



Bi-objective optimization of multi-server intermodal hub-location-allocation problem in congested systems: modeling and solution

Mahdi Rashidi Kahag¹ · Seyed Taghi Akhavan Niaki² · Mehdi Seifbarghy³ · Sina Zabihi⁴

Received: 29 July 2017 / Accepted: 25 July 2018 / Published online: 22 August 2018
© The Author(s) 2018

Abstract

A new multi-objective intermodal hub-location-allocation problem is modeled in this paper in which both the origin and the destination hub facilities are modeled as an $M/M/m$ queuing system. The problem is being formulated as a constrained bi-objective optimization model to minimize the total costs as well as minimizing the total system time. A small-size problem is solved on the GAMS software to validate the accuracy of the proposed model. As the problem becomes strictly NP-hard, an MOIWO algorithm with an efficient chromosome structure and a fuzzy dominance method is proposed to solve large-scale problems. Since there is no benchmark available in the literature, an NSGA-II and an NPGA are developed to validate the results obtained. The parameters of all algorithms are tuned using the Taguchi method and their performances are statistically compared in terms of some multi-objective metrics. Finally, the entropy-TOPSIS method is applied to show that MOIWO is the best in terms of simultaneous use of all the metrics.

Keywords Intermodal P-hub median problem · Queuing systems · Multi-objective invasive weed optimization · Entropy-TOPSIS method

Introduction and literature review

Aiming to minimize the total network cost (transportation and fixed cost) while the customer demand is met, the hub-location-allocation problem (HLAP) is generally stated to determine the locations of hub nodes from a set of potential

hub nodes in a network and to allocate demand nodes to the selected hub nodes. HLAPs are classified into four major groups of problems including (1) capacitated and incapacitated HLAP, (2) P-hub-covering location problems, (3) P-hub center problems, and (4) P-hub median problems (Campbell 1994). In an incapacitated HLAP, the capacities of the hub nodes are not limited, while in a capacitated HLAP, it is limited and all the non-hub nodes may not be dedicated to their nearest hub nodes. In the P-hub-covering location problems, the number of hub nodes is limited. This constraint limits the areas covered by each hub node. The P-hub center problems are defined to find the best positions of the hubs in a network and to allocate non-hub nodes to them so that the maximum travel time (or distance) between any origin–destination pair is minimized. Finally, analogous to the P-median problem, the P-hub median problem is a hub location problem in which the number of hubs (P) is predetermined. The aim is to locate the hubs, to allocate the non-hubs to the hubs, and to determine the path for each origin–destination pair such that the total transportation cost for satisfying the demand of the customers, each supplied from its closest facility, is minimized.

✉ Seyed Taghi Akhavan Niaki
Niaki@Sharif.edu

Mahdi Rashidi Kahag
m.rashidi.k.eng@gmail.com

Mehdi Seifbarghy
M.Seifbarghy@alzahra.ac.ir

Sina Zabihi
SinaZabihi85@gmail.com

¹ Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran
² Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11155-9414, Azadi Ave, Tehran, Iran
³ Department of Industrial Engineering, Alzahra University, Tehran, Iran
⁴ Department of Industrial Engineering, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

Hub-location-allocation models were firstly developed by O’Kelly (1986, 1987), where a quadratic integer model for the hub location problem was presented and shown to belong to the class of NP-Hard problems. Campbell (1994) used some flow variables to present a new linear formulation for multiple-allocation P-hub problems. Lin et al. (2012) developed a mathematical formulation for the capacitated P-hub median problem. The objective of their model was to minimize the total costs consisting of the fixed costs of creating the hub facilities along with the transportation costs. They added some singular path constraints to the model in order to prevent storing commodities on hub facilities. García et al. (2012) presented a mixed-integer linear program for the incapacitated multiple-allocation P-hub median problem. Their model aimed to determine the optimal location of the hub facilities among some candidate sites so that the total transportation cost is minimized.

Ghodratnama et al. (2013) proposed a fuzzy possibilistic bi-objective nonlinear mixed-integer programming formulation for the hub-covering problem. They considered the major parameters of their mathematical formulation to be fuzzy in order to model the uncertainties involved. The first objective function of their formulation intended to minimize six major cost components comprising opening cost, reopening cost, transportation cost, covering cost, activating cost, and cost of purchasing transportation vehicles. The second objective function of their proposed model designed to minimize the transportation of the commodities from their origin nodes to their destination nodes using the hub facilities. They used a linearization method to convert the problem into a linear optimization problem, based on which they utilized four fuzzy approaches to solve the problem. Peiró et al. (2014) introduced a novel mixed-integer programming model for the P-hub median problem and extended it to an R-allocation P-hub median problem. Based on this extension, they assumed that every spoke node could be allocated to R of P chosen active hub nodes. Their model meant to minimize the total routing cost. They proposed a heuristic algorithm to solve the proposed problem. Habibzadeh Boukani et al. (2016) developed two mathematical models, one for a capacitated single, and the other for a capacitated multiple-allocation P-hub median problems. They used a robust optimization approach to consider uncertainties involved in the parameters of the models including the capacity of the hub nodes and the fixed costs of establishing the hub facilities. They showed that the consequences of considering the uncertainties were to reduce the total cost as well as to reduce delays in delivering the commodities to the customers. Parvaresh et al. (2014) proposed a mixed-integer programming formulation for the multiple-assignment P-hub median problem. They developed a bi-objective bi-level model to

consider the occurrence of intentional disruptions on the hub facilities. The upper-level model aimed to optimize two different objectives of minimizing the total transportation costs in normal and worst cases. Meanwhile, the lower level aimed to minimize the maximum damage to the network. A bi-objective nonlinear binary programming model was presented by Zade et al. (2014) for the maximal hub-covering problem. The first objective function of their formulation intended to maximize the total utility of the hub network and the second objective was to maximize the safety of the network’s weakest path. They also considered uncertainties involved in transporting the commodities from their origins to their destinations via the hub nodes. Ghaffari-Nasab et al. (2015) developed two mixed-integer linear programming models for a single and a multiple-allocation capacitated hub location problem with stochastic demands. Their models aimed to minimize the total cost (fixed, transportation, and capacity acquisition costs). They employed a robust optimization approach to account for the uncertainties in the parameters and used a standard optimization package to solve the problems.

Various exact and heuristic methods were developed in the literature to solve hub location problems. O’Kelly (1986, 1987) proposed an enumeration-based heuristic to solve the problem. Skorin-Kapov et al. (1996) proposed an exact method to solve P-hub median problem by using a tight linear relaxation method presented by Campbell (1992). A hybrid branch-and-bound and simulated annealing (SA)-based greedy interchange heuristic was proposed by Aykin (1995) to solve an HLP. Campbell (1996) introduced another heuristic based on the greedy interchange algorithm to solve a P-hub median problem. Puerto et al. (2011) presented a new branch-and-bound and cut method for the single-assignment P-hub median problem. They concluded that their proposed algorithm could find efficient solutions for small- and medium-sized problems. García et al. (2012) used a branch-and-cut algorithm to solve an incapacitated multiple-allocation P-hub median problem modeled by a mixed-integer linear programming formulation and showed that it performs well especially when it solves large-scale instances. Peiró et al. (2014) proposed a heuristic algorithm to solve a P-hub median problem modeled by a mixed-integer programming model.

Since the time O’Kelly (1987) proved that the hub-location-allocation problem is NP-hard, many meta-heuristic algorithms have been developed to find efficient feasible solutions. The genetic algorithms (GA) of Topcuoglu et al. (2005), Kratica et al. (2007), Cunha and Silva (2007), Takano and Arai (2009) and Lin et al. (2012); the Tabu search (TS) algorithms of Calik et al. (2009), Ishfaq and Sox (2010), Shahvari et al. (2012) and Shahvari and Logendran (2016); the evolutionary algorithm (EA) of Kratica et al. (2011); the ant colony optimization (ACO)

algorithm of Ting and Chen (2013); the particle swarm optimization (PSO) algorithm of Yang et al. (2013); and the variable neighborhood search (VNS) algorithm of Ilić et al. (2010) are some examples among others to solve hub location problems. In multi-objective optimization of the hub-location-allocation problems, Parvaresh et al. (2014) utilized two multi-objective meta-heuristic algorithms called multi-objective simulated annealing (MOSA) and multi-objective Tabu search (MOTS) algorithms. Zade et al. (2014) used a non-dominated sorting genetic algorithm (NSGA-II) to solve a maximal hub-covering problem.

To the best of the authors' knowledge, only a few researchers proposed hub location models that consider congestion. Marianov and Serra (2003) proposed a hub location model for an airline industry under congestion. They used an M/D/c queuing system to impose a probabilistic capacity constraint to the hub location problem first. Then, they turned the probabilistic capacity constraint into a linear form. Elhedhli and Hu (2005) presented a congestion-based hub-location-allocation model. Their model aimed to minimize the total cost of the hub network including congestion cost in the hub nodes along with the transportation cost. As their proposed cost function was nonlinear, they linearized the objective function and solved the linearized model using a Lagrange heuristic method. de Camargo et al. (2009) presented a model for a multiple-allocation hub location problem under congestion. Their objective function aimed to minimize the total cost (fixed, transportation, and congestion costs). The benders decomposition algorithm was utilized in their work to solve the problem. de Camargo et al. (2011) developed a model for a single-allocation hub-location-allocation problem with congestion. Their model aimed to minimize the total costs (congestion and transportation costs). They solved the problem utilizing a hybrid outer approximation method and Benders decomposition algorithm. Mohammadi et al. (2011) introduced an M/M/c queuing model for a hub-covering location problem under capacity constraint. They linearized the probabilistic capacity constraint using M/M/c queuing formulation. de Camargo and Miranda (2012) developed a single-allocation hub location model with congestion. They proposed a nonlinear mixed-integer programming model and considered congestion in the form of a convex cost function. Their model aimed to minimize the total cost (transportation and fixed costs) along with minimizing the maximum congestion effect. A Benders decomposition algorithm was used in their work to solve the problem. Seifbarghi and Mansouri (2016) presented a model to determine the locations of immobile service facilities congested by stochastic demands originated by nearby customer locations. Recently, Maghsoudlou et al. (2016) utilized the concept of congested systems to design

a three-echelon multi-server supply chain network. This concept will also be used in this paper to propose a multi-server intermodal hub-location-allocation model.

The above literature review reveals that there is no research work performed on a multi-objective intermodal hub-location-allocation problem under a capacity constraint that causes congestion. Consequently, in order to fill the gap, a new stochastic multi-objective P-hub median multiple-location problem is formulated in this paper. In other words, the traditional intermodal hub location model is extended in this paper by considering M/M/c queuing systems in the hub nodes. The proposed multi-objective formulation includes two objectives. The first objective intends to minimize the total idle time of the servers located in the hub nodes along with the total waiting time of the customers arriving at the hub nodes. The second objective function involves minimizing the total network costs including transportation, fixed cost of locating the hub nodes, and the fixed cost of locating the servers in the hub nodes.

The rest of the paper is organized as follows. The problem is stated in “[The problem and its formulation](#)” section, where a bi-objective optimization model is proposed. A small-size problem is solved in “[Model validation](#)” section to validate the mathematical formulation and to show the NP-Hardness of the problem. “[Solving methodologies](#)” section is devoted to the meta-heuristic solution approaches. “[Applications](#)” section presents the validation and the application of the proposed methodology. Computational results and performance comparison of the solution methods are presented in “[Computational results](#)” section. Finally, conclusions are brought in “[Conclusion and future research](#)” section, where some topics are suggested for future research.

The problem and its formulation

This section is devoted to describing the problem along with its assumptions and mathematical formulation. To this aim, the problem and the assumptions are first stated in “[Problem definition and assumptions](#)” section. Then, the notation including the indices, the parameters, and the decision variables are given in “[Notation](#)” section. At the end, the model is derived in “[The mathematical formulation](#)” section.

Problem definition and assumptions

The problem being investigated in this paper is an intermodal congested multi-echelon hub-location-allocation problem. In this structure, servers are fixed at their locations in their origin and destination hubs, where



commodities are dedicated to open origin–destination hub pairs to be shipped to their destinations. The aims are to determine the number of required hub facilities for each origin and destination layers, to obtain the number of servers in the hub nodes, and to allocate flows to the open hub facilities. Moreover, to bring the problem more applicable to real-world networks, two objectives are considered based on some real constraints such as the capacities of the hub nodes located at both origin and destination layers. The first objective function minimizes the total costs including transportation and fixed costs and the second minimizes the total system time (including waiting, service, and idle times) in both origin and destination hubs. The total system time is obtained using some formulations in M/M/c queuing systems. Both the origin and the destination hub layers can be mapped in many real-world transportation networks such as airports, rail, and road networks. In these systems, commodities arrive at their origin hub nodes based on a Poisson process with a specific rate and the service time to deliver the commodities to their destination follows an exponential distribution with a given mean. Moreover, it is assumed that all the servers located in a given hub node are similar and share a unique service rate, while the service rates of the servers located in the other hub nodes are different. The transportation process is performed in the context of an intermodal network so that the commodities can be shipped using either a rail or a road network. Furthermore, every origin–destination node pair can be assigned to more than one hub pair.

Notation

The notations including the indices, the parameters, and the decision variables are as follows.

Indices and sets

- i : An index for an origin node.
- j : An index for a destination node.
- k : An index for an origin hub node.
- m : An index for a destination hub node.
- N : The set of all nodes indexed by either i or j or k or m ;
 $N = \{1, 2, \dots, 2|C|\}$.
- C : Set of all cities indexed by c' , $C = \{1, 2, \dots, |C|\}$.
- S : The set of index to calculate π_{0k} ; $S = \{1, 2, \dots, u_k\}$
and π_{0m} ; $S = \{1, 2, \dots, u_m\}$.

Parameters

- λk_k : The arrival rate of the commodities to origin hub node k .

λm_m : The arrival rate of the commodities to destination hub node m .

μk_k : The service rate of the servers located at the origin hub node k .

μm_m : The service rate of the servers located at the destination hub node m .

$u k_k$: The maximum number of servers that can be used in each origin hub node k .

$u m_m$: The maximum number of servers that can be used in each destination hub node m .

$P1$: The number of hub facilities located at the origin layer.

$P2$: The number of hub facilities located at the destination layer.

$A k_k$: The productivity factor of origin hub node k ,
 $A k_k = \frac{\lambda k_k}{\mu k_k}$.

$A m_m$: The productivity factor of destination hub node m ,
 $A m_m = \frac{\lambda m_m}{\mu m_m}$.

α : The cost discount factor.

C : The number of all cities.

νk_k : A multiplier for calculating fixed establishment costs of both types of road/train origin hub node k .

νm_m : A multiplier for calculating fixed establishment costs of both types of road/train destination hub node m .

$F k_k$: The fixed cost of establishing a facility at a potential origin hub node k .

$F m_m$: The fixed cost of establishing a facility at a potential destination hub node m .

$c k_k$: The unit staffing cost at the origin hub node k .

$c m_m$: The unit staffing cost at the destination hub node m .

$C1_{ik}$: The unit transportation cost from the origin node i to the origin hub node k .

$C2_{km}$: The unit transportation cost from the origin hub node k to the destination hub node m .

$C3_{mj}$: The unit transportation cost from the destination hub node m to the destination node j

$$f_{ij} = \begin{cases} \text{Flows from the origin node } i \text{ to the destination node } j; \\ \{i \neq j \text{ and } i, j \leq C\} \\ 0; & \{i = j \text{ or } i > C, \text{ or } j > C\} \end{cases}$$

Decision variables

$$y k_k = \begin{cases} 1 & \text{if node } k \text{ is an origin hub node} \\ 0 & \text{otherwise} \end{cases}$$

$$y m_m = \begin{cases} 1 & \text{if node } m \text{ is a destination hub node} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijkm} = \begin{cases} 1 & \text{if the shipments for the origin destination pair } (i, j) \text{ are assigned to the hubs } k \text{ and } m \\ 0 & \text{otherwise} \end{cases}$$

lk_k : An integer variable indicating the number of required servers in the origin hub node k .

lm_m : An integer variable indicating the number of required servers in the destination hub node m .

$\pi k_{0,k}$: The idle probability of the open origin hub facility k .

$\pi m_{0,m}$: The idle probability of the open destination hub facility m .

Wk_k : The expected waiting time of the commodities at the open origin hub facility k .

Wm_m : The expected waiting time of the commodities at the open destination hub facility m .

The mathematical formulation

A node identification method is applied in the proposed model to distinguish between rail and road nodes. In this method, while the road nodes are numbered from 1 to C , and the rail nodes are numbered from $C + 1$ to $2C$, where C is the number of all cities in the network under investigation. As the fixed cost of opening a rail hub node is more than that of a road hub node, a multiplier $\nu k_k, \nu m_m$ is used to calculate the fixed establishment costs of both types of hub nodes. This multiplier takes the value of 1 for the road nodes and 1.5 for the intermodal nodes (Ishfaq and Sox 2010). Note that a single objective of minimizing the waiting times in both origin and destination hub layers will probably leads into an increased number of servers used in each hub node with higher fixed cost and higher idle time. Therefore, minimizing the cost is considered as another objective function. These objective functions are clearly in conflict with each other.

The mathematical formulation of the intermodal hub-location-allocation problem is as follows.

$$\text{Min } f_1 = \sum_k^N \left[\lambda k_k \times Wk_k + (1 - Ak_k) \times \frac{lk_k}{\mu k_k} \right] + \sum_m^N \left[\lambda m_m \times Wm_m + (1 - Am_m) \times \frac{lm_m}{\mu m_m} \right] \quad (1)$$

$$\text{Min } f_2 = \sum_i^N \sum_j^N \sum_k^N \sum_m^N f_{ij} \times x_{ijkm} \times (C1_{ik} + \alpha \times C2_{km} + C3_{mj}) + \sum_k^N \nu k_k \times Fk_k \times yk_k + lk_k \times ck_k + \sum_m^N \nu m_m \times Fm_m \times ym_m + lm_m \times cm_m \quad (2)$$

Subject to:

$$\pi k_{0k} = \frac{yk_k}{\left(\sum_{s=0}^{lk_k-1} \frac{Ak_k^s}{s!} + \left(\frac{Ak_k^{lk_k}}{lk_k!} \times \frac{lk_k}{(lk_k - Ak_k)} \right) \right)}; \quad \forall k \in N \quad (3)$$

$$\pi m_{0m} = \frac{ym_m}{\left(\sum_{s=0}^{lm_m-1} \frac{Am_m^s}{s!} + \left(\frac{Am_m^{lm_m}}{lm_m!} \times \frac{lm_m}{(lm_m - Am_m)} \right) \right)}; \quad \forall m \in N \quad (4)$$

$$Wk_k = \frac{Ak_k^{lk_k} \times \mu k_k \times lk_k \times \pi k_{0k} + yk_k}{lk_k \times (\mu k_k \times lk_k - \lambda k_k)^2 + \mu k_k}; \quad \forall k \in N \quad (5)$$

$$Wm_m = \frac{Am_m^{lm_m} \times \mu m_m \times lm_m \times \pi m_{0m} + ym_m}{lm_m \times (\mu m_m \times lm_m - \lambda m_m)^2 + \mu m_m}; \quad \forall m \in N \quad (6)$$

$$\sum_k^N yk_k = P_1; \quad (7)$$

$$\sum_m^N ym_m = P_2; \quad (8)$$

$$\sum_{\substack{k \neq i \\ k \neq i\text{-city} \\ k \neq i\text{+city}}}^N \sum_{\substack{m \neq i \\ m \neq i\text{-city} \\ m \neq i\text{+city}}}^N x_{ijkm} = 1; \quad \forall i \cdot j \in N \text{ and } \{i \neq j \ \& \ i + |C| \neq j \ \& \ i - |C| \neq j\} \quad (9)$$

$$\sum_m^N x_{ijkm} \leq yk_k; \quad \forall i \cdot j \cdot k \in N \quad (10)$$

$$\sum_k^N x_{ijkm} \leq ym_m; \quad \forall i \cdot j \cdot m \in N \quad (11)$$

$$yk_k + yk_{k+|C|} \leq 1; \quad \forall k \in N, k \leq |C| \quad (12)$$

$$ym_m + ym_{m+|C|} \leq 1; \quad \forall m \in N, m \leq |C| \quad (13)$$

$$yk_i + ym_i \leq 1; \quad \forall i \in N \quad (14)$$

$$Ak_k \times \left(\sum_i^N \sum_j^N \sum_m^N x_{ijkm} \right) \leq lk_k; \quad \forall k \in N \quad (15)$$

$$Am_m \times \left(\sum_i^N \sum_j^N \sum_k^N x_{ijkm} \right) \leq lm_m; \quad \forall m \in N \quad (16)$$

$$lk_k \leq uk_k \times yk_k; \quad \forall k \in N \quad (17)$$

$$lm_m \leq um_m \times ym_m; \quad \forall m \in N \quad (18)$$

$$x_{ijkm} = 0; \quad \forall i \cdot j \cdot k \cdot m \in N \text{ and } \{k = m\} \quad (19)$$

$$x_{ijkm} = 0; \quad \forall i \cdot j \cdot k \cdot m \in N \text{ and } \{i = j\} \quad (20)$$

$$yk_k \in \{0.1\}; \quad \forall k \in N \quad (21)$$

$$ym_m \in \{0.1\}; \quad \forall m \in N \quad (22)$$

$$x_{ijkm} \in \{0.1\}; \quad \forall i \cdot j \cdot k \cdot m \in N \quad (23)$$

$$lk_k \geq 0 \cdot \text{integer}; \quad \forall k \in N \quad (24)$$

$$lm_m \geq 0 \cdot \text{integer}; \quad \forall m \in N \quad (25)$$

$$\pi k_{0k} \geq 0 \cdot \text{positive}; \quad \forall k \in N \quad (26)$$

$$\pi m_{0m} \geq 0 \cdot \text{positive}; \quad \forall m \in N \quad (27)$$

$$Wk_k \geq 0 \cdot \text{positive}; \quad \forall k \in N \quad (28)$$

$$Wm_m \geq 0 \cdot \text{positive}; \quad \forall m \in N \quad (29)$$

The first objective function in (1) minimizes the total waiting times of the commodities alongside the servers' total idle times in both origin and destination layers. Some of the terms in this equation are obtained using the steady-state performance measures of M/M/c queuing systems discussed later in this paragraph. The second objective function in (2) minimizes the total transportation costs for all origin–destination flows along with the fixed costs of establishing hub nodes. The fixed cost of the hub nodes includes costs of establishing the hub nodes and the staffing costs. Constraints (3) and (4) calculate the idle probabilities of open origin and open destination hub facilities, respectively (Gross and Harris 1998). Constraints (5) and (6) calculate the waiting times of the commodities at the origin or destination hub facilities, respectively (Gross and Harris 1998). Constraints (7) and (8) ensure that exactly P_1 and P_2 hub facilities are used in origin and destination hub layers, respectively. Constraint (9) ensures that every origin–destination pair is assigned to the hub pairs. Constraints (10) and (11) guarantee that an origin–destination flow can be assigned to nodes k and m if the origin and destination hub facilities are located at nodes k and m , respectively. Constraints (12) and (13) forces one node in each city c' to be selected as a hub origin or destination node. Constraint (14) forces each node to be selected as either one origin or one destination hub node. Constraints (15) and (16) imply that the servers' service rates must be greater than the arrival rates of the commodities at all origin and destination hub facilities. Constraints (17) and (18) restrict the number of servers at every origin and destination hub node, respectively. Constraint (19) guarantees that hub pairs cannot be the same node. Constraint (20) guarantee that one node cannot be considered as both origin and destination node, indeed there is no loop. Constraints (21)–(29) define the types of the variables used in the proposed formulation.

Model validation

A small-size problem with eight cities and two road and train nodes is solved in this section to validate the accuracy of the mathematical formulation developed in “[The problem and its formulation](#)” section. This problem is solved using the GAMS software, where the optimal solution is shown in Table 1. In addition, the optimal locations of the hub nodes along with the spoke nodes connected to them are schematically shown in Fig. 1. It should be noted that the required computational time of solving this problem on a PC with core i5 CPU and 4 GB RAM was 19,052.31 s. This fact somehow shows that the intricacy degree of the problem at hand is too high so as its large instances cannot be solved using an exact solution algorithm in a reasonable computational time. As such, one needs to utilize a meta-heuristic to find near optimal solutions of the problems.

Solving methodologies

As the multi-objective intermodal hub location-allocation problem modeled in “[Model validation](#)” section is strictly NP-hard, a population-based meta-heuristic optimization algorithm called multi-objective invasive weeds optimization (MOIWO) algorithm is utilized in this section to solve the problem. In addition, in order to evaluate the performance of the MOIWO, two other population-based multi-objective optimization algorithms including a non-dominated ranking genetic algorithm (NRGA) and a non-dominated sorting genetic algorithm (NSGA-II) are used.

The multi-objective invasive weeds optimization (MOIWO) algorithm

The MOIWO algorithm has been presented by Kundu et al. (2011). As an initial population in an IWO algorithm, a set of weeds are generated randomly. Each weed is a plant that grows globally in a specified region so that it cannot be controlled or removed by a human being. A popular claim about weed is that they always are a winner and the more farmers try, the more they grow on the land. The pruning system of the weeds is the base for the weed's behavior to occupy soil and to generate new colony. Using this system, they initially find appropriate regions to attack, to invade and generate new colonies. Consequently, new plants are generated to use appropriate regions.

The invasive weeds optimization (IWO) algorithm simulates the colonizing behavior of the weeds. This behavior can be used as a method to search the solution space. In this method, the seeds that are scattered in a specified region will turn in weeds, where they grow better

Table 1 Result of solving a small test problem on the GAMS software

	Solver 1	Solver 2
f_1	$Z_1^* = 6.237$	6.593
f_2	129,070.52	$Z_2^* = 74,557.768$
yk_k	City 5 = $yk_5 = 1$ City 4 = $yk_{12} = 1$ City 7 = $yk_{15} = 1$	City 3 = $yk_3 = 1$ City 4 = $yk_{12} = 1$ City 6 = $yk_{14} = 1$
ym_m	City 8 = $ym_8 = 1$ City 1 = $ym_9 = 1$ City 2 = $ym_{10} = 1$	City 4 = $ym_4 = 1$ City 6 = $ym_6 = 1$ City 3 = $ym_{11} = 1$
lk_k	City 5 = $lk_5 = 17$ City 4 = $lk_{12} = 100$ City 7 = $lk_{15} = 100$	City 3 = $lk_3 = 98$ City 4 = $lk_{12} = 56$ City 6 = $lk_{14} = 58$
lm_m	City 8 = $lm_8 = 100$ City 1 = $lm_9 = 100$ City 2 = $lm_{10} = 17$	City 4 = $lm_4 = 51$ City 6 = $lm_6 = 52$ City 3 = $lm_{11} = 100$
πk_{0k}	City 5 = $\pi k_{05} = 0.681$ City 4 = $\pi k_{012} = 0.615$ City 7 = $\pi k_{015} = 0.614$	City 3 = $\pi k_{03} = 0.644$ City 4 = $\pi k_{012} = 0.615$ City 6 = $\pi k_{014} = 0.646$
πm_{0m}	City 8 = $\pi m_{08} = 0.615$ City 1 = $\pi m_{09} = 0.614$ City 2 = $\pi m_{010} = 0.681$	City 4 = $\pi m_{04} = 0.725$ City 6 = $\pi m_{06} = 0.648$ City 3 = $\pi m_{011} = 0.685$
Wk_k	City 5 = $wk_5 = 0.032$ City 4 = $wk_{12} = 0.034$ City 7 = $wk_{15} = 0.033$	City 3 = $wk_3 = 0.031$ City 4 = $wk_{12} = 0.034$ City 6 = $wk_{14} = 0.032$
Wm_k	City 8 = $wm_8 = 0.034$ City 1 = $wm_9 = 0.033$ City 2 = $wm_{10} = 0.032$	City 4 = $wm_4 = 0.032$ City 6 = $wm_6 = 0.033$ City 3 = $wm_{11} = 0.031$
City 1, node1; $x_{1jkm}=1$	node1.node2.node12.node8 = 1, node1.node3.node15.node8 = 1, node1.node4.node15.node8 = 1, node1.node5.node15.node8 = 1, node1.node6.node12.node8 = 1, node1.node7.node12.node8 = 1, node1.node8.node12.node10 = 1, node1.node10.node15.node8 = 1, node1.node11.node15.node8 = 1, node1.node12.node15.node8 = 1, node1.node13.node15.node10 = 1, node1.node14.node12.node8 = 1, node1.node15.node12.node8 = 1, node1.node16.node15.node10 = 1;	node1.node2.node3.node11 = 1, node1.node3.node12.node4 = 1, node1.node4.node3.node11 = 1, node1.node5.node3.node11 = 1, node1.node6.node3.node11 = 1, node1.node7.node12.node4 = 1, node1.node8.node12.node4 = 1, node1.node10.node3.node11 = 1, node1.node11.node12.node4 = 1, node1.node12.node3.node11 = 1, node1.node13.node3.node11 = 1, node1.node14.node3.node11 = 1, node1.node15.node3.node11 = 1, node1.node16.node3.node11 = 1;



Table 1 (continued)

	Solver 1	Solver 2
City 2, node2; $x_{2jkm}=1$	node2.node1.node12.node8 = 1, node2.node3.node15.node9 = 1, node2.node4.node5.node8 = 1, node2.node5.node15.node8 = 1, node2.node6.node12.node8 = 1, node2.node7.node12.node8 = 1, node2.node8.node15.node9 = 1, node2.node9.node15.node8 = 1, node2.node11.node12.node8 = 1, node2.node12.node15.node8 = 1, node2.node13.node15.node9 = 1, node2.node14.node12.node8 = 1, node2.node15.node12.node8 = 1, node2.node16.node15.node9 = 1;	node2.node1.node3.node11 = 1, node2.node3.node14.node6 = 1, node2.node4.node3.node11 = 1, node2.node5.node3.node11 = 1, node2.node6.node3.node11 = 1, node2.node7.node3.node4 = 1, node2.node8.node3.node11 = 1, node2.node9.node3.node11 = 1, node2.node11.node14.node6 = 1, node2.node12.node3.node11 = 1, node2.node13.node3.node11 = 1, node2.node14.node3.node11 = 1, node2.node15.node3.node11 = 1, node2.node16.node3.node11 = 1,
City 3, node3; $x_{3jkm} = 1$	node3.node1.node12.node8 = 1, node3.node2.node15.node8 = 1, node3.node4.node15.node9 = 1, node3.node5.node12.node10 = 1, node3.node6.node12.node9 = 1, node3.node7.node12.node9 = 1, node3.node8.node15.node9 = 1, node3.node9.node12.node8 = 1, node3.node10.node12.node8 = 1, node3.node12.node5.node9 = 1, node3.node13.node15.node9 = 1, node3.node14.node15.node8 = 1, node3.node15.node5.node9 = 1, node3.node16.node15.node9 = 1;	node3.node1.node12.node4 = 1, node3.node2.node14.node6 = 1, node3.node4.node14.node6 = 1, node3.node5.node14.node6 = 1, node3.node6.node12.node4 = 1, node3.node7.node14.node6 = 1, node3.node8.node14.node6 = 1, node3.node9.node12.node4 = 1, node3.node10.node14.node6 = 1, node3.node12.node14.node6 = 1, node3.node13.node14.node6 = 1, node3.node14.node12.node4 = 1, node3.node15.node12.node4 = 1, node3.node16.node14.node6 = 1,
City 4, node4; $x_{4jkm} = 1$	node4.node1.node15.node8 = 1, node4.node2.node15.node9 = 1, node4.node3.node15.node8 = 1, node4.node5.node15.node9 = 1, node4.node6.node15.node9 = 1, node4.node7.node5.node9 = 1, node4.node8.node15.node9 = 1, node4.node9.node15.node8 = 1, node4.node10.node5.node9 = 1, node4.node11.node15.node8 = 1, node4.node13.node15.node8 = 1, node4.node14.node15.node9 = 1, node4.node15.node5.node8 = 1, node4.node16.node15.node9 = 1;	node4.node1.node3.node11 = 1, node4.node2.node3.node11 = 1, node4.node3.node14.node6 = 1, node4.node5.node3.node11 = 1, node4.node6.node3.node11 = 1, node4.node7.node3.node11 = 1, node4.node8.node3.node11 = 1, node4.node9.node3.node11 = 1, node4.node10.node3.node11 = 1, node4.node11.node14.node6 = 1, node4.node13.node3.node11 = 1, node4.node14.node3.node11 = 1, node4.node15.node3.node11 = 1, node4.node16.node3.node11 = 1,



Table 1 (continued)

	Solver 1	Solver 2
City 5, node5; $x_{5jkm} = 1$	node5.node1.node15.node8 = 1, node5.node2.node15.node8 = 1, node5.node3.node15.node9 = 1, node5.node4.node15.node8 = 1, node5.node6.node12.node10 = 1, node5.node7.node12.node8 = 1, node5.node8.node12.node9 = 1, node5.node9.node15.node8 = 1, node5.node10.node12.node8 = 1, node5.node11.node15.node9 = 1, node5.node12.node15.node8 = 1, node5.node14.node15.node9 = 1, node5.node15.node12.node9 = 1, node5.node16.node12.node9 = 1;	node5.node1.node3.node11 = 1, node5.node2.node3.node11 = 1, node5.node3.node14.node6 = 1, node5.node4.node3.node11 = 1, node5.node6.node3.node11 = 1, node5.node7.node14.node6 = 1, node5.node8.node3.node11 = 1, node5.node9.node3.node11 = 1, node5.node10.node3.node11 = 1, node5.node11.node14.node6 = 1, node5.node12.node3.node11 = 1, node5.node14.node3.node11 = 1, node5.node15.node3.node11 = 1, node5.node16.node3.node11 = 1,
City 6, node6; $x_{6jkm} = 1$	node6.node1.node12.node8 = 1, node6.node2.node12.node9 = 1, node6.node3.node12.node8 = 1, node6.node4.node15.node8 = 1, node6.node5.node15.node8 = 1, node6.node7.node12.node9 = 1, node6.node8.node15.node9 = 1, node6.node9.node15.node8 = 1, node6.node10.node12.node9 = 1, node6.node11.node12.node8 = 1, node6.node12.node15.node9 = 1, node6.node13.node12.node10 = 1, node6.node15.node12.node8 = 1, node6.node16.node12.node10 = 1;	node6.node1.node3.node11 = 1, node6.node2.node3.node11 = 1, node6.node3.node12.node4 = 1, node6.node4.node3.node11 = 1, node6.node5.node3.node11 = 1, node6.node7.node3.node4 = 1, node6.node8.node3.node11 = 1, node6.node9.node3.node11 = 1, node6.node10.node3.node11 = 1, node6.node11.node12.node4 = 1, node6.node12.node3.node11 = 1, node6.node13.node3.node11 = 1, node6.node15.node3.node11 = 1, node6.node16.node3.node11 = 1,
City 7, node7; $x_{7jkm} = 1$	node7.node1.node12.node8 = 1, node7.node2.node12.node8 = 1, node7.node3.node12.node8 = 1, node7.node4.node5.node9 = 1, node7.node5.node12.node9 = 1, node7.node6.node12.node8 = 1, node7.node8.node12.node9 = 1, node7.node9.node12.node8 = 1, node7.node10.node12.node8 = 1, node7.node11.node12.node8 = 1, node7.node12.node5.node9 = 1, node7.node13.node12.node8 = 1, node7.node14.node5.node10 = 1, node7.node16.node12.node9 = 1;	node7.node1.node12.node4 = 1, node7.node2.node3.node11 = 1, node7.node3.node12.node4 = 1, node7.node4.node3.node11 = 1, node7.node5.node14.node6 = 1, node7.node6.node3.node11 = 1, node7.node8.node12.node4 = 1, node7.node9.node12.node4 = 1, node7.node10.node3.node11 = 1, node7.node11.node12.node4 = 1, node7.node12.node3.node11 = 1, node7.node13.node3.node11 = 1, node7.node14.node3.node11 = 1, node7.node16.node3.node11 = 1,



Table 1 (continued)

	Solver 1	Solver 2
City 8, node8; $x_{8jkm} = 1$	node8.node1.node5.node10 = 1, node8.node2.node15.node9 = 1, node8.node3.node12.node9 = 1, node8.node4.node15.node9 = 1, node8.node5.node15.node9 = 1, node8.node6.node15.node9 = 1, node8.node7.node12.node9 = 1, node8.node9.node15.node10 = 1, node8.node10.node15.node9 = 1, node8.node11.node12.node9 = 1, node8.node12.node15.node9 = 1, node8.node13.node15.node9 = 1, node8.node14.node12.node9 = 1, node8.node15.node12.node9 = 1;	node8.node1.node12.node4 = 1, node8.node2.node3.node11 = 1, node8.node3.node14.node6 = 1, node8.node4.node3.node11 = 1, node8.node5.node14.node6 = 1, node8.node6.node3.node11 = 1, node8.node7.node12.node4 = 1, node8.node9.node12.node4 = 1, node8.node10.node3.node11 = 1, node8.node11.node14.node6 = 1, node8.node12.node3.node11 = 1, node8.node13.node3.node11 = 1, node8.node14.node3.node11 = 1, node8.node15.node12.node4 = 1,
City 1, node9; $x_{9jkm} = 1$	node9.node2.node12.node8 = 1, node9.node3.node15.node8 = 1, node9.node4.node15.node8 = 1, node9.node5.node12.node8 = 1, node9.node6.node12.node10 = 1, node9.node7.node12.node8 = 1, node9.node8.node5.node10 = 1, node9.node10.node12.node8 = 1, node9.node11.node12.node8 = 1, node9.node12.node15.node8 = 1, node9.node13.node15.node8 = 1, node9.node14.node15.node8 = 1, node9.node15.node12.node8 = 1, node9.node16.node12.node10 = 1;	node9.node2.node3.node11 = 1, node9.node3.node14.node6 = 1, node9.node4.node14.node6 = 1, node9.node5.node14.node6 = 1, node9.node6.node3.node11 = 1, node9.node7.node12.node4 = 1, node9.node8.node14.node6 = 1, node9.node10.node3.node11 = 1, node9.node11.node14.node6 = 1, node9.node12.node3.node11 = 1, node9.node13.node3.node11 = 1, node9.node14.node3.node11 = 1, node9.node15.node12.node4 = 1, node9.node16.node3.node11 = 1,
City 2, node10; $x_{10jkm} = 1$	node10.node1.node15.node8 = 1, node10.node3.node15.node9 = 1, node10.node4.node15.node8 = 1, node10.node5.node12.node9 = 1, node10.node6.node12.node9 = 1, node10.node7.node12.node9 = 1, node10.node8.node12.node9 = 1, node10.node9.node15.node8 = 1, node10.node11.node12.node9 = 1, node10.node12.node15.node9 = 1, node10.node13.node12.node8 = 1, node10.node14.node12.node9 = 1, node10.node15.node12.node9 = 1, node10.node16.node15.node9 = 1;	node10.node1.node12.node4 = 1, node10.node3.node12.node4 = 1, node10.node4.node3.node11 = 1, node10.node5.node12.node11 = 1, node10.node6.node12.node4 = 1, node10.node7.node12.node4 = 1, node10.node8.node12.node4 = 1, node10.node9.node12.node4 = 1, node10.node11.node12.node4 = 1, node10.node12.node3.node11 = 1, node10.node13.node12.node11 = 1, node10.node14.node12.node4 = 1, node10.node15.node12.node4 = 1, node10.node16.node12.node4 = 1,



Table 1 (continued)

	Solver 1	Solver 2
City 3, node11; $x_{11jkm} = 1$	node11.node1.node12.node10 = 1, node11.node2.node12.node8 = 1, node11.node4.node15.node9 = 1, node11.node5.node15.node9 = 1, node11.node6.node5.node10 = 1, node11.node7.node12.node9 = 1, node11.node8.node15.node9 = 1, node11.node9.node12.node8 = 1, node11.node10.node12.node9 = 1, node11.node12.node15.node8 = 1, node11.node13.node15.node8 = 1, node11.node14.node15.node8 = 1, node11.node15.node5.node8 = 1, node11.node16.node15.node9 = 1;	node11.node1.node12.node4 = 1, node11.node2.node14.node6 = 1, node11.node4.node14.node6 = 1, node11.node5.node14.node6 = 1, node11.node6.node12.node4 = 1, node11.node7.node12.node4 = 1, node11.node8.node12.node4 = 1, node11.node9.node12.node4 = 1, node11.node10.node14.node6 = 1, node11.node12.node14.node6 = 1, node11.node13.node14.node6 = 1, node11.node14.node12.node4 = 1, node11.node15.node12.node4 = 1, node11.node16.node12.node4 = 1,
City 4, node12; $x_{12jkm} = 1$	node12.node1.node15.node8 = 1, node12.node2.node15.node8 = 1, node12.node3.node15.node9 = 1, node12.node5.node15.node8 = 1, node12.node6.node15.node8 = 1, node12.node7.node5.node8 = 1, node12.node8.node15.node9 = 1, node12.node9.node15.node8 = 1, node12.node10.node15.node9 = 1, node12.node11.node15.node9 = 1, node12.node13.node15.node9 = 1, node12.node14.node15.node8 = 1, node12.node15.node5.node8 = 1, node12.node16.node15.node9 = 1;	node12.node1.node14.node11 = 1, node12.node2.node14.node6 = 1, node12.node3.node14.node6 = 1, node12.node5.node14.node6 = 1, node12.node6.node3.node11 = 1, node12.node7.node14.node6 = 1, node12.node8.node14.node6 = 1, node12.node9.node14.node6 = 1, node12.node10.node14.node6 = 1, node12.node11.node14.node6 = 1, node12.node13.node14.node11 = 1, node12.node14.node3.node11 = 1, node12.node15.node14.node6 = 1, node12.node16.node14.node11 = 1,
City 5, node13; $x_{13jkm} = 1$	node13.node1.node12.node8 = 1, node13.node2.node12.node8 = 1, node13.node3.node15.node8 = 1, node13.node4.node15.node8 = 1, node13.node6.node12.node9 = 1, node13.node7.node12.node9 = 1, node13.node8.node15.node9 = 1, node13.node9.node12.node8 = 1, node13.node10.node15.node9 = 1, node13.node11.node12.node9 = 1, node13.node12.node15.node8 = 1, node13.node14.node15.node9 = 1, node13.node15.node12.node9 = 1, node13.node16.node12.node9 = 1;	node13.node1.node12.node4 = 1, node13.node2.node3.node11 = 1, node13.node3.node12.node4 = 1, node13.node4.node3.node11 = 1, node13.node6.node12.node4 = 1, node13.node7.node12.node4 = 1, node13.node8.node12.node4 = 1, node13.node9.node12.node4 = 1, node13.node10.node3.node11 = 1, node13.node11.node12.node4 = 1, node13.node12.node3.node11 = 1, node13node14.node12.node4 = 1, node13.node15.node12.node4 = 1, node13.node16.node12.node4 = 1,



Table 1 (continued)

	Solver 1	Solver 2
City 6, node14; $x_{14jkm}=1$	node14.node1.node15.node8 = 1, node14.node2.node12.node8 = 1, node14.node3.node15.node8 = 1, node14.node4.node15.node8 = 1, node14.node5.node15.node9 = 1, node14.node7.node12.node9 = 1, node14.node8.node12.node9 = 1, node14.node9.node15.node8 = 1, node14.node10.node12.node9 = 1, node14.node11.node12.node9 = 1, node14.node12.node15.node8 = 1, node14.node13.node15.node9 = 1, node14.node15.node12.node9 = 1, node14.node16.node12.node9 = 1;	node14.node1.node3.node11 = 1, node14.node2.node3.node11 = 1, node14.node3.node12.node4 = 1, node14.node4.node3.node11 = 1, node14.node5.node3.node11 = 1, node14.node7.node12.node4 = 1, node14.node8.node3.node11 = 1, node14.node9.node12.node4 = 1, node14.node10.node3.node11 = 1, node14.node11.node12.node4 = 1, node14.node12.node3.node11 = 1, node14.node13.node3.node11 = 1, node14.node15.node3.node11 = 1, node14.node16.node3.node11 = 1,
City 7, node15; $x_{15jkm}=1$	node15.node1.node12.node8 = 1, node15.node2.node12.node8 = 1, node15.node3.node12.node8 = 1, node15.node4.node5.node8 = 1, node15.node5.node12.node8 = 1, node15.node6.node12.node8 = 1, node15.node8.node12.node9 = 1, node15.node9.node12.node10 = 1, node15.node10.node12.node9 = 1, node15.node11.node12.node9 = 1, node15.node12.node5.node9 = 1, node15.node13.node12.node8 = 1, node15.node14.node12.node8 = 1, node15.node16.node12.node9 = 1;	node15.node1.node14.node11 = 1, node15.node2.node14.node6 = 1, node15.node3.node14.node6 = 1, node15.node4.node14.node6 = 1, node15.node5.node14.node6 = 1, node15.node6.node3.node11 = 1, node15.node8.node14.node6 = 1, node15.node9.node14.node6 = 1, node15.node10.node14.node6 = 1, node15.node11.node14.node6 = 1, node15.node12.node14.node11 = 1, node15.node13.node14.node11 = 1, node15.node14.node3.node11 = 1, node15.node16.node14.node11 = 1;
City 8, node16; $x_{16jkm}=1$	node16.node1.node15.node10 = 1, node16.node2.node12.node9 = 1, node16.node3.node5.node9 = 1, node16.node4.node15.node9 = 1, node16.node5.node12.node9 = 1, node16.node6.node12.node9 = 1, node16.node7.node12.node9 = 1, node16.node9.node15.node10 = 1, node16.node10.node15.node9 = 1, node16.node11.node15.node9 = 1, node16.node12.node15.node9 = 1, node16.node13.node15.node9 = 1, node16.node14.node12.node9 = 1, node16.node15.node12.node9 = 1;	node16.node1.node12.node4 = 1, node16.node2.node14.node6 = 1, node16.node3.node14.node6 = 1, node16.node4.node14.node6 = 1, node16.node5.node14.node6 = 1, node16.node6.node3.node11 = 1, node16.node7.node14.node6 = 1, node16.node9.node12.node4 = 1, node16.node10.node3.node11 = 1, node16.node11.node14.node6 = 1, node16.node12.node3.node11 = 1, node16.node13.node3.node11 = 1, node16.node14.node3.node11 = 1, node16.node15.node14.node6 = 1;



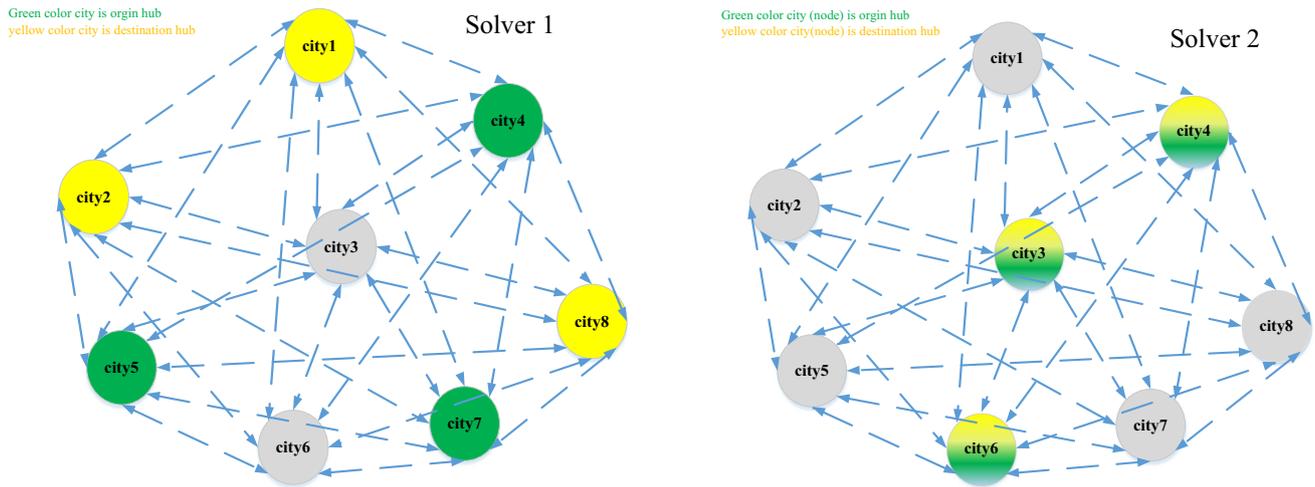


Fig. 1 Schematic representation of the solution of a small-size problem on the GAMS software (solver 1 and 2)

in a more fertile region and hence will have higher competency and will have more chance to survive. Thus, higher breeding is reached in the vicinity of these weeds. An important point is that as the number of iterations increases, the distance of the generated weeds from their parent weed will decrease remarkably. This process that is called competitive elimination will continue until better weeds are obtained.

The following steps are taken in the IWOA of this paper:

- Generate an initial random population and evaluate its fitness using the objective functions.
- Use a fuzzy sorting method (explained in the next section) to prioritize the population.
- Permit all members of the population to generate several seeds with better properties. The seeds will be produced based on Eq. (30).

$$seeds_i = \text{floor} \left(S_{\min} + (S_{\max} - S_{\min}) \times \left(\frac{np - rank_i}{np} \right) \right), \quad (30)$$

where the grade of i th population member, the number of seeds generated by i th population member, the maximum number of seeds generated by a weed, the minimum number of seeds generated by every weed, and the population number are shown, respectively, by $rank_i$, $seeds_i$, S_{\max} , S_{\min} , and np .

- Breed based on the competency and the updated standard deviation. The corresponding value of the standard deviation is calculated by Eq. (31).

$$\sigma_{\text{iter}} = (\text{iter}_{\max} - \text{iter})^n \times \frac{\sigma_{\text{initial}} - \sigma_{\text{final}}}{(\text{iter}_{\max})^n} + \sigma_{\text{final}}, \quad (31)$$

where nonlinear modulation index, the standard deviation of the seeds in each iteration, the maximum number of iterations, initial, and final standard deviation are denoted, respectively, by n , σ_{iter} , iter_{\max} , σ_{initial} and σ_{final} .

- Competitive elimination.
- Control the stopping criterion.

The pseudocode of the proposed MOIWO algorithm is presented below:

1. Assess proportion of members of np population.
2. Use the fuzzy sorting method to prioritize the population.
3. Let all members of the population make several seeds. More proportionate members of the population make a greater number of seeds in accordance with Eq. (30).
4. Seeds that have been produced are scattered on the search space using a normal random variable with mean zero and varying standard deviations. The corresponding value of the standard deviation is obtained using Eq. (31).
5. When the population size of the weeds becomes greater than the upper limit (p_{\max}), apply the fuzzy sorting method and keep p_{\max} seeds with better competencies.
6. Resume until the stopping criterion is met.

Fuzzy dominance-based sorting

In order to utilize the multi-objective version of the IWOA discussed in “The multi-objective invasive weeds optimization (MOIWO) algorithm” section, we start with an elementary phase of routine using the fuzzy dominance of the solutions in the population, based on which the solutions are first sorted in ascending order. Then, regarding the

crowding distance, the diversity is obtained through sorted solutions. Obtaining the greatest n -dimensional hyper cube over objective space per solution enables the hypercube to find similar solutions. This is achieved using Eq. (32) (Kundu et al. 2011).

$$I(\vec{v}) = \sum_{i=1}^n (y_i(\vec{u}) - y_i(\vec{w})) / (\max(y_i) - \min(y_i)). \quad (32)$$

In this equation, the neighboring solutions of \vec{v} are demonstrated by \vec{u} and \vec{w} , where the combined population is sorted in ascending order based on the i th objective function y_i . In Eq. (32), the dominator is used for normalization and the boundary solutions in the denominator include the extremes of the assigned values. Clearly, a higher value of $I(\vec{v})$ indicates more dispersal over the solution space. The non-dominated solutions are placed in the archive alongside those solutions which were placed in sparser region.

The pseudocode of utilized MOIWA using the fuzzy dominance method for each solution is presented in Algorithm 1. In this algorithm, the membership degree of i^{th} population member's domination to other weeds and membership degree of weed i to weed k are, respectively, denoted by $\mu(i)$ and $\mu_i^{\text{dom}}(\vec{x}_k \prec_i^F \vec{x}_i)$.

```

For k = 1: n //calculate fuzzy J-dominance of the solution//
  μ(k) = 0;
  For i = 1: n
    μ(i) = 1;
    For j = 1: m // compute fuzzy J-dominance of the solution//
      if y_j(̄x_i) - y_j(̄x_k) < 0
        μ_i^dom(̄x_k <_i^F ̄x_i) = 0
      else if y_j(̄x_i) - y_j(̄x_k) < p_j
        μ_i^dom(̄x_k <_i^F ̄x_i) = (y_j(̄x_i) - y_j(̄x_k)) / p_j
      else
        μ_i^dom(̄x_k <_i^F ̄x_i) = 1
    end
    μ^dom(̄x_k <_i^F ̄x_i) = μ^dom(̄x_k <_i^F ̄x_i) × μ_i^dom(̄x_k <_i^F ̄x_i)
  end
  μ(k) = μ(k) + μ^dom(̄x_k <_i^F ̄x_i) - μ(k) × μ^dom(̄x_k <_i^F ̄x_i)
end
end

```

Algorithm1. The pseudo code of the MOIWO algorithm

The coding method

To avoid generating infeasible solutions, a new structure is designed for a chromosome to assure satisfaction of all constraints all the times. This structure consists of six vectors. The lengths of these vectors represent the number of origin nodes, the number of destination nodes, the

number of servers located in origin hub nodes, the number of servers located in destination hub nodes, the number of origin nodes allocated to origin hub nodes, and the number of destination nodes allocated to the destination hub nodes. An instance of this structure is shown in Fig. 2, where Vector 1 is a 1×5 vector of origin hub nodes demonstrating origin hub node's assignment. Vector 2 is a 1×5 vector of destination hub nodes indicating destination hub node's assignment. Vector 3 is a 1×5 vector of the servers located in the origin hub nodes. Vector 4 is a 1×5 vector of the servers located in destination hub nodes. Vector 5 is a 1×5 vector of origin nodes allocated to origin hub nodes, and Vector 6 is a 1×5 vector of destination nodes allocated to destination hub nodes. The elements of all the above vectors are random samples taken from a uniform distribution in $[0, 1]$.

The decoding method

An example of the decoding method is demonstrated in this section to show the way the constraints are satisfied. Assuming the numbers of origin and destination nodes both are 5 and that the maximum number of origin and destination hub nodes is 2, the following procedure is used to generate feasible solutions with respect to Constraint (7).

First, the elements of Vector 1 in Fig. 3 are sorted in ascending order. Then, the lowest two elements are considered as open (active) origin hub nodes. This method is shown in Fig. 4, in which the origin hubs 2 and 5 are chosen as the active origin hub nodes.

Similarly, the elements of Vector 2 are first sorted in ascending order. Then, the lowest two elements are chosen

Fig. 2 Representation of the chromosome structure

Layer 1:origin hub nodes	0.8147	0.2785	0.9572	0.7922	0.6778
Layer 2:destination hub nodes	0.9058	0.5469	0.4854	0.9595	0.7577
Layer 3: servers in origin hub	0.1270	0.9575	0.8003	0.6557	0.7431
Layer 4: servers in destination hub	0.9134	0.9649	0.1419	0.0357	0.3922
Layer 5: allocated to origin hub	0.6324	0.1576	0.4218	0.8491	0.6555
Layer 6: allocated to destination hub	0.0975	0.9706	0.9157	0.9340	0.1712

Fig. 3 Decoding active origin hub nodes

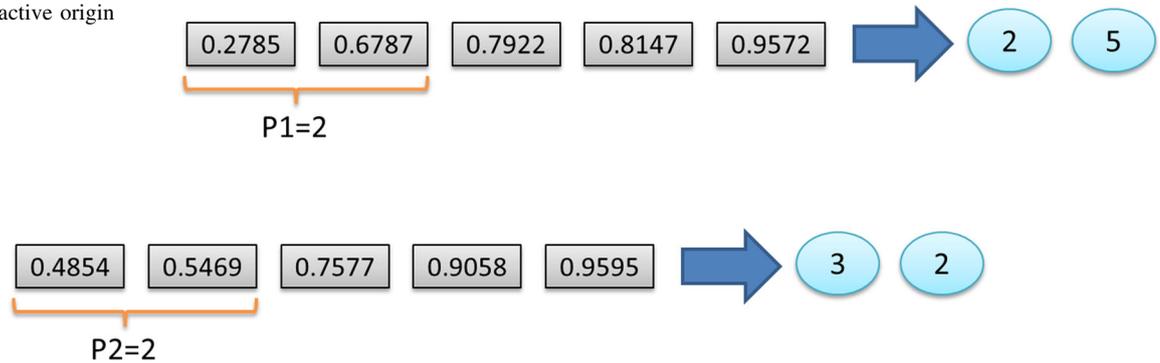
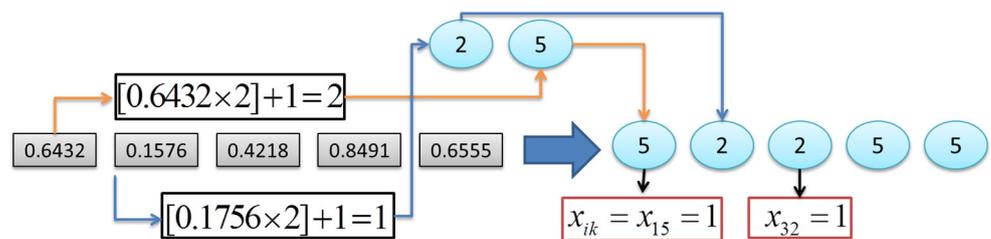


Fig. 4 Decoding active destination hub nodes

Fig. 5 Allocating origin nodes to origin hub nodes



as open destination hub nodes in order to generate the solutions that satisfy Constraint (8). According to the procedure shown in Fig. 4, destination hub nodes 3 and 2 are selected as active destination hub facilities.

Vector 5 is used to satisfy Constraints (9) and (10) to determine the origin nodes that are allocated to origin hub nodes. As shown in Fig. 5, the elements of Vector 5 are first multiplied by P_1 and are added by 1, where the resulted values are rounded down. If the resulted number becomes 1, then the origin node will be assigned to the origin hub node 2, which is shown as an active origin hub node in Fig. 3. However, if the resulted number becomes 2, then the origin node will be assigned to the origin hub node 5, which is also shown as an active origin hub node in Fig. 3.

Similarly, in order to generate solutions that satisfy Constraints (9) and (11) to determine the origin hub nodes allocated to destination hub nodes, the elements of Vector

6 are first multiplied by P_2 and then are added by 1, where resulted values are rounded down. If the resulted number becomes 1, then the origin hub node is assigned to destination hub node 3, which is shown as an active destination hub node in Fig. 4. However, if the resulted number becomes 2, then the origin hub node is assigned to destination hub node 2. This procedure is shown in Fig. 6.

Vector 3 is used to determine the number of allocated servers to active hubs. As an obvious fact, servers can only be located in active origin hub nodes. Moreover, the lower and the upper bounds on the number of servers located in each active origin hub node can be obtained using Constraints (15) and (17). Hence, Eq. (33) is used to determine the exact number of the servers that should be located in each origin hub node

$$l_k = \text{round}(\text{LB}_k + x_k \times (\text{UB}_k - \text{LB}_k)), \tag{33}$$

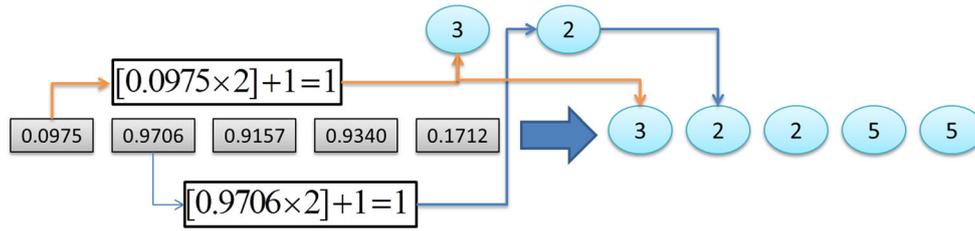


Fig. 6 Allocating origin hub nodes to destination hub nodes

Fig. 7 Encoding the number of servers located in active origin hub facilities

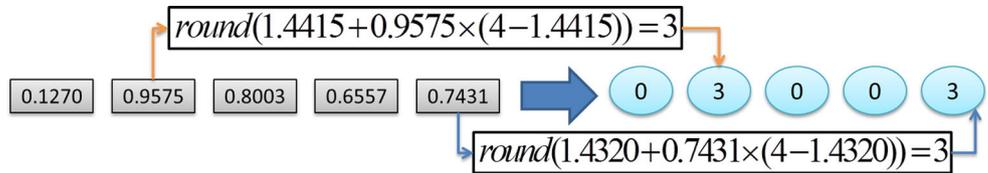
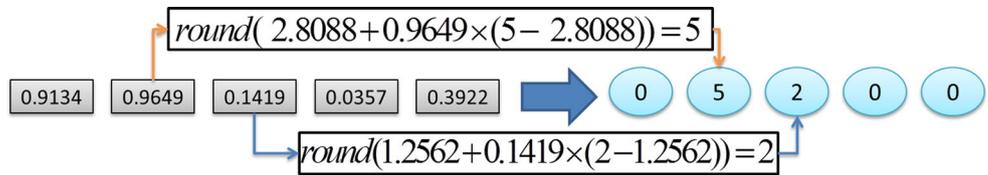


Fig. 8 Encoding the number of servers located in active destination hub facilities



where, LB_k and UB_k are the lower and upper bounds of the active number of servers located in origin hub k , l_k is the number of servers located in origin hub k and x_k is the k th element of the Vector 3. The procedure used to determine the number of servers that are located in every open origin hub is presented schematically in Fig. 7.

Similarly, Vector 4 is used to identify the number of the servers assigned to each active destination hub node. To do this, Constraints (16) and (18) are used to determine the lower and the upper bounds on the number of servers that should be located in each active destination hub node. The exact number of these servers is obtained using Eq. (34),

$$l_m = \text{round}(LB_m + x_m \times (UB_m - LB_m)), \tag{34}$$

where LB_m and UB_m represent the lower and the upper bounds on the number of active servers located in destination hub m , l_m is the number of servers located in destination hub m , and x_m is the m th element of Vector 4. The procedure of satisfying Constraints (16) and (18) and determining the number of servers assigned to each active destination hub node is depicted in Fig. 8.

NRGA and NSGA-II

The chromosome structure of both NSGA-II (Deb et al. 2002) and NRGA (Al Jadaan et al. 2008) algorithms is the one proposed for the MOIWO algorithm. Moreover, both NSGA-II and NRGA algorithms use the same mutation and

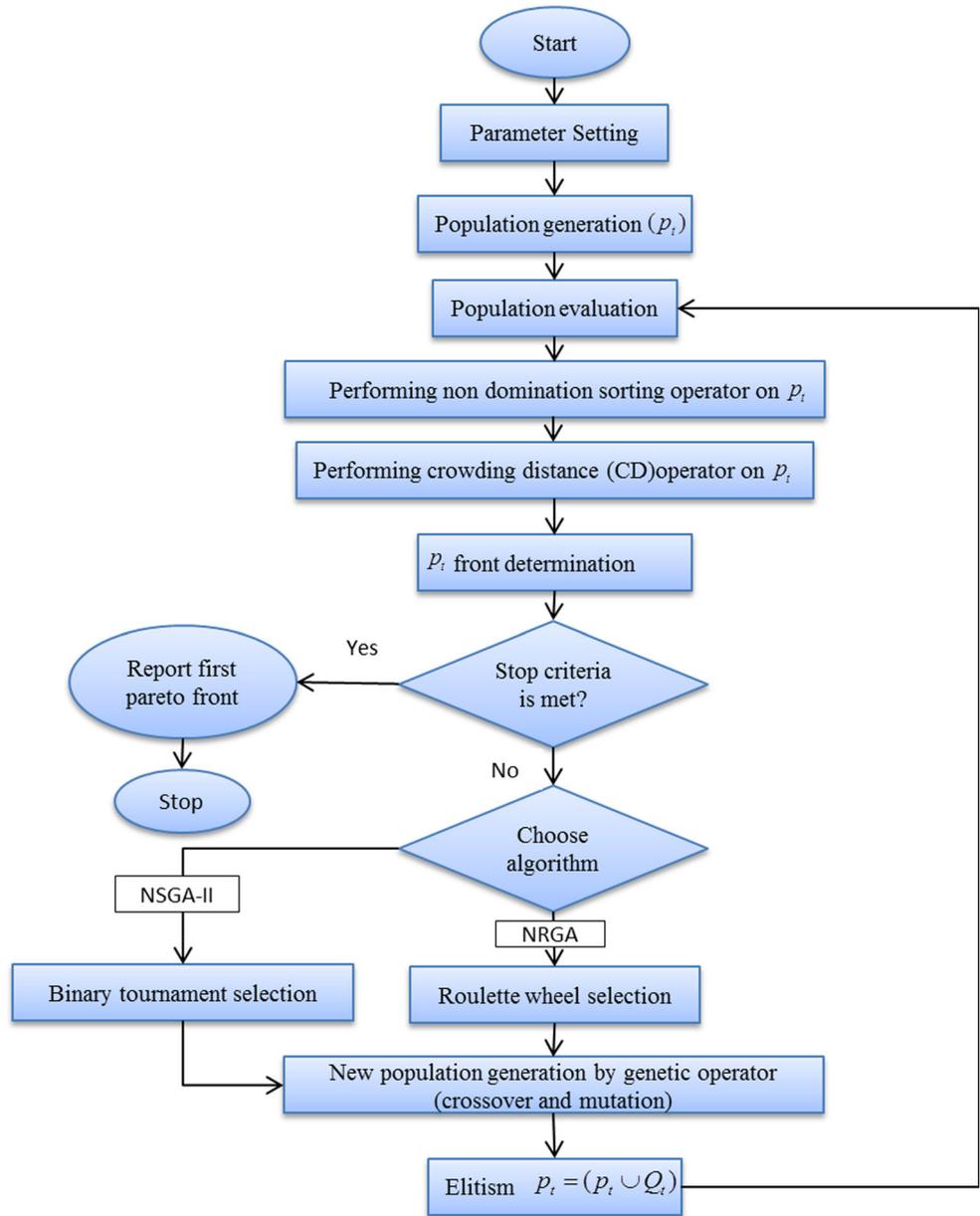
crossover operator. Their mutation operator is a combination of three operators; reversion, insertion, and swap mutations. Besides, in order to enhance the performances, a combination of three different crossover operators (single-point, double-point, and uniform crossover) is used in these two algorithms. However, as the main difference, NSGA-II employs the binary tournament selection, while NRGA uses the Roulette wheel selection strategy. The flowchart of NSGA-II and NRGA algorithms is shown in Fig. 9.

Applications

The implementation of the three utilized meta-heuristic algorithms called MOIWO, NSGA-II, and NRGA on 20 different test problems are discussed in this section. To do so, four different metrics called mean ideal distance (MID), the rate of achievement to two objectives simultaneously (RAS), the diversification matrix (DM), and the number of Pareto solutions (NPS) are used to evaluate the performances of the above algorithms. These metrics are:

- *MID* used to assess the closeness of the Pareto solutions obtained to an ideal point using Eq. (35) (Karimi et al. 2010).

Fig. 9 Flowchart of the NSGA-II and NPGA



$$MID = \frac{\sum_{i=1}^n \sqrt{\left(\frac{f1_i - f1_{best}}{f1_{total}^{max} - f1_{total}^{min}}\right)^2 + \left(\frac{f2_i - f2_{best}}{f2_{total}^{max} - f2_{total}^{min}}\right)^2}}{n}, \quad (35)$$

where, $f1_{total}^{max}$, $f2_{total}^{min}$ and n denote, respectively, the smallest and the biggest values of the two fitness functions among all non-dominated solutions and the number of non-dominated solutions.

- *RAS* Eq. (36) is used to calculate the value of this metric (Asefi et al. 2014):

$$RAS = \frac{\sum_{i=1}^n \left[\left| \frac{f1_i(x) - f1_i^{best}(x)}{f1_i^{best}(x)} \right| + \left| \frac{f2_i(x) - f2_i^{best}(x)}{f2_i^{best}(x)} \right| \right]}{n} \quad (36)$$

where the value of the first and the second objective functions of the i th non-dominated solution and the number of non-dominated solutions are, respectively, denoted by $f1_i(x)$, $f2_i(x)$ and n .

- *DM* this metric is employed to obtain the diversity of the solutions in Pareto front using Eq. (37) (Mohammadi et al. 2013).

$$DM = \sqrt{\sum_{i=1}^I (\text{Min } f_i - \text{Max } f_i)^2} \quad (37)$$

where $\text{Min } f_i$ and $\text{Max } f_i$ are the minimum and the maximum values of the two fitness functions among all non-dominated solutions attained by the algorithm, respectively.

- *NPS* this metric is used to count the number of non-dominated solutions attained by each algorithm (Sarrafha et al. 2015).

Parameter setting

Since the performance of a meta-heuristic algorithm is significantly affected by the choice of its parameters, this section is dedicated for parameter calibration. To do this, some test problems are first generated in “Randomly generated test problems” section based on their parameters randomly. Then, the calibration process is explained in “Calibrating the parameters of the algorithms” section.

Randomly generated test problems

In this section, 20 test problems are generated randomly. These problems are categorized based on number of origin nodes (I), number of origin hub nodes (K), number of destination hub nodes (M), number of destination nodes (J), number of hub facilities located at origin layer (P_1) and number of hub facilities located at destination layer (P_2). The characteristics of these problems are presented in Table 2. Moreover, the following distributions are used to set the parameters of the test problems:

- The demand for commodities from the origin node i to the destination node j follows a uniform distribution in $[100, 1000]$.
- Unit transportation costs from the origin node i to the origin hub node k , from the origin hub node k to the destination hub node m , and from the destination hub node m to the destination node j follow a uniform distribution in $[100, 1000]$.
- Unit staffing costs at the origin hub node k and the destination hub node m follow a uniform distribution in $[10, 15]$.
- Fixed establishment costs of a facility at the potential origin hub node k and the potential destination hub node m follow a uniform distribution in $[1000, 10,000]$.
- The discount factor is set to 0.3 for the costs.
- Maximum numbers of servers to be located in open origin hub node k and open destination hub node m are set to 4 and 5, respectively.
- Arrival rate of commodities to the origin hub node k follows a uniform distribution in $[10, 30]$.

Table 2 General data of the test problems

	N	C	i	j	k	m	P_1	P_2
1	1	7	14	14	14	14	2	2
2	2	9	18	18	18	18	2	3
3	3	10	20	20	20	20	2	3
4	4	12	24	24	24	24	2	3
5	5	14	28	28	28	28	3	3
6	6	15	30	30	30	30	3	4
7	7	17	34	34	34	34	3	4
8	8	19	38	38	38	38	3	4
9	9	20	40	40	40	40	4	4
10	10	22	44	44	44	44	4	4
11	11	25	50	50	50	50	5	5
12	12	26	52	52	52	52	6	6
13	13	28	56	56	56	56	6	6
14	14	30	60	60	60	60	6	6
15	15	32	64	64	64	64	7	8
16	16	34	68	68	68	68	7	8
17	17	35	70	70	70	70	7	8
18	18	37	74	74	74	74	8	9
19	19	39	78	78	78	78	8	9
20	20	40	80	80	80	80	8	9
21	21	45	90	90	90	90	10	10
22	22	50	100	100	100	100	10	10
23	23	55	110	110	110	110	12	13
24	24	60	120	120	120	120	12	13
25	25	65	130	130	130	130	12	13
26	26	70	140	140	140	140	15	15
27	27	75	150	150	150	150	15	15
28	28	80	160	160	160	160	17	18
29	29	90	180	180	180	180	20	20
30	30	100	200	200	200	200	22	23

- Arrival rate of commodities to the destination hub node m follows a uniform distribution in $[10, 20]$.
- Servers’ service rate at the origin hub node k and the destination hub node m follows a uniform distribution in $[20, 25]$.

Calibrating the parameters of the algorithms

In this section, the Taguchi method is utilized to tune the parameters of the three solution algorithms. This method has been proposed by Taguchi (1986) as an efficient alternative for the full-factorial experimental design. Taguchi method is aimed to investigate a group of decision variables using orthogonal arrays. As the main advantage, this method decreases the number of experiments required

Table 3 Parameters of the algorithm and their chosen levels

Multi-objective algorithm	Parameter	Parameter level		
		Level 1	Level 2	Level 3
NRGA	Percentage of crossover (Pc)	0.7	0.8	0.9
	Percentage of mutation (Pm)	0.05	0.1	0.15
	Number of solutions in the population (N-pop)	50	100	150
	Maximum number of iterations (max iteration)	5*N	10*N	15*N
NSGA-II	Percentage of crossover (Pc)	0.7	0.8	0.9
	Percentage of mutation (Pm)	0.05	0.1	0.15
	Number of solutions in the population (N-pop)	50	100	150
	Maximum number of Iterations (max iteration)	5*N	10*N	15*N
MOIWO	Maximum number of Iterations (max iteration)	15*N	10*N	5*N
	Number of weeds (N-weeds)	200	150	100
	P max	400	300	200
	Percentage of mutation (Pm)	0.6	0.5	0.4
	Initial value of standard deviation (initial sigma)	0.3	0.4	0.5
	Final value of standard deviation (final sigma)	0.05	0.03	0.01
	Minimum number of seeds (S min)	3	2	1
	Maximum number of seeds (S max)	10	8	5
	Nonlinear modulation index (<i>n</i>)	4	3	2

to calibrate the parameters. The main feature of these arrays is to set a family of experiences. These arrays should be designed and selected so that they can be considered as representative of all the possible arrays. Using this method, the performances of the algorithms are evaluated by a statistical measure called signal-to-noise ratio (S/N). As multi-objective algorithms are evaluated based on multi-objective measures, the metric proposed by Rahmati et al. (2013) is employed in this paper as the response for the Taguchi method. This response that is shown in Eq. (38) considers two important features of meta-heuristic algorithms called diversity and convergence, simultaneously.

$$R = \frac{MID}{DM} \quad (38)$$

In Eq. (38), MID is used as the convergence metric and DM is used as the diversity metric. In addition, as both the objective functions of the mathematical formulation of the problem are of a minimization type, “the smaller is better” is employed to select the proper levels of the parameters.

As the first step of using the Taguchi method, the parameter levels of the algorithms are defined in Table 3. In this table, the low, the medium, and the high levels of the parameters are coded as “1”, “2”, and “3”, respectively.

The experiments are designed in Minitab software. For the NRGA and NSGA-II, the L^9 design is selected (Rahmati, et al. 2013), meanwhile, the L^{27} design is chosen for

the MOIWO algorithm (Maghsoudlou, et al. 2016). The resulted response values obtained by NSGA-II and NRGA when they are employed to solve the second problem (as an example) are shown in Table 4. Moreover, Table 5 contains the response values obtained by MOIWO. Using the Taguchi method, the levels with smaller values of (S/N) are selected as the best levels of the parameters. The corresponding value of (S/N) for each algorithm is illustrated in Fig. 10, based on which the best parameter levels are presented in bold in Table 3.

Computational results

The implementation results of the three parameter-tuned meta-heuristics on test problems with the general data in Table 2 are shown in Table 6. These algorithms were coded in MATLAB software and were executed on a PC with 2 GB RAM and Dual 2 GHz CPU. The problems are classified into three groups of small, medium, and large, based on which the test problems 1 to 6 are considered small, the test problems 7–14 are considered medium, and the rest are considered large-size problems. The average of each of the four metrics defined in “Applications” section for small-, medium-, and large-size problems are shown in the bottom three rows of this table. In addition, Fig. 11 shows the Pareto fronts obtained by each algorithm for small, medium and large problems. In order to compare the



Table 4 Results to calibrate the parameters of NSGA-II and NPGA

Run order	Algorithm parameters				Response	
	P_{cr}	P_{mut}	N-pop	Max-iteration	NRGA	NSGA-II
1	1	1	1	1	30.25	21.98
2	1	2	2	2	27.30	33.79
3	1	3	3	3	30.15	28.91
4	2	1	2	3	29.00	27.83
5	2	2	3	1	17.04	26.47
6	2	3	1	2	19.16	15.55
7	3	1	3	2	31.75	48.05
8	3	2	1	3	19.66	19.34
9	3	3	2	1	30.20	20.02

performances of the meta-heuristics in terms of the four metrics, the Fisher test is used. The p values of the Fisher statistics shown in Table 7 indicate that at a 95%

confidence level, there are no significant differences among the algorithms in terms of all metrics, except MID. However, the box plots of the metrics in Fig. 12 show that MOIWO performs the best in term of MID, diversification (DM), and RAS metrics, while NPGA is the best in terms of the NPS metric.

Ranking the algorithms

As none of the three utilized parameter-calibrated algorithms is superior in terms of all of the four metrics, a hybrid multi-attribute decision-making (MADM) method called Entropy-TOPSIS is used to identify the best in terms of all metrics, simultaneously. In this method, the algorithms are considered as the alternatives and the metrics act as criteria. Moreover, the entropy method proposed by Hwang and Yoon (1981) is utilized to identify the weights of the criteria, meanwhile, the TOPSIS method presented by Hwang and Yoon (1981) is employed to identify the

Table 5 Results to tune the parameters of MOIWO

Run order	Max-iteration	Number of weeds	P_{max}	P_{mut}	Initial sigma	Final sigma	S_{min}	S_{max}	Nonlinear modulation index	MOIWO response
1	1	1	1	1	1	1	1	1	1	1.68335
2	1	1	1	1	2	2	2	2	2	1.66503
3	1	1	1	1	3	3	3	3	3	1.12930
4	1	2	2	2	1	1	1	2	2	1.61157
5	1	2	2	2	2	2	2	3	3	1.31550
6	1	2	2	2	3	3	3	1	1	1.45001
7	1	3	3	3	1	1	1	3	3	1.19566
8	1	3	3	3	2	2	2	1	1	1.49482
9	1	3	3	3	3	3	3	2	2	1.45047
10	2	1	2	3	1	2	3	1	2	1.22645
11	2	1	2	3	2	3	1	2	3	1.42086
12	2	1	2	3	3	1	2	3	1	1.78438
13	2	2	3	1	1	2	3	2	3	1.30581
14	2	2	3	1	2	3	1	3	1	1.50994
15	2	2	3	1	3	1	2	1	2	1.50176
16	2	3	1	2	1	2	3	3	1	1.51293
17	2	3	1	2	2	3	1	1	2	1.58298
18	2	3	1	2	3	1	2	2	3	1.54248
19	3	1	3	2	1	3	2	1	3	1.25910
20	3	1	3	2	2	1	3	2	1	1.38598
21	3	1	3	2	3	2	1	3	2	1.86841
22	3	2	1	3	1	3	2	2	1	1.69817
23	3	2	1	3	2	1	3	3	2	1.25912
24	3	2	1	3	3	2	1	1	3	1.45219
25	3	3	2	1	1	3	2	3	2	1.19657
26	3	3	2	1	2	1	3	1	3	1.38338
27	3	3	2	1	3	2	1	2	1	1.78587



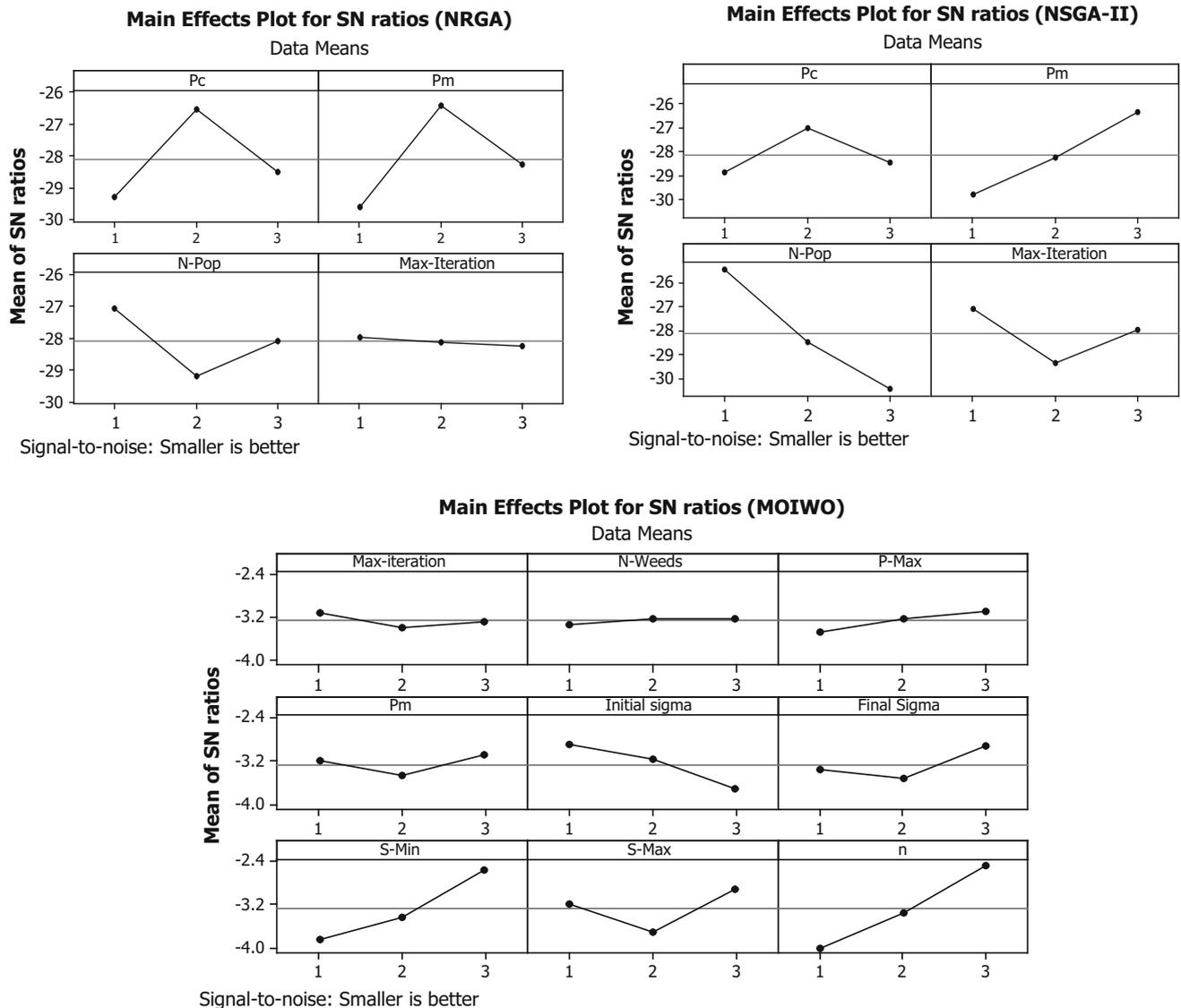


Fig. 10 Signal-to-noise ratios obtained by the algorithms

priority of the alternatives in solving the problems. The steps of the hybrid Entropy-TOPSIS method are:

- Create the decision matrices of the criteria for small-, medium-, and large-size problems
- Normalize the elements of the decision matrices using Eq. (39).

$$p_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}}, \quad \forall j, \quad (39)$$

where, p_{ij} is the normalized element and a_{ij} is an element of a decision matrix.

- Calculate the entropy of the j th criterion using Eq. (40).

$$E_j = -\frac{1}{\ln m} \times \sum_{i=1}^n [p_{ij} \ln p_{ij}] \quad \forall j \quad (40)$$

- Compute the deviation or the uncertainty degree of a criterion using Eq. (41).

$$d_j = 1 - E_j \quad \forall j \quad (41)$$

- Obtain the weights of the criteria using Eq. (42).

$$w_j = \frac{d_j}{\sum_{j=1}^n d_j} \quad \forall j \quad (42)$$

- Calculate the normalized decision matrices, r , in the TOSIS method using (43).

Table 6 Implementation results

MOIWO	NRGA			NSGA-II								
	MID	DM	RAS	NPS	MID	DM	RAS	NPS	MID	DM	RAS	NPS
1	14,5797.02	131,427.45	0.40	3	3,895,310.77	131,509.95	0.83	4	5,230,511.39	135,782.58	0.56	3
2	326,125.35	239,682.02	0.50	7	3,389,101.46	251,542.56	0.87	8	5,478,031.01	234,469.44	0.66	6
3	540,670.57	438,975.46	0.40	7	7,081,969.75	366,151.22	0.55	8	7,903,251.82	416,820.27	0.35	5
4	980,227.99	809,143.42	0.31	12	6,304,484.77	689,376.22	0.33	10	5,833,880.76	690,098.63	0.50	8
5	1,291,284.43	934,606.18	0.45	11	9,360,322.28	744,203.08	0.43	11	8,130,932.35	800,704.09	0.38	11
6	1,572,342.57	1,242,426.08	0.33	8	10,749,581.10	916,773.48	0.33	9	4,631,936.54	944,722.84	0.43	15
7	1,909,708.29	1,408,958.40	0.43	14	10,515,097.14	1,054,079.00	0.34	10	6,839,332.19	1,113,971.21	0.37	14
8	2,085,405.97	1,771,358.26	0.28	13	12,183,370.50	1,301,056.44	0.37	14	13,107,682.87	1,217,529.87	0.24	10
9	2,513,822.42	2,037,443.45	0.32	13	6,742,237.48	1,490,442.39	0.48	13	8,906,309.44	1,502,146.42	0.36	15
10	2,862,735.17	2,413,238.45	0.27	18	9,449,736.66	1,719,067.77	0.33	17	17,536,250.64	1,605,893.54	0.27	11
11	3,218,662.75	2,726,211.36	0.24	9	11,497,718.52	2,025,707.32	0.38	16	18,463,268.12	1,924,784.53	0.28	13
12	3,481,289.81	2,981,884.61	0.24	15	12,540,765.98	2,122,459.19	0.33	16	9,624,932.39	2,109,699.14	0.30	13
13	3,956,685.75	3,245,213.79	0.32	18	16,110,135.90	2,330,416.86	0.31	17	10,099,438.44	2,397,786.27	0.35	17
14	3,960,026.02	3,380,123.38	0.29	18	18,678,455.63	2,416,800.07	0.32	13	13,846,808.20	2,563,862.33	0.26	16
15	4,559,387.63	3,820,348.8	0.26	14	13,508,653.33	2,732,337.49	0.32	20	14,728,055.37	2,836,050.29	0.34	20
16	4,720,835.08	4,003,112.66	0.29	17	13,099,015.85	3,137,558.88	0.25	18	14,927,802.21	3,110,634.63	0.32	15
17	5,127,697.51	4,533,786.93	0.21	16	13,060,574.95	3,170,186.57	0.35	27	15,580,593.56	3,340,536.62	0.26	21
18	5,655,287.39	4,886,069.22	0.24	16	18,451,369.81	3,502,510.30	0.34	20	22,497,484.10	3,564,745.44	0.21	16
19	5,773,480.38	5,059,399.67	0.24	17	14,219,795.87	3,535,900.79	0.39	24	17,039,715.11	3,565,041.10	0.31	19
20	5,918,516.27	5,376,930.00	0.19	14	16,528,754.20	3,849,816.05	0.29	26	14,140,586.54	3,749,769.99	0.28	22
21	145,797.02	131,427.45	0.40	3	3,895,310.77	131,509.95	0.83	4	5,230,511.39	135,782.58	0.56	3
22	326,125.35	239,682.02	0.50	7	3,389,101.46	251,542.56	0.87	8	5,478,034.01	234,469.44	0.66	6
23	540,670.57	438,975.46	0.40	7	7,081,969.71	366,153.22	0.55	8	7,903,251.82	416,820.27	0.35	5
24	448,125.33	298,782.06	0.60	4	3,265,954.75	458,751.02	0.77	8	5,478,032.01	398,541.33	0.46	7
25	552,470.17	524,674.47	0.55	6	7,520,966.15	387,542.30	0.66	7	7,145,876.45	325,874.54	0.55	6
26	358,925.85	421,682.02	0.55	5	5,842,979.75	398,542.92	0.60	7	5,247,885.02	458,412.09	0.48	6
27	413,625.39	356,975.76	0.48	4	8,541,960.05	498,135.45	0.77	6	4,569,822.08	556,820.25	0.60	7
28	511,170.22	366,682.08	0.62	7	6,981,969.75	257,591.46	0.61	5	7,425,413.01	385,425.82	0.58	7
29	469,825.01	425,978.55	0.48	5	6,054,965.21	502,122.55	0.58	6	5,247,895.36	389,521.35	0.74	6
30	528,770.03	514,475.49	0.50	5	5,456,994.75	426,411.02	0.62	7	4,879,852.25	554,162.75	0.65	7
AV1	809,407.988	632,710.1	0.4	8	6,796,795	516,592.75	0.56	8	6,201,424	537,099.64	0.48	8
AV2	2,998,542.02	2,495,554	0.3	15	12,214,690	1,807,503.6	0.36	14	12,303,003	1,804,459.2	0.31	14
AV3	2,099,421.44	1,838,575.6	0.42	9	8,892,778.9	1,391,618.3	0.57	12	9,519,517	1,412,437.2	0.47	10

$$r_{ij} = \frac{f_{ij}}{\sqrt{\sum_{j=1}^n f_{ij}^2}} \quad \forall i, \tag{43}$$

where r_{ij} , f_{ij} and n denote an element of the normalized decision matrix, an element of the decision matrix, and the number of alternatives, respectively.

- Obtain the weighted normalized decision matrices using Eq. (44):

$$V = r * W_{n \times n}, \tag{44}$$

where V and $W_{n \times n}$ are the weighted normalized decision matrix and the diagonal weight matrix, respectively. Please note that the elements in the main diagonal of the weight matrix are the weights of the criteria calculated by the entropy approach.

- Compute the positive and the negative ideal solutions using Eqs. (45)–(46):

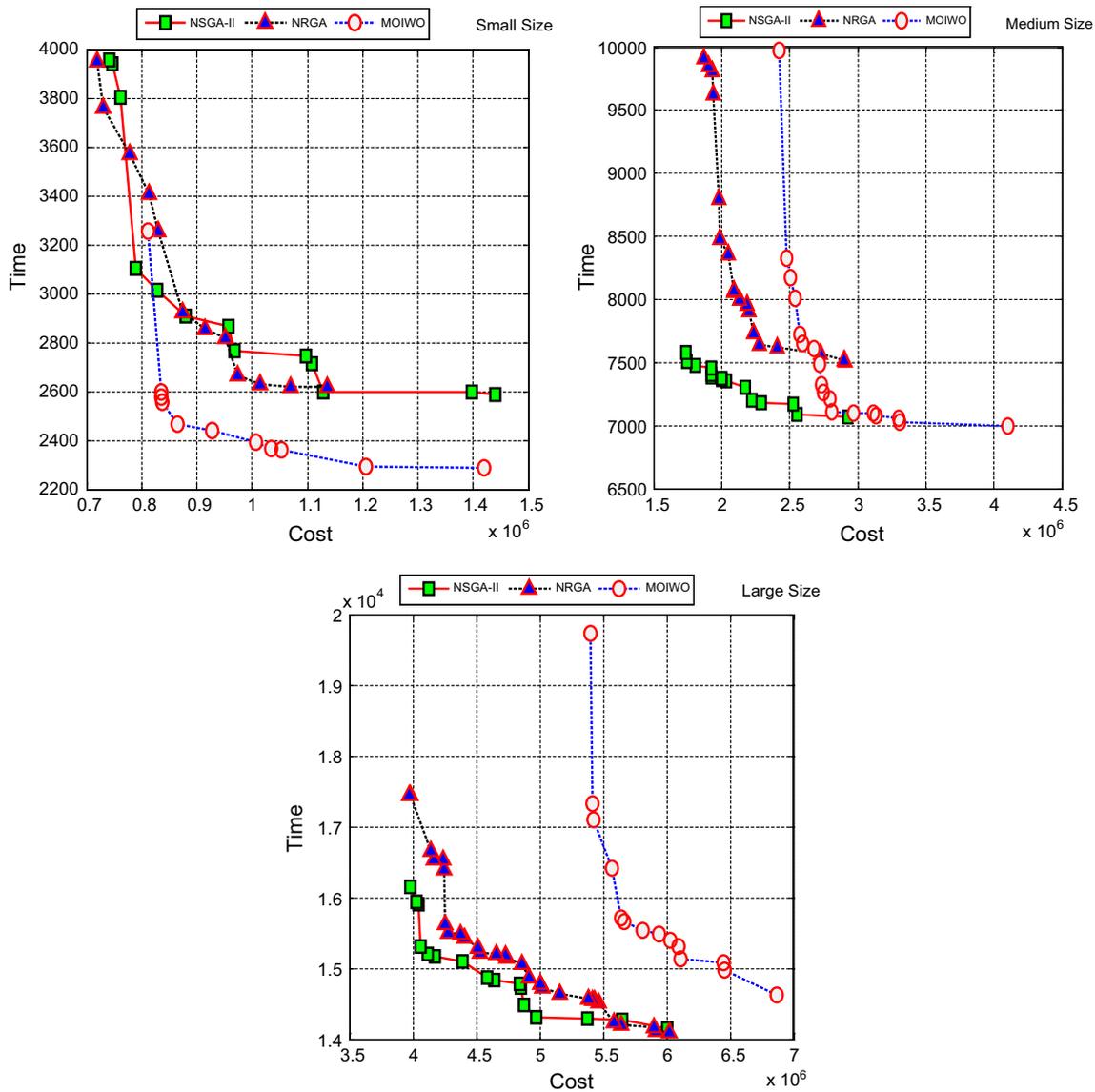


Fig. 11 Pareto fronts obtained by the solution algorithms

Table 7 Statistical comparisons of the algorithms

Metric	P value	Test result
MID	0.000	The algorithms perform different
DM	0.347	No significant difference found
NPS	0.472	No significant difference found
RAS	0.013	No significant difference found

$$v_j^+ = \begin{cases} \max_i(v_{ij}) & \text{for benefit criteria} \\ \min_i(v_{ij}) & \text{for cost criteria} \end{cases} \quad (45)$$

$$v_j^- = \begin{cases} \max_i(v_{ij}) & \text{for benefit criteria} \\ \min_i(v_{ij}) & \text{for cost criteria} \end{cases} \quad (46)$$

- Use Eqs. (47) and (48) to obtain the distances of the alternatives from the positive and the negative ideal solutions.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad i = 1, \dots, m \quad (47)$$

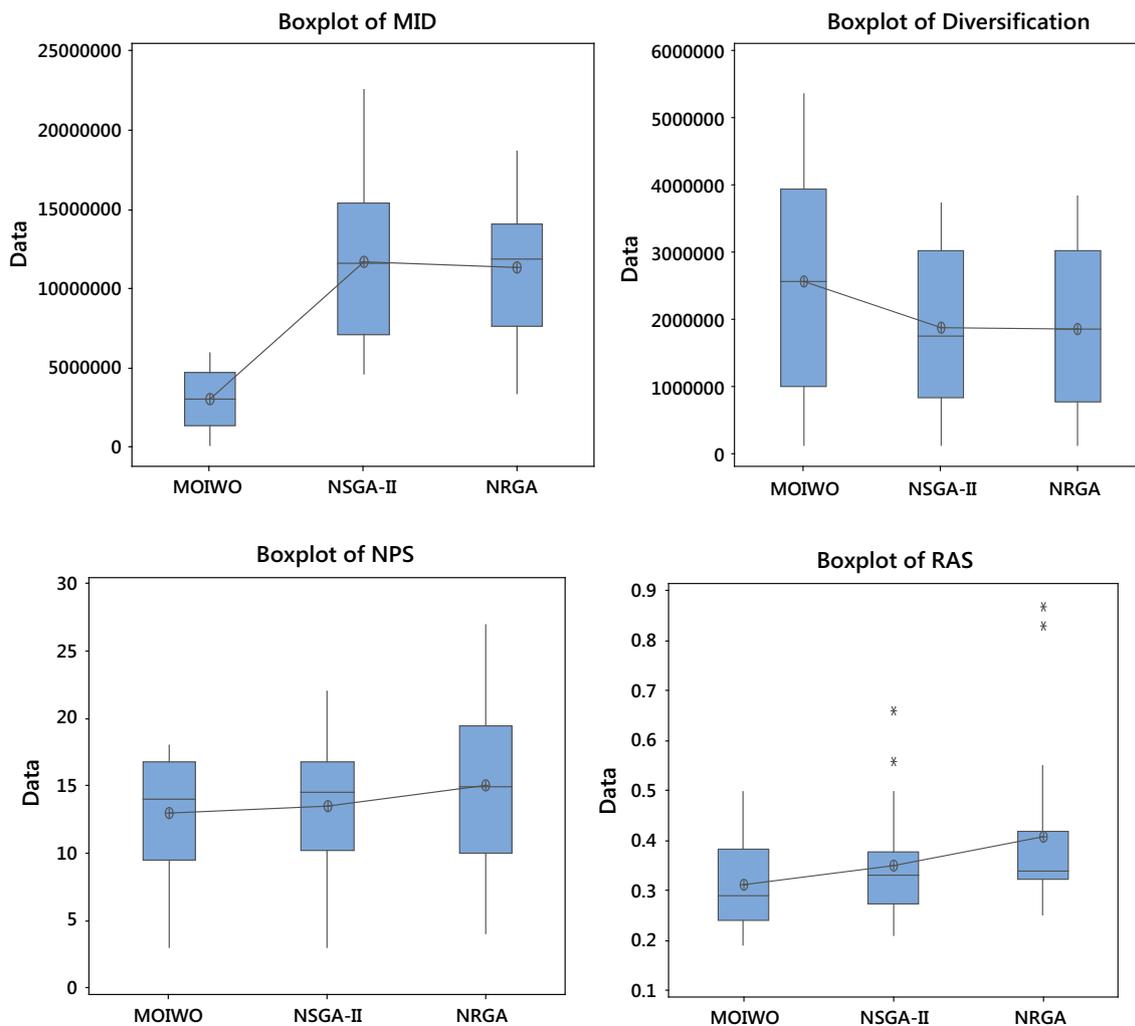


Fig. 12 Box plot of the four metrics obtained for the solution algorithms

Table 8 Decision and the normalized decision matrices (small-size problems)

Problem number	Decision matrix				Normalized decision matrix			
	MID	DM	NPS	RAS	MID	DM	NPS	RAS
1	3,090,539.72	132,906.66	3.33	0.597	0.23	0.04	0.07	0.21
2	3,064,419.27	241,898.00	7.00	0.676	0.23	0.07	0.14	0.10
3	5,175,297.38	407,315.65	6.67	0.431	0.14	0.12	0.14	0.16
4	4,372,864.50	729,539.42	10.00	0.377	0.16	0.22	0.21	0.18
5	6,260,846.35	826,504.45	11.00	0.420	0.11	0.25	0.23	0.16
6	5,651,286.74	1,034,640.80	10.67	0.360	0.13	0.31	0.22	0.19

Table 9 Weights of the criteria (small-size problems)

	MID	DM	NPS	RAS
E	0.978	0.899	0.965	0.988
$1 - E$	0.022	0.101	0.035	0.012
Weight	0.130	0.594	0.206	0.070

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad i = 1, \dots, m \tag{48}$$

- Obtain the closeness of the alternatives using Eq. (49).

Table 10 Decision and the normalized decision matrices (medium-size problems)

Problem number	Decision matrix				Normalized decision matrix			
	MID	DM	NPS	RAS	MID	DM	NPS	RAS
7	6,421,379.20	1,192,336.20	12.67	0.378	0.17	0.07	0.11	0.10
8	9,125,486.44	1,429,981.52	12.33	0.295	0.12	0.09	0.11	0.13
9	6,054,123.11	1,676,677.42	13.67	0.384	0.18	0.10	0.12	0.10
10	9,949,574.16	1,912,733.25	15.33	0.288	0.11	0.12	0.13	0.14
11	11,059,883.13	2,225,567.73	12.67	0.298	0.10	0.14	0.11	0.13
12	8,548,996.06	2,404,680.98	14.67	0.290	0.13	0.15	0.13	0.14
13	10,055,420.03	2,657,805.64	17.33	0.324	0.11	0.16	0.15	0.12
14	12,161,763.28	2,786,928.59	15.67	0.289	0.09	0.17	0.14	0.14

Table 11 Weights of the criteria (medium-size problems)

	MID	DM	NPS	RAS
<i>E</i>	0.986	0.982	0.997	0.997
1 – <i>E</i>	0.014	0.018	0.003	0.003
Weight	0.368	0.469	0.085	0.077

$$CL_i = \frac{d_i^-}{d_i^- + d_i^+} \quad i = 1, \dots, m \quad (49)$$

- Rank the alternatives based on their closeness. The best alternative (algorithm) is the one with the highest closeness.

A brief implementation of the above steps to rank the three meta-heuristics in terms of four metrics is shown in Tables 8, 9, 10, 11, 12, 13, 14, 15, and 16. According to the

closeness of the alternatives obtained at the end, the MOIWO algorithm is the best to solve the problems.

Conclusion and future research

In this paper, a new bi-objective intermodal hub-location-allocation problem within queuing framework was investigated. The developed model of this problem intended to simultaneously minimize the total network costs and the total time spent in the network. The solution of a small-size problem using the GAMS software was presented to show the accuracy of the developed formulation. As the problem belongs to the class of NP-Hard problems with conflicting objectives, an MOIWO algorithm with a new chromosome structure was utilized to solve the problem. Moreover, a fuzzy ranking method was used in MOIWO to prioritize the solutions obtained in each iteration. Since there was no

Table 12 Decision and the normalized decision matrices (large-size problems)

Problem number	Average decision matrix				Normalized decision matrix			
	MID	DM	NPS	RAS	MID	DM	NPS	RAS
15	10,932,032	3,129,579	18	0.31	0.096	0.118	0.103	0.040
16	10,915,884	3,417,102	16	0.29	0.096	0.129	0.095	0.037
17	11,256,289	3,681,503	21	0.27	0.099	0.139	0.122	0.036
18	15,534,714	3,984,442	17	0.26	0.137	0.151	0.099	0.034
19	12,344,330	4,053,447	20	0.31	0.108	0.153	0.115	0.041
20	12,195,952	4,325,505	20	0.25	0.107	0.164	0.119	0.033
21	3,090,540	132,906.7	3	0.60	0.027	0.005	0.019	0.078
22	3,064,420	241,898	7	0.68	0.027	0.009	0.040	0.089
23	5,175,297	407,316.3	6	0.43	0.045	0.015	0.038	0.057
24	3,064,037	385,358.1	6	0.61	0.026	0.014	0.036	0.080
25	5,073,104	412,697.1	6	0.59	0.045	0.015	0.036	0.077
26	3,816,597	426,212.3	6	0.54	0.034	0.016	0.0345	0.071
27	4,508,469	470,643.8	5	0.62	0.039	0.017	0.032	0.081
28	4,972,851	336,566.5	6	0.60	0.043	0.012	0.036	0.079
29	3,924,229	439,207.5	5	0.6	0.034	0.016	0.032	0.079
30	3,621,872	498,349.8	6	0.59	0.031	0.018	0.03	0.078

Table 13 Weights of the criteria (large-size problems)

	MID	DM	NPS	RAS
<i>E</i>	0.941	0.815	0.938	0.979
$1 - E$	0.058	0.184	0.061	0.020
Weight	0.179	0.568	0.189	0.063

available benchmark in literature, two other meta-heuristics called NSGA-II and NPGA with a unique chromosome structure were designed to qualify the solutions obtained by MOIWO. The parameters of the algorithms were tuned using the Taguchi method. The results obtained using these calibrated solution algorithms were compared statistically. The comparison results showed that the MOIWO algorithm acts better than the other two solution algorithms in terms of RAS, MID, and diversification metrics, while the NPGA

algorithm is the best in term of the NPS metric. Finally, an Entropy-TOPSIS method was used to find the best algorithm in terms of all metrics. The results showed that the MOIWO algorithm was the best to solve all problems.

The following works can be recommended in future research:

- Formulating the uncertainties involved in the main parameters of the model using some other methods such as the robust optimization approach.
- Taking into consideration the failure probabilities of the origin and the destination hub facilities.
- Utilizing some other meta-heuristic solution algorithms to solve the problem.
- Employing a different approach to compare the performances of the solution algorithms.

Table 14 Entropy-TOPSIS results (small-size problems)

	Decision matrix				Normalized decision matrix				Weighted normalized decision matrix				d_i^+	d_i^-	CL
	MID	DM	NPS	RAS	MID	DM	NPS	RAS	MID	DM	NPS	RAS			
NRGA	6,796,795.02	516,592.75	15	0.56	0.74	0.53	0.58	0.67	0.10	0.31	0.12	0.05	0.13	0	0
NSGA-II	6,201,423.98	537,099.64	14	0.48	0.67	0.55	0.58	0.57	0.09	0.33	0.12	0.04	0.10	0.02	0.17
MOIWO	809,407.99	632,710.1	15	0.40	0.09	0.65	0.58	0.19	0.01	0.39	0.12	0.01	0.00	13.0	1.00

Table 15 Entropy-TOPSIS results (medium-size problems)

	Decision matrix				Normalized decision matrix				Weighted normalized decision matrix				d_i^+	d_i^-	CL
	MID	DM	NPS	RAS	MID	DM	NPS	RAS	MID	DM	NPS	RAS			
NRGA	12,214,689.73	1,807,503.6	15	0.36	0.69	0.51	0.59	0.65	0.25	0.24	0.05	0.05	0.21	0.01	0.05
NSGA-II	12,303,002.79	1,804,459.1	14	0.30	0.7	0.51	0.55	0.54	0.26	0.24	0.05	0.04	0.41	0.01	0.02
MOIWO	2,998,542.02	2,495,553.9	15	0.30	0.17	0.70	0.59	0.54	0.06	0.33	0.05	0.04	0.00	0.21	1.00

Table 16 Entropy-TOPSIS results (large-size problems)

	Decision matrix				Normalized decision matrix				Weighted normalized decision matrix				d_i^+	d_i^-	CL
	MID	DM	NPS	RAS	MID	DM	NPS	RAS	MID	DM	NPS	RAS			
NRGA	9,181,271.02	1,475,413.3	13	0.5	0.67	0.51	0.68	0.66	0.12	0.29	0.04	0.1	0.14	0.01	0.08
NSGA-II	9,845,050.64	1,501,413	11	0.45	0.72	0.52	0.56	0.56	0.13	0.30	0.03	0.1	0.13	0.02	0.14
MOIWO	2,253,169.32	1,962,436.4	10	0.40	0.16	0.68	0.47	0.5	0.03	0.38	0.03	0.09	0.009	0.14	0.93

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Al Jadaan O, Rajamani L, Rao CR (2008) Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NPGA. *J Theor Appl Inf Technol* 4:60–67
- Asefi H, Jolai F, Rabiee M, Tayebi Araghi ME (2014) A hybrid NSGA-II and VNS for solving a bi-objective no-wait flexible flowshop scheduling problem. *Int J Adv Manuf Technol* 75:1017–1033
- Aykin T (1995) Networking policies for hub-and-spoke systems with application to the air transportation system. *Transp Sci* 29:201–221
- Calik H, Alumur SA, Kara BY, Karasan OE (2009) A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Comput Oper Res* 36:3088–3096
- Campbell J (1992) Location and allocation for distribution systems with transshipments and transportation economies of scale. *Ann Oper Res* 40:77–99
- Campbell JF (1994) Integer programming formulations of discrete hub location problems. *Eur J Oper Res* 72:387–405
- Campbell JF (1996) Hub location and the p-hub median problem. *Oper Res* 44:923–935
- Cunha CB, Silva MR (2007) A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *Eur J Oper Res* 179:747–758
- de Camargo RS, Miranda G (2012) Single allocation hub location problem under congestion: network owner and user perspectives. *Expert Syst Appl* 39:3385–3391
- de Camargo RS, Miranda G Jr, Ferreira RPM, Luna HP (2009) Multiple allocation hub-and-spoke network design under hub congestion. *Comput Oper Res* 36:3097–3106
- de Camargo RS, de Miranda Jr G, Ferreira RPM (2011) A hybrid outer-approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Oper Res Lett* 39:329–337
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197
- Elhedhli S, Hu FX (2005) Hub-and-spoke network design with congestion. *Comput Oper Res* 32:1615–1632
- García S, Landete M, Marín A (2012) New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem. *Eur J Oper Res* 220:48–57
- Ghaffari-Nasab N, Ghazanfari M, Teimoury E (2015) Robust optimization approach to the design of hub-and-spoke networks. *Int J Adv Manuf Technol* 76:1091–1110
- Ghodratnama A, Tavakkoli-Moghaddam R, Azaron A (2013) A fuzzy possibilistic bi-objective hub covering problem considering production facilities, time horizons and transporter vehicles. *Int J Adv Manuf Technol* 66:187–206
- Gross D, Harris CM (1998) *Fundamental of queuing theory*, 3rd edn. Wiley, New York
- Habibzadeh Boukani F, Farhang Moghaddam B, Pishvaei M (2016) Robust optimization approach to capacitated single and multiple allocation hub location problems. *Comput Appl Math* 35:45–60
- Hwang CL, Yoon K (1981) *Multiple attribute decision making: methods and applications—a state-of-the-art survey*. Springer, New York
- Ilić A, Urošević D, Brimberg J, Mladenović N (2010) A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *Eur J Oper Res* 206:289–300
- Ishfaq R, Sox CR (2010) Intermodal logistics: the interplay of financial, operational and service issues. *Transp Res Part E Logist Transp Rev* 46:926–949
- Karimi N, Zandieh M, Karamooz HR (2010) Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. *Expert Syst Appl* 37:4024–4032
- Kraticevic J, Stanimirović Z, Tošić D, Filipović V (2007) Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *Eur J Oper Res* 182:15–28
- Kraticevic J, Milanović M, Stanimirović Z, Tošić D (2011) An evolutionary-based approach for solving a capacitated hub location problem. *Appl Soft Comput* 11:1858–1866
- Kundu D, Suresh K, Ghosh S, Das S, Panigrahi BK, Das S (2011) Multi-objective optimization with artificial weed colonies. *Inf Sci* 181:2441–2454
- Lin C-C, Lin J-Y, Chen Y-C (2012) The capacitated p-hub median problem with integral constraints: an application to a Chinese air cargo network. *Appl Math Model* 36:2777–2787
- Maghsoudlou H, Kahag MR, Niaki STA, Pourvaziri H (2016) Bi-objective optimization of a three-echelon multi-server supply chain problem in congested systems: modeling and solution. *Comput Ind Eng* 99:41–62
- Marianov V, Serra D (2003) Location models for airline hubs behaving as M/D/c queues. *Comput Oper Res* 30:983–1003
- Mohammadi M, Jolai F, Rostami H (2011) An M/M/c queue model for hub covering location problem. *Math Comput Model* 54:2623–2638
- Mohammadi M, Jolai F, Tavakkoli-Moghaddam R (2013) Solving a new stochastic multi-mode p-hub covering location problem considering risk by a novel multi-objective algorithm. *Appl Math Model* 37:10053–10073
- O’Kelly ME (1986) The location of interacting hub facilities. *Transp Sci* 20:92–106
- O’Kelly ME (1987) A quadratic integer program for the location of interacting hub facilities. *Eur J Oper Res* 32:393–404
- Parvaresh F, Husseini SMM, Golpayegany SAH, Karimi B (2014) Hub network design problem in the presence of disruptions. *J Intell Manuf* 25:755–774
- Peiró J, Corberán Á, Martí R (2014) GRASP for the uncapacitated r-allocation p-hub median problem. *Comput Oper Res* 43:50–60
- Puerto J, Ramos AB, Rodríguez-Chía AM (2011) Single-allocation ordered median hub location problems. *Comput Oper Res* 38:559–570
- Rahmati SHA, Hajipour V, Niaki STA (2013) A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. *Appl Soft Comput* 13:1728–1740
- Sarrafha K, Rahmati SHA, Niaki STA, Zaretalab A (2015) A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: a new tuned MOEA. *Comput Oper Res* 54:35–51
- Seifbarghi M, Mansouri A (2016) Modelling and solving a congested facility location problem considering systems’ and customers’ objectives. *Int J Ind Syst Eng* 22:281–304
- Shahvari O, Logendran R (2016) Hybrid flow shop batching and scheduling with a bi-criteria objective. *Int J Prod Econ* 179:239–258



- Shahvari O, Salmasi N, Logendran R, Abbasi B (2012) An efficient tabu search algorithm for flexible flow shop sequence-dependent group scheduling problems. *Int J Prod Res* 50:4237–4254
- Skorin-Kapov D, Skorin-Kapov J, O’Kelly M (1996) Tight linear programming relaxations of uncapacitated p-hub median problems. *Eur J Oper Res* 94:582–593
- Taguchi G (1986) Introduction to quality engineering: designing quality into products and processes. Asian Productivity Organization
- Takano K, Arai M (2009) A genetic algorithm for the hub-and-spoke problem applied to containerized cargo transport. *J Mar Sci Technol* 14:256–274
- Ting C-J, Chen C-H (2013) A multiple ant colony optimization algorithm for the capacitated location routing problem. *Int J Prod Econ* 141:34–44
- Topcuoglu H, Corut F, Ermis M, Yilmaz G (2005) Solving the uncapacitated hub location problem using genetic algorithms. *Comput Oper Res* 32:967–984
- Yang K, Liu Y, Yang G (2013) An improved hybrid particle swarm optimization algorithm for fuzzy p-hub center problem. *Comput Ind Eng* 64:133–142
- Zade AE, Sadegheih A, Lotfi MM (2014) A modified NSGA-II solution for a new multi-objective hub maximal covering problem under uncertain shipments. *J Ind Eng Int* 10:185–197

