimizes the maximum distances, and the third part minimizes all the time needed for processing the goods. In their model, demand was considered as an uncertain parameter, and due to the influence of some of its parameters, these parameters were also uncertain, and under different scenarios and with regard to the capacity Constraints of the model, they sought to achieve the objectives. Rodríguez-Martín et al. (2014) considered a hub location-routing problem taking into account the decisions of hubs and allocations. They used AP and CAB data to solve their problem and used a branch-and-cut algorithm to solve their problem. Zhai et al. (2016) examined a two-level model of hub location in uncertainty conditions where demand was in an uncertain condition and was analyzed using a fuzzy approach. Gao and Qin (2016) focused on hub locations and the allocation of nodes to hubs in uncertain times, and aimed at reducing the maximum travel time so that justice was also respected. Their hub location model included a central p-hub location model. Silva et al. (2017) presented a heuristic method based on taboo search for solving the P-hub problem. They performed their problem with AP data sets with 200 nodes and showed that the method they provided had high efficiency in finding the answer. Ghaffarinasab Nasab et al., (2018) modeled the multiple-assignment and single-assignment hubs and solved these two models using simulated annealing algorithm. By demonstrating this model on several data and solving with their proposed algorithm, the simulated annealing algorithm is optimized at a shorter time than other algorithms. Seifbarghy et al. (2018) studied the modeling of hub covering location problem considering queuing systems and capacity constraints. The whole network goal was to minimize network costs. They used genetic algorithms and particle swarm optimization to solve their model and showed that the genetic algorithm had a high efficiency to solve this problem compared to other algorithm. In the literature, we come up with articles that model the hub covering location problems in certainty, but so far, hub covering location problems considering the queuing system in uncertainty mode (using a robust-box optimization method for controlling non-deterministic parameters) and problem solving has not been studied using ant lion multi-objective algorithm.

3. Statement of the problem and modeling

This section addresses the definition of the proposed problem. Corresponding to Fig. 1, several nodes are considered for the problem. The main purpose of selecting some of these nodes are as hub locations, which are responsible for receiving and sending the demand flow between each pair of nodes. The problem is that each node is only allowed to be assigned to one hub, and full connection among all the hubs is set. Each hub has a cover radius that only the nodes in this range can connect to the hub. A

queuing system is also defined for each selected hub whose task is to streamline the queue stream and reduce waiting times in the hub queues for service. Therefore, waiting time in the queue is considered as a cost in the cost objective function. Hence, the assumptions of hub covering location problem considering the queuing system in the uncertainty mode are summarized as follows:

- A. The proposed model is a nonlinear mathematical programming model with the aim of reducing network costs and reducing greenhouse gas emissions.
- B. Hubs have capacity constraints.
- C. The demand for all nodes must be met and no shortage is not allowed;
- D. Demand and transportation parameters are considered uncertain,
- E. To control uncertain parameters robust-box optimization method will be used, and
- F. The rate of service in the hubs is defined and clear.

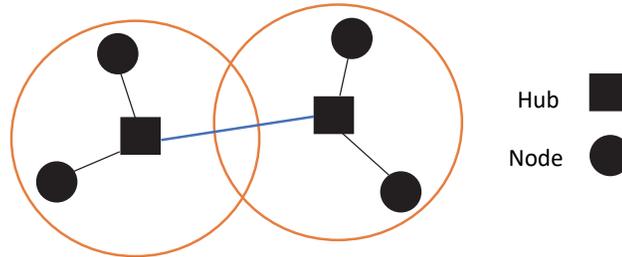


Fig. 1. The proposed hub covering location model

For problem modeling, set, parameter, and decision variables are defined as follows

Sets

I Network node set: $i, j, k, m \in \{1, \dots, I\}$

Parameters

- F_k The fixed cost of choosing and establishing a hub k
- Dis_{ij} The distance between two pairs of nodes i and j
- \widetilde{TC}_{ij} Shipping cost per unit distance between nodes i and j
- C_{ij} he amount of greenhouse gas emissions caused by a unit of distance between the nodes i and j
- \widetilde{FL}_i Flow rate (demand) node i
- Cap_K Serving capacity of hub k
- Γ_k Maximum covering radius of hob k
- CT_k The cost of waiting time for service in hub k
- μ_k Service rate of hub k (exponential distribution)
- B_k The upper limit of the queue length to serve in hub k
- θ_k The probability of the upper limit for queuing lengths in hub k
- ϑ_k Number of servers in hub k

Given the nature of the hub location problem, if a product is shipped or services rendered between two hubs, a discount is applied to costs, so parameter χ is defined as a discount parameter to increase the transfer of goods-services between hubs. Therefore, the total cost of moving between two corresponding nodes in hub covering location problem can be defined as follows:

$$\widetilde{TR}_{ikmj} = TC_{ik} + \chi TC_{km} + TC_{mj} \quad (1)$$

Decision Variables

X_{ikmj}	$\begin{cases} \text{If the path between the node } i \text{ and the node } j \text{ is taken from the } k \text{ and } m \text{ hubs} & 1 \\ \text{Otherwise} & 0 \end{cases}$
Y_{ik}	$\begin{cases} \text{If the node } i \text{ allocated to hub } k & 1 \\ \text{Otherwise} & 0 \end{cases}$
λ_k	Customer login rate to hub
π_{0k}	The probability of occurrence of the construction of the hub
W_k	Customer waiting time in the hub

3.1 Modeling hub covering location problem in an uncertain state

Given the sets, parameters, and decision variables, the problem of designing a hub covering location problem with a queuing system in terms of uncertainty as a mixed integer nonlinear mathematical programming model is as follows:

$$\min Z_1 = \sum_{k=1}^K F_k Y_{kk} + \sum_{i=1}^I \sum_{k=1}^I \sum_{m=1}^I \sum_{j=1}^I \widetilde{TR}_{ikmj} X_{ikmj} \quad (2)$$

$$\min Z_2 = \sum_{i=1}^I \sum_{k=1}^I \sum_{m=1}^I \sum_{j=1}^I \left(\frac{C_{ik}}{Dis_{ik}} + \chi \frac{C_{km}}{Dis_{km}} + \frac{C_{mj}}{Dis_{mj}} \right) X_{ikmj} \quad (3)$$

s. t.

$$\sum_{k=1}^I \sum_{m=1}^I X_{ikmj} = 1, \quad \forall i, j, i \neq j \quad (4)$$

$$\sum_{m=1}^I X_{ikmj} + \sum_{\substack{m=1 \\ m \neq k}}^I X_{imkj} \leq Y_{kk}, \quad \forall i, j, k \quad (5)$$

$$\sum_{i=1}^I \widetilde{FL}_i Y_{ik} = Cap_K Y_{kk}, \quad \forall k \quad (6)$$

$$X_{ikmj} \leq Y_{ik}, \quad \forall i, j, k, m \quad (7)$$

$$X_{ikmj} \leq Y_{jm}, \quad \forall i, j, k, m \quad (8)$$

$$\sum_{k=1}^I Y_{ik} = 1, \quad \forall i \quad (9)$$

$$Dis_{ik} Y_{ik} \leq \Gamma_k Y_{kk}, \quad \forall i, k \quad (10)$$

$$P\{k \text{ Queue of Hub} > B_k\} \leq \theta_k, \quad \forall k \quad (11)$$

$$X_{ikmj}, Y_{ik} \in \{0,1\}, \quad \forall i, j, k, m \quad (12)$$

Eq. (2) represents the first objective function of the model and includes minimizing the total network costs (hub construction costs and shipping costs). Eq. (3) minimizes the total amount of greenhouse gas emissions per unit of distance. Constraint (4) ensures that the path of the beginning and the end of each pair of nodes is selected from the selected hubs. Constraint (5) shows that if the node is selected as a hub, then the other hubs will also be connected to it. Constraint (6) ensures that the amount of transfer flow to the selected hub does not exceed its service capacity. Constraints (7) and (8) show that if nodes are selected as hubs, other nodes are only allowed to connect to those hubs. Constraint (9) ensures that each node only has to be connected to one hub. Constraint (10) shows the maximum coverage constraint of the cover and ensures that only nodes are assigned to the hub that is in its predefined radius. Constraint (11) shows the queue length of the hubs. Constraint (12) shows the type and material of the decision variables.

3.2. Jackson Network

The Jackson Network under study consists of the selected hubs shown in Fig. 2. The suggested network equations are as follows.

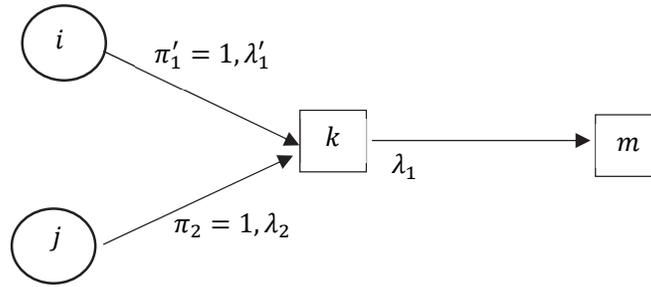


Fig. 2. Jackson network under study in the proposed model

$$\pi_1' \lambda_1' + \pi_2 \lambda_2 = \lambda_1 \tag{13}$$

Therefore, according to the values given in the figure, the input rate to the hub is as follows:

$$\lambda_1' + \lambda_2 = \lambda_1 \tag{14}$$

As a result, the input rate corresponding to the hub covering location model will be described in the following equation:

$$\lambda_k = \sum_{i=1}^I \widetilde{F}L_i Y_{ik}, \quad \forall k \tag{15}$$

According to constraint (11), it is necessary to transform the model into a certain state, so a queuing model $M/M/m/C$ is used to transform this constraint into a certain one. In this queuing model $M/M/m/C$ for the hub, the input rate is defined by λ (Poisson distribution) and service time by the rate μ (exponential distribution). The queuing model includes m servitors and limited capacity C . The hub is assumed to have ϑ_k servitors, as a result, constraint (11) is converted as follows (Marianov & Serra, 2003):

$$\sum_{k'=\vartheta_k+B_k+1}^C Pr_{k'k} \leq \theta_k \text{ or } 1 - \sum_{k'=0}^{\vartheta_k+B_k} Pr_{k'k} \leq \theta_k, \quad \forall k \tag{16}$$

The first term of constraint (16) shows the probability that more than n customers is less than the serving hub k with ϑ_k service provider, and the second expression indicates that the sum of all probabilities is equal to 1. Therefore, the service rate is obtained as follows:

$$\mu_{nk} = \begin{cases} n\mu_k & n \leq \vartheta_k \\ \vartheta_k \mu_k & \vartheta_k < n < C \end{cases}, \quad \forall k \tag{17}$$

By combining the above constraints, π_{0k} (the probability of node selection as a hub) is obtained as follows.

$$\pi_{0k} = \left[\sum_{k'=0}^{\vartheta_k-1} \frac{1}{k'!} \left(\frac{\lambda_k}{\mu_k}\right)^{k'} + \frac{1}{\vartheta_k!} \left(\frac{\lambda_k}{\mu_k}\right)^{\vartheta_k} \left(\frac{\vartheta_k \mu_k}{\vartheta_k \mu_k - \lambda_k}\right) \right]^{-1} Y_{kk}, \quad \forall k \tag{18}$$

In the probability results, a robust state for the existence of n customers in the queue for hub service (constraint 11) is obtained as follows:

$$Pr_{nk} = \begin{cases} \frac{\lambda_k^n}{n! \mu_k^n} \pi_{0k} & 1 \leq n \leq \vartheta_k \\ \frac{\lambda_k^n \vartheta_k^{n-\vartheta_k}}{n! \mu_k^n} \pi_{0k} & \vartheta_k < n \leq C \end{cases} \quad \forall k \tag{19}$$

$$Pr_{k'k} = \sum_{k'=0}^{\vartheta_k} \frac{\lambda_k^{k'}}{k'! \mu_k^{k'}} \pi_{0k} Y_{kk} + \sum_{k'=\vartheta_k+1}^{\vartheta_k+B_k} \frac{\lambda_k^{k'} \vartheta_k^{k'-\vartheta_k}}{k'! \mu_k^{k'}} \pi_{0k} Y_{kk} \geq (1 - \theta_k) Y_{kk}, \quad \forall k \tag{20}$$

$$Pr_{k'k} = \left(\sum_{k'=0}^{\vartheta_k} \frac{\lambda_k^{k'}}{k'! \mu_k^{k'}} + \sum_{k'=\vartheta_k+1}^{\vartheta_k+B_k} \frac{\lambda_k^{k'} \vartheta_k^{k'-\vartheta_k}}{k'! \mu_k^{k'}} \right) \pi_{0k} Y_{kk} \geq (1 - \theta_k) Y_{kk}, \quad \forall k \tag{21}$$

Also, with respect to the above, the customer's waiting time in the hub k is also given as equation (21):

$$W_k = \left[\frac{\pi_{0k}}{\vartheta_k!} \left(\frac{\lambda_k}{\mu_k} \right)^{\vartheta_k!} \frac{\vartheta_k \mu_k}{(\vartheta_k \mu_k - \lambda_k)^2} + \frac{1}{\mu_k} \right] Y_{kk}, \quad \forall k \tag{22}$$

3.3 A robust-box optimization model

The proposed model presented in the previous section is related to the model in an uncertain state. This model has been developed based on the robust optimization model provided by Ben Tal and Nemirovski (2009). Ben Tal et al. showed that in a limited framework, a robust model can be transformed into an equilibrium problem from a semi-immutable problem where the set u_{box} is replaced by the boundary set u_{ext} . In this problem, u_{ext} includes the maximum values in the set u_{box} . Regarding the index, the parameter and decision variables of the proposed non-linear math programming model are the integer as follows:

$$\min Z_1 = \sum_{k=1}^K F_k Y_{kk} + \sum_{i=1}^I \sum_{k=1}^I \sum_{m=1}^I \sum_{j=1}^I (TR_{ikmj} X_{ikmj} + \eta_{ikmj}^{TR}) + \sum_{k=1}^K CT_k W_k \tag{23}$$

$$\min Z_2 = \sum_{i=1}^I \sum_{k=1}^I \sum_{m=1}^I \sum_{j=1}^I \left(\frac{C_{ik}}{Dis_{ik}} + \chi \frac{C_{km}}{Dis_{km}} + \frac{C_{mj}}{Dis_{mj}} \right) X_{ikmj} \tag{24}$$

s. t.

$$\rho_{TR} TR_{ikmj} X_{ikmj} \leq \eta_{ikmj}^{TR}, \quad \forall i, j, k, m \tag{25}$$

$$\rho_{TR} TR_{ikmj} X_{ikmj} \geq -\eta_{ikmj}^{TR}, \quad \forall i, j, k, m \tag{26}$$

$$\sum_{k=1}^I \sum_{m=1}^I X_{ikmj} = 1, \quad \forall i, j, i \neq j \tag{27}$$

$$\sum_{m=1}^I X_{ikmj} + \sum_{\substack{m=1 \\ m \neq k}}^I X_{imkj} \leq Y_{kk}, \quad \forall i, j, k \tag{28}$$

$$\sum_{i=1}^I \overline{FL}_i (1 + \rho_{FL}) Y_{ik} = Cap_K Y_{kk}, \quad \forall k \tag{29}$$

$$X_{ikmj} \leq Y_{ik}, \quad \forall i, j, k, m \tag{30}$$

$$X_{ikmj} \leq Y_{jm}, \quad \forall i, j, k, m \tag{31}$$

$$\sum_{k=1}^I Y_{ik} = 1, \quad \forall i \tag{32}$$

$$Dis_{ik} Y_{ik} \leq \Gamma_k Y_{kk}, \quad \forall i, k \tag{33}$$

$$\lambda_k = \sum_{i=1}^I \overline{FL}_i (1 + \rho_{FL}) Y_{ik}, \quad \forall k \quad (34)$$

$$W_k = \left[\frac{\pi_{0k}}{\vartheta_k!} \left(\frac{\lambda_k}{\mu_k} \right)^{\vartheta_k} \frac{\vartheta_k \mu_k}{(\vartheta_k \mu_k - \lambda_k)^2} + \frac{1}{\mu_k} \right] Y_{kk}, \quad \forall k \quad (35)$$

$$\pi_{0k} = \left[\sum_{k'=0}^{\vartheta_k-1} \frac{1}{k'!} \left(\frac{\lambda_k}{\mu_k} \right)^{k'} + \frac{1}{\vartheta_k!} \left(\frac{\lambda_k}{\mu_k} \right)^{\vartheta_k} \left(\frac{\vartheta_k \mu_k}{\vartheta_k \mu_k - \lambda_k} \right) \right]^{-1} Y_{kk}, \quad \forall k \quad (36)$$

$$\left(\sum_{k'=0}^{\vartheta_k} \frac{\lambda_k^{k'}}{k'! \mu_k^{k'}} + \sum_{k'=\vartheta_k+1}^{\vartheta_k+B_k} \frac{\lambda_k^{k'} \vartheta_k^{k'-\vartheta_k}}{k'! \mu_k^{k'}} \right) \pi_{0k} Y_{kk} \geq (1 - \theta_k) Y_{kk}, \quad \forall k \quad (37)$$

$$\lambda_k, W_k, \pi_{0k}, \eta_{ikmj}^{TR} \geq 0, \quad \forall i, j, k, m \quad \text{and } X_{ikmj}, Y_{ik} \in \{0,1\}, \quad \forall i, j, k, m \quad (38)$$

4. Solution

In this paper, the MOALO meta-heuristic algorithm is used to solve the two-objective model and is compared with two meta-heuristic algorithms NSGA II and MOPSO. Therefore, this section describes the MOALO algorithm and the initial solution used in solving the model.

4.1 Process of the MOALO algorithm

The principles of this algorithm must first be addressed (Mirjalili et al., 2017) for the purpose of designing a multi-objective version of ant-lion algorithm. An algorithm must use the same search behavior in the extended version of the same algorithm. The ant-lion algorithm emulates the ant lion hunting mechanism and interacts with the bait, the desired ant and imitate with all of them. As with other population-based algorithms, the ant-lion algorithm approximates optimal solutions for optimization problems by applying a set of random solutions. This set is promoted based on sciences inspired by the interaction between the ants and the ant lion. There are two populations in the ant-lion algorithm: ants set and ant-lions set. The general stages of the ant-lion algorithm for changing these sets and, finally, the global optimization estimation for the optimization problem are as follows:

- A. The ant set is initiated with random values and are the main factors in the search for the ant-lion algorithm.
- B. The value of the fit of each ant is evaluated using the objective function in each replication,
- C. The ants in the search space are moving randomly around the ant lions
- D. The ant lion population is never evaluated. In fact, the ant lions is supposed to be positioned in the first reps and will move to the new position of the ants in the remaining repeats, of course, if the ants get better,
- E. One ant lion is assigned to any ant and its position is manifested as the ant becomes more fit.
- F. There is an ant lion that, regardless of its distance, affects the movement of the ants.
- G. If any ant lion is better than the selected ant, it will be replaced by the selected ant.
- H. The steps (B) to (E) are repeated to the extent that the final criterion is observed as satisfactory, and
- I. The position and value of the fitting of the selected ant lion is returned as the best estimate for global optimization.

The main responsibility of ants is to examine the search space. They should move around the search space using a random step. The ant lions retains the best of the ants and locates the ants to the promising areas of search space. In order to solve the optimization problems, the ant lion algorithm emulates the

random step of the ants, immersion in the ant lion cavity, making the cavity, moving the ant to ant lions, catching the bait and repairing the cavity, and selecting the preferred one. The mathematical model and programming modules provided for each of these steps are presented in the next paradiagrams. The original randomized step used in the ant lion algorithm to simulate the random step of the ants is as follows:

$$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \dots, \text{cumsum}(2r(t_n) - 1)] \quad (39)$$

In which the cumsum is the cumulative sum, n calculates the maximum repeat count and t shows the step of a random step

$$r(t) = \begin{cases} 1 & \text{if } rand > 0.5 \\ 0 & \text{if } rand \leq 0.5 \end{cases} \quad (40)$$

t is (replicate in this study) and rand is random function in which t is a random step (repetition in this study) and rand is random number of births with a uniform distribution in the interval [0,1]. Random steps should be normalized to the following equation in order to maintain a random step on the boundaries of the search space and avoid over-hunting.

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \quad (41)$$

where c_i^t is the least significant variable in repeat t, d_i^t is the maximum variable i in repeat t, a_i is the minimum random variable i variable, and b_i is the maximum random step in the variable i. Ant-lion algorithm simulates the ants' clogging in the ant lion hole by changing the random steps around the ant lions. The following equations are presented in this regard:

$$c_i^t = Antlion_j^t + c^t \quad (42)$$

$$d_i^t = Antlion_j^t + d^t \quad (43)$$

where c^t is the minimum variables in repetition t, d^t is vector containing the maximum of all variables in repetition t, c_i^t is the minimum variables for variable i, d_i^t is maximum of all the variables for ant i, $Antlion_j^t$ is the position of the ant lion chosen in repetition t. In nature, larger ant lions represent larger cavities to increase the chance of survival. In order to simulate this, the ant-lion algorithm uses a roulette wheel operator that selects the ant lions based on their fitting value. The roulette wheel helps to attract better ants to fit better-fitted lion ants. To mimic ants from falling to ant lions, the boundaries of random steps should be reduced as follows:

$$c^t = \frac{c^t}{I} \quad (44)$$

$$d^t = \frac{d^t}{I} \quad (45)$$

Where I is a ratio, c^t is minimum of all variables in repetition t, and d^t shows the vector representing the maximum of all the variables in repetition t. In the above equations $I = 1 + 10^w \frac{t}{T}$, in which t is the current repetition, T is the maximum number of repetitions and w is defined based on the current replication (w=2 when $t > 0.1T$, w=3 when $t > 0.5T$, w=4 when $t > 0.75T$, w = 5 when $t > 0.9T$ and w = 6 when $t > 0.95T$). Parameter W in the equation I can be used to adjust the level of operation accuracy. The second stage up to the last stage in the ant-lion algorithm is to catch the ant and rebuild the pit. The following equation simulates this process:

$$Antlion_j^t = Ant_i^t \quad \text{if } f(Ant_i^t) < f(Antlion_j^t), \quad (46)$$

where t shows the current repetition, $Antlion_j^t$ indicates the position of the selected ant lion i in repetition t, Ant_i^t indicates position i in repetition t. The last operator in the ant-lion algorithm is the

elitism in which the best-fitted ant lion formed during optimization is stored. This is the only ant lion that can affect all ants. This means moving the selected ant lion (randomly selected using a Rolette wheel) to the ant lion. The equation follows both of them:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (47)$$

where Ant_i^t shows the position of ant i on repetition t . R_A^t is random motion in the selected ant lion by the roulette wheel in repetition t and R_E^t is the random motion around the elite in repetition t . As noted in the literature review, there are different methods for finding and storing Pareto optimal solutions using heuristic algorithms. In this work, we will use an archive to store the best Pareto solutions. Obviously, the convergence of the multi-objective ant-lion algorithm inherits from the ant-lion algorithm. If we choose a solution from the archive, the ant-lion algorithm will be able to improve its quality. However, it is challenging to find optimal Pareto solutions with high diversity. To overcome this challenge, we have inspired a multi-objective ant-lion algorithm and we use leadership selection and archive maintenance. Obviously, there should be a limit to the archive and solution to be selected from the archive. To measure the distribution of solutions in the archive, we use niching. In this approach, the proximity of each solution is examined according to the predefined radius. The number of solutions in the proximity is then counted and considered as distribution measure to improve the distribution of solutions in the archive, we considered two similar mechanisms, similar to the multi-objective particle swarm algorithm. At first, the ants are selected from a solution to a small community. The following equation is used to define the probability of choosing a solution in the archive:

$$P_i = \frac{c}{N_i}, \quad (48)$$

where c is constant and must be greater than 1, and N_i is the number of solutions adjacent to solution i . Secondly, when the archive is complete, the solution to the most populated neighborhood is removed from the archive. The following equation is used in this context that defines the possibility of deleting a solution from the archive:

$$P_i = \frac{N_i}{c}, \quad (49)$$

where t is constant and should be greater than 1, and N_i is the number of solutions is in the vicinity of solution i . Ant-lion Algorithm for solving multi-objective problems, Eqs. (2-8) should be corrected due to the multi-objective nature of the problem and its specific problems.

$$Antlion_j^t = Ant_i^t \quad \text{if } f(Ant_i^t) < f(Antlion_j^t) \quad (50)$$

where t indicates the current repeat, $Antlion_j^t$ indicates the position of the selected j^{th} ant lion in t^{th} repetition, and Ant_i^t is the position of the ant lion in t^{th} repetition. Another correction to select the random ant lion and elite selection is in Eq. (49), which uses a roulette wheel to select a non-defeating solution from the archive. The rest of the operators in the multi-objective ant-lion algorithm are similar to the operators in the ant-lion algorithm. The pseudo-code of the multi-objective ant lion algorithm is shown in Fig. 3.

```

Until the moratorium condition is established
For any ant
Choose a random ant lion from the archive
Select the favorite ant using the roulette wheel from the archive
Update c and d using equations (44) and (45)
Create a random and abnormal motion using equations (39) and (40).
Update the position of each ant using Equation (47)
End
Calculate the value of the objective function for all ants
Update the archive
If the archive is full
Remove some solutions using the roulette wheel and equation (49) and replace the new solutions.
End
End

```

Fig. 3. Pseudo code of Multi-objective ant-lion algorithm

4.2 Designing the initial answer to hub covering location problem

The initial answer used in this paper is a series of random numbers in the number of nodes under study. For example, in accordance with Fig. 4, the initial answer is shown for 6 customers.

Nod	1	2	3	4	5	6
Initial response	2	2	3	6	3	6

Fig. 4. Initial response designed for hub covering location problem

According to Fig. 4, number generation is in such a way that each generated number is considered as a hub for its customer. For example, in the above figure, nodes 2, 3, and 6 are selected as main hubs and it is observed that nodes 1 and 2 are assigned to hubs 2, nodes 3 and 5 to hubs 3, and nodes 4 and 6 to hubs 6. After the network is established, the demand for each hub is specified and if two of the following conditions are not met, fines are added to the objective function:

- A) The demand for nodes connected to the hub is greater than the capacity of the hub; and
- B) Serving nodes by hub servers is less than the demand for nodes connected to the hub.

4.3 Evaluation criteria for multi-objective meta-heuristic algorithms

One of the problems that exists in solving multi-objective problems is how to evaluate the quality of the final solutions, which is sometimes complicated by the contradiction of the objectives used. To this end, in the early 1990s, visual methods were used to compare Pareto sets. The convergence to Pareto's optimal solutions and the provision of density and diversity among the set of answers are two main objectives of each multi-objective evolutionary algorithm, but because these two goals are somewhat in contradiction to one another, there is no criterion that can solve individually and absolutely the algorithm's function. Therefore, in order to evaluate the performance of the proposed algorithms, the following criteria are used:

A: CPU Time: An algorithm with less computational time will be more desirable.

B: Number of Parrot Responses (NPFs): The number of undefined answers in the Pareto set for each problem is shown, and each value is greater than the number of points, which means that the algorithm is more efficient.

C: Maximum Spread Index (MSI): This criterion shows how much of the answer to a Pareto set is distributed in the distributed space, which is calculated from the following equation. The larger value of this criterion indicates the proper variation of the Pareto set.

$$MSI = \sqrt{\sum_{m=1}^M (\max_{i=1:|Q|} f_m^i - \min_{i=1:|Q|} f_m^i)^2} \quad (51)$$

D: Spacing Metric (SM): Indicates the amount of answers placed equally together, which is calculated from the following equation.

$$SM = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (52)$$

In the above, $|Q|$ is the size of the Pareto archive, and the values d_i and \bar{d} can be calculated from the following equations, respectively. An algorithm that is less than this criterion will be more desirable.

$$\bar{d} = \sum_{i=1}^{|Q|} \frac{\min_{k \in Q \cap k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|}{|Q|} \quad (53)$$

E: Mean Ideal Distance (MID): This criterion is used to measure the closeness to Pareto real level, which is calculated as follows:

$$MID = \frac{\sum_{i=1}^n \sqrt{(f_{1i} - f_1^*)^2 + (f_{2i} - f_2^*)^2 + \dots + (f_{mi} - f_m^*)^2}}{n} \quad (54)$$

In this equation, n is the number of solutions in the Pareto optimal set.

5. Data analysis

In the previous section, the modeling of hub covering location problem considering the queuing system under uncertainty conditions was investigated and robust-box optimization method was used to control its non-deterministic parameters (demand and shipping cost). In this section, the exact method and meta-heuristic algorithms are used to evaluate the model and study the output variables of the problem and solve it. The main objective of this section is to evaluate the efficiency of the MOLAO Multi-objective Algorithm (MOLAO) for obtaining efficient two-objective models. Therefore, in this section, a small sample size is firstly designed and the two-objective model is solved using a multi-criteria decision-making method. The analysis of sensitivity of the problem is also discussed below. In the next section, the model is solved using the meta-heuristic algorithms, and the MOALO performance is compared with the NSGA II and MOPSO algorithms.

5.1 Solving sample problem in small size

In this section, in order to evaluate the model and examine the output variables, a small sample problem is designed taking into account 6 nodes. Due to the lack of access to real data, random data is used based on uniform distribution as shown in Table (1) below.

Table 1

The boundaries of the parameters produced on the basis of uniform distribution

Parameter	Range interval	Parameter	Range interval
F_k	$\sim U(10000, 12000)$	Cap_K	$\sim U(100, 160)$
Dis_{ij}	$\sim U(40, 50)$	Γ_k	$\sim U(47, 20)$
TC_{ij}	$\sim U(20, 50)$	CT_k	$\sim U(10, 20)$
C_{ij}	$\sim U(120, 180)$	μ_k	$\sim U(80, 90)$
FL_i	$\sim U(30, 40)$	B_k	4
ϑ_k	3	θ_k	0.5
χ	0.6		

As the designed model is two-objective, multi-objective decision-making methods should be used to solve the model. In this paper, a multi-objective decision method is used to solve the two-objective model. Also due to the nonlinearity of the model, Solver Baron on the GAMS 24.8.5 software has been used. In the comprehensive decision-making process, it is necessary to obtain the best value of each objective function by means of individual optimization. That is, one must first obtain the value of each objective function, regardless of the other objective function, by the software to be used in the calculation. Equation (55) shows multi-dimensional decision-making method of the comprehensive benchmark.

$$L_p = \left\{ \sum_{i=1}^n W_i \left[\frac{(f_i - f_i^*)}{(f_i^*)} \right]^p \right\}^{\frac{1}{p}}, p \geq 1 \quad (55)$$

Also in this article, $p = 0.1$ has been used. Table (2) shows the optimal value of each objective function using individual optimization method.

Table 2

Optimal value of first and second objective functions using individual optimization method

Objective Function	Best Value	Cpu time
FOBV1	23012.128	1041.10
SOBV2	244.556	343.94

The high computational time obtained for solving the two-objective problem is due to the nonlinearity and complexity of the use of queuing system equations in the problem. In the following, considering the use of the comprehensive benchmarking method, the nodes selected as hubs as well as the optimal assignment of non-hub nodes to the hub nodes are shown in Fig. 5.

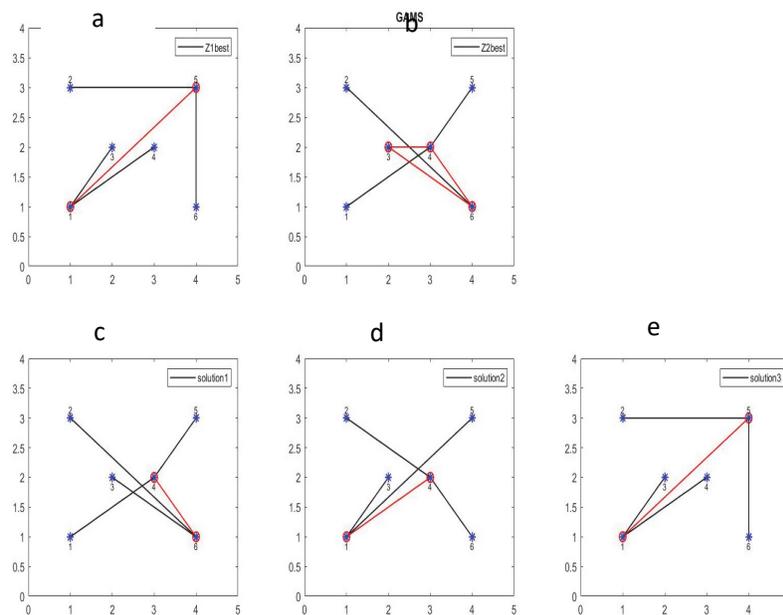


Fig. 5: How to assign non-hub nodes to hub nodes in problem-solving using a comprehensive benchmarking method

According to Fig. 5, it is observed that in solving the problem, considering the first objective function of nodes 1 and 5 as hubs, and considering the second objective function, nodes 3, 4 and 6 are selected as hubs. Also, after solving the above model with comprehensive benchmarking method, 3 efficient responses were obtained according to Table 3. Output results of decision variables are also shown in Table 4.

Table 3

Efficient solutions obtained from solving a small size sample using a comprehensive benchmarking method

Solution	FOBV1	SOBV2	CPU time
1	23653.03	245.069	
2	23371.84	250.496	2071.14
3	23012.12	265.920	

Table 4

How to assign non-hub nodes to different hubs in efficient solutions

Solution	Fig.	Selected nodes as hubs	Allocation of non-hub nodes to hubs						
1	C	4,6	Node	1	2	3	4	5	6
			Hub	4	6	6	4	4	6
2	D	1,4	Node	1	2	3	4	5	6
			Hub	1	4	1	4	4	1
3	E	1,5	Node	1	2	3	4	5	6
			Hub	1	5	1	1	5	5

5.2 Sensitivity analysis

After verifying and validating the designed model, in order to analyze the sensitivity of the model, several parameters of the model are selected and the changes in the value of the first and second objective functions of the problem solving are shown. First, the number of service providers in the three different scenarios is equal to the values of 2, 3, and 4, and the value of the first and second objective functions is shown in Table 5.

Table 5

Changing the values of objective functions by applying changes to the number of service providers

Scenario	Number of servers	Selected nodes as hubs	FOBV1	SOBV2
1	2	1,4	23418.24	252.751
2	3	1,4	23371.84	250.496
3	4	1,4	23231.17	249.471

According to the results of Table 5, with increasing the number of service providers in selected hubs, overall costs and greenhouse gas emissions are reduced. Fig. 6 shows the trend of changes to the first and second objective functions in different scenarios.

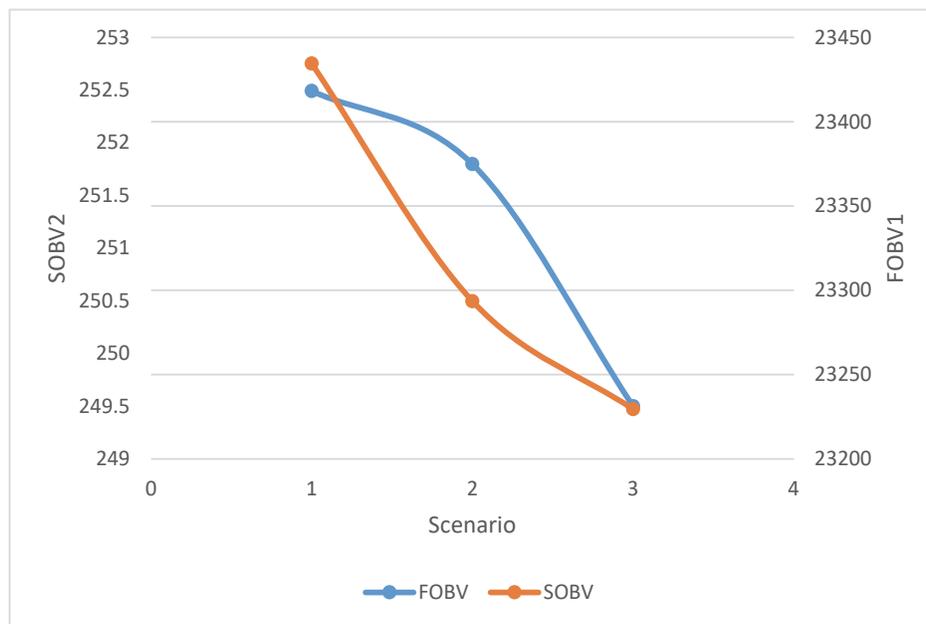


Fig. 6. Changes in the value of the first and second objective functions by changing the number of service providers

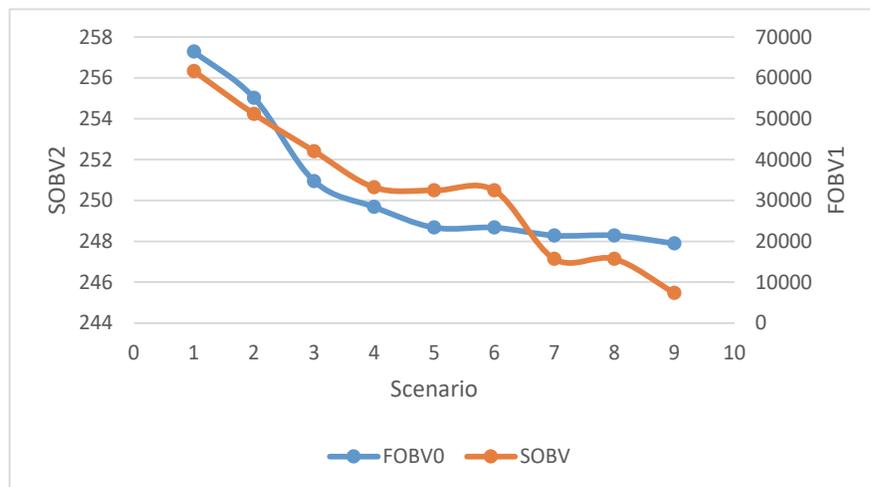
In the second analysis, the change in the capacity of the selected hubs is considered. For this purpose, the capacity of the hubs is as high as 20% to 50% lower and more than the base value, and changes in the objective functions 1 and 2 are presented in Table 6 in nine different scenarios.

Table 6

Changing the values of objective functions applying changes to the capacity of the hubs

Scenario	Percent change in capacity	Average hub capacity	Selected nodes as hubs	FOB V1	SOBV2
1	-50%	58.55	2,3,4,6	66415.88	256.33
2	-40%	70.26	1,3,4,6	55077.48	254.23
3	-30%	81.97	3,4,6	34732.78	252.41
4	-20%	93.68	3,4,6	28415.71	250.64
5	Base	117.10	1,4	23371.84	250.49
6	+20%	140.52	1,4	23371.84	250.49
7	+30%	152.23	2,4	21426.14	247.14
8	+40%	163.94	2,4	21426.14	247.14
9	+50%	175.56	5	19471.41	245.47

Given the changes in the hub capacity, it can be seen that by reducing the capacity of the hubs, the number of hubs required to meet the customers are increased, and the cost and greenhouse gas emissions are proportionally increased. Conversely, by increasing the capacity of the hubs, the number of hubs reduced, the amount of transport decreased, resulting in total transport costs as well as reduced greenhouse gas emissions. Fig. 7 shows the trend of changes in the value of the first and second objective functions for capacity changes in different scenarios.

**Fig. 7.** Changes in the value of the first and second objective functions by changing the capacity of the hubs

Finally, in order to investigate the effect of the uncertainty rate in robust-box, the uncertainty rate from 0 to 0.5 is considered in 6 different scenarios. The value of the first and second objective functions derived from the certain and uncertain model is given in Table 7, below.

Table 7

Change the values of objective functions applying changes to uncertainty rate

Scenario	Uncertainty rate	Selected nodes as hubs	FOB V1	SOBV2
1	0 (Deterministic)	1,4	23347.17	250.14
2	0.1	1,4	23613.84	251.74
3	0.2	2,5	25146.47	252.19
4	0.3	2,5	25478.63	254.47
5	0.4	1,3,5	47076.37	256.25
6	0.5	1,2,3,5	57706.27	258.99

According to the results of the above table, it can be concluded that increasing the uncertainty rate of costs and greenhouse gas emissions has increased due to the direct effect of the uncertainty rate on demand. However, the robustness of the model has been proven. With an increase in the uncertainty rate as high as 50%, the cost of network design has increased by 1.47 times. Figure (8) shows the trend

of changes in the value of the first and second objective functions in certain and uncertain conditions in different scenarios.

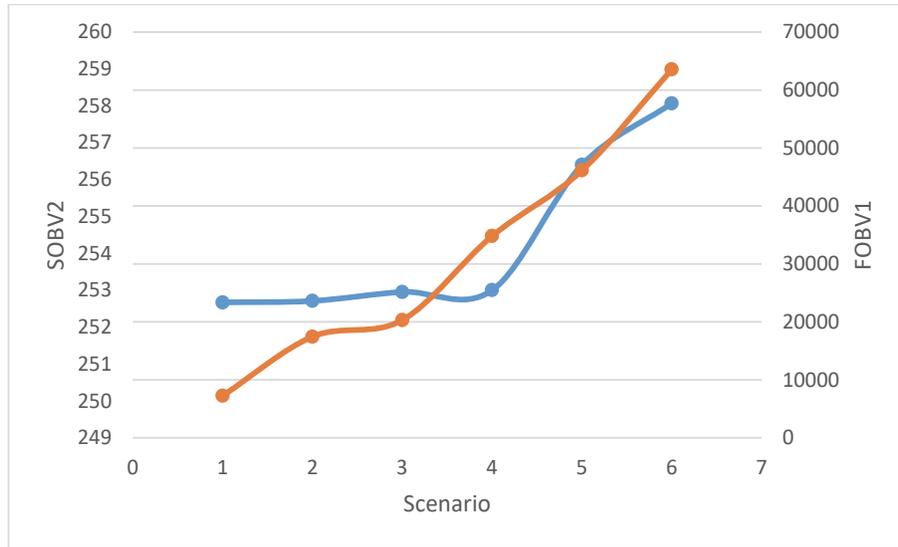


Fig. 8. Changes in the value of the first and second objective functions in certain and uncertain conditions

5.3.3 Adjustment of the Algorithm using Taguchi Method

Before solving the problem using meta-heuristic methods, the parameters of the MOALO, MOPSO and NSGA II algorithms were determined by Taguchi method. In Taguchi process, the appropriate factors must first be identified and then the levels of each of the factors should be selected and then the appropriate test design for these control factors should be determined. After determining the design of the experiment, the experiments are carried out and the tests are analyzed with the aim of finding the best combination of parameters. In this paper, for each factor, 3 levels are considered in accordance with Table 8. For each algorithm, considering the number of factors and the number of their levels, they have determined the design of the experiment and their implementation.

Table 8

The value of proposed and optimized parameters of meta-heuristic algorithms

Algorithm	Parameter	Level 1	Level 2	Level 3	Optimum
NSGA II	Max it	50	100	200	200
	Npop	50	100	200	100
	Pc	0.3	0.5	0.7	0.3
	Pm	0.3	0.5	0.7	0.7
MOPSO	Max it	50	100	200	200
	Nparticle	50	100	200	50
	C1	1	1.5	2	2
	C2	1	1.5	2	1.5
MOALO	W	0.8	0.9	1	0.9
	Max it	50	100	200	200
	Nant	50	100	200	100

Given that the proposed model is a two objective function, for the analysis of the data, first, the value of each experiment is to be calculated from equation (58). In this equation, in the case of the deduction, these indicators are used in the comparison of meta-heuristic algorithms including (the number of efficient responses, the maximum spread index, the spacing metric, the mean ideal distance, and the computational time). After determining the amount of each test, using equation (59), RPD is calculated for the analysis of the design of the Taguchi test.

$$S_i = \frac{NPF + MSI - SM - MID - CPU}{5} \quad (58)$$

$$RPD = \frac{S_i - S_i^*}{S_i^*} \quad (59)$$

In this section, after providing the parameter adjustment of the meta-heuristic algorithm, the two-objective model is solved using the NSGA II, MOPSO and MOALO algorithms. Therefore, in order to examine the output variables and the accuracy of the coding provided, the small sample of 6 nodes with random data generated according to table (1) and the effective solutions from the problem solving is shown in Table 9.

Table 9

Efficient solutions obtained from solving a small sample problem using meta-heuristic algorithms

Algorithm	Solution	FOBV1	SOBV2	Cpu time
NSGA II	1	23340.15	240.56	73.79
	2	34431.83	239.87	
	3	34906.71	238.88	
	4	44840.28	233.45	
	5	44884.11	230.43	
MOPSO	1	23340.15	240.56	35.72
	2	34431.83	239.87	
	3	44932.86	235.29	
	4	54970.76	232.67	
	5	56074.07	232.08	
MOALO	1	22785.14	264.14	9.82
	2	33373.48	242.14	
	3	34431.83	239.87	
	4	44840.28	233.45	
	5	56074.07	232.08	

Table 9 shows the efficient solutions obtained from problem solving with the NSGA II, MOPSO and MOALO algorithms. According to the results, each algorithm has 5 different responses that the MOALO algorithm has been able to achieve in desirable solutions in the least time. Therefore, in order to show the output variables of each efficient answer, for each algorithm, Fig. (9) to Fig. (11) are presented. In these figures, blue stars show problem nodes, red circles show the selected hubs, red lines show the links between hubs and black lines represent the connection between nodes and hubs.

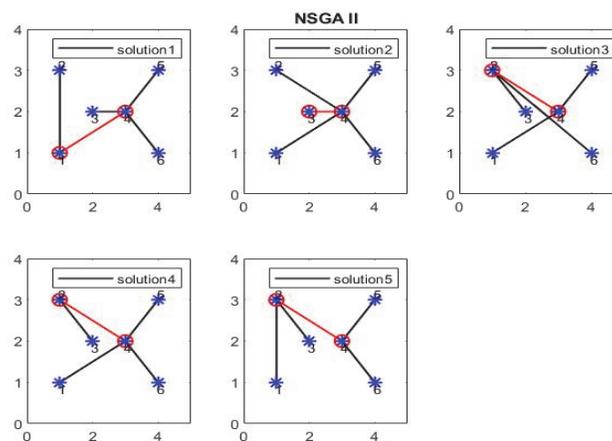


Fig. 9. How to assign non-hub nodes to hub nodes in problem solving with NSGA II algorithm

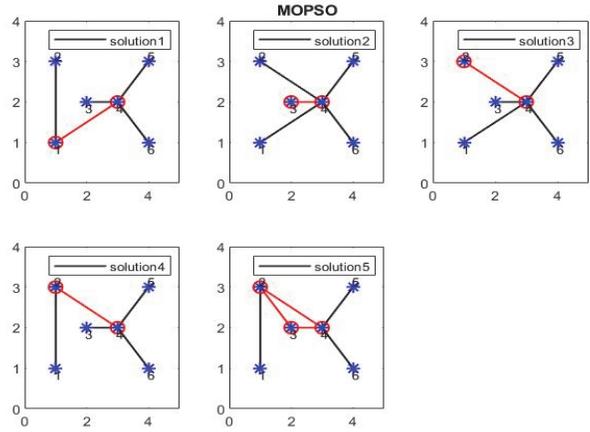


Fig. 10. How to assign non-hub nodes to hub nodes in problem solving using the MOPSO algorithm

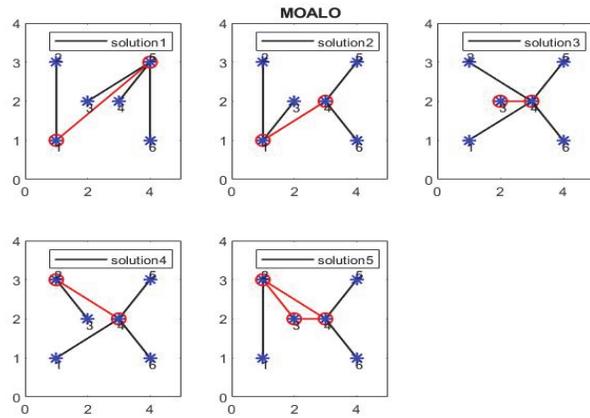


Fig. 11. How to assign non-hub nodes to hub nodes in problem solving with the MOALO algorithm

Also, for the overall comparison of the results obtained from the solution of this model using the precise method and the meta-heuristic algorithms, Table 10 shows the general indices used in comparing the efficient solutions for the four solution methods. These indicators include (the number of efficient responses, the maximum spread index, the spacing metric, the mean ideal distance, and the computational time)).

Table 10

Comparison of solving problem indices in small sample solution

Algorithm	NSGA II	MOPSO	MOALO	GAMS
NPF	5	5	5	3
MSI	21543.96	32733.92	33288.94	640.58
SM	1.53	0.62	0.68	0.10
MID	13468.49	19737.81	15380.0	340.68
Cpu time	73.79	35.72	9.82	2071.14

By comparing the indexes of the above table, it can be concluded that the efficiency of all algorithms is better than GAMS software for obtaining various responses. Thus, according to the most exponential index, it can be concluded that the range of efficient solutions presented to the decision maker is higher than the GAMS. Among the above mentioned meta-heuristic algorithms, the NSGA II algorithm for obtaining indices (average objective function and mean ideal distance), the MOPSO algorithm for obtaining indices (average of the second objective function and the spacing metric) and the MOALO algorithm has worked more efficiently than other algorithms in obtaining the (maximum spread index) and average computational time indices. Fig. (12) also presents the Pareto front of the solution of the small sample problem solving with all the solving methods.

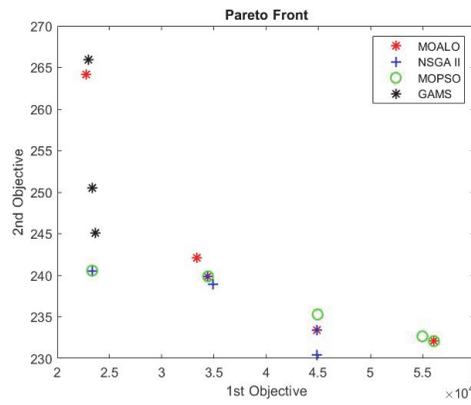


Fig. 12. Comparison of efficient solutions to solving a small sample problem using the solution methods

Due to the high computing time of GAMS software in solving a small sample problem and its inability to solve larger-size problems, only NSGA II, MOPSO and MOALO algorithms have been used to solve problems.

4.5 solving sample problems in larger sizes using meta-heuristic algorithms

As stated, due to the NP-hardness and non-linearity of the model, GAMS spent a lot of time to solve a small sample problem, and the problem with the GAMS software problem did not resolve. For this reason, larger-scale algorithms have been used to solve the problem. For this purpose, 9 sample problems are designed according to Table 11 and are based on random data generated in accordance with Table 1.

Table 11

Size of sample issues designed in larger sizes

Sample Problem	Nodes	Sample Problem	Nodes	Sample Problem	Nodes
1	10	4	25	7	40
2	15	5	30	8	45
3	20	6	35	9	50

After design, each sample problem is executed 5 times by any algorithm, which means the obtained functions averages of 5 runs are considered as the benchmark of comparison. Fig. 13 also shows the process of variations in the total average of the first and second objective function by increasing the size of sample problems in larger sizes for the NSGA II algorithm.

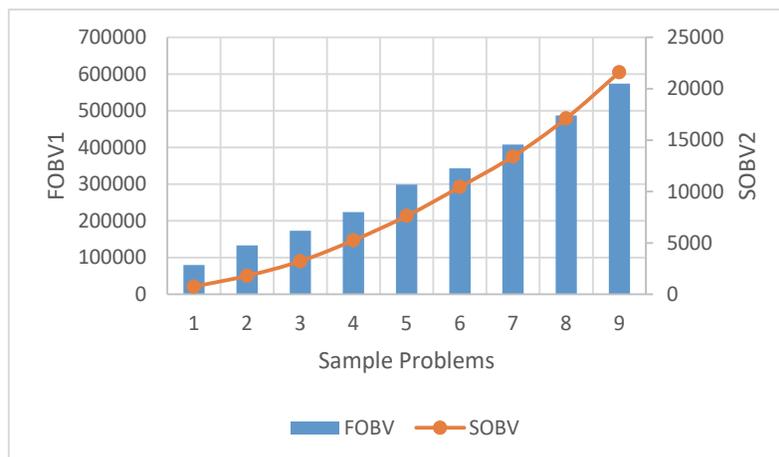


Fig. 13. Changes in the first and second objective function in different sample problems by NSGA II algorithm

According to the results of Fig. 13, it can be seen that by increasing the size of the problem, the values of the first and second objective functions, due to increased demand, increased and the trend was uptrend. Fig. 14 shows the average changes in the total of the first and second objective functions by increasing the size of sample problems in larger sizes for the MOPSO algorithm.

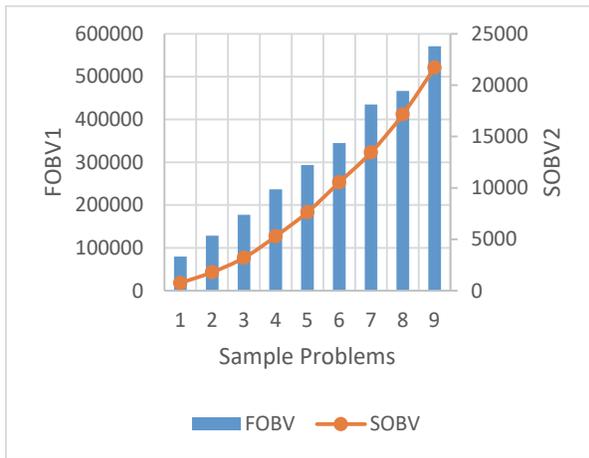


Fig. 14. Changes in the first and second objective function in different sample problems by MOPSO algorithm

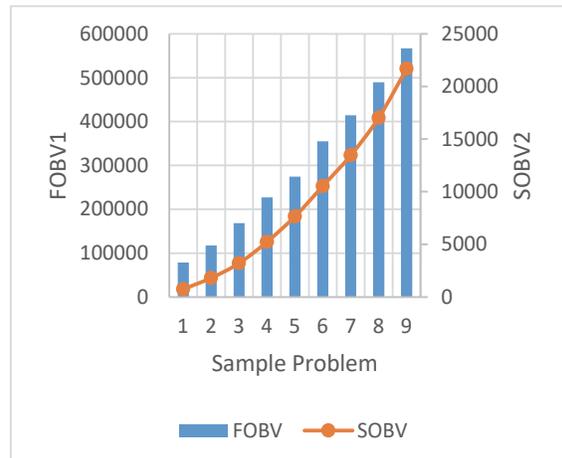


Fig. 15. Changes of the first and second objective function in different sample problems by MOALO algorithm

Finally, Fig. 15 shows the process of variations in the total average of the first and second objective function by increasing the size of sample problems in larger sizes for the MOALO algorithm. According to the results of the diagram of Fig. 15, by increasing the size of the problem, the value of the first objective function and the second objective function is also increased and the trend is upward due to increased demand. To compare the significant difference between the first, second, and computational time averages obtained by the NSGA II, MOPSO and MOALO algorithms, the Tukey multiplex test was used at 95% confidence level. According to the hypothesis test, if the P value is less than 0.05, then there is a significant difference between the averages of that computational index between the used algorithms. Table 12 shows the results of the Tukey multiple test output for large sample size problems with these algorithms.

Table 12

Tukey multiple test outputs at 95% confidence level

Algorithm	Index	Means Different	Confidence interval 95%		T-value	P-value
			L bound	U bound		
NSGA II-MOPSO	FOBV1	1458.47	-10470	7553	0.33	0.746
NSGA II-MOALO		3232.47	-3118	9583	1.03	0.311
MOPSO-MOALO		4690.94	-4432	13814	1.04	0.306
NSGA II-MOPSO	SOBV2	37.55	-11	862	1.56	0.124
NSGA II-MOALO		19.03	-54	519	1.1	0.279
MOPSO-MOALO		18.52	-24	61	0.88	0.385
NSGA II-MOPSO	Cpu_time	165.92	40	291	3.04	0.016
NSGA II-MOALO		397.58	145	650	3.63	0.007
MOPSO-MOALO		231.66	102	361	4.13	0.003

According to the results obtained from statistical analysis, all P values for the mean index of the first and second objective functions are greater than the value of 0.05, which indicates that there is no significant difference between the mean of this index among the NSGA II, MOPSO and MOALO algorithms. Is. While the P value in the computing time average is less than 0.05, which means that there is a significant difference between the computational time averages among all the algorithms used. Summarizing the above table, it can be concluded that there is a significant difference between the computational time and the small computational time of the MOALO algorithm in solving sample problems, indicating the high efficiency of this algorithm in comparison with the NSGA II and MOPSO algorithms. This solution is also used to generalize other computational indices (such as the number of

efficient responses, the most exponential index, the metric distance index, and the distance indicator from the ideal point). In the following, in Fig. (16) through Fig. (18), the process of changes in the computing index average is shown in all sample problems by the NSGA II, MOPSO and MOALO algorithms.

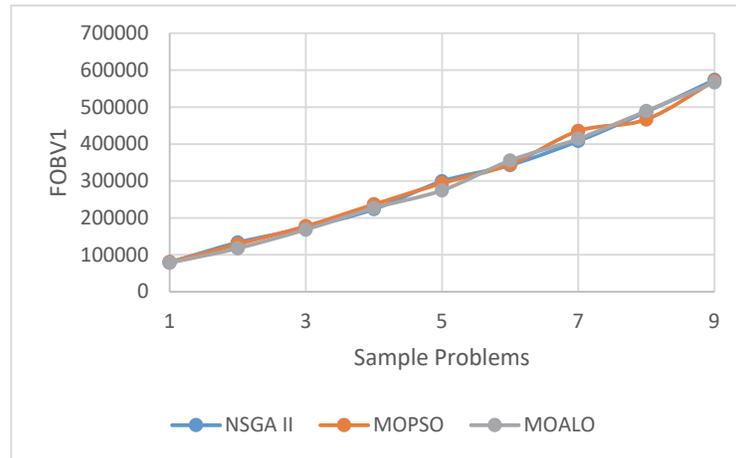


Fig. 16. Changes in averages of the first objective function in large sample sizes

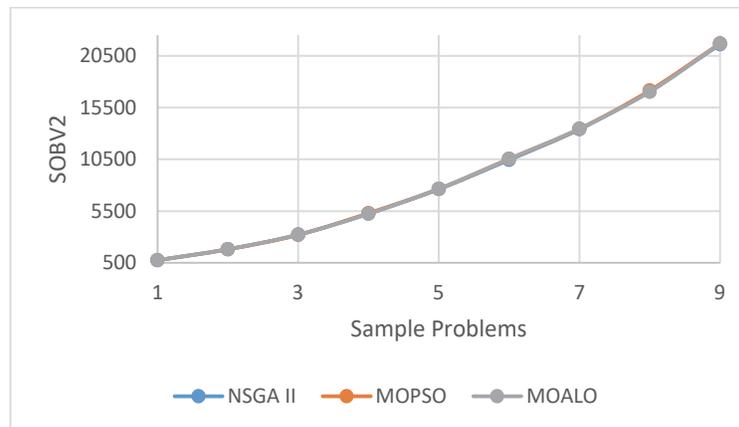


Fig. 17. Changes in averages of the second objective function in large sample sizes

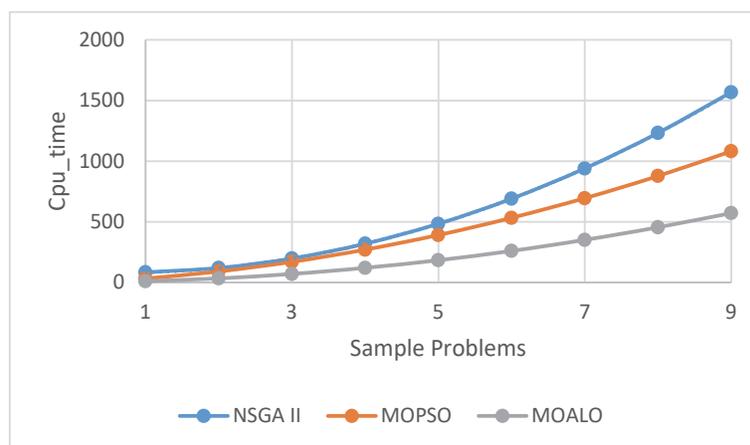


Fig. 18. Changes in computational time averages in large sample cases

According to the above diagrams, in particular diagram of Fig. 18, it can be stated that as the size of the problem is increased, the computational time is exponentially increasing, as the process of increasing the computational time of the MOALO algorithm is less than the other algorithm.

5.5 Choosing the most efficient algorithm using TOSIS method

Given all the results obtained from solving sample problems in different sizes, the average of all indices in all sample problems 1 to 9 is used to select the most efficient algorithm. Accordingly, Table 13 shows the averages of the calculated indexes for the NSGA II, MOPSO and MOALO algorithms. The entropy weight mentioned is also chosen according to the importance of the indicators.

Table 13

Total Averages of Problem Solving Indicators

Index	NSGA II	MOPSO	MOALO	Entropy
NPF	13.66	13.88	15	0.2
MSI	47339.99	49812.41	41379.49	0.2
SM	0.623	0.687	0.865	0.2
MID	26224.58	24986.99	23983.32	0.2
Cpu time	626.03	460.10	228.44	0.2
Utility Weight	0.1603	0.4378	0.8073	

After weighing, the desirability weight of each of the algorithms was calculated. The MOALO algorithm is known to be the most efficient algorithm with the desirability weight of 0.8073, and then the MOPSO and NSGA II algorithms are known to be the most efficient algorithms with the desirability weight of 0.4378 and 0.1603, respectively.

6. Conclusion

In this paper, modeling the hub covering location problem with respect to the queuing system was dealt with, whose objective functions were to minimize total network costs and minimize greenhouse gas emissions. Because hub covering network models are multi-objective and NP-hard, a multi-objective meta-heuristic ant-lion algorithm was used. First, a hub covering location problem in an uncertain mode was modeled considering the queue system. In this step, demand parameters and transport costs were considered uncertain, using a robust-box optimization method to control these parameters and convert the problem to a certain version. In this regard, the queuing system was modeled in hub centers and the costs of the queuing system were calculated and added to the objective function. According to the two objective functions of the proposed model for problem solving, the exact method (GAMS software) and the meta-heuristic algorithms NSGA II, MOPSO and MOALO were used. The result was the high GAMS software computing time for problem solving. In this phase, the outputs from the small size problem were compared using a comprehensive benchmarking method with the above algorithms. In the next step, 9 sample problems were designed in larger sizes and the model was solved by meta-heuristic algorithms NSGA II, MOPSO and MOALO. To compare the effectiveness of meta-heuristic algorithms, Tukey multiple test was used. According to the results of statistical analysis, all P values for mean indexes of the first and second objective functions were more than 0.05, which indicates that there was no significant difference between the mean of this index among NSGA II, MOPSO and MOALO algorithms. However, the P value in the computing time average is less than 0.05, which means that there is a significant difference between the computational time averages among all the algorithms used. It was also concluded that there is a significant difference between the computational time and the small computational time of the MOALO algorithm in solving sample problems, indicating the high efficiency of this algorithm compared to the NSGA II and MOPSO algorithms. Finally, for selecting the most efficient algorithm, five indicators (the number of efficient responses, the maximum spread index, the spacing metric, the mean ideal distance, and the computational time) were considered. The results obtained by the TOPSIS method showed that the MOALO algorithm with the desirability weight of 0.8073 was considered as the most efficient algorithm, and then the MOPSO and NSGA II algorithms with the desirability weight 0.4378 and 0.1603 as the most efficient algorithms for solving the proposed model. Following this article, these are presented as suggestions for other researchers to conduct future research:

- A. Taking into account several products in the model instead of a single product,

- B. Using vehicle routing in hub location instead of direct allocation
- C. Using a fuzzy-robust method instead of a robust-box optimization method, and
- D. Use of other meta-heuristic algorithms

References

- Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1), 1-21.
- Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization* (Vol. 28). Princeton University Press.
- Calik, H., Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, 36(12), 3088-3096.
- Campbell, J., Ernst, A., & Krishnamoorthy, M. (2002). *Hub location problems*.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2), 387-405.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4), 1096-1109.
- Gao, Y., & Qin, Z. (2016). A chance constrained programming approach for uncertain p-hub center location problem. *Computers & Industrial Engineering*, 102, 10-20.
- Gavriliouk, E. O. (2009). Aggregation in hub location problems. *Computers & Operations Research*, 36(12), 3136-3142.
- Ghaffarinasab, N., Motallebzadeh, A., Jabarzadeh, Y., & Kara, B. Y. (2018). Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems. *Computers & Operations Research*, 90, 173-192.
- Goldman, A. J. (1969). Optimal locations for centers in a network. *Transportation Science*, 3(4), 352-360.
- Hasanzadeh, H., Bashiri, M., & Amiri, A. (2018). A new approach to optimize a hub covering location problem with a queue estimation component using genetic programming. *Soft Computing*, 22(3), 949-961.
- Huang, J., & Qingyun, W. A. N. G. (2009). Robust optimization of hub-and-spoke airline network design based on multi-objective genetic algorithm. *Journal of Transportation Systems Engineering and Information Technology*, 9(3), 86-92.
- Makui, A., Rostami, M., Jahani, E., & Nikui, A. (2011). A multi-objective robust optimization model for the capacitated P-hub location problem under uncertainty. *Management Science Letters*, 2(2), 525-534.
- Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as M/D/c queues. *Computers & Operations Research*, 30(7), 983-1003
- Merakli, M., & Yaman, H. (2017). A capacitated hub location problem under hose demand uncertainty. *Computers & Operations Research*, 88, 58-70.
- Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*, 46(1), 79-95.
- Nickel, S., Schöbel, A., & Sonneborn, T. (2001). Hub location problems in urban traffic networks. *Mathematical Methods on Optimization in Transportation Systems*, 48, 95.
- O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation science*, 20(2), 92-106.
- O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3), 393-404.
- O'Kelly, M. E. (1992). Hub facility location with fixed costs. *Papers in Regional Science*, 71(3), 293-306.
- Peker, M., & Kara, B. Y. (2015). The P-Hub maximal covering problem and extensions for gradual decay functions. *Omega*, 54, 158-172.

- Rodríguez-Martín, I., Salazar-González, J. J., & Yaman, H. (2014). A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, 50, 161-174.
- Seifbarghy, M., Hemmati, M., & Soltan Karimi, S. (2018). Hub Covering Location Problem Considering Queuing and Capacity Constraints. *Journal of Optimization in Industrial Engineering*, 11(1), 143-156.
- Silva, M. R., & Cunha, C. B. (2017). A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem. *European Journal of Operational Research*, 262(3), 954-965.
- Yu, J., Liu, Y., Chang, G. L., Ma, W., & Yang, X. (2009). Cluster-based hierarchical model for urban transit hub location planning: formulation, solution, and case study. *Transportation Research Record*, 2112(1), 8-16.
- Zhai, H., Liu, Y. K., & Yang, K. (2016). Modeling two-stage UHL problem with uncertain demands. *Applied Mathematical Modelling*, 40(4), 3029-3048.



© 2020 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).