Eydi, Alireza; Khaleghi, Amir; Barzegar, Khaled

Article

Ring hierarchical hub network design problem: Exact and heuristic solution methods

EURO Journal on Transportation and Logistics (EJTL)

Provided in Cooperation with:
Association of European Operational Research Societies (EURO), Fribourg

Suggested Citation: Eydi, Alireza; Khaleghi, Amir; Barzegar, Khaled (2022) : Ring hierarchical hub network design problem: Exact and heuristic solution methods, EURO Journal on Transportation and Logistics (EJTL), ISSN 2192-4384, Elsevier, Amsterdam, Vol. 11, Iss. 1, pp. 1-23, https://doi.org/10.1016/j.ejtl.2022.100096

This Version is available at:
https://hdl.handle.net/10419/325166

EURO Journal on Transportation and Logistics 11 (2022) 100096

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.sciencedirect.com/journal/euro-journal-on-transportation-and-logistics

Ring hierarchical hub network design problem: Exact and heuristic solution methods

Alireza Eydi [a,*], Amir Khaleghi [b], Khaled Barzegar [c]

[a] Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran
[b] PhD in Industrial Engineering, University of Kurdistan, Sanandaj, Iran
[c] MSc in Industrial Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Keywords:
Transportation
Ring hierarchical hub network
Ring network
Benders decomposition
Genetic algorithm
Variable neighborhood search algorithm
Relax-and-fix algorithm

ABSTRACT

In this paper, we study the hierarchical hub network design problem with a ring-star-star structure. In this problem, the hubs are located in two layers. In the first layer, central hubs (main hubs) are located in a ring structure, and in the second layer, secondary hubs are located. Each secondary hub is allocated to a central hub. Other demand points can also be allocated to each of these hubs (central hubs or secondary hubs). The objective is to minimize transportation costs on the network. This problem applies to communication networks when establishing direct links between demand nodes is not cost-effective, and there are two levels of service for customers. Also, the ring structure is used to reduce costs associated with full communication between central hubs. We present a mixed-integer programming model for the problem and report the results of the problem solving for a numerical example on the CAB dataset. Also, we present three solution methods for the problem. First, by exploiting the decomposable structure of the proposed model, we introduce an accelerated Benders decomposition algorithm. Then, we present a hybrid genetic algorithm that uses the Dijkstra algorithm to evaluate solutions. Next, we present a hybrid variable neighborhood search algorithm that uses the Dijkstra algorithm to calculate the cost of each solution. Also, a relax-and-fix algorithm is proposed to find near-optimal feasible solutions for the problem. Computational results are presented on the USA423 dataset. The results show that the proposed relax-and-fix algorithm can solve instances with up to 100 nodes. Also, the proposed hybrid algorithms can solve instances with up to 423 nodes. The performance of the proposed hybrid variable neighborhood search is better than the proposed hybrid genetic algorithm in solving large-sized instances.

1. Introduction

Hubs are known as collection, transferring, and distribution centers in communication systems used for providing indirect connections between demand points when establishing direct links is not cost-effective. In hub systems, some hubs are selected as intermediaries between demand points, and other demand points are allocated to these hubs. The flows between the origin and destination first enter the hubs, are organized at hubs, and finally distributed to the destination. Hub systems have many applications in transportation networks, telecommunications, message transmission networks, logistics, and postal services. The hub-and-spoke structure allows the use of economies of scale on the inter-hub connections. Also, this structure reduces network development costs by reducing the number of communication links in the network.

In the context of hub location, (O'Kelly 1987) presented a mathematical model for the hub location problem and investigated several classes of this problem. Generally, hub location problems can be categorized into four types: 1) hub median location problem, 2) hub location problem with fixed costs, 3) hub center location problem, and 4) hub covering location problem. In these variants, the objective function in the first and second types of the problem is to minimize the total costs. In the first type, only the total flow cost is minimized; this is because the number of facilities (hubs) is exogenous. However, in the second type, since the number of hubs is endogenous, the cost of establishing hubs is also added to the objective function (Farahani et al., 2013). In the hub center location problem, the objective function is to minimize the maximum length or cost of each path in the network. In the hub covering location problem, the objective function is to minimize the number of hubs established, while constraints of covering the demands in the network are satisfied.

* Corresponding author.
E-mail addresses: alireza.eydi@uok.ac.ir (A. Eydi), A.khaleghi@eng.uok.ac.ir (A. Khaleghi), Barzegar.kh@gmail.com (K. Barzegar).

https://doi.org/10.1016/j.ejtl.2022.100096
Received 27 March 2022; Received in revised form 24 September 2022; Accepted 18 October 2022
Available online 22 October 2022

More detailed categories can be made for the mentioned problems based on the type of allocation: single allocation and multiple allocation. In the first case, each non-hub node is allocated to exactly one hub. However, in the multiple allocation case, this constraint is relaxed, and each non-hub node can be allocated to more than one hub. Also, in some problems, there are flow constraints for hubs and arcs. These problems are called capacitated problems. However, in uncapacitated problems, there are no such constraints for the flows on the network. Interested readers can refer to (Alumur and Kara, 2008; Campbell et al., 2002; Campbell and O'Kelly, 2012; Farahani et al., 2013; Kara and Taner, 2011) to learn more about different types of hub location problems, applications, and solution algorithms.

One of the assumptions of the classical hub location problem is the completeness of the hub network. In the classical hub location problem, each hub is connected to all other hubs by direct links, and a complete hub graph is formed. Given the many constraints faced by decision-makers, especially concerning the budget, in many cases, it is not feasible having a complete hub graph structure, and we should rely on fewer communication links to cover the demands in the system. In such cases, the use of ring structure in the backbone network (which interconnects the hubs) is applied. The ring hub network design problem is a single allocation problem in which the hubs in the network are connected with a ring structure, and the non-hub nodes are allocated to the selected hubs. In this case, the network structure is ring-star. The ring hub network is suitable for developing expensive hub networks.

Another structure used in hub networks is the hierarchical structure, which is also suitable for designing telecommunication networks. In this structure, there are different levels of service in the network, and the customers are allocated to service centers (hubs) at different levels. This provides a variety of customer services. By combining ring structure and hierarchical structure for hub location problem, the ring hierarchical hub location problem is obtained. This structure has the advantages of the ring and hierarchical structures. We deal with such a network in some real-world applications such as goods shipment and public transportation. For example, if we use the fast trains or subways as the main hub network and use other facilities as secondary hubs in public transportation, we will have a ring hierarchical hub network. Another use of ring hierarchical structures can be found in telecommunication networks in which telecommunication equipment is used as hubs. For example, when there are two types of branching in the network in optical fiber networks and the costs of developing equipment are high, the ring structure will have the advantage over other structures.

This research provides a mathematical model for designing a ring hierarchical hub network, in which it is decided about the location of hubs at each level of the hierarchy, allocations, how the ring is formed, and routing of the flows on the network to minimize total transportation costs.

2. Literature review

In the context of the incomplete hub location problem, (Klincewicz 1998) presented a review of the hub location problem in designing the backbone network and tributary network (which connects demand nodes to hubs). Also, the authors studied applications of this problem in facility location, telecommunications, computer networks, and transportation systems. Later on, the hub arc location problem was introduced by (Campbell et al., 2005a, b), in which the hub arcs instead of the hub nodes are located. In their problem, a certain number of hub arcs (with discounted shipping costs) are located in a way that minimizes the shipping costs imposed on the system. The assumption of the completeness of the hub network and the connectivity of the backbone network is not included in their work. They proposed integer programming models with multiple allocation for four specific cases and used an enumeration-based algorithm to solve the problem.

The hub covering network design problem with an incomplete structure has been investigated by (Alumur and Kara 2009). In this problem, it is assumed that there could not be more than three hubs in paths designed between each origin-destination pair. (Calik et al., 2009) proposed a tabu search-based algorithm to solve a single allocation hub location problem with an incomplete structure of the hub network. Also, (Yoon and Current 2008) discussed the problem of designing an incomplete hub network for two modes of arc costs (fixed and variable costs). The proposed model minimizes the total costs of transportation, hub location, and hub arc location.

Later on, (Alumur et al. 2009) presented efficient mathematical models for all previous versions of the single allocation hub network design problem. They also showed that in most cases, the completeness of the hub network is not necessary to achieve a specific service level (the delivery of goods or passengers at a specified time). As one of the applied researches in the transportation scope, (Gelareh and Nickel 2011) studied multiple allocation hub network design in an incomplete network and used the Benders decomposition method and a greedy algorithm to solve the model.

In some studies, a pre-defined topology is considered for the incomplete hub network. Modeling the hub center problem with star topology was done by (Yaman 2008). The author applied a Lagrangian relaxation-based algorithm to solve the problem. Then, (Gelareh and Pisinger 2011) applied the chain structure of the hub network for maritime transportation. The authors used a decomposition approach to solve the model. Later on, (Karimi and Setak 2014) studied a multiple allocation hub location problem considering an incomplete hub network to minimize proprietor costs. These costs include the fixed costs of hubs, hub links, and spoke links. The researchers developed a set of valid inequalities to strengthen the formulation of the problem and also presented a lower bounding procedure using the Lagrangian relaxation algorithm.

Hub location problem considering tree structure of hub network was studied by (Lee et al., 1996). In this structure, system costs are mainly costs of establishing the hubs. Later on, a tree hub location problem was studied by (Contreras et al., 2010). In this research, a tree topology is considered for the backbone network based on the location of a certain number of hubs, and each non-hub node is allocated to exactly one hub. In other words, in their study, the hub network is two-layered and has a tree-star structure, which itself represents a larger tree. The objective function is to minimize total transportation costs. To solve this problem, the authors proposed a branch-and-bound algorithm. They also provided a flow-based four-index model for this problem in (Contreras et al., 2009) and showed that this model gives tighter bounds. They used a Lagrangian relaxation algorithm to decompose the mentioned model and also used a heuristic algorithm to obtain a feasible solution (upper bound on the minimization problem) at each stage of the Lagrangian dual solution process in such a way that the Lagrangian relaxation method can solve large-sized problems in a reasonable time. Also, researchers in (de Sá et al., 2013) solved problems with 100 nodes on a tree network by presenting an exact algorithm based on the Benders decomposition.

A multi-modal hub location problem with a tree structure under fuzzy uncertainty was studied by (Sedehzadeh et al., 2016). The authors considered two objectives: minimizing fuel consumption and minimizing transportation costs in the network. A Multi-Objective Imperialist Competitive Algorithm (MOICA) was used to solve the model, and its results were compared with a Non-dominated Sorting Genetic Algorithm version II (NSGAII). Concerning ring networks, ring-star problems were introduced by (Labbé et al., 2004) and (Labbé et al., 2005). These problems aim to form a ring of hub nodes and allocate each non-hub node to the nearest hub node in the ring, subject to the costs of network design are minimized.

Another related problem is the median tour problem introduced in (Current and Schilling 1994). This problem is used for tour infrastructure facilities and aims to find a tour of nodes so that the total cost of assigning nodes outside of the tour to selected tour nodes does not exceed a specified budget. Also, (Current and Schilling 1994) and

(Gendreau et al., 1997) studied the covering ring-star problem. In this problem, the distance between each node outside of the selected ring and the nearest node in the ring should not exceed a certain value. All of the above models only consider network design costs, and flows are not included in the network design.

In the context of the hierarchical hub location problem, authors in (Kuby and Gray 1993) studied hub networks with stopovers and feeders. In this case, the location of the central hub is already known, and the aim is to determine the route of flow from the origin to the destination, taking into account the capacity constraints of the arcs, so that the sum of line-haul costs of arcs and fixed costs of routes are minimized. The researchers examined the application of this problem in the Federal Express (FedEx) system. They showed that their proposed network is more cost-effective than the pure hub-and-spoke network. Later on, (Horner and O'Kelly, 2005) introduced a hierarchical assignment problem, which combines the clustering problem and interaction between facilities. The facilities are individual geographic entities. This problem aims to find the optimal flow patterns between the origin and destination points, so that grouping or aggregating can be done between the origin or destination points. This problem can be used to design hub-and-spoke networks. A hierarchical median hub location problem with three layers was studied by (Yaman 2009). The author considered a complete structure for the central hubs. Each secondary hub is allocated to exactly one central hub, and each non-hub node is allocated to one of these two types of hubs. The author presented a quadratic integer programming model for the problem and linearized the objective function of the model. This model is applied in cargo shipment networks. The network extended in this case is a three-layered star-star structure. An integer programming model was also proposed by (Alumur et al., 2012) for the hierarchical multi-modal hub location problem. In their work, a given city is considered as the central hub, and the hub network structure is star-tour-star. A mathematical model for a capacitated single allocation hierarchical hub median location problem was proposed by (Karimi et al., 2014). Recently, (Shang et al., 2021) proposed two heuristics (a variable neighborhood search and an improved NSGAII algorithm) for a bi-objective hierarchical multi-modal hub location problem with applications in cargo delivery systems. The authors considered the following objectives: minimizing system-wide costs and minimizing the maximum delivery time.

In the field of developing meta-heuristic algorithms for solving the hierarchical hub location problem, (Zarandi et al., 2015) performed an empirical comparison of the simulated annealing algorithm and an iterated local search to solve the hierarchical hub median location problem with single allocation. Results showed that solution time in both algorithms on the CAB dataset is negligible and is significantly lower than the exact method. In the context of the hierarchical facility location and hub location problem on the network, (Torkestani et al., 2016) published a review paper. A hub covering location problem with service time bound was proposed by (Dukkanci and Kara 2017). In their work, the hub network is a hierarchical multi-modal network with three layers and a ring-star-star structure. They supposed that in each layer, there are different types of vehicles. Decisions in their problem were scheduling and routing the flows on the network to minimize the number of airline links between the airports (hubs). Also, (Zhong et al., 2018) worked on applications of the hierarchical hub location problem in integrating urban and rural public transport. Later on, (Eghbali-Zarch et al., 2019) proposed a mixed-integer linear programming model for a ring hub location problem under uncertainty and considering congestion at hubs. The hubs were modeled as an M/G/c queue system. To cope with uncertainty, researchers applied a robust possibilistic programming approach.

According to the previous studies, some of the most relevant works related to the subject of the present study were selected and more closely examined. Table 1 lists the relevant works and shows the differences between the present study and previous studies.

Given the above discussions, this research aims to design a hierarchical hub network in which the hubs in the first layer have a ring structure. The proposed three-layered ring hierarchical hub network is shown in Fig. (1). According to Fig. (1), in the first layer, the ring formed by the main hubs (hexagons) represents the main part that must necessarily be connected. Among the examples, we can refer to subway lines in the public transportation network of a large city. The secondary hubs (the feeder of the main network-the big circles) are allocated to the main hubs, and demand nodes (small circles) are allocated to one of the selected hubs. Thus, the proposed network has a ring-star-star structure. It should be noted that in Fig. (1), the thick lines are the main hub arcs, and dashed lines are the second layer or secondary hub arcs. The aim of the ring hierarchical hub network design, as discussed earlier, is to find

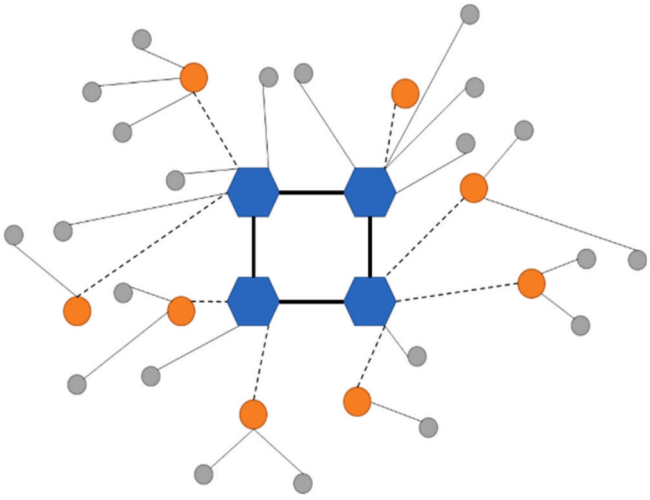

Fig. 1. The ring hierarchical hub network.

Table 1
A comparison between the present study and previous studies.

Problem features	Labbé et al. (2004)	Yaman (2009)	Alumur et al. (2012)	Dukkanci and Kara (2017)	Shang et al. (2021)	This study
Problem type	Median	Median	Covering	Covering	Median-Center	Median
Network structure	Ring-star	Complete-star-star	Star-tour-star	Ring-star-star	Complete-star-star	Ring-star-star
Facilities type	Single level	Hierarchical	Hierarchical	Hierarchical	Hierarchical	Hierarchical
Allocation	Single	Single	Single	Single	Single	Single
Objective function	Minimization	Minimization	Minimization	Minimization	Minimization	Minimization
Cost items	RC+AC	TC	TC+OC	NAL	SC and MDT	TC
Solution method	B & C	Mixed integer programming	Mixed integer programming	LR-based algorithm	VNS and NSGAII	ABD, HGA, HVNS, and R&F

RC: Ring Costs; **AC**: Allocation Costs; **TC**: Transportation Costs; **OC**: Operating Costs; **SC**: System-wide Costs; **MDT**: Maximum Delivery Time; **NAL**: Number of Airline Links; **ABD**: Accelerated Benders Decomposition; **HGA**: Hybrid Genetic Algorithm; **HVNS**: Hybrid Variable Neighborhood Search; **R&F**: Relax-and-Fix algorithm

the optimal location of the main and secondary hubs, allocations, and routing flow in such a way that the total transportation costs through the network to be minimized.

The remainder of the present research is organized as follows: in the next section, we propose a mixed-integer programming model for the problem. Then, in section 4, the results of solving the model for small-sized instances are reported. In section 5, the solution methods are presented. Then, the results of the proposed solution methods are reported in section 6. Finally, the conclusions and future research are presented in section 7.

3. Problem description

In the present study, the problem definition is as follows: a transportation network containing some demand nodes is given. Candidate locations for establishing central hubs and secondary hubs are already known. Also, the flow that needs to be transferred between the demand nodes, the unit transportation cost between the demand nodes, and the discount factors of inter-hub connections for central and secondary hubs are predetermined. Some nodes in this network must be selected as central hubs, and some nodes must be selected as secondary hubs. The aim is to determine the location of the central and secondary hubs, allocation of the secondary hubs to the central hubs, allocation of the demand nodes to each hub (central or secondary), how to connect the central hubs in a ring structure, and routing the flows on the network, to minimize the total transportation costs. In the following, notations of the mathematical model are introduced:

$N = \{1, 2, \ldots, n\}$ Set of demand nodes

H Set of candidate nodes for secondary hubs $(H \subseteq N)$

C Set of candidate nodes for main hubs $(C \subseteq H)$

p Number of main hubs that must be located

q Number of secondary hubs that must be located

c_{ij} Unit transportation cost for one unit of distance of one unit of flow, from node $i \in N$ to node $j \in N$

f_{ij} Amount of flow to be sent from node $i \in N$ to node $j \in N$

$O_i = \sum_{j \in N} f_{ij}$ Total flows originating from node $i \in N$

$D_i = \sum_{j \in N} f_{ji}$ Total flows which delivered at node $i \in N$

α_c Discount factor used in the main layer

α_h Discount factor used in the secondary layer

At first, a mathematical model with three-index variables is presented to formulate the flow in the network. These variables represent the amount of flow that originates from a particular node and passes through a hub arc. In general, the three-index model has two binary variables to determine locations and allocations and two continuous variables for routing the flows on the network. The proposed formulation is based on the formulation presented by (Yaman 2009). In the model proposed by (Yaman 2009), the central hub network is fully-connected, while in the present model, the ring structure is considered for the central hubs. In the present model, we assume that $f_{ij}+f_{ji} > 0$ for $i, j \in N$. If a particular application requires a single ring and the graph of flows contains more than one connected component, we can replace these flows of value zero with $f_{ij} = \varepsilon > 0$ sufficiently small (see (Contreras et al., 2016) for more details). In our model, two binary variables and two continuous variables are defined. Binary variables determine how demand nodes are allocated to secondary hubs and secondary hubs to central hubs and how the ring is formed between the central hubs. Continuous variables are related to the routing of flows between secondary hubs and central hubs and between central hubs. The decision variables are as follows: The binary variable z_{ijl} takes value 1, if demand node $i \in N$ is allocated to hub $j \in H$ and hub $j \in H$ is allocated to central hub $l \in C$, otherwise 0. If z_{jjl} is equal to 1, node $j \in H$ is selected as a secondary hub and allocated to central hub $l \in C$. If z_{lll} is 1, node $l \in C$ will be a central hub. Also, if z_{ill} is 1, node $i \in N$ is allocated to central hub $l \in C$. The binary variable y_{kl} is equal to 1, if a main (central) hub arc is established on the arc (k,l), for $k,l \in C$, $k<l$, otherwise 0. The continuous variable w_{ijl} represents the amount of flow that originates from node $i \in N$ and passes through the secondary hub arc (j,l), for $j \in H$, $l \in C$. The continuous variable t_{ikl} represents the amount of flow that originates from node $i \in N$ and passes through the central hub arc (k,l), for $k,l \in C$, $k \neq l$. The proposed model is formulated as follows:

$$\min \sum_{i \in N} \sum_{j \in H} \sum_{l \in C} (c_{ij} O_i + c_{ji} D_i) z_{ijl} + \sum_{i \in N} \sum_{j \in H} \sum_{l \in C \setminus \{j\}} \alpha_h c_{jl} w_{ijl} + \sum_{i \in N} \sum_{k \in C} \sum_{l \in C \setminus \{k\}} \alpha_c c_{kl} t_{ikl} \quad (1)$$

$$z_{ijl} \leq z_{jjl} \forall i \in N, j \in H \setminus \{i\}, l \in C \quad (2)$$

$$\sum_{j \in H} \sum_{l \in C} z_{ijl} = 1 \forall i \in N \quad (3)$$

$$y_{kl} \leq z_{kkk} \forall k, l \in C : k < l \quad (4)$$

$$y_{kl} \leq z_{lll} \forall k, l \in C : k < l \quad (5)$$

$$\sum_{j \in H} z_{ijl} \leq z_{lll} \quad \forall i \in N, l \in C \setminus \{i\} \quad (6)$$

$$\sum_{l \in C} z_{lll} = p \quad (7)$$

$$\sum_{j \in H} \sum_{l \in C \setminus \{j\}} z_{jjl} = q \quad (8)$$

$$\sum_{k \in C} \sum_{l \in C:\ k<l} y_{kl} = p \quad (9)$$

$$\sum_{l \in C: k<l} y_{kl} + \sum_{l \in C: k>l} y_{lk} = 2 z_{kkk} \forall k \in N \quad (10)$$

$$z_{ljl} = 0 \forall j \in H, l \in C \setminus \{j\} \quad (11)$$

$$\sum_{k \in C \setminus \{l\}} t_{ilk} - \sum_{k \in C \setminus \{l\}} t_{ikl} = \sum_{j \in H} \sum_{r \in N} f_{ir} (z_{ijl} - z_{rjl}) \forall i \in N, l \in C \quad (12)$$

$$w_{ijl} \geq \sum_{r \in N \setminus \{j\}} (f_{ir} + f_{ri}) (z_{ijl} - z_{rjl}) \forall i \in N, j \in H, l \in C \setminus \{j\} \quad (13)$$

$$t_{ikl} + t_{ilk} \leq O_i y_{kl} \forall i \in N, k, l \in C : k < l \quad (14)$$

$$w_{ijl} \geq 0 \forall i \in N, j \in H, l \in C \quad (15)$$

$$t_{ikl} \geq 0 \forall i \in N, k, l \in C, k \neq l \quad (16)$$

$$z_{ijl} \in \{0, 1\} \forall i \in N, j \in H, l \in C \quad (17)$$

$$y_{kl} \in \{0, 1\} \forall k, l \in C : k < l \quad (18)$$

In the above formulation, the objective function of the model in **Eq. (1)** consists of three cost items. The first item calculates the total costs of the flow from the demand nodes to the hubs. The second item is the total costs of the flow from the secondary hubs to the main hubs. Finally, the third term is total flow costs on the central ring. Constraint (2) ensures that if the demand node is allocated to the secondary hub j and secondary hub j is allocated to central hub l, then the node j must be a secondary hub. Constraint (3) ensures that each node in the network allocates to a secondary hub or a central hub or that it is a hub itself.

Constraints (4) and (5) ensure that the central hub arc can be established among the points k and l only if the start and end of this arc are main hubs. If node i is allocated to main hub l, l must be a central hub which is guaranteed by constraint (6). Constraints (7) and (8) declare that p hubs in the central ring and q hubs in the second layer must be selected. To form a ring in the central layer, the number of arcs in this layer must be equal to the number of main hubs that can be realized through constraints (9) and (10). Constraint (10) states that each hub of the main layer must be connected exactly to two other central hubs (note that this constraint is valid for $p > 2$). Constraint (11) is not necessary but is used for preprocessing and can improve the quality of the lower bound of the relaxed linear programming problem. Constraint (12) is the flow balance constraint at central hubs. The outgoing flows from node i, which enter the central hub l through the secondary hub j, are calculated by constraint (13). Note that if two nodes are simultaneously allocated to a secondary hub, the flow between them will not enter the central hub, which is included in constraint (13). Constraint (14) states that the flow between the nodes can only pass from the main hub arc (k,l), only if a main hub arc is established between these two points (note that because it is assumed that $f_{ij}+f_{ji} > 0$, constraints (9), (10), and (14) ensure that the central hubs graph is not divided into separate rings or sub-tours). Constraints (15)–(18) represent the type of decision variables.

In the worst case, when $|N| = |H| = |C| = n$, the above formulation has $3n^3$-n^2/2-n/2 decision variables, of which n^3+n^2/2-n/2 variables are binary, and $2n^3$-n^2 variables are continuous. Thus, the number of variables in the worst case is $O(n^3)$. The number of constraints of the problem is $5n^3/2+3n^2/2-n+3$. Thus, in the worst case, the number of constraints is $O(n^3)$.

4. Computational experiments

Computational experiments of the mathematical model are presented in this section. The proposed model is coded in the GAMS software environment version 27.3.0. The CPLEX commercial solver version 12.9.0 is used to solve instances. Computations are carried out on a laptop with the features: Intel (R) Core (TM) i5-5200U CPU 2.20 GHz processor, 6 gigabytes of RAM on Windows 8.1 Enterprise 64 bit. The CAB dataset is used in experiments (Beasley 2008). The first fifteen cities in the CAB dataset are selected as candidate nodes for hubs, and the first ten cities are selected as candidate nodes for main hubs. Calculations for instances on the CAB dataset are shown in Table 2. Columns in Table 2 represent the solved problem features, including the discount factors for main and secondary hubs, number of main and secondary hubs, the optimal objective function value, the location of main hubs, the location of secondary hubs, number of branch-and-bound tree nodes, and CPLEX solver solution time, respectively. Note that in the column "Location of main hubs", the order of the appearance of the main hubs represents the ring structure. For example, if the location of the main hubs is 2, 6, 4, and 8, then the ring is formed as 2-6-4-8-2.

According to Table 2, for a specific number of main hubs, by increasing the number of secondary hubs, total transportation cost (the objective function value) decreases, and in some cases, by increasing the number of secondary hubs, total transportation cost does not change (cases with $p = 4$ and $q = 2, 3, 4$). For a specific number of secondary hubs ($q = 0, 1, 2, 3$), transportation cost in cases with $p = 4$ is lower than in cases with $p = 3$. This means that by increasing the number of main hubs, transportation costs decrease. The minimum transportation cost is related to the case with $\alpha_c = 0.9$, $\alpha_h = 0.8$, $p = 4$, and $q = 4$, and the maximum transportation cost is related to the case with $\alpha_c = 1$, $\alpha_h = 1$, $p = 3$, and $q = 0$. Note that the lowest transportation cost are obtained when the total number of hubs is maximum and discount factors are lower than 1. Also, the highest transportation cost occurs when the total number of hubs in the network is minimum and the discount factors are maximum. Also, total transportation cost when discount factors are $\alpha_c = 1$ and $\alpha_h = 1$ is bigger than in two other cases. It seems that as the number of hubs in the network increases and the amount of discount factors decreases, most flows are routed through inter-hub links to take advantage of the economies of scale to reduce transportation costs. In Fig. (2), a representation of the optimal solution for the case with four main hubs and four secondary hubs for different values of discount factors is provided.

According to Fig.2, for $\alpha_c = 0.8$ (case b), the ring of central hubs has maximum length, and for $\alpha_h = 0.8$ (case c), the total distances of connections between secondary hubs and main hubs is maximum. Also, transportation cost in case (a) is higher than case (b) and transportation cost in case (b) is higher than case (c). Values of α_c and α_h affect the unit

Table 2
Calculations on the CAB dataset.

(α_c,α_h)	p	q	Optimal obj. value	Location of main hubs	Location of secondary hubs	Nodes	CPLEX Time (s)
(1,1)	3	0	10826334067.16	2, 4, 8	–	0	1.09
		1	10520877129.32	1, 2, 4	8	26	59.23
		2	10434096418.07	4, 7, 8	2, 12	84	79.66
		3	10369055082.17	4, 6, 8	2, 12, 13	48	76.27
	4	0	10535560637.18	2, 6, 4, 8	–	0	5.16
		1	10324886450.51	4, 6, 7, 8	2	80	92.91
		2	10228686307.12	4, 6, 7, 8	2, 12	144	141.56
		3	10228686307.12	4, 6, 7, 8	2, 10, 12	350	112.95
		4	10228686307.12	4, 6, 7, 8	2, 9, 10, 12	728	106.00
(0.8,0.9)	3	0	10036336175.43	2, 4, 8	–	0	0.81
		1	9767245481.06	1, 2, 4	8	19	34.20
		2	9602756903.05	4, 7, 8	2, 12	13	65.41
		3	9480577465.44	4, 6, 8	2, 12, 13	47	102.50
	4	0	9729751091.31	2, 4, 8, 7	–	27	7.56
		1	9486742602.12	2, 4, 8, 7	12	165	174.36
		2	9274041453.58	4, 6, 7, 8	2, 12	32	75.98
		3	9234841867.37	4, 6, 7, 8	1, 2, 12	312	205.47
		4	9213619452.05	2, 6, 4, 8	3, 12, 13, 14	228	95.56
(0.9,0.8)	3	0	10431335121.30	2, 4, 8	–	0	1.16
		1	9742342537.15	1, 2, 4	12	17	26.42
		2	9472655811.08	4, 7, 8	2, 12	24	27.02
		3	9388018738.95	1, 4, 6	2, 7, 12	40	37.59
	4	0	10136469363.77	2, 6, 4, 8	–	15	7.61
		1	9535898851.13	1, 2, 6, 4	12	39	39.69
		2	9351383103.89	1, 2, 6, 4	7, 12	174	129.41
		3	9239337995.08	1, 2, 6, 4	7, 12, 14	348	124.86
		4	9162926171.19	1, 2, 6, 4	3, 7, 12, 14	323	135.38

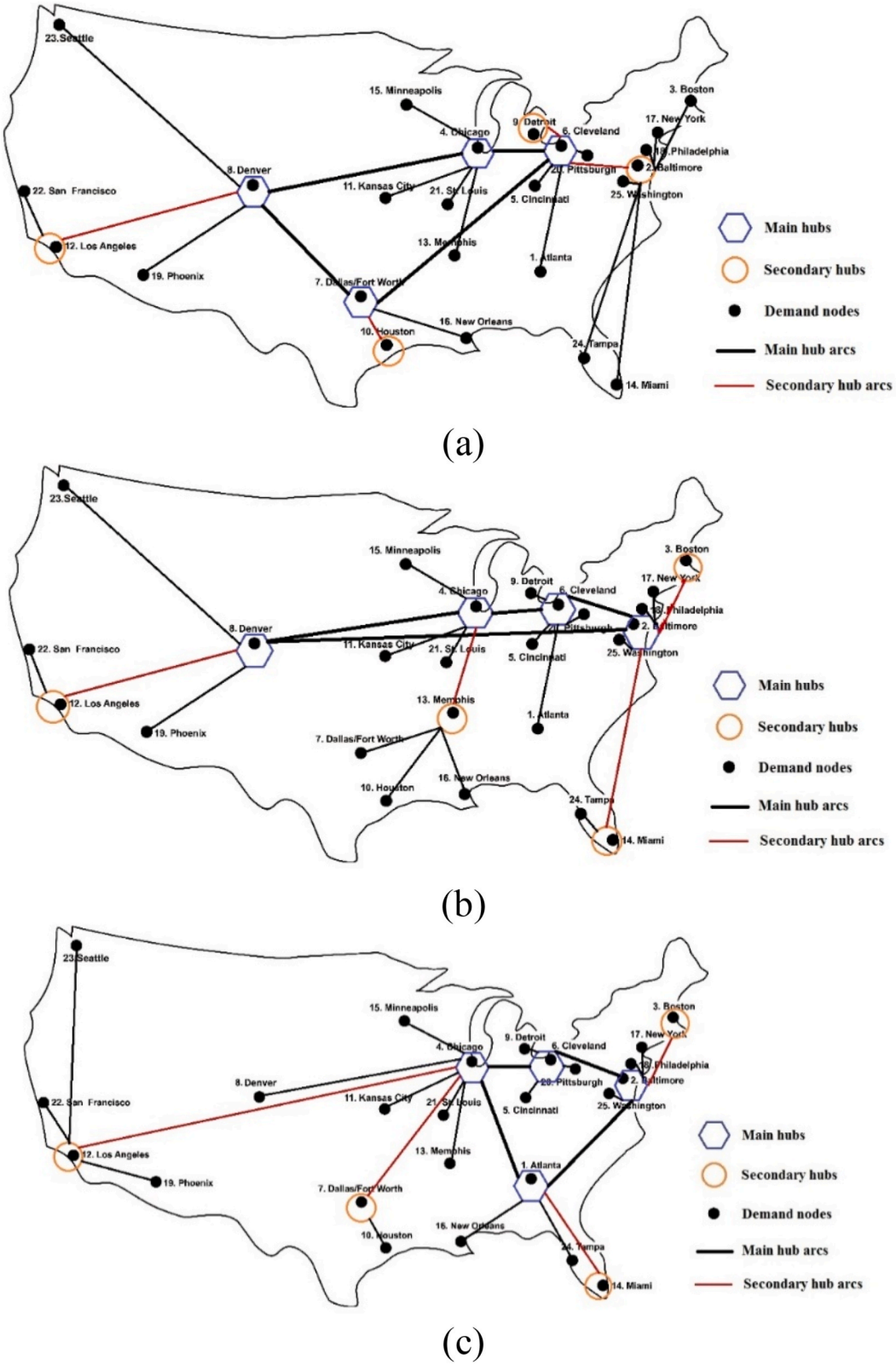

Fig. 2. Representation of the optimal solution for instance with $p = 4$, $q = 4$ for a) $\alpha_c = 1$, $\alpha_h = 1$ b) $\alpha_c = 0.8$, $\alpha_h = 0.9$ c) $\alpha_c = 0.9$, $\alpha_h = 0.8$.

transportation costs in inter-hub connections. Also, decisions about the location of hubs in the model affect the routing of flows. Therefore, in each of the above three cases, the location of the hubs is done in such a way that most of the flows are concentrated in less costly links for different values of discount factors.

5. Solution methods

The classical hub location problem is NP-hard (Skorin-Kapov et al., 1996). The hierarchical hub location problem is also NP-hard; because the p-hub median problem is a special case of the hierarchical hub

location problem (Yaman 2009). The ring hierarchical hub location problem, which has more constraints than the hierarchical hub location problem, is also NP-hard. As the size of the problem increases, its solution time exponentially increases. Thus, in this section, we propose four solution methods to solve larger instances of the problem. These methods include an accelerated Benders decomposition algorithm, a hybrid genetic algorithm, a hybrid variable neighborhood search algorithm, and a relax-and-fix algorithm. In the following, we will explain each of these solution methods.

5.1. Benders decomposition algorithm

Benders decomposition algorithm or Benders partitioning algorithm usually is used for solving MIP models (Benders 1962). In this approach, the primal MIP model is decomposed into two problems (namely, master problem and sub-problem). Master Problem (MP) is a relaxed version of the primal model, and Sub-Problem (SP) is a Linear Programming problem that is formulated after relaxing complicating variables (integer or binary variables). The dual SP is used to find the cutting plane. In each iteration of the Benders algorithm, MP is solved and provides complicated variables value, and then SP is solved to find the cutting plane for MP. This process continues until reaching the optimality conditions (based on relationships between primal and dual problems). In the proposed model of the present study, if the binary variables are fixed at a specified level, the problem becomes a minimum cost flow problem that is easy to solve. Therefore, the binary variables cause the complexity of the model, and the model is decomposable to implement Benders decomposition.

5.1.1. Dual sub-problem

In the present study, SP is formulated as **Eqs. (1), (12)–(16)**. In this formulation, binary variables are fixed at a specified level (denoted by the hat symbol). By introducing dual variables $v^1_{il}, v^2_{ijl}, v^3_{ikl}$, respectively for constraints (12)–(14), dual of SP is formulated as **Eqs. 19–23**. Note that in this formulation, binary variables fixed at a specific level are denoted by the hat symbol.

$$\max \sum_{i\in N}\sum_{l\in C} v^1_{il}\left[\sum_{j\in H}\sum_{r\in N} f_{ir}\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+ \sum_{i\in N}\sum_{j\in H}\sum_{l\in C\setminus\{j\}} v^2_{ijl}\left[\sum_{r\in N\setminus\{j\}}\left(f_{ir}+f_{ri}\right)\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+ \sum_{i\in N}\sum_{k\in C}\sum_{l\in C,l>k} v^3_{ikl}[O_i\widehat{y}_{kl}] \tag{19}$$

$$v^1_{ik}-v^1_{il}+v^3_{ikl}\le \alpha_c C_{kl}\forall i\in N,k,l\in C,k<l \tag{20}$$

$$v^1_{ik}-v^1_{il}+v^3_{ilk}\le \alpha_c C_{kl}\forall i\in N,k,l\in C,k>l \tag{21}$$

$$v^2_{ijl}\le \alpha_h C_{jl}\forall i\in N,j\in H,l\in C \tag{22}$$

$$v^1_{il}\ is\ free, v^2_{ijl}\ge 0 \quad and \quad v^3_{ikl}\le 0 \tag{23}$$

5.1.2. Master problem

MP formulation includes constraints with only integer variables, optimality cuts, and feasibility cuts (for unbounded SP). Optimality cuts are formulated as **Eq. (25)**. Note that once the integer variables are fixed, the SP becomes a minimum cost flow problem. Because there are no capacity constraints in the model, each produced solution by MP is feasible for dual of SP. Thus, there is no need for feasibility cuts in MP. Note that in this formulation, the dual variables whose values are derived from the solution of the Benders SP are represented by the star symbol.

$$\min z^{MP} \tag{24}$$

s.t.

$$z^{MP}\ge \sum_{i\in N}\sum_{j\in H}\sum_{l\in C}\left(C_{ij}O_i+C_{ji}D_i\right)z_{ijl}+ \sum_{i\in N}\sum_{l\in C} v^{*1}_{il}\left[\sum_{j\in H}\sum_{r\in N} f_{ir}\left(z_{ijl}-z_{rjl}\right)\right]+ \sum_{i\in N}\sum_{j\in H}\sum_{l\in C\setminus\{j\}} v^{*2}_{ijl}\left[\sum_{r\in N\setminus\{j\}}\left(f_{ir}+f_{ri}\right)\left(z_{ijl}-z_{rjl}\right)\right]+ \sum_{i\in N}\sum_{k\in C}\sum_{l\in C,l>k} v^{*3}_{ikl}[O_i y_{kl}] \tag{25}$$

And **Eqs. (2)–(11), (17), (18)**

5.1.3. Accelerating Benders decomposition algorithm

In the present study, straightforward implementation of Benders decomposition leads to a low convergence rate and high solution time of the algorithm. Therefore, it is necessary to use techniques to accelerate the Bender process. To do this, we use the high-density Pareto-optimal cut generation approach developed by (Tang et al., 2013). To illustrate this method, suppose that the general form of the SP is as follows:

$$\min cx \tag{26}$$

s.t.

$$Ax\le b-B\widehat{y} \tag{27}$$

$$x\ge 0 \tag{28}$$

In which, $c\in\mathbb{R}^{n_1}$, $x\in\mathbb{R}^{n_1}_{+}$, $A\in\mathbb{R}^{m\times n_1}$, $b\in\mathbb{R}^m$ and $B\in\mathbb{R}^{m\times n_2}$. Then, the dual of SP is formulated as follows:

$$\max v^T(b-B\widehat{y}) \tag{29}$$

s.t.

$$A^T v\le c \tag{30}$$

$$v\le 0 \tag{31}$$

After solving the dual of SP, we get the optimal objective function value as $v^{*T}(b-B\widehat{y})$. For generating high-density Pareto-optimal cuts, the following problem must be solved to optimality:

$$\max \sum_{k=1}^{m} v_k\left(\frac{1}{b_k}-\frac{1}{\sum_{s=1}^{m}B_{ks}}\right) \tag{32}$$

s.t.

$$A^T v\le c \tag{33}$$

$$v^T(b-B\widehat{y})=v^{*T}(b-B\widehat{y}) \tag{34}$$

$$v\le 0 \tag{35}$$

In this formulation, only v_k can be incorporated into the objective function for which we have $\sum_{s=1}^{m}B_{ks}\neq 0$. Also, if $b_k=0$, then $\frac{1}{b_k}$ is an infinite number (a sufficiently big number). In the following formulation, this big number is denoted by M. With this preface, in the present model, for generating high-density Pareto-optimal cuts, after solving the SP, the following problem must be solved to optimality:

$$\max \sum_{i\in N}\sum_{k\in C}\sum_{l\in C,l>k} v^3_{ikl}\left(M-\frac{1}{O_i}\right) \tag{36}$$

s.t.

$$\sum_{i\in N}\sum_{l\in C} v^1_{il}\left[\sum_{j\in H}\sum_{r\in N} f_{ir}\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+\sum_{i\in N}\sum_{j\in H}\sum_{l\in C\setminus\{j\}} v^2_{ijl}\left[\sum_{r\in N\setminus\{j\}}\left(f_{ir}+f_{ri}\right)\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+$$
$$\sum_{i\in N}\sum_{k\in C}\sum_{l\in C,l>k} v^3_{ikl}[O_i\widehat{y}_{kl}]=\sum_{i\in N}\sum_{l\in C} v^{*1}_{il}\left[\sum_{j\in H}\sum_{r\in N} f_{ir}\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+$$
$$\sum_{i\in N}\sum_{j\in H}\sum_{l\in C\setminus\{j\}} v^{*2}_{ijl}\left[\sum_{r\in N\setminus\{j\}}\left(f_{ir}+f_{ri}\right)\left(\widehat{z}_{ijl}-\widehat{z}_{rjl}\right)\right]+\sum_{i\in N}\sum_{k\in C}\sum_{l\in C,l>k} v^{*3}_{ikl}[O_i\widehat{y}_{kl}] \quad (37)$$

And Eqs. 19–23

5.2. Hybrid genetic algorithm

One of the most powerful and most widely used meta-heuristic algorithms, whose effectiveness in solving continuous and discrete problems has been proven, is the Genetic Algorithm (Holland 1992). In this subsection, we present a hybrid method based on genetic algorithm, which is a combination of a genetic algorithm and a Dijkstra algorithm. In this method, the Dijkstra algorithm is used to evaluate the solutions generated in the genetic algorithm. The following are the details of this hybrid method.

5.2.1. Coding

Coding the chromosomes is one of the most important steps in the implementing the genetic algorithm. Each chromosome represents a solution to the problem. Our strategy in producing solutions in the initial population in this research is that no infeasible solution is generated. In this paper, decisions about locating central and secondary hubs, allocating demand nodes to the located hubs, and the ring structure between the central hubs are considered in the chromosome representation. To display the solution, the chromosomes should be divided into two parts. The first part relates to the allocation of demand nodes to hubs (including central hubs and secondary hubs), and the second part describes how secondary hubs are allocated to the central hubs, as well as how to create arcs in the ring structure. For example, suppose that we have a network with fourteen nodes, four central hubs, and four secondary hubs. A solution representation can be as Fig. (3).

According to the represented chromosome in Fig. (3), central hubs 1, 4, 3, and 2 and the secondary hubs 5, 9, 11, and 6 are located. Note that to separate the secondary and central hubs in the second part of the chromosome, the numbers related to the central hubs appear twice. The secondary hubs that precede a central hub are assigned to that central hub. Also, in the first part of the chromosome, the demand nodes that precede a hub (central or secondary) are assigned to that hub. Therefore, the allocations are as follows: nodes 12 and 13 to secondary hub 11, node 10 to secondary hub 9, nodes 7 and 8 to secondary hub 6, and node 14 to central hub 3. Note that a ring structure should be created between the central hubs. The appearance of the central hubs in the second part of the chromosome illustrates the links between them. In this case, since the order of appearance of central hubs is 1-4-3-2, the ring created in the network includes arcs (1,4), (3,4), (2,3), and (1,2). The graphical representation of this chromosome is shown in Fig. (4). In this way of solution representation, constraints (2)–(11) that are related to the design of the network are taken into account. That is, the network structure for each chromosome can be determined from the information available on each chromosome.

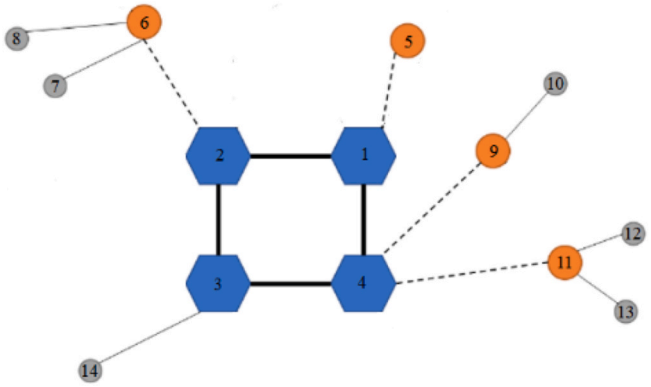

Fig. 4. The schematic representation of the solution in Fig. (3).

5.2.2. Creating initial population

In this step, initial population is randomly generated, and the number of its members is considered to be *npop*. Given that in the present model, the capacity constraints are not considered for hubs, all the solutions generated in the initial population are feasible.

5.2.3. Evaluation

In this step, the chromosomes must be decoded, and the fitness function value must be calculated for them. In this research, we interpret each chromosome into the decision variables; so that the values of the decision variables are determined from the genes of each chromosome. To calculate the objective function value for each chromosome, we first introduce two new binary variables. Variable $x_{ij}\forall i\in N, j\in H$ and variable $r_{jl}\forall j\in H, l\in C$. x_{ij} takes value of 1 if node $i\in N$ is allocated to hub $j\in H$, otherwise 0. r_{jl} takes value of 1 if node $j\in H$ is allocated to hub $l\in C$, otherwise 0. The transportation cost is calculated as the sum of the costs related to the flows on the arcs. These arcs include arcs between non-hub nodes and hubs (secondary and central), arcs between secondary hubs and central hubs, and arcs between central hubs. In calculating the costs related to the flows on the central hub arcs, it is assumed that central hubs are fully-connected, and the unit transportation cost between central hubs is equal to the cost of the shortest path between them. The minimum-cost routes between central hubs are calculated through Dijkstra algorithm. In this way, for each chromosome, the y_{kl} values must be determined according to the ring structure and using these values, an adjacency matrix is determined (concerning the central hubs graph). Also, a weight matrix is obtained by using unit transportation costs and discount factor related to the central hubs. The adjacency matrix and weight matrix are inputs of Dijkstra algorithm. Because in the present study, capacity constraints are not considered for hubs, the flow between the origin and destination is shipped through the minimum-cost route. The complexity of the proposed evaluation algorithm is $O(n^2)$. The pseudocode of the evaluation algorithm is shown in Fig. (5).

5.2.4. Selection

In this research, the parent selection mechanism is done through the roulette wheel selection method. In this regard, the better chromosomes have a greater chance of choosing. The probability of selection for chromosome i (Pr_i) is calculated based on Eq. (38).

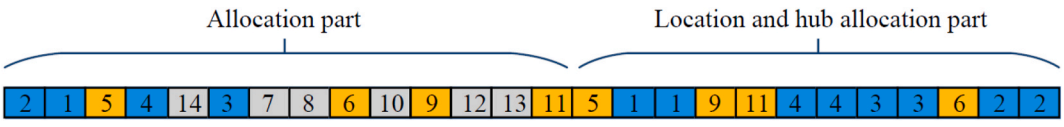

Fig. 3. The solution representation.

Inputs: x_{ij}, r_{jl}, and y_{kl} values related to a solution, matrix c, matrix f, α_c, *and* α_h
Output: Optimal objective function value for the solution

$AdjacancyMatrix=O_{|C|\times|C|}$;
$WeightMatrix=O_{|C|\times|C|}$;
$flow=O_{|C|\times|C|}$;
$TC=0$;

foreach $i\in N$
  foreach $j\in H$
    if $x_{ij}=1$
      $TC=TC+c_{ij}O_i+c_{ji}D_i$;
    end if
  end for
end for

foreach $j\in H$
  foreach $l\in C$
    if $r_{jl}=1$
      $TC=TC+\alpha_h\left[c_{jl}\left(O_j-\sum_{i\in N\setminus\{j\}}f_{ji}x_{ij}\right)+c_{lj}\left(D_j-\sum_{i\in N\setminus\{j\}}f_{ij}x_{ij}\right)+c_{jl}\left(\sum_{i\in N}x_{ij}\left(O_i-\sum_{s\in N}f_{is}x_{sj}\right)\right)+c_{lj}\left(\sum_{i\in N}x_{ij}\left(D_i-\sum_{s\in N}f_{si}x_{sj}\right)\right)\right]$;
    end if
  end for
end for

foreach $k\in C$
  foreach $l\in C$
    if $y_{kl}=1$
      $AdjacancyMatrix_{kl}=1$;
      $WeightMatrix_{kl}=c_{kl}\times\alpha_c$;
      $AdjacancyMatrix_{lk}=1$;
      $WeightMatrix_{lk}=c_{lk}\times\alpha_c$;
    end if
  end for
end for
$MinCostRoutes$=Dijkstra*(AdjacancyMatrix, WeightMatrix);*

foreach $k\in C$
  foreach $l\in C$
    if $r_{kk}=1$ and $r_{ll}=1$ and $k<l$
      $flow_{kl}$=Total flow from (central hub k, nodes and secondary hubs allocated to central hub k, and nodes allocated to secondary hubs connected to central hub k) to (central hub l, nodes and secondary hubs allocated to central hub l, and nodes allocated to secondary hubs connected to central hub l);
      $flow_{lk}$=Total flow from (central hub l, nodes and secondary hubs allocated to central hub l, and nodes allocated to secondary hubs connected to central hub l) to (central hub k, nodes and secondary hubs allocated to central hub k, and nodes allocated to secondary hubs connected to central hub k);
      $TC=TC+MinCostRoutes_{kl}\times flow_{kl}+MinCostRoutes_{lk}\times flow_{lk}$;
    end if
  end for
end for

Stop. The optimal objective function value is obtained

Fig. 5. Pseudocode of the evaluation algorithm.

$$Pr_i=\frac{F_i}{\sum_{j=1}^{S}F_j} \tag{38}$$

In which S is the number of chromosomes, and F_i is the fitness value of the chromosome i (fitness value is obtained by using the proposed evaluation algorithm in section 5.2.3).

5.2.5. Crossover

The offspring chromosomes should be determined in this step according to the selected parents at the selection stage. In this step, we use the one-point crossover in such a way that a random point is selected from 1 to n. Then the contents of the chromosomes before and after that point are exchanged. Thus, two child chromosomes are formed. See Fig. (6).

If the nth gene is selected and the nth gene is not a member of the second part of the chromosome (located hubs), it is randomly swapped with one of the genes of the first part of the chromosome, which is a hub. However, there may be some duplicate numbers at this step, and some numbers are not in the sequence. Therefore, changes should be made to the children, which can be seen in Fig. (7).

According to Fig. (7), the duplicated numbers must be removed, and the missing numbers must be added to the sequence in a random order to achieve a feasible solution. For child 1, numbers 4 and 7 are repetitive in the sequence. Therefore, these numbers must be deleted, and numbers 1 and 6 that are not in the sequence must be added. For child 2, numbers 6

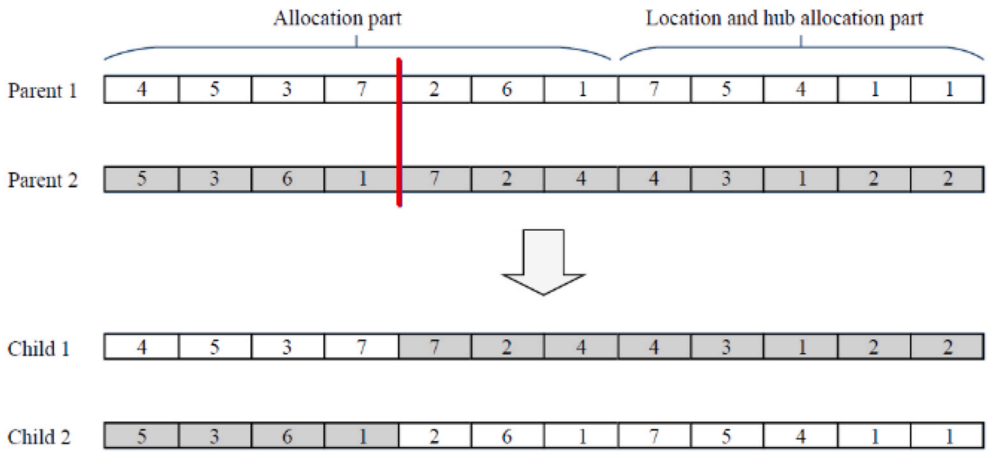

Fig. 6. Crossover.

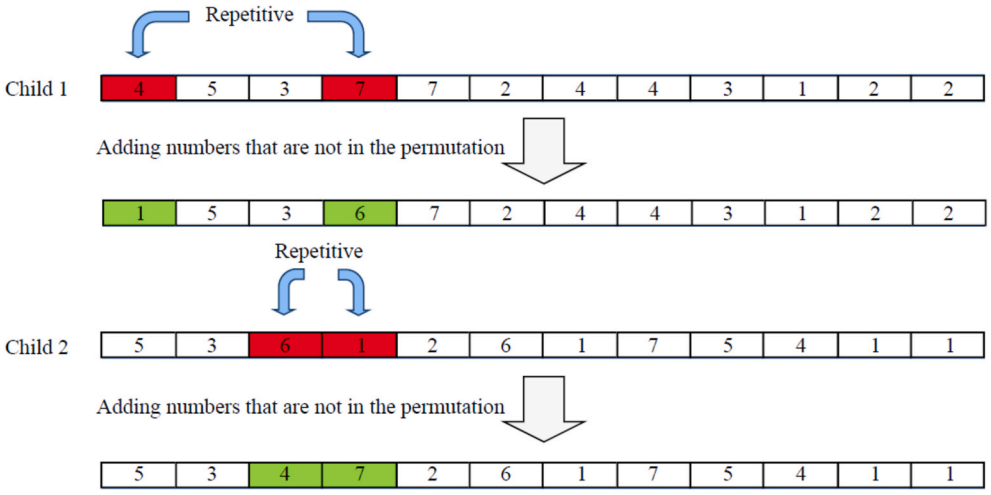

Fig. 7. The amending crossover.

and 1 are repetitive in the sequence. Therefore, they must be deleted, and numbers 4 and 7 that are not in the sequence should be added so that all nodes is allocated to the hubs and a feasible solution is obtained.

5.2.6. *Evaluating children from crossover*

Evaluation is required after generating children in the crossover method. In this regard, the children are decoded, and the objective function value is calculated for them (based on the proposed evaluation algorithm in section 5.2.3).

5.2.7. *Mutation*

In this study, the mutation operator is applied in such a way that two genes are selected from the first part of the chromosome (allocation part). Then, one of the operators of swap, reversion, and scramble is applied to the chromosome with equal probability. In the swap operator, the contents of the two selected genes are swapped. In the reversion operator, the content between the two selected genes is reversed, and in the scramble operator, the content between the two selected genes is randomly distorted. In this study, the mutation is also applied on the second part of the chromosome. In this way, duplicate numbers are first removed from the second part of the chromosome, and the mutation operator is applied. Then, the initially repeated numbers are repeated in the sequence created after the mutation. The mutation process is shown in Fig. (8).

Fig. 8. Mutation.

5.2.8. Evaluating mutated chromosomes

In this step, the decoding process is performed, and the objective function value for each mutated chromosome is calculated (using the proposed evaluation algorithm in section 5.2.3).

5.2.9. Merging solutions and population management

In this step, the children from crossover and mutation are added to the main population, and a larger population is created. Then, the members of the created population are sorted in ascending order according to the fitness value, and the first *npop* members are transferred to the next iteration of the algorithm.

5.2.10. Stopping criteria

In the final step, the stopping condition for the genetic algorithm is determined. In this research, the algorithm continues to reach a certain number of iterations.

5.3. Hybrid variable neighborhood search

In this subsection, we present a variable neighborhood search algorithm that uses a Dijkstra algorithm to calculate the cost of each solution (based on the evaluation algorithm in section 5.2.3). In the proposed variable neighborhood search method, the initial solution is generated similar to the HGA (section 5.2.1). Then the initial solution is stored as a structure with fields x, r, and y. In this structure, x is an $n \times |H|$ matrix that specifies how nodes are allocated to hubs. It is determined from the first part of the initial solution. r is a $|H| \times |C|$ matrix that specifies how hubs are allocated to central hubs. y is a $|C| \times |C|$ matrix that specifies the ring structure. The second and third fields are determined from the second part of the initial solution. Neighborhood search is applied to the obtained structure. In this algorithm, seven neighborhood structures are used to create the neighborhoods of a solution. The neighborhood structures #1 to #6 are adapted from (Shang et al., 2021). Neighborhood structure #7 is concerned with how to create a ring between central hubs.

To avoid being trapped in the local optima, a shaking operator is used, which is a method of creating a neighborhood that is randomly defined for each neighborhood structure separately. In the following, we describe the details of neighborhood structures and shaking operators.

Neighborhood Structure #1. Each non-hub node is allocated to a different hub.

Neighborhood Structure #2. Each secondary hub is allocated to a different central hub.

Neighborhood Structure #3. Each of the non-hub nodes that are members of the set H is replaced by a hub (both central and secondary). Thus, the non-hub node becomes a hub, and the hub becomes a non-hub node. The allocation of nodes should be modified in such a way that no infeasible solution is generated. The allocation modification method is as follows: If the non-hub node is replaced with a secondary hub, the secondary hub, which is now a non-hub node, and the nodes that are currently allocated to the secondary hub are assigned to the new hub. The new hub is assigned to a central hub to which the secondary hub was previously assigned. If the non-hub node is replaced by a central hub, the central hub, which is now a non-hub node, and the secondary hubs and other demand nodes that are currently assigned to the central hub are assigned to the new hub. The new hub connects directly to central hubs to which the central hub was previously connected.

Neighborhood Structure #4. Each of the secondary hubs that are members of the set C is replaced by a central hub. Thus, the secondary hub becomes a central hub, and the central hub becomes a secondary hub. The allocation of nodes should be modified in such a way that no infeasible solution is generated. To modify the allocation, the central hub, which is now a secondary hub, and the secondary hubs that are currently allocated to the central hub are assigned to the new hub. The new hub connects directly to central hubs to which the central hub was previously connected.

Neighborhood structure #5. The allocation for each pair of non-hub nodes changes.

Neighborhood structure #6. The allocation for each pair of secondary hubs changes. **Neighborhood structure #7**. The structure of the ring created between the central hubs changes for each pair of central hubs. First, the ring structure is converted to a permutation, and one of the swap, reversion, and insertion operators are applied to it, and a new ring structure is obtained. For this purpose, the points corresponding to the central hubs are selected from the generated permutation. In the swap operator, two central hubs are swapped. In the reversion operator, the order of the hubs between the two central hubs is reversed. In the insertion operator, the central hub in the second order is immediately placed after the central hub in the first order. Each operator has the same chance to be selected. If one of the seven neighborhoods is selected in one iteration, one of these operators is applied on the solution.

Shaking Operator for Neighborhood Structure #1. A non-hub node and a different hub from the hub to which the non-hub node is allocated are randomly selected, and the non-hub node is allocated to that hub.

Shaking Operator for Neighborhood Structure #2. A secondary hub and a central hub different from the central hub to which the secondary hub is allocated are randomly selected, and the secondary hub is allocated to the central hub.

Shaking Operator for Neighborhood Structure #3. A non-hub node that is a member of the set H and a hub is randomly selected, and their role changes. Thus, the demand point becomes a hub, and the hub becomes a non-hub node. The allocation of nodes should be modified in such a way that no infeasible solution is generated.

Shaking Operator for Neighborhood Structure #4. A secondary hub that is a member of the set C and a central hub are randomly selected, and their role changes. Thus, the secondary hub becomes the central hub, and the central hub becomes the secondary hub. The allocation of nodes should be modified in such a way that no infeasible solution is generated.

Shaking Operator for Neighborhood structure #5. The allocation of a pair of randomly selected non-hub nodes changes.

Shaking Operator for Neighborhood structure #6. The allocation of a pair of randomly selected secondary hubs changes.

Shaking Operator for Neighborhood structure #7. The structure of the ring created between the central hubs changes for two randomly selected central hubs. First, the ring structure is converted to a permutation, and one of the swap, reversion, and insertion operators with equal probability are applied to it, and a new ring structure is obtained.

In the implementation of the algorithm, namely HVNS, to generate the initial solution, first, a solution is randomly generated, and then, by using an improvement algorithm, the generated solution is improved. The improvement algorithm is based on the neighborhood structures #1, #2 and #7, and its steps are as follows:

1 Generate a solution randomly
2 Allocate each non-hub node to the hub that provides the greatest reduction in the objective function value.
3 Allocate each secondary hub to the central hub that provides the greatest reduction in the objective function value. (note that this step is applied for $q > 0$)
4 Change the ring structure related to the solution using neighborhood structure #7 with the best-improvement strategy (note that this step is applied for $p \geq 4$).
5 Stop. An improved solution is obtained.

In the proposed HVNS, the search strategy is a combination of first-improvement and best-improvement strategies. The first-improvement strategy is used for neighborhood structures #1, #3, and #5. The best-improvement strategy is used for neighborhood structures #2, #4, #6, and #7. In the first-improvement strategy, we move to a neighbor that has better objective function value. In the best-improvement strategy,

we move to the best of all neighboring solution (in terms of objective function value).

The flowchart of the proposed HVNS is shown in Fig. (9).

5.4. Relax-and-fix algorithm

Due to the high computational complexity of the model, in this subsection, a Relax-and-Fix (R&F) algorithm is presented to find good feasible solutions to the problem in a reasonable amount of time, which is based on (Ferreira et al., 2010; Marufuzzaman and Eksioglu 2014; Khaleghi and Eydi 2022). In our implementation, the model (**Eqs. 1–18**) is decomposed into some sub-problems based on levels of hierarchy and ring structure, in which binary variables related to certain levels of hierarchy and ring structure are relaxed from the integrality. The algorithm stops when all the sub-problems are solved. The R&F provides an upper bound for the problem. The pseudocode of the proposed R&F algorithm is shown in Fig. (10).

According to Fig. (10), the proposed R&F consists of three iterations. In the first iteration, the integrality is considered only for the binary variables related to locating the central hubs and allocating the non-hub nodes to the central hubs. Other binary variables are relaxed from the integrality. Then the model is solved, and the values of these variables

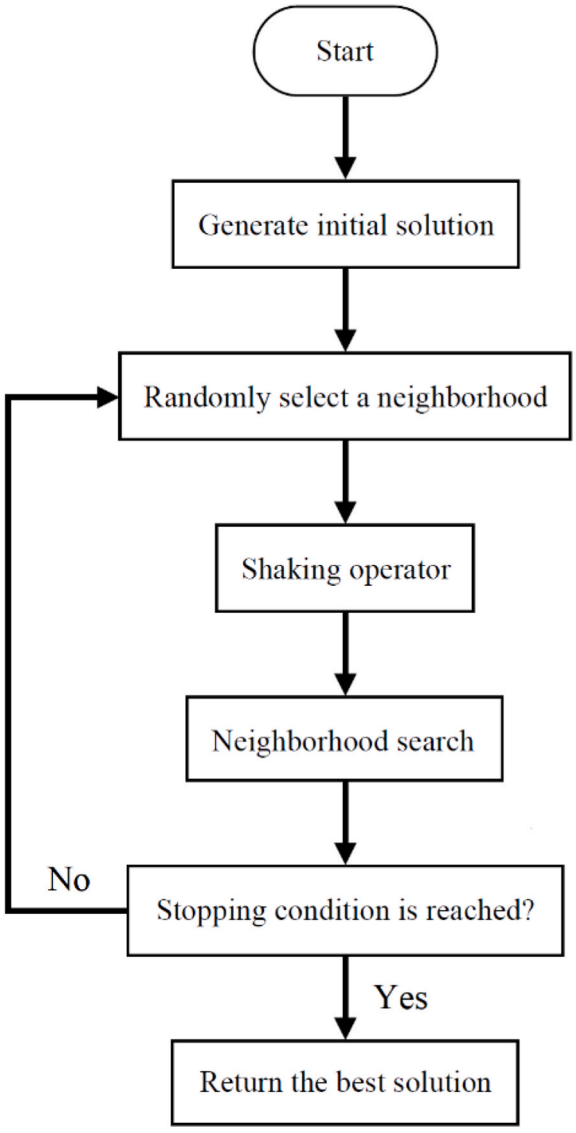

Fig. 9. Flowchart of the proposed HVNS.

Input: Problem instance
Output: A feasible solution for the problem
Set:
$z_{ill} \in \{0,1\}\ \forall\ i \in N, l \in C.$
Set:
$0 \le z_{ijl} \le 1\ \forall\ i \in N, j \in H, l \in C\backslash\{j\}, 0 \le y_{kl} \le 1\ \forall\ k, l \in C, k < l.$
Solve the model and fix the values of $z_{ill}\ \forall\ i \in N, l \in C.$
Set:
$y_{kl} \in \{0,1\}\ \forall\ k \in C, l \in C, k < l.$
Set:
$0 \le z_{ijl} \le 1\ \forall\ i \in N, j \in H, l \in C\backslash\{j\}.$
Solve the model and fix the values of $y_{kl} \in \{0,1\}\ \forall\ k \in C, l \in C, k < l.$
Set:
$z_{ijl} \in \{0,1\}\ \forall\ i \in N, j \in H, l \in C\backslash\{j\}.$
Solve the model
Stop. A feasible solution for the problem is obtained.

Fig. 10. Pseudocode of the proposed R&F algorithm.

are fixed in the model. In the second iteration, the integrality is considered only for variables related to the ring structure, and the remaining binary variables are relaxed from the integrality. Then the model is solved and the variables related to the ring structure are fixed. In the third iteration, the integrality is considered only for variables $z_{ijl}\forall i \in N, j \in H, l \in C\backslash\{j\}$. By solving the model, a feasible solution is obtained to the problem.

5.4.1. Example

In this subsection, to illustrate the procedure of the proposed R&F algorithm, we present the results of a sample problem with fifteen demand nodes based on the CAB dataset (Beasley 2008). The demand nodes are the first fifteen cities in the CAB dataset. The first ten cities in the CAB dataset are selected as candidate nodes for hubs, and the first seven cities are selected as candidate nodes for main hubs. The flow and cost parameters are extracted from the CAB dataset. Four central hubs and four secondary hubs must be located. The discount factors for main hubs and secondary hubs are considered to be 0.9. The results of the R&F algorithm are presented in Tables 3 and 4. In Tables 3 and 4, the values of the variables that are subject to the integrality are bolded. Variables that are not in Tables 3 and 4 take value of zero. According to Tables 3 and 4, in the first iteration, variables $z_{ill}\forall i \in N, l \in C$ have integer values and nodes 1, 4, 6, and 7 are selected as central hubs. Also, the allocation of the demand nodes to central hubs is also specified: nodes 13 and 14 to hub 1, nodes 11 and 15 to hub 4, node 2 to hub 6, and node 10 to hub 7. The values of other variables are between zero and one. In the second iteration, the values of variables $z_{ill}\forall i \in N, l \in C$ of the first iteration are fixed, and the integrality is considered for variables $y_{kl}\forall k \in C, l \in C, k < l$. The formed ring includes arcs (1,6), (1,7), (4,6), and (4,7). In the third iteration, the values of variables $y_{kl}\forall k \in C, l \in C, k < l$ of the second iteration are fixed (note that in the third iteration, the value of the integer variables of the first iteration are also fixed) and the integrality is considered for other integer variables (i.e., $z_{ijl}\forall i \in N, j \in H, l \in C\backslash\{j\}$). Nodes 3, 5, 8, and 9 are selected as secondary hubs. In this iteration, the allocation of secondary hubs to main hubs and the allocation of demand nodes to the secondary hubs are also determined: secondary hub 8 to central hub 4, secondary hubs 3, 5, and 9 to hub 6, and node 12 to secondary hub 8. Therefore, a feasible solution is obtained for the problem. The objective function value is 2606115969.90.

6. Computational results

In this section, we present the computational results of the solving model by using the proposed Benders decomposition algorithm, HGA, HVNS, and R&F algorithm for fifty instances of the problem (based on

Table 3
Results of variables z_{ijl} in different iterations of the R&F.

Iteration #1				Iteration #2				Iteration #3			
i	j	l	Value	i	j	l	Value	i	j	l	Value
1	1	1	**1.000**	1	1	1	1.000	1	1	1	1.000
2	6	6	**1.000**	2	6	6	1.000	2	6	6	1.000
3	3	6	1.000	3	3	6	1.000	3	3	6	**1.000**
4	4	4	**1.000**	4	4	4	1.000	4	4	4	1.000
5	5	1	0.187	5	5	4	0.548	5	5	6	**1.000**
5	5	4	0.548	5	5	6	0.452	6	6	6	1.000
5	5	6	0.266	6	6	6	1.000	7	7	7	1.000
6	6	6	**1.000**	7	7	7	1.000	8	8	4	**1.000**
7	7	7	**1.000**	8	8	4	0.718	9	9	6	**1.000**
8	8	4	0.799	8	8	7	0.282	10	7	7	1.000
8	8	7	0.201	9	9	4	0.592	11	4	4	1.000
9	9	4	0.592	9	9	6	0.408	12	8	4	**1.000**
9	9	6	0.408	10	7	7	1.000	13	1	1	1.000
10	7	7	**1.000**	11	4	4	1.000	14	1	1	1.000
11	4	4	**1.000**	12	8	4	0.718	15	4	4	1.000
12	8	4	0.799	12	8	7	0.282	–	–	–	–
12	8	7	0.201	13	1	1	1.000	–	–	–	–
13	1	1	**1.000**	14	1	1	1.000	–	–	–	–
14	1	1	**1.000**	15	4	4	1.000	–	–	–	–
15	4	4	**1.000**	–	–	–	–	–	–	–	–

Table 4
Results of variables y_{kl} in different iterations of the R&F.

Iteration #1			Iteration #2			Iteration #3		
k	l	Value	k	l	Value	k	l	Value
1	4	0.450	1	6	**1.000**	1	6	1.000
1	6	1.000	1	7	**1.000**	1	7	1.000
1	7	0.550	4	6	**1.000**	4	6	1.000
4	6	0.550	4	7	**1.000**	4	7	1.000
4	7	1.000	–	–	–	–	–	–
6	7	0.450	–	–	–	–	–	–

six test problems). The computations are done on the USA423 dataset. This dataset includes distances and volume of demand flow between 423 demand points of US airlines collected over three months (Peiró et al., 2014). Also, discount factors for main hubs and secondary hubs are set at 0.9. Candidate nodes for secondary hubs and central hubs are the first $|H|$ and the first $|C|$ nodes of the related test problem, respectively. The number of nodes and candidate nodes for hubs for different test problems are listed in Table 5.

6.1. Results of the proposed Benders decomposition algorithm

In this subsection, we present the computational results of the solving model by using the proposed Benders decomposition algorithm. The time limit for the CPLEX solver and Benders process is set at 14400 s. Note that to get the initial integer solution for the Benders algorithm, we solve the **Eqs. (2)–(11) and (17), (18)** as a feasibility problem without any objective function. The results of solving test Problems #1 to #6 using the CPLEX solver and the proposed accelerated Benders decomposition algorithm are presented in **Tables 6–11** Columns in these tables are respectively the number of main and secondary hubs, the optimal objective function value, the optimal locations of the main and secondary hubs, CPLEX solution time, the number of branch-and-bound nodes used in the CPLEX solver, the solution time of the accelerated Benders decomposition algorithm and the number of iterations of the Benders algorithm. Also, the last column is related to the obtained lower and upper bounds for the cases that are not solved to optimality within the time limit. Note that in the last column, the cases solved to optimality are shown by "-". Also, in **Table 9**, in the CPLEX solution time column, cases that cannot be solved by the CPLEX solver due to the out-of-memory error are denoted by "Memory".

Table 5
Characteristics of test problems.

Test problem number	$\|N\|$	$\|H\|$	$\|C\|$
1	15	10	5
2	25	15	10
3	50	20	15
4	100	25	20
5	200	30	25
6	423	35	30

According to Tables 6–11, all instances of test problem #1 have been optimized using the proposed Benders decomposition algorithm within the time limit. Also, three instances of test problem #2 and two instances of test problem #3 have been solved to optimality. For problems with less than a hundred nodes, the performance of the CPLEX is better than the Benders decomposition in terms of solution time. Also, all instances with less than a hundred nodes and two instances of test problem #4 with a hundred nodes have been optimized using CPLEX. The average solution time of the CPLEX is 284.76 s. By increasing the instance size, CPLEX solution time increases. However, for seven instances of test problem #4 and all instances of test problem #5 and test problem #6, CPLEX cannot handle the instance size. For these cases, only the results of the Benders decomposition algorithm have been reported. Generally, by increasing the number of nodes, the performance of the Benders decomposition algorithm deteriorates, and the number of iterations decreases within the time limit. It can be concluded that the proposed Benders decomposition algorithm can solve more instances of the problem. Although, in some cases, it does not prove the optimality within our time limit, it produces solutions with known quality. We can say that the efficiency of the proposed Benders decomposition algorithm directly depends on $|N|$. Fig. (11) provides a comparison between the lower bound and upper bound of the Benders decomposition algorithm and the optimal objective function value derived from the CPLEX solver for instances that have not been solved to optimality within the time limit by Benders decomposition but have been optimized by CPLEX.

Fig. 12 provides a comparison between convergence rates in classical Benders decomposition and proposed accelerated Benders decomposition for test problem #1 with $p = 3$ and $q = 1$.

6.2. Results of the proposed heuristic algorithms

6.2.1. Parameters setting of the proposed HGA

Calibrating the parameters of meta-heuristic algorithms has a direct effect on improving the performance of these algorithms. In this

Table 6
Results of accelerated Benders decomposition for test problem #1.

Instance number	p	q	Optimal obj. value	Main hubs	Secondary hubs	CPLEX		Accelerated Benders decomposition		
						CPU (s)	Nodes	CPU (s)	Iterations	[LB,UB]
1	3	0	64398855.90	2,4,5	–	0.28	0	3.01	7	–
2		1	62535494.00	2,4,5	10	0.45	0	56.43	58	–
3		2	62366530.60	2,4,5	3,10	0.63	0	621.27	231	–
4		3	62228254.60	2,4,5	3,8,10	0.44	0	1493.89	346	–
5	4	0	63834439.00	2,3,4,5	–	0.14	0	6.13	13	–
6		1	61971077.80	2,3,4,5	10	0.45	0	110.48	100	–
7		2	61832801.80	2,3,4,5	8,10	0.55	0	1065.60	323	–
8		3	61779568.60	2,3,4,5	1,8,10	0.61	0	2403.98	478	–
9		4	61754974.80	2,3,4,5	1,7,8,10	0.48	0	3172.12	550	–

Table 7
Results of accelerated Benders decomposition for test problem #2.

Instance number	p	q	Optimal obj. value	Main hubs	Secondary hubs	CPLEX		Accelerated Benders decomposition		
						CPU (s)	Nodes	CPU(s)	Iterations	[LB,UB]
10	3	0	73147976.10	2,4,5	–	0.30	0	10.11	13	–
11		1	71093039.10	2,4,5	10	5.64	0	5442.18	239	–
12		2	70293126.40	2,4,5	10,15	7.49	0	TimeLimit	248	[57024795.70,72348410.00]
13		3	70067738.50	2,4,5	3,10,15	8.94	0	TimeLimit	273	[51673669.30,70944550.00]
14	4	0	71885233.80	2,3,4,5	–	0.66	0	37.28	36	–
15		1	69830296.80	2,3,4,5	10	8.48	0	TimeLimit	335	[68021219.20,70546420.00]
16		2	69030384.10	2,3,4,5	10,15	6.44	0	TimeLimit	259	[51909108.10,73295530.00]
17		3	68889210.10	2,3,4,5	8,10,15	10.39	0	TimeLimit	241	[45013932.70,74131700.00]
18		4	68755840.50	2,3,4,5	8,10,11,15	9.16	0	TimeLimit	220	[42416732.40,76852200.00]

Table 8
Results of accelerated Benders decomposition for test problem #3.

Instance number	p	q	Optimal obj. value	Main hubs	Secondary hubs	CPLEX		Accelerated Benders decomposition		
						CPU (s)	Nodes	CPU(s)	Iterations	[LB,UB]
19	3	0	1735768948.50	2,7,12	–	5.83	0	594.42	71	–
20		1	1705392341.70	4,7,12	2	531.860	0	TimeLimit	104	[1497259667.30,1737039000.00]
21		2	1688869545.79	4,7,12	2,10	287.56	0	TimeLimit	66	[1132884917.90,1795767000.00]
22		3	1684825937.00	4,7,12	2,9,10	436.70	3	TimeLimit	47	[911734899.89,1823558000.00]
23	4	0	1708362168.89	2,4,7,12	–	22.59	0	5558.74	182	–
24		1	1691839373.00	2,4,7,12	10	685.23	0	TimeLimit	91	[1332661211.10,1727056000.00]
25		2	1684473600.50	4,7,12,13	2,10	592.42	27	TimeLimit	48	[940297675.30,1763996000.00]
26		3	1680429991.70	4,7,12,13	2,9,10	449.27	29	TimeLimit	33	[746588673.10,1968938000.00]
27		4	1677150232.70	4,7,12,13	2,9,10,15	570.39	25	TimeLimit	27	[636240359.30,1976206000.00]

Table 9
Results of accelerated Benders decomposition for test problem #4.

Instance number	p	q	Optimal obj. value	Main hubs	Secondary hubs	CPLEX		Accelerated Benders decomposition		
						CPU (s)	Nodes	CPU(s)	Iterations	[LB,UB]
28	3	0	12247695091.29	2,7,12	–	360.34	0	TimeLimit	120	[12147674491.40,1.22765E10]
29		1	–	–	–	Memory	–	TimeLimit	27	[8792751045.40,1.21105E10]
30		2	–	–	–	Memory	–	TimeLimit	10	[5815333894.90,1.47599E10]
31		3	–	–	–	Memory	–	TimeLimit	11	[4972704675.29,2.55051E10]
32	4	0	12038633898.29	2,4,7,12	–	4254.27	32	TimeLimit	92	[11605762287.50,1.20985E10]
33		1	–	–	–	Memory	–	TimeLimit	25	[7889805555.20,1.24427E10]
34		2	–	–	–	Memory	–	TimeLimit	12	[5569660347.79,1.45308E10]
35		3	–	–	–	Memory	–	TimeLimit	11	[4577068833.69,1.48255E10]
36		4	–	–	–	Memory	–	TimeLimit	8	[3390253747.90,1.52453E10]

research, we use the Taguchi design of experiment method to tune the parameters of the proposed HGA in solving test problems #1 to #4. The parameters of test problems #5 and #6 are tuned by the trial and error method. To apply the Taguchi method, we first identify factors and response. In this study, the factors are parameters of the HGA that have different levels, and the response is the objective function value of the problem in each experiment. One way to study the effect of different levels of factors on the mean of the response variable is to use the "Factorial design", which has a high computational cost. Because all possible combinations for the parameters need to be considered. To save computational costs and do fewer experiments, Taguchi proposes the "Fractional Factorial experiments" scheme (Roy 2001). According to this approach, the factors influencing the experiment include controllable factors (S) and noise factors (N). Of these, only controllable factors can be adjusted. To this end, a process was proposed by Taguchi that reduces the variations of the response variable by controlling N with the

Table 10
Results of accelerated Benders decomposition for test problem #5.

Instance number	p	q	Optimal obj. value	Accelerated Benders decomposition		
				CPU (s)	Iterations	[LB,UB]
37	3	0	–	TimeLimit	38	[68706095136.19,7.03591E10]
38		1	–	TimeLimit	10	[42040344423.92,1.14064E11]
39		2	–	TimeLimit	6	[33095844816.39,1.04087E11]
40		3	–	TimeLimit	6	[26436682418.14,1.05650E11]
41	4	0	–	TimeLimit	59	[57288907348.55,7.24936E10]
42		1	–	TimeLimit	8	[39463385066.79,8.12074E10]
43		2	–	TimeLimit	5	[25436341564.90,1.11690E11]
44		3	–	TimeLimit	7	[22072317926.30,8.80848E10]
45		4	–	TimeLimit	5	[18135622021.33,1.28094E11]

Table 11
Results of accelerated Benders decomposition for test problem #6.

Instance number	p	q	Optimal obj. value	Accelerated Benders decomposition		
				CPU (s)	Iterations	[LB,UB]
46	3	0	–	TimeLimit	11	[543313047329.89,5.98141E11]
47		1	–	TimeLimit	2	[320168751293.50,7.77035E11]
48		2	–	TimeLimit	1	[226812431861.89,9.95301E11]
49	4	0	–	TimeLimit	11	[387737898585.59,6.52505E11]
50		1	–	TimeLimit	2	[24403509172.72,1.07774E12]

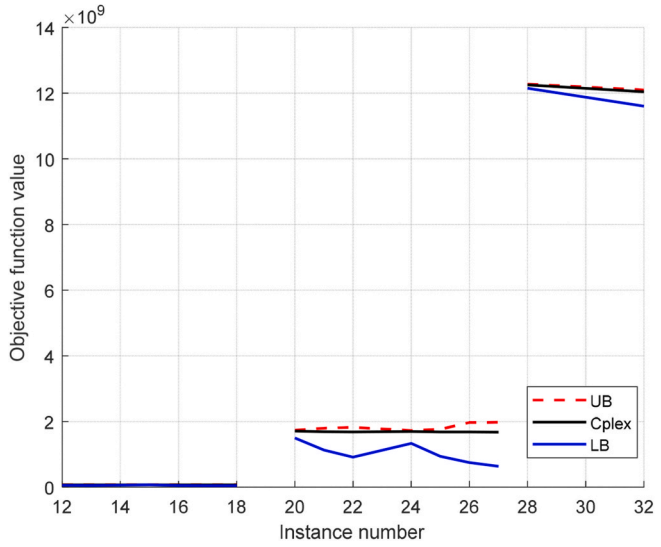

Fig. 11. A comparison between the lower and upper bound of Benders decomposition algorithm and optimal objective value.

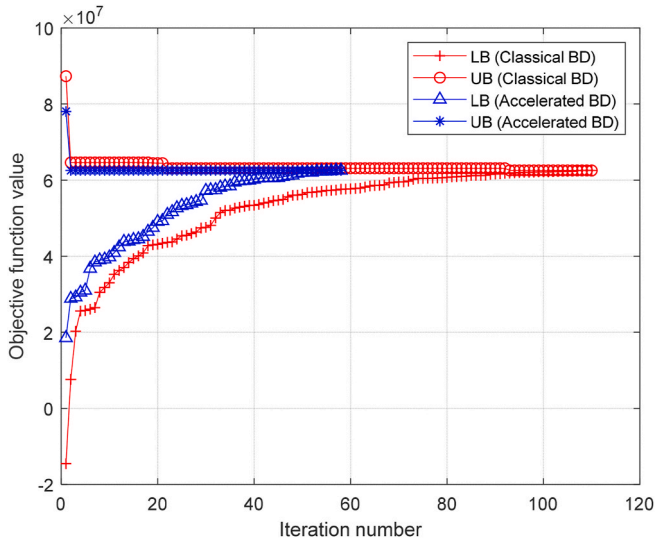

Fig. 12. A comparison between the performance of classical and accelerated Benders decomposition algorithms for test instance #2.

help of orthogonal arrays. In the present study, we use a Signal to Noise (S/N) ratio to adjust the parameters by considering several replications for each experiment. The (S/N) ratio is calculated as **Eq. (39)**:

$$S/N = -10\log\left(\frac{1}{r}\sum_{i=1}^{r} y_i^2\right) \tag{39}$$

In which r is the number of replications in experiments and y_i is the response in replication i.

The different levels considered for HGA parameters are shown in Table 12. Depending on the number of factors and their levels, the orthogonal array L^9 is used. The (S/N) ratio chart for the HGA is shown in Fig. (13). According to Fig. (13), in the proposed HGA, parameters Np, Pc, Pm, and MaxIt should be set at levels 3, 2, 1, and 2. In Table 13, the values of the tuned parameters for test problems #1 to #6 are reported.

Table 12
Different levels of parameters of GA.

Parameter	L1	L2	L3
Population size (Np)	80	100	120
Crossover rate (Pc)	0.7	0.8	0.9
Mutation rate (Pm)	0.1	0.2	0.3
Maximum number of iterations (MaxIt)	100	150	200

6.2.2. *Results of the proposed HGA*

In this subsection, we present the results of the HGA and compare them with the results of the CPLEX. In the HGA, we run each instance of the problem ten times, and the best solution is reported. The results of solving test problems #1 to #6 are reported in Tables 14–19. According to Tables 14–19, for all instances that can be solved to optimality using the CPLEX, the average gap of the proposed HGA is 2.55%. The minimum gap of the HGA is 0.00%, and the maximum gap of the HGA is

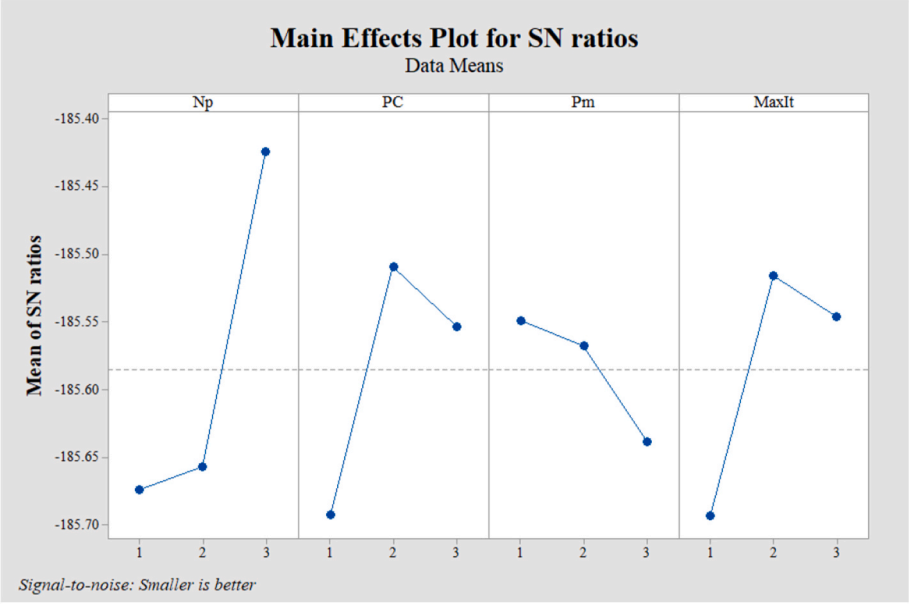

Fig. 13. S/N chart.

Table 13
Tuned parameters of HGA.

Parameter	Value for test problems #1-#4	Value for test problem #5	Value for test problem #6
Np	120	145	170
Pc	0.8	0.8	0.7
Pm	0.1	0.4	0.3
MaxIt	150	170	145

7.95%. The average solution time of the HGA is 64.49 s. In six instances of the problem, the HGA has found the optimal solution to the problem. The average solution time of the HGA algorithm for instances #1 to #50 is 440.52 s. As the size of the problem increases, the solution time of the HGA increases.

6.2.3. *Results of the proposed HVNS*

In this subsection, we present the results of the HVNS algorithm and compare them with the results of the CPLEX. In the HVNS algorithm, after performing various experiments, the maximum number of iterations was considered to be 250. Also, we introduce a new parameter, namely G. If in G consecutive iterations of the algorithm, no improvement in the objective function value is observed, the algorithm stops. Based on some computational experiments on instances with different sizes, parameter G is considered to be $3 \times$ (the number of possible neighborhood structures for each instance). To obtain the possible

Table 14
Results of HGA for test problem #1.

Instance number	p	q	Optimal obj. value	CPLEX		HGA		
				CPU (s)	Nodes	Best solution	CPU (s)	Gap (%)
1	3	0	64398855.90	0.28	0	64398855.90	26.07	0.00
2		1	62535494.00	0.45	0	62535494.00	22.26	0.00
3		2	62366530.60	0.63	0	62397218.70	22.81	0.05
4		3	62228254.60	0.44	0	62394869.80	25.17	0.27
5	4	0	63834439.00	0.14	0	63834439.00	23.25	0.00
6		1	61971077.80	0.45	0	61971077.80	24.62	0.00
7		2	61832801.80	0.55	0	61847834.20	26.94	0.02
8		3	61779568.60	0.61	0	61808208.00	28.52	0.05
9		4	61754974.80	0.48	0	61754974.80	29.64	0.00

Table 15
Results of HGA for test problem #2.

Instance number	p	q	Optimal obj. value	CPLEX		HGA		
				CPU (s)	Nodes	Best solution	CPU (s)	Gap (%)
10	3	0	73147976.10	0.30	0	73147976.10	40.31	0.00
11		1	71093039.10	5.64	0	72408101.00	46.26	1.85
12		2	70293126.40	7.49	0	72220955.70	43.07	2.74
13		3	70067738.50	8.94	0	72300744.10	42.66	3.19
14	4	0	71885233.80	0.66	0	71968313.60	43.59	0.12
15		1	69830296.80	8.48	0	71277944.20	45.71	2.07
16		2	69030384.10	6.44	0	71618795.00	46.23	3.75
17		3	68889210.10	10.39	0	72139718.00	46.96	4.72
18		4	68755840.50	9.16	0	69680379.40	53.03	1.34

Table 16
Results of HGA for test problem #3.

Instance number	p	q	Optimal obj. value	CPLEX		HGA		
				CPU (s)	Nodes	Best solution	CPU (s)	Gap (%)
19	3	0	1735768948.50	5.83	0	1788792830.90	91.82	3.05
20		1	1705392341.70	531.860	0	1739635641.10	94.97	2.01
21		2	1688869545.79	287.56	0	1784983602.50	76.62	5.69
22		3	1684825937.00	436.70	3	1818704887.90	78.34	7.95
23	4	0	1708362168.89	22.59	0	1759675026.30	78.14	3.00
24		1	1691839373.00	685.23	0	1785079125.00	79.04	5.51
25		2	1684473600.50	592.42	27	1772497269.20	80.89	5.23
26		3	1680429991.70	449.27	29	1759078186.00	81.47	4.68
27		4	1677150232.70	570.39	25	1769641837.90	90.12	5.51

Table 17
Results of HGA for test problem #4.

Instance number	p	q	Optimal obj. value	CPLEX		HGA		
				CPU (s)	Nodes	Best solution	CPU (s)	Gap (%)
28	3	0	12247695091.29	360.34	0	12767510279.00	284.00	4.24
29		1	–	Memory	–	13172000595.80	194.35	–
30		2	–	Memory	–	12935593302.30	203.92	–
31		3	–	Memory	–	13043503815.00	190.90	–
32	4	0	12038633898.29	4254.27	32	12862281189.70	197.68	6.84
33		1	–	Memory	–	13039712828.00	210.12	–
34		2	–	Memory	–	13206218601.30	205.57	–
35		3	–	Memory	–	13095796795.50	195.17	–
36		4	–	Memory	–	13313717989.30	196.07	–

Table 18
Results of HGA for test problem #5.

Instance number	p	q	Optimal obj. value	HGA		
				Best solution	CPU (s)	Gap (%)
37	3	0	–	80366528412.20	938.01	–
38		1	–	81875323402.00	930.40	–
39		2	–	79712568878.60	913.84	–
40		3	–	79798619230.00	961.85	–
41	4	0	–	82855331558.40	925.61	–
42		1	–	79216141604.30	980.17	–
43		2	–	81486357431.40	928.60	–
44		3	–	81781033663.10	968.73	–
45		4	–	81599076228.40	916.03	–

Table 19
Results of HGA for test problem #6.

Instance number	p	q	Optimal obj. value	HGA		
				Best solution	CPU (s)	Gap (%)
46	3	0	–	743805403820.30	2022.97	–
47		1	–	677079982480.10	2189.13	–
48		2	–	757077203124.80	2046.78	–
49	4	0	–	752492119980.00	2008.83	–
50		1	–	751615068952.20	2028.69	–

neighborhood structures, note that when $p < 4$, only one ring structure is possible, and therefore, there is no need to use neighborhood structure #7. When $q = 0$, the neighborhood structures #2, #4, and #6 are not possible; because these neighborhood structures are defined for secondary hubs. When $q = 1$, neighborhood structure #6 is not possible; because this neighborhood structure is defined for at least two secondary hubs. The possible neighborhood structures for different values of p and q for the HVNS are presented in Table 20.

In the proposed HVNS, we run each instance of the problem ten times, and the best solution is reported. The results of solving test problems #1 to #6 are reported in Tables 21–26. In Tables 21–26, in columns "initial solution" and "Improved solution", the initial random solution and the improved solution by using the proposed improvement algorithm are presented, respectively. According to Tables 21–26, for all instances that can be solved to optimality by the CPLEX, the average gap of the proposed HVNS is 0.41%. The minimum gap is 0.00%, and the maximum gap is 2.98%. The average solution time of the HVNS is 56.35 s. In twelve instances of the problem, the HVNS algorithm has found the optimal solution to the problem. The average solution time of the HVNS algorithm for instances #1 to #50 is 404.08 s. As the size of the problem increases, the solution time of the HVNS increases.

6.2.4. *Results of the proposed R&F*

In this subsection, we present the results of the R&F algorithm and compare them with the results of the CPLEX. The results of solving test problems #1 to #4 are reported in Tables 27–30. Note that test problems #5 and #6 could not be solved due to the out-of-memory error. According to Tables 27–30, for all instances that can be solved to optimality using the CPLEX, the average gap of the proposed R&F is 0.22%. The minimum gap of the R&F is 0.00%, and the maximum gap of the R&F is 3.10%. The average solution time of the R&F is 63.85 s. In twenty-one instances of the problem, the R&F has found the optimal solution to the problem. The average solution time of the R&F algorithm for instances #1 to #36 is 2791.63 s. As the size of the problem increases, the solution time of the R&F increases.

Table 20
Possible neighborhood structures for different values of p and q.

Number of central hubs	Number of secondary hubs	Possible neighborhood structures
$p < 4$	$q = 0$	#1, #3, and #5
$p \geq 4$	$q = 0$	#1,#3, #5, and #7
$p < 4$	$q = 1$	#1,#2,#3, #4, and #5
$p \geq 4$	$q = 1$	#1,#2,#3,#4, #5, and #7
$p < 4$	$q > 1$	#1,#2,#3,#4, #5, and #6
$p \geq 4$	$q > 1$	#1,#2,#3,#4,#5, #6, and #7

Table 21
Results of HVNS for test problem #1.

Instance number	p	q	Optimal obj. value	CPLEX		HVNS				
				CPU (s)	Nodes	Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
1	3	0	64398855.90	0.28	0	72990234.10	72019790.70	64398855.90	0.75	0.00
2		1	62535494.00	0.45	0	82463544.40	78403226.40	62535494.00	1.20	0.00
3		2	62366530.60	0.63	0	82115169.00	75838186.30	62366530.60	1.42	0.00
4		3	62228254.60	0.44	0	86979062.60	81543897.10	62302995.70	1.84	0.12
5	4	0	63834439.00	0.14	0	79442087.20	73082443.90	63834439.00	1.09	0.00
6		1	61971077.80	0.45	0	113788490.80	96204070.60	61971077.80	1.29	0.00
7		2	61832801.80	0.55	0	118797516.80	79127624.10	61832801.80	2.65	0.00
8		3	61779568.60	0.61	0	136798485.90	135499380.80	61808208.00	2.05	0.05
9		4	61754974.80	0.48	0	359046343.90	340591910.00	61754974.80	1.61	0.00

Table 22
Results of HVNS for test problem #2.

Instance number	p	q	Optimal obj. value	CPLEX		HVNS				
				CPU (s)	Nodes	Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
10	3	0	73147976.10	0.30	0	98282018.30	97668848.50	73147976.10	2.66	0.00
11		1	71093039.10	5.64	0	159744654.10	158748009.60	72064784.90	4.82	1.37
12		2	70293126.40	7.49	0	140048027.20	91344281.90	70867651.20	5.82	0.82
13		3	70067738.50	8.94	0	92612243.70	87836476.30	70768119.10	11.59	1.00
14	4	0	71885233.80	0.66	0	484240563.80	412952744.50	71885233.80	8.25	0.00
15		1	69830296.80	8.48	0	152029388.40	98318847.60	69830296.80	8.71	0.00
16		2	69030384.10	6.44	0	99339104.90	93303781.40	69696927.20	8.78	0.97
17		3	68889210.10	10.39	0	403532222.70	98886119.10	69621811.10	8.56	1.06
18		4	68755840.50	9.16	0	183368531.70	128431178.50	68755840.50	16.34	0.00

Table 23
Results of HVNS for test problem #3.

Instance number	p	q	Optimal obj. value	CPLEX		HVNS				
				CPU (s)	Nodes	Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
19	3	0	1735768948.50	5.83	0	4374841015.00	2884113940.00	1744613396.50	35.01	0.51
20		1	1705392341.70	531.860	0	3917373205.90	3577448204.50	1728424225.60	38.46	1.35
21		2	1688869545.79	287.56	0	12510910962.90	3578341900.30	1689203170.80	98.68	0.02
22		3	1684825937.00	436.70	3	5809257189.70	3176305494.90	1687125621.40	56.42	0.14
23	4	0	1708362168.89	22.59	0	4154096993.00	3245694627.30	1708908564.10	75.79	0.03
24		1	1691839373.00	685.23	0	3189384917.40	3119396065.50	1691839373.00	131.52	0.00
25		2	1684473600.50	592.42	27	3879900686.30	3117636287.10	1688753487.90	111.15	0.25
26		3	1680429991.70	449.27	29	10880043645.90	5584615117.40	1685629055.70	107.73	0.31
27		4	1677150232.70	570.39	25	12287731571.70	3403421666.40	1684345594.50	101.24	0.43

Table 24
Results of HVNS for test problem #4.

Instance number	p	q	Optimal obj. value	CPLEX		HVNS				
				CPU (s)	Nodes	Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
28	3	0	12247695091.29	360.34	0	29358911308.80	27281805989.10	12310707418.80	567.37	0.51
29		1	–	Memory	–	36636925888.20	25968405004.10	11898133247.20	197.68	–
30		2	–	Memory	–	51637294038.40	41658647786.70	12964563017.90	137.97	–
31		3	–	Memory	–	29204153529.00	19201474402.80	12359623494.90	254.38	–
32	4	0	12038633898.29	4254.27	32	30168812809.60	14950600294.20	12397203504.50	221.33	2.98
33		1	–	Memory	–	18837893580.10	13333999576.30	12550096866.60	460.64	–
34		2	–	Memory	–	61250724837.50	17306780614.10	12388525463.40	147.77	–
35		3	–	Memory	–	33415034232.00	17114042697.30	12293246601.30	144.12	–
36		4	–	Memory	–	85493371784.60	19321801146.00	12571922397.70	364.21	–

6.2.5. *Comparative study*

In this subsection, we provide a comparison between the CPLEX and the proposed heuristic solution methods in terms of objective function value and efficiency. Table 31 provides a comparison between the HGA, HVNS, and R&F for the instances that cannot be solved by the CPLEX. In Table 31, for each algorithm, the gap between the solutions from the algorithm and the best-known solution is reported. Note that for the cases with 200 nodes and more for $q = 0$, the best-known solutions are obtained by the Benders decomposition algorithm. According to Table 31, the average gaps from the best-known solution for the HGA, HVNS, and R&F are 8.14%, 2.17%, and 0.52%, respectively. The minimum gap for all algorithms is 0.00%. The maximum gap for the HGA, HVNS, and R&F are 24.35%, 7.41%, and 3.63%, respectively. The average solution times of the HGA, HVNS, and R&F are 959.80, 884.28, and 14092.46 s, respectively.**OFV**: Objective Function Value; **CT**: CPU Time (seconds); **GBKS**: Gap from the Best-Known Solution.

Table 25
Results of HVNS for test problem #5.

Instance number	p	q	Optimal obj. value	HVNS				
				Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
37	3	0	–	171260350469.50	101540020814.60	72619825797.10	2450.09	–
38		1	–	157928448245.90	117284931003.60	77469303284.60	909.33	–
39		2	–	609540378877.80	103726875589.80	76534974820.70	636.61	–
40		3	–	162943223298.40	128885675146.00	75786745327.10	393.98	–
41	4	0	–	152147001407.90	102049661129.10	77344789511.80	1166.35	–
42		1	–	224846342152.60	118769549143.70	75841235328.40	581.55	–
43		2	–	179594647624.60	101538201660.40	77911519192.90	347.43	–
44		3	–	217264363374.20	123553954446.90	78057073357.10	686.57	–
45		4	–	255164974581.60	223632930475.80	79476799537.70	211.85	–

Table 26
Results of HVNS for test problem #6.

Instance number	p	q	Optimal obj. value	HVNS				
				Initial solution	Improved solution	Best solution	CPU (s)	Gap (%)
46	3	0	–	4044142821877.20	822048293511.70	631663136901.29	2798.43	
47		1	–	1105438537127.60	918564167793.10	700606688928.20	1811.86	
48		2	–	1145574798213.60	776603594611.50	727795264622.90	1352.27	
49	4	0	–	1060845643354.00	905706655965.00	690287375066.50	2281.75	
50		1	–	1017644448476.90	948966184011.20	704166773708.70	1235.03	

Table 27
Results of R&F for test problem #1.

Instance number	p	q	Optimal obj. value	CPLEX		R&F		
				CPU (s)	Nodes	Feasible solution	CPU (s)	Gap (%)
1	3	0	64398855.90	0.28	0	64398855.90	0.17	0.00
2		1	62535494.00	0.45	0	62535494.00	0.29	0.00
3		2	62366530.60	0.63	0	62366530.60	0.39	0.00
4		3	62228254.60	0.44	0	62228254.60	0.43	0.00
5	4	0	63834439.00	0.14	0	63834439.00	0.17	0.00
6		1	61971077.80	0.45	0	61971077.80	0.24	0.00
7		2	61832801.80	0.55	0	61832801.80	0.26	0.00
8		3	61779568.60	0.61	0	61779568.60	0.25	0.00
9		4	61754974.80	0.48	0	61754974.80	0.26	0.00

Table 28
Results of R&F for test problem #2.

Instance number	p	q	Optimal obj. value	CPLEX		R&F		
				CPU (s)	Nodes	Feasible solution	CPU (s)	Gap (%)
10	3	0	73147976.10	0.30	0	73147976.10	0.70	0.00
11		1	71093039.10	5.64	0	71093039.10	7.84	0.00
12		2	70293126.40	7.49	0	70293126.40	13.02	0.00
13		3	70067738.50	8.94	0	70774813.90	5.29	1.01
14	4	0	71885233.80	0.66	0	74117185.10	2.54	3.10
15		1	69830296.80	8.48	0	69830296.80	5.77	0.00
16		2	69030384.10	6.44	0	69030384.10	3.23	0.00
17		3	68889210.10	10.39	0	69030384.10	3.29	0.20
18		4	68755840.50	9.16	0	68755840.50	3.73	0.00

The convergence diagrams for the two proposed hybrid algorithms for instance #2 with $p = 3$ and $q = 1$ are shown in Fig. (14).

Fig. (15) provides a comparison between the results of the CPLEX, HGA, HVNS, and R&F for test problems #1 to #6 with different values of p and q. According to Fig. (15), in terms of median value, the performance of the proposed R&F is better than the proposed HGA in solving test instances #1, #2, and #4. Also, the performance of the proposed HVNS is better than the HGA and R&F in solving test problem #3. In test problems #5 and #6, the performance of the HVNS is better than HGA.

Fig. (16) provides a comparison between the objective function value and solution time in the heuristic algorithms and the CPLEX solver. According to Fig. (16b), by increasing the size of the problem from instance #28 to #32, a significant increase in the CPLEX solution time is observed (solution time increases from 360.34 s to 4254.27 s). While the solution time in heuristic algorithms does not increase exponentially.

Fig. (17) provides a comparison between the objective function value and solution time in the proposed solution methods for the instances that cannot be solved to optimality using the CPLEX. According to Fig. (17a), in all instances except instances #30 and #47, the objective function value of the HVNS is lower than that of the HGA. In instances #29-#31 and #33-#36, the objective function value of the R&F is lower than that of the HGA. Also, in instances #29, #30 and #33-#36, the objective function value of the R&F is lower than that of the HVNS. According to Fig. (17b), the solution time in the R&F algorithm is higher than the

Table 29
Results of R&F for test problem #3.

Instance number	p	q	Optimal obj. value	CPLEX		R&F		
				CPU (s)	Nodes	Feasible solution	CPU (s)	Gap (%)
19	3	0	1735768948.50	5.83	0	1735768948.50	10.22	0.00
20		1	1705392341.70	531.860	0	1705392341.70	515.74	0.00
21		2	1688869545.79	287.56	0	1688869545.79	184.05	0.00
22		3	1684825937.00	436.70	3	1693142204.30	205.82	0.49
23	4	0	1708362168.89	22.59	0	1708362168.89	7.26	0.00
24		1	1691839373.00	685.23	0	1691839373.00	48.94	0.00
25		2	1684473600.50	592.42	27	1694497815.80	134.76	0.60
26		3	1680429991.70	449.27	29	1689720055.20	121.23	0.55
27		4	1677150232.70	570.39	25	1685676446.40	167.79	0.51

Table 30
Results of R&F for test problem #4.

Instance number	p	q	Optimal obj. value	CPLEX		R&F		
				CPU (s)	Nodes	Feasible solution	CPU (s)	Gap (%)
28	3	0	12247695091.29	360.34	0	12247695091.29	166.59	0.00
29		1	–	Memory	–	11834792471.40	TimeLimit	–
30		2	–	Memory	–	12069733080.90	TimeLimit	–
31		3	–	Memory	–	12808816263.50	TimeLimit	–
32	4	0	12038633898.29	4254.27	32	12040251817.10	241.31	0.01
33		1	–	Memory	–	12326033106.00	TimeLimit	–
34		2	–	Memory	–	12047212188.01	12247.24	–
35		3	–	Memory	–	11768358340.69	TimeLimit	–
36		4	–	Memory	–	12126357767.53	TimeLimit	–

Table 31
A comparison between the proposed methods for instances that cannot be optimized by the CPLEX.

Instance number	p	q	HGA		HVNS	R&F					
			OFV	CT	GBKS (%)	OFV	CT	GBKS (%)	OFV	CT	GBKS (%)
29	3	1	13172000595.80	194.35	11.30	11898133247.20	197.68	0.54	11834792471.40	TimeLimit	0.00
30		2	12935593302.30	203.92	7.17	12964563017.90	137.97	7.41	12069733080.90	TimeLimit	0.00
31		3	13043503815.00	190.90	5.53	12359623494.90	254.38	0.00	12808816263.50	TimeLimit	3.63
33	4	1	13039712828.00	210.12	5.79	12550096866.60	460.64	1.82	12326033106.00	TimeLimit	0.00
34		2	13206218601.30	205.57	9.62	12388525463.40	147.77	2.83	12047212188.01	12247.24	0.00
35		3	13095796795.50	195.17	11.28	12293246601.30	144.12	4.46	11768358340.69	TimeLimit	0.00
36		4	13313717989.30	196.07	9.79	12571922397.70	364.21	3.67	12126357767.53	TimeLimit	0.00
37	3	0	80366528412.20	938.01	14.22	72619825797.10	2450.09	3.21	–	Memory	–
38		1	81875323402.00	930.40	5.69	77469303284.60	909.33	0.00	–	Memory	–
39		2	79712568878.60	913.84	4.15	76534974820.70	636.61	0.00	–	Memory	–
40		3	79798619230.00	961.85	5.29	75786745327.10	393.98	0.00	–	Memory	–
41	4	0	82855331558.40	925.61	14.29	77344789511.80	1166.35	6.69	–	Memory	–
42		1	79216141604.30	980.17	4.45	75841235328.40	581.55	0.00	–	Memory	–
43		2	81486357431.40	928.60	4.59	77911519192.90	347.43	0.00	–	Memory	–
44		3	81781033663.10	968.73	4.77	78057073357.10	686.57	0.00	–	Memory	–
45		4	81599076228.40	916.03	2.67	79476799537.70	211.85	0.00	–	Memory	–
46	3	0	743805403820.30	2022.97	24.35	631663136901.29	2798.43	5.60	–	Memory	–
47		1	677079982480.10	2189.13	0.00	700606688928.20	1811.86	3.47	–	Memory	–
48		2	757077203124.80	2046.78	4.02	727795264622.90	1352.27	0.00	–	Memory	–
49	4	0	752492119980.00	2008.83	15.32	690287375066.50	2281.75	5.79	–	Memory	–
50		1	751615068952.20	2028.69	6.74	704166773708.70	1235.03	0.00	–	Memory	–

HGA and HVNS for instances #29-#31 and #33-#36.

Fig. 18 compares the objective function value of the proposed solution methods and the lower and upper bound of the Benders decomposition algorithm for the instances that cannot be solved to optimality by CPLEX. According to Fig. 18, in fifteen instances of the problem, the objective function value of the HGA is between the lower and upper bound. Also, in sixteen instances of the problem, the objective function value of the HVNS is between the lower and upper bound. In all instances that can be solved by using the R&F (instances #29-#31 and #33-#36), the objective function value is between the lower and upper bound.

7. Conclusions and future research

In this paper, we studied the ring hierarchical hub network design problem. This problem can be applied when it is impossible or very expensive to develop a complete hub network (a network in which hubs are fully interconnected). The ring structure of the backbone network can be used in the development of telecommunications and fast transportation networks. To design the ring hierarchical hub network, we proposed a mixed-integer programming model called the flow-based three-index model based on the type of decision variables. To validate the proposed model, numerical experiments were performed using the CPLEX on the CAB dataset. The results showed that the CPLEX can obtain the optimal solution for small-sized instances of the problem in a short computational time. But as the size of the problem increases, the

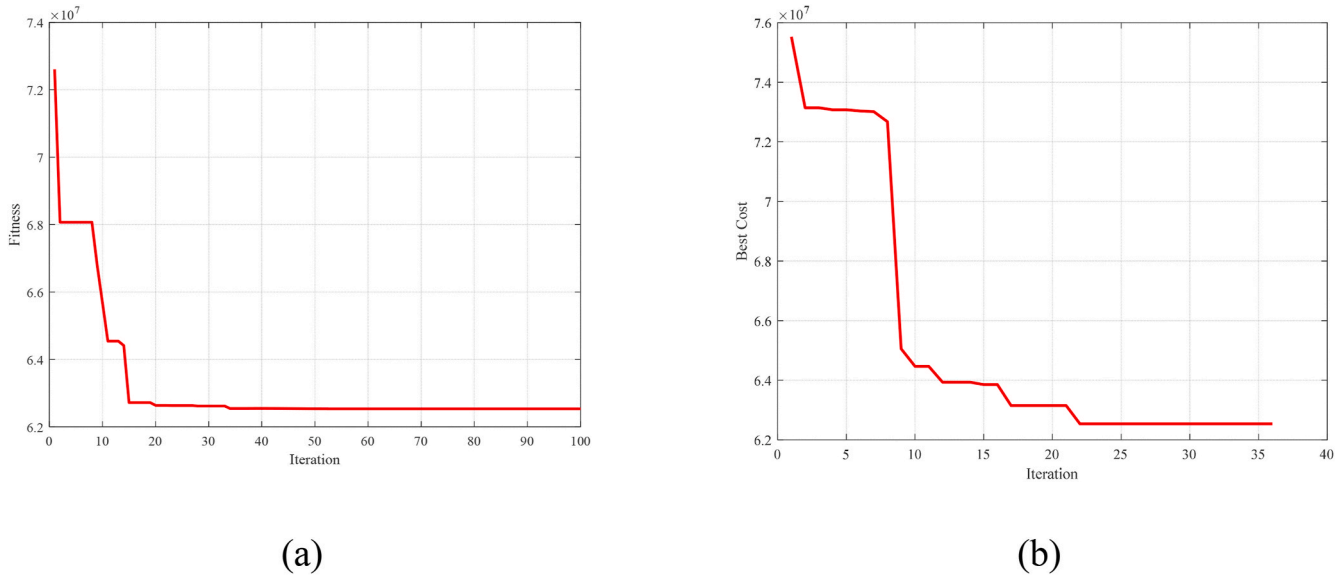

Fig. 14. The convergence chart of a) HGA b) HVNS for instance #2.

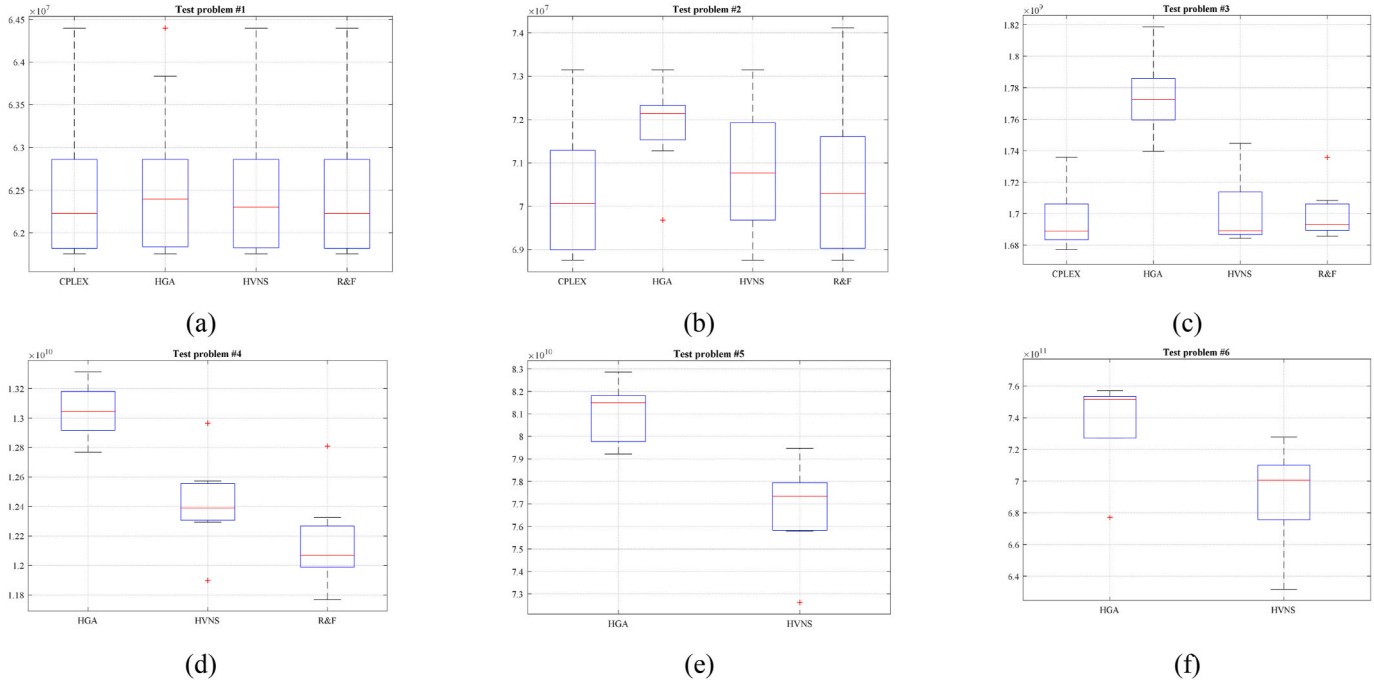

Fig. 15. Comparing results of the proposed solution methods for a) test problem #1 b) test problem #2 c) test problem #3 d) test problem #4 e) test problem #5 f) test problem #6.

solution time and the required computational space increase dramatically. Therefore, to solve larger instances of the problem, an accelerated Benders decomposition algorithm was proposed, and its results were compared with the results of the CPLEX on the USA423 dataset. Also, the lower and upper bounds of the proposed Benders decomposition algorithm for a predefined time limit were reported. Also, two hybrid algorithms, including a hybrid genetic algorithm (HGA) and a hybrid variable neighborhood search (HVNS) algorithm, and a relax-and-fix (R&F) heuristic were proposed to solve more instances of the problem. Then, the results of the proposed solution algorithms were compared with the results of the CPLEX on the USA423 dataset. The results showed that the proposed R&F algorithm can solve instances with up to 100 demand nodes. Also, the HGA and HVNS can solve instances with up to 423 nodes and the performance of the HVNS is better than the HGA in solving large instances. For future research, the following can be considered by researchers: proposing tighter formulation for the problem, improving the cut generation process of the Benders decomposition algorithm, studying multiple allocation and multi-modal versions of the problem, considering capacity constraints for hubs (in this case, it is necessary to develop methods for evaluating the solutions in the proposed hybrid methods, considering the capacity constraints of the hubs), and considering uncertainty for the parameters of the problem.

Declaration of competing interest

The authors declare that they have no known competing financial

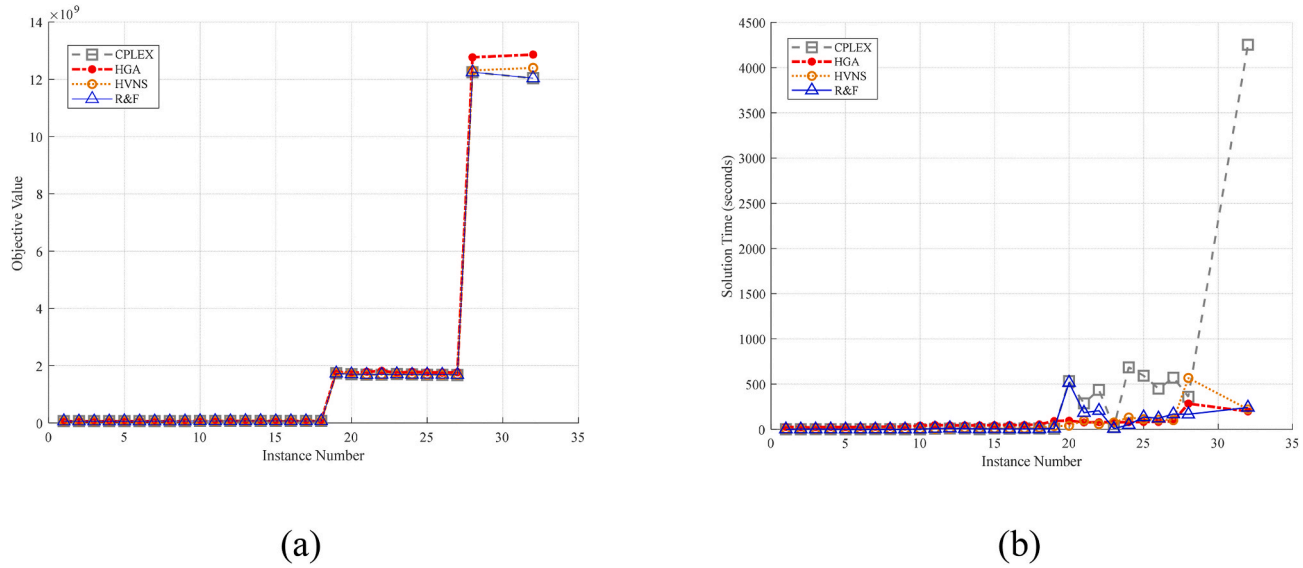

Fig. 16. A comparison between the proposed solution methods and CPLEX in terms of a) objective function value b) solution time.

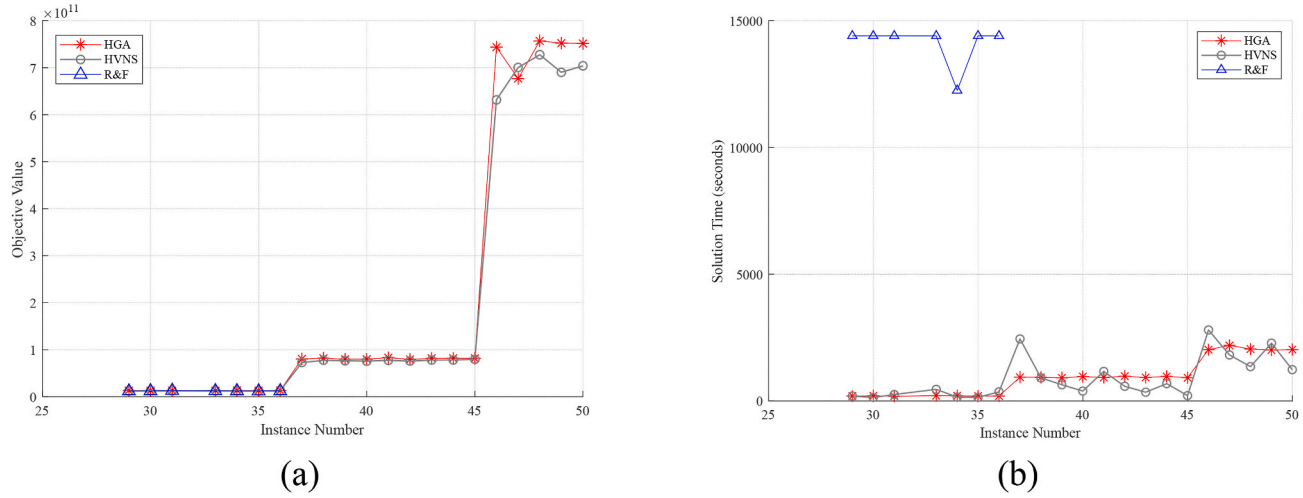

Fig. 17. A comparison between the proposed solution methods in terms of a) objective function value b) solution time.

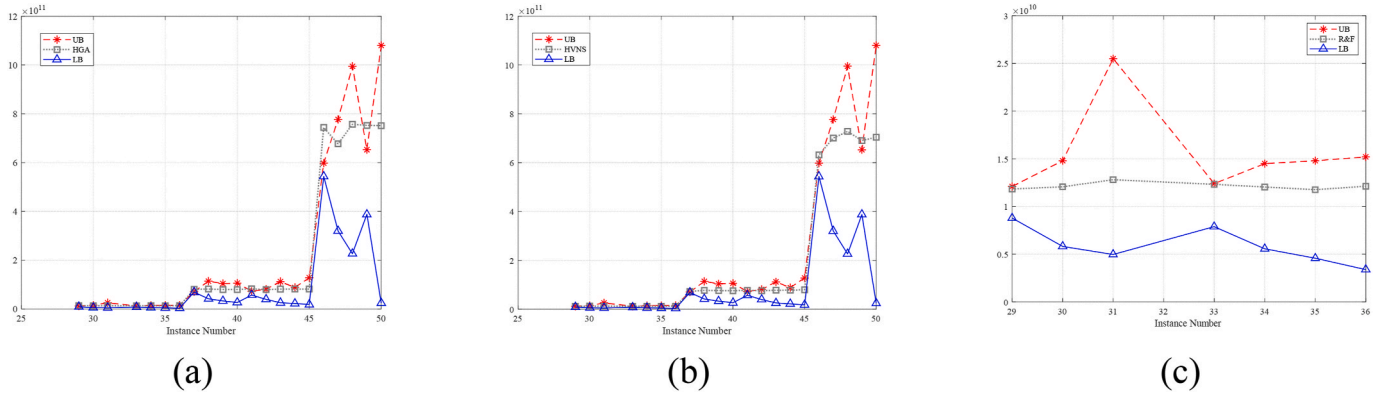

Fig. 18. A comparison between the lower and upper bound of the Benders decomposition algorithm and the objective value of the a) HGA b) HVNS c) R&F.

interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alumur, S.a., Kara, B.Y., Karasan, O.E., 2009. The design of single allocation incomplete hub networks. Transp. Res. Part B Methodol. 43, 936–951. https://doi.org/10.1016/j.trb.2009.04.004.

Alumur, S., Kara, B.Y., 2009. A hub covering network design problem for cargo applications in Turkey. J. Oper. Res. Soc. 60, 1349–1359.

Alumur, S., Kara, B.Y., 2008. Network hub location problems: the state of the art. Eur. J. Oper. Res. 190, 1–21.

Alumur, S.A., Yaman, H., Kara, B.Y., 2012. Hierarchical multimodal hub location problem with time-definite deliveries. Transp Res Part E Logist Transp Rev 48, 1107–1120.

Beasley, J.E., 2008. OR-Library: Hub Location.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252.

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. Comput. Oper. Res. 36, 3088–3096. https://doi.org/10.1016/j.cor.2008.11.023.

Campbell, J., Ernst, A., Krishnamoorthy, M., 2002. Hub location problems. In: Facility Location : Application and Theory. Springer, Berlin.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005a. hub arc location problems: Part I: introduction and results. Manag. Sci. 51, 1540–1555. https://doi.org/10.2307/20110442.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005b. hub arc location problems: Part II: formulations and optimal algorithms. Manag. Sci. 51, 1556–1571. https://doi.org/10.2307/20110443.

Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transport. Sci. 46, 153–169.

Contreras, I., Fernández, E., Marín, A., 2010. The tree of hubs location problem. Eur. J. Oper. Res. 202, 390–400. https://doi.org/10.1016/j.ejor.2009.05.044.

Contreras, I., Fernández, E., Marín, A., 2009. Tight bounds from a path based formulation for the tree of hub location problem. Comput. Oper. Res. 36, 3117–3127. https://doi.org/10.1016/j.cor.2008.12.009.

Contreras, I., Tanash, M., Vidyarthi, N., 2016. Exact and heuristic approaches for the cycle hub location problem. Ann. Oper. Res. https://doi.org/10.1007/s10479-015-2091-2.

Current, J.R., Schilling, D.A., 1994. The median tour and maximal covering tour problems: formulations and heuristics. Eur. J. Oper. Res. 73, 114–126. https://doi.org/10.1016/0377-2217(94)90149-X.

de Sá, E.M., de Camargo, R.S., de Miranda, G., 2013. An improved Benders decomposition algorithm for the tree of hubs location problem. Eur. J. Oper. Res. 226, 185–202. https://doi.org/10.1016/j.ejor.2012.10.051.

Dukkanci, O., Kara, B.Y., 2017. Routing and scheduling decisions in the hierarchical hub location problem. Comput. Oper. Res. 85, 45–57.

Eghbali-Zarch, M., Tavakkoli-Moghaddam, R., Jolai, F., 2019. A robust-possibilistic programming approach for a hub location problem with a ring-structured hub network under congestion: an m/g/c queue system. Int J Ind Eng 26.

Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E., 2013. Hub location problems: a review of models, classification, solution techniques, and applications. Comput. Ind. Eng. 64, 1096–1109.

Ferreira, D., Morabito, R., Rangel, S., 2010. Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. Comput. Oper. Res. 37, 684–691.

Gelareh, S., Nickel, S., 2011. Hub location problems in transportation networks. Transp Res Part E Logist Transp Rev 47, 1092–1111. https://doi.org/10.1016/j.tre.2011.04.009.

Gelareh, S., Pisinger, D., 2011. Fleet deployment, network design and hub location of liner shipping companies. Transp Res Part E Logist Transp Rev 47, 947–964. https://doi.org/10.1016/j.tre.2011.03.002.

Gendreau, M., Laporte, G., Semet, F., 1997. The covering tour problem. Oper. Res. 45, 568–576. https://doi.org/10.1287/opre.45.4.568.

Holland, J.H., 1992. Genetic algorithms. Sci. Am. 267, 66–73.

Horner, M.W., O'Kelly, M.E., 2005. A combined cluster and interaction model: the hierarchical assignment problem. Geogr. Anal. 37, 315–335.

Kara, B.Y., Taner, M.R., 2011. Hub location problems: the location of interacting facilities. In: Foundations of Location Analysis. Springer, pp. 273–288.

Karimi, H., Setak, M., 2014. Proprietor and customer costs in the incomplete hub location-routing network topology. Appl. Math. Model. 38, 1011–1023.

Karimi, M., Eydi, A.R., Korani, E., 2014. Modeling of the capacitated single allocation hub location problem with a hierarchical approch. Int J Eng Trans A Basics 27, 573–586.

Khaleghi, A., Eydi, A., 2022. Hybrid solution methods for a continuous-time multi-period hub location problem with time-dependent demand and sustainability considerations. J. Ambient Intell. Hum. Comput. 1–41.

Klincewicz, J.G., 1998. Hub location in backbone/tributary network design: a review. Locat. Sci. 6, 307–335.

Kuby, M.J., Gray, R.G., 1993. The hub network design problem with stopovers and feeders: the case of Federal Express. Transp Res Part A Policy Pract 27, 1–12.

Labbé, M., Laporte, G., Martín, I.R., González, J.J.S., 2004. The ring star problem: polyhedral analysis and exact algorithm. Networks 43, 177–189. https://doi.org/10.1002/net.10114.

Labbé, M., Laporte, G., Rodriguez Martin, I., González, J.J.S., 2005. Locating median cycles in networks. Eur. J. Oper. Res. 160, 457–470. https://doi.org/10.1016/j.ejor.2003.07.010.

Lee, Y., Lim, B.H., Park, J.S., 1996. A hub location problem in designing digital data service networks: Lagrangian relaxation approach. Locat. Sci. 4, 185–194. https://doi.org/10.1016/S0966-8349(96)00009-5.

Marufuzzaman, M., Eksioglu, S.D., 2014. Developing a reliable and dynamic intermodal hub and spoke supply chain for biomass. In: IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), p. 2417.

O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. Eur. J. Oper. Res. 32, 393–404.

Peiró, J., Corberán, Á., Marti, R., 2014. GRASP for the uncapacitated r-allocation p-hub median problem. Comput. Oper. Res. 43, 50–60.

Roy, R.K., 2001. Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement. John Wiley & Sons.

Sedehzadeh, S., Tavakkoli-Moghaddam, R., Baboli, A., Mohammadi, M., 2016. Optimization of a multi-modal tree hub location network with transportation energy consumption: a fuzzy approach. J. Intell. Fuzzy Syst. 30, 43–60.

Shang, X., Yang, K., Jia, B., et al., 2021. Heuristic algorithms for the bi-objective hierarchical multimodal hub location problem in cargo delivery systems. Appl. Math. Model. 91, 412–437.

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.

Tang, L., Jiang, W., Saharidis, G.K.D., 2013. An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. Ann. Oper. Res. 210, 165–190.

Torkestani, S.S., Seyedhosseini, S.M., Makui, A., Shahanaghi, K., 2016. Hierarchical facility location and hub network problems: a literature review. J Ind Syst Eng 9, 1–22.

Yaman, H., 2008. Star p-hub median problem with modular arc capacities. Comput. Oper. Res. 35, 3009–3019. https://doi.org/10.1016/j.cor.2007.01.014.

Yaman, H., 2009. The hierarchical hub median problem with single assignment. Transp. Res. Part B Methodol. 43, 643–658. https://doi.org/10.1016/j.trb.2009.01.005.

Yoon, M.-G., Current, J., 2008. The hub location and network design problem with fixed and variable arc costs: formulation and dual-based solution heuristic. J. Oper. Res. Soc. 59, 80–89.

Zarandi, M.H.F., Davari, S., Sisakht, S.A.H., 2015. An empirical comparison of simulated annealing and iterated local search for the hierarchical single allocation hub median location problem. Sci Iran Trans E, Ind Eng 22, 1203.

Zhong, W., Juan, Z., Zong, F., Su, H., 2018. Hierarchical hub location model and hybrid algorithm for integration of urban and rural public transport. Int. J. Distributed Sens. Netw. 14, 1550147718773263.