Computers & Operations Research 157 (2023) 106252

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

ILS-based algorithms for the profit maximizing uncapacitated hub network design problem with multiple allocation

Fabricio Alves Oliveira [a,b], Elisangela Martins de Sá [b], Sérgio Ricardo de Souza [b,*], Marcone Jamilson Freitas Souza [c]

[a] Federal Institute Catarinense (IFC), Rua Bernardino José de Oliveira, 81 - Badenfurt - CEP 89070-270, Blumenau, SC, Brazil
[b] Federal Center of Technological Education of Minas Gerais (CEFET-MG), Av. Amazonas, 7675 - Nova Gameleira - CEP 30510-000, Belo Horizonte, MG, Brazil
[c] Federal University of Ouro Preto (UFOP), Morro do Cruzeiro - CEP 35400-000, Ouro Preto, MG, Brazil

ARTICLE INFO

Keywords:
Hub network design problem
Profit maximization
Multiple allocation
Iterated local search
Variable neighborhood descent
Metaheuristics

ABSTRACT

This study addresses a hub network design problem to maximize net profit. This problem considers an incomplete hub network with multiple allocation that does not impose capacity constraints, does not allow direct connections between non-hub nodes, and accepts the demand to be partially met, being satisfied only when profitable. To tackle this problem, which is NP-hard, we propose two heuristic algorithms based on the Iterated Local Search (ILS) metaheuristic, a standard ILS algorithm, and an Enhanced ILS algorithm, which increases the perturbation level only after a few unsuccessful attempts at improvement. Both algorithms use Random Variable Neighborhood Descent in the local search. Computational experiments were performed using benchmark instances for hub location problems, and statistical analyzes of the algorithms were presented. Numerical results confirm that both algorithms yield good-quality solutions with an acceptable runtime. In particular, the proposed algorithms obtain the optimal solution for most instances with up to 150 nodes, which have known optimal solutions. Furthermore, the proposed algorithms were able to handle instances with up to 500 nodes.

1. Introduction

Hub networks are often used in passenger and freight transportation systems and telecommunications networks. In such networks, the demand between origin and destination pairs is routed through intermediate facilities, known as hubs. Hubs are responsible for receiving, aggregating, transferring, and distributing the demand flow in the network. In this way, origin and destination nodes can be connected using fewer connections, which reduces the cost of establishing the network. In addition, consolidating demand flow at hubs can allow economies of scale to be applied when routing flows through hub arcs, i.e., arcs connecting a pair of hubs, providing a reduction in transportation costs.

Hub location problems (HLPs) focus on locating the hubs, allocating nodes to these hubs, and routing the demand flows in the network while optimizing a given objective. For HLP, the hub network design is determined by the nodes selected to be hubs and how demand nodes are allocated to them, assuming that hubs are fully interconnected. In general, HLPs assume that demands are routed only through trivial routes, i.e., routes having up to two hub nodes connecting each origin and destination pair of nodes. On the other hand, there is a more complex class of problems in hub literature, known as hub network design problems (HNDPs), which explicitly consider network design decisions, such as determining which hub arc will be installed as well as nontrivial routing decisions. The current study addresses a HNDP.

A hub network is composed of two types of nodes (hub nodes and non-hub nodes) and also the arcs connecting these nodes. This network is composed of two kinds of networks: *(i)* access or distribution networks, which connects non-hub nodes to hubs, and *(ii)* the network at the hub level (inter-hub network), which interconnects the hubs. Fig. 1 illustrates a hub network, where circles represent non-hub nodes and triangles indicate hubs. In this figure, simple segments, called allocation arcs, represent the allocation of non-hub nodes to hubs. In turn, the segments in red, called hub arcs, represent connections between the hubs. Note that, in this network, there are demand nodes allocated to more than one hub and that the hubs are not fully interconnected, characterizing an HLP with multiple allocation and an incomplete network, which is addressed in HNDPs.

In the literature, there is a wide range of HLPs and HNDPs, as well as different mathematical models to describe these problems. Their

* Corresponding author.
E-mail addresses: fabricio.oliveira@ifc.edu.br (F.A. Oliveira), elisangelamartins@cefetmg.br (E.M. de Sá), sergio@cefetmg.br (S.R. de Souza), marcone@ufop.edu.br (M.J.F. Souza).

https://doi.org/10.1016/j.cor.2023.106252
Received 20 June 2022; Received in revised form 19 April 2023; Accepted 19 April 2023
Available online 5 May 2023

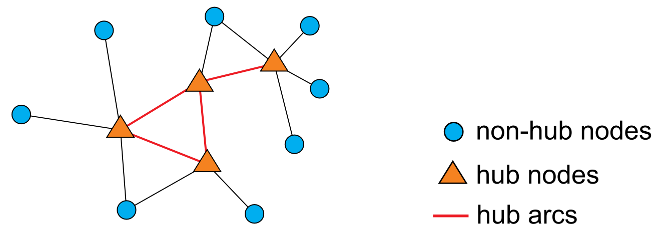


Fig. 1. Incomplete hub network with multiple allocation.

characterization and classification are related to several factors, such as the definition of the problem, the nature of the demand, the objectives to be optimized, and the types of restrictions imposed. In general, such factors are related to the application area. A review of the various types of problems, models, and taxonomy employed in the area of hub location can be found in Alumur and Kara (2008), Campbell and O'Kelly (2012), Contreras (2015) and Alumur et al. (2021).

Most hub location problems aim to minimize the total cost of the network, assuming that all origin and destination pairs have their demand satisfied. However, from the point of view of profit, it may not be advantageous to meet all the demand, but rather to satisfy only the demand whose revenue is greater than the costs involved, i.e., when it is profitable. Furthermore, most of the studies addressing profit maximization consider that all hubs in the network are interconnected, that is, that the inter-hub network is complete.

This study addresses the profit maximizing uncapacitated hub network design problem with multiple allocation (PMHNDP). This problem aims to determine the quantity and location of hubs, select the origin and destination pairs that will be served, establish the arcs that will be installed, and determine the optimal routing for each served demand flow, allowing non-hub nodes to be allocated to more than one hub, to maximize the total profit given by the difference between the total revenue and the total costs of the network design and operation. Thus, this problem does not impose that the demand of the network is completely satisfied and does not impose any conditions on the topology of the hub network, so the hub nodes can be partially interconnected. This study also considers that the problem is uncapacitated, i.e., it is assumed that the hubs and the hub arcs have sufficient capacity to handle the demand flow as necessary.

Potential applications of the PMHNDP arise, for example, in the design of air transportation networks since profit is a decisive impact factor in the consolidation and maintenance of these networks. The objective, in this case, is to find an ideal hub network structure to maximize the total net profit to provide air travel services to a set of flights, taking into account the total cost of the network (Alibeyg et al., 2016). The PMHNDP can also be used in any other areas where profit is an essential factor in the design of the hub network.

Many procedures have been proposed to solve HLPs and HNDPs. We can highlight exact algorithms based on the Benders decomposition method (Camargo and Miranda, 2012; Camargo et al., 2017; Martins de Sá et al., 2018a; Taherkhani et al., 2020) and methods that employ the branch-and-bound structure together with other techniques such as cutting planes, partial enumeration, and tests to reduce the problem dimension (Contreras et al., 2011b; Rodríguez-Martí et al., 2014; Alibeyg et al., 2017). Due to the complexity of these problems, several types of heuristic approaches have also been used to solve them, including local search methods (Rodríguez-Martín and Salazar-González, 2006; Fazel Zarandi et al., 2015), tabu search (Calık et al., 2009; Ghaffarinasab, 2020), evolutionary and genetic algorithms (Kratica et al., 2011; Shang et al., 2021), variable neighborhood search (Davari et al., 2013; Todosijević et al., 2017), Lagrangian relaxation (Contreras et al., 2009), and other heuristics (Hoff et al., 2017; Dai et al., 2019).

Since HDNPs are NP-hard problems (Alumur and Kara, 2008; Campbell and O'Kelly, 2012), the resolution of large instances of these problems often requires the use of heuristic methods. In this study, we propose two heuristic algorithms based on local search methods, that is, methods that consist in exploring the solution space through perturbations in locally optimal solutions. More specifically, the first algorithm uses the Iterated Local Search (ILS) (Lourenço et al., 2003) metaheuristic framework in its standard version. The second algorithm, called Enhanced Iterated Local Search (E-ILS), is an adaptation of the ILS method that employs more intensification in the search. This adaptation was initially used by Reinsma et al. (2018). The two algorithms use as a local search procedure the Random Variable Neighborhood Descent (RVND) method, proposed by Souza et al. (2010) and Subramanian et al. (2010). These algorithms were proposed due to the successful application of local search methods in other types of HNDPs, such as the ILS-VND method employed by Martins de Sá et al. (2018a).

Extensive computational experiments were performed using the benchmark instances from the *Australian Post* (AP) data set, introduced by Ernst and Krishnamoorthy (1996), with up to 200 nodes and from the data set proposed by Contreras et al. (2011a) with up to 500 nodes. Furthermore, the result of the statistical tests used to compare the two developed algorithms is presented.

The main contribution of the current article is to present two efficient heuristic algorithms able to handle large instances of the PMHNDP. This problem was initially proposed by Taherkhani and Alumur (2018), which solved only the AP instances with 40 nodes. This problem was also addressed by Oliveira et al. (2022), which proposed an exact algorithm, based on the Benders decomposition method, able to solve the AP instances with up to 150 nodes. However, the proposed Benders decomposition algorithm was not able to solve all instances with 125 and 150 nodes, nor was able to solve any AP instance with 200 nodes, which refers to a real application of the Australian post services, due to exceeding the memory limits. Zhang et al. (2023) proposed an algorithm based on Variable Neighborhood Search to handle the PMHNDP, which only solved instances with up to 60 nodes. Hence, according to our knowledge, this study is the only one that tackles the AP instances with 200 nodes. Furthermore, the proposed procedure solves instances with up to 500 nodes to attest to its scalability.

The remainder of this article is structured as follows. Section 2 briefly reviews the literature on related works. Section 3 describes the characteristics of the PMHNDP and presents a MIP formulation. Section 4 details the proposed algorithms. Section 5 shows the computational experiments and statistical analysis of the heuristic algorithms. Finally, Section 6 presents the final remarks and future work.

2. Literature review

In this section, we review the literature related to our work. Initially, studies involving HNDPs with an incomplete inter-hub network are addressed. Then, works focusing on profit maximization and studies with similar objectives are highlighted.

In general, HLPs assume that the hub network is fully connected, meaning that all hubs are interconnected. Although this assumption makes the models simpler, many real-world applications may not be satisfactorily described by such models. Furthermore, there are situations in which the installation of many arcs between hubs is not feasible, for example, when high installation costs are incurred in these connections. Thus, many researchers have turned their attention to HNDPs with incomplete networks in recent years.

Nickel et al. (2001) presented two formulations for a HNDP applied to urban public transportation networks to minimize transportation costs and the fixed costs of installing hubs and hub arcs. These formulations allow direct connections between non-hub nodes and use the multiple allocation strategy. Campbell et al. (2005a) introduced the hub arc location problems, which aim to locate a fixed number of hub arcs, minimizing total transportation costs. Campbell et al. (2005b) proposed enumeration-based solution methods for this type of problem and solved instances with up to 25 nodes. Yoon and Current (2008) addressed a HNDP associating fixed and variable costs to the hub arcs. They developed dual-based heuristics and solved instances with up to 25 nodes, varying the cost structures. Alumur and Kara (2009) proposed a formulation for the hub coverage problem to minimize the hubs and hub arcs installation costs while designing a network that serves each origin and destination pair within a time limit, considering single allocation and routes with a maximum of three hubs. The authors used the CPLEX solver to solve instances with 81 nodes. This type of problem was also addressed by Calık et al. (2009), who developed a heuristic based on tabu search, used to solve instances with up to 81 nodes. Alumur et al. (2009) proposed mathematical models for different versions of HNDPs, considering incomplete inter-hub networks and single allocation, and they used the CPLEX solver to solve the proposed formulations.

Gelareh and Nickel (2011) addressed an uncapacitated HNDP with multiple allocation applied to urban and maritime transportation network design. They propose an accelerated Benders decomposition method and a greedy heuristic algorithm, being able to solve instances with up to 50 nodes. Davari et al. (2013) addressed a hub coverage problem, in which the demands are unknown and estimated with fuzzy variables. This work considered the objective of maximizing the premise that each flow in the network can be satisfied in a determined fixed time. An algorithm based on the Variable Neighborhood Search (VNS) was developed and instances with up to 25 nodes were solved. Camargo et al. (2017) proposed formulations and developed specialized algorithms based on Benders decomposition to solve HNDPs with incomplete networks with and without hop-constraints. Benders decomposition was also applied to solve HNDPs with capacity constraints by Xu et al. (2017). Martins de Sá et al. (2018a) addressed a robust HNDP assuming uncertainties in demand flows and fixed costs of installation of the hubs. The authors developed methods based on Benders decomposition and also presented an ILS-VND heuristic, solving instances with up to 100 nodes. Dai et al. (2019) implemented a heuristic algorithm, named HUBBI, to handle a p-hub location problem with incomplete network, which focuses on minimizing the transportation and configuration costs. Öztürk et al. (2021) also dealt with HNDPs where the number of hubs is determined beforehand. They considered the single allocation strategy and presented heuristic methods that use measures of centrality. In both works, they performed computational experiments using benchmark instances with up to 200 nodes. Wandelt et al. (2022) studies 12 versions of hub location problems that include four variants of HNDPs, presenting an experimental benchmark for hub location problems.

Table 1 summarizes these studies, presenting the method used to solve the problems and the largest instance solved in each study, where the instance size is given by the number of nodes in the network. Note that the largest instance solved has 200 nodes. It is worth pointing out here that, although the studies addressing HNDPs solved instances with up to 200 nodes, there are studies in the literature that solved HLPs instances with 500 nodes (Contreras et al., 2011a,b; Taherkhani et al., 2020). Indeed, it is well known in the hub location area that HNDPs are more difficult to solve than HLPs since they also include complex network designing decisions and have to handle non-trivial routes (O'Kelly and Miller, 1994; Alumur et al., 2021).

Classical HLPs and HNDPs aim to minimize the total cost of the hub network under the premise of serving the entire demand. However, this hypothesis may not always be the most adequate when dealing with profit maximization. In this case, it may be more advantageous to serve only a portion of the origin and destination pairs corresponding to those whose service revenue is greater than the incident costs. Thus, as a relatively new approach, some studies in the literature have discussed the *trade-off* between revenue and total costs in the design of hub networks.

Alibeyg et al. (2016) introduced the first study considering profit maximization, which proposed several versions of the multiple allocation problem. Additional constraints and/or decisions were considered, such as the imposition of fully meeting the demand in the network, and, although an HNDP was addressed, they also assumed that the route between the origin and destination pairs should contain at least one and at most two hubs. The proposed formulations were evaluated using instances with up to 70 nodes using the CPLEX solver. Alibeyg et al. (2017) presented an exact algorithm, based on branch-and-bound and Lagrangian relaxation, to solve the models proposed by Alibeyg et al. (2016). The proposed algorithm was able to solve instances with up to 100 nodes.

Taherkhani and Alumur (2018) proposed new formulations for profit-maximizing HNDPs. Models for different versions of the problem were presented, obtained according to the type of allocation strategy (single, multiple, or r-allocation), allowing or not direct connections between non-hub nodes and assuming that the inter-hub network is complete or incomplete. The CPLEX solver was used to evaluate the performance of the proposed formulations with the CAB data set and AP instances with 40 nodes. Taherkhani et al. (2020) addressed a profit-maximizing capacitated HLP with multiple classes of demand. They developed two models, a deterministic and a stochastic, considering multiple allocation, a complete inter-hub network, and assuming that there are at most two hubs on the paths connecting the origin and destination pairs. To solve the proposed models, the authors proposed algorithms based on Benders decomposition that use tests to reduce the problem size and variable pre-fixation. With the deterministic model, instances of up to 500 nodes were solved. It is important to highlight that, although large instances have been solved in this study, complete inter-hub network problems are less challenging than incomplete inter-hub network problems, as already mentioned. Oliveira et al. (2022) addressed an uncapacitated HNDP with an incomplete network, multiple allocation, and focused on profit maximization. The authors presented different versions of exact algorithms based on Benders decomposition that apply Pareto-optimal cuts. The most promising method, which adds Benders cuts into a branch-and-cut structure, was able to solve AP instances with up to 150 nodes.

Recently, Zhang et al. (2023) applied a Variable Neighborhood Search procedure to solve some variants of uncapacitated HNDPs with profits and incomplete hub network. They addressed problems with single allocation, multiple allocation, and r-allocation strategies with and without direct connections. The authors presented computational results with CAB instances containing up to 60 nodes.

In addition to these studies that addressed the design of hub networks, selecting the origin and destination pairs with profitable demand flows, there are also works dealing with maximization objectives associated with the concept of profitability. Lür-Villagra and Marianov (2013) used a genetic algorithm to deal with a HLP incorporating price decisions, considering a competitive environment. Lin and Lee (2018) addressed a HNDP applied to freight transport, which was solved with implicit enumeration, applying a built-in pricing sub-problem. Čvokić and Stanimirović (2020) incorporate price decisions in an uncapacitated single-allocation HLP, in which the amount of commodities flows between the pairs of nodes has a stochastic nature. A deterministic and a robust model were proposed. An equivalent conic-quadratic formulation was presented for the deterministic model, while the robust version was solved through a two-phase matheuristic approach.

Other studies, such as Tikani et al. (2018), Huo et al. (2019) and Salehi and Tikani (2020) have integrated revenue management techniques into HLPs and HNDPs. In these studies, genetic algorithms or hybrid algorithms that apply genetic operators were used to solve instances of the proposed problems. Studies related to profit maximization also often arise in problems involving a competitive environment. For example, Neamatian Monemi et al. (2017), Mahmoodjanloo et al. (2020) and Tiwari et al. (2021) deal with the design of hub networks applied to transportation systems. A matheuristic approach combined with Lagrangian relaxation was employed by Neamatian Monemi et al. (2017). To solve the problem, Mahmoodjanloo et al. (2020) performed

Table 1
Studies addressing incomplete inter-hub network.

Article	Solution Method	Instance size
Nickel et al. (2001)	Shortest path algorithms	10
Campbell et al. (2005b)	Enumeration based algorithms	25
Yoon and Current (2008)	Dual-based heuristic algorithms	25
Alumur and Kara (2009)	The CPLEX solver	81
Calık et al. (2009)	Heuristic algorithms based on tabu search	81
Alumur et al. (2009)	The CPLEX solver	81
Gelareh and Nickel (2011)	Benders decomposition and greedy algorithms	50
Davari et al. (2013)	VNS-based heuristic algorithms	25
Camargo et al. (2017)	Benders decomposition	100
Xu et al. (2017)	Benders decomposition	25
Martins de Sá et al. (2018a)	Benders decomposition and ILS-VND	100
Martins de Sá et al. (2018b)	Benders decomposition	50
Dai et al. (2019)	VNS-based heuristic algorithms	200
Öztürk et al. (2021)	Heuristic algorithms based on centrality measures	200

Table 2
Studies addressing profit maximization HNDPs.

Article	Profit maximization HNDP	Route restriction	Solution method	Instance size
Alibeyg et al. (2016)	UMAI, CapMAI	Yes	CPLEX	70
Alibeyg et al. (2017)	UMAI, CapMAI	Yes	Branch-and-bound and Lagrangian relaxation	100
Taherkhani and Alumur (2018)	UMAI, USAI, UrAI, UMAIDC, USAIDC, UrAIDC	No	CPLEX	40
Taherkhani et al. (2020)	CapMAComp	Yes	Benders decomposition	500
Oliveira et al. (2022)	UMAI	No	Benders decomposition	150
Zhang et al. (2023)	UMAI, USAI, UrAI, UMAIDC, USAIDC, UrAIDC	No	VNS algorithm	60
This study	UMAI	No	ILS algorithm	500

Note. U/Cap=Uncapacitated/Capacitated, SA/MA/rA=Multiple/Single/r-allocation.
I/Comp=Incomplete/Complete hub network, DC=Direct connections allowed.

a two-level decomposition of the model, and Tiwari et al. (2021) proposed different approaches, involving conical reformulation, Kelley's cutting plane method, and Lagrangian relaxation.

Table 2 presents a summary of the works that address profit maximization HNDPs. This table provides, for each article, the characteristics of the problems addressed, whether the number of hubs in a route is restricted, the solution method applied, and the size of the largest instance that was solved.

Note that the only works that dealt with the problem with the same characteristics as the one addressed in this study were Taherkhani and Alumur (2018), Oliveira et al. (2022), and Zhang et al. (2023). The first two works used exact methods and solved instances of the problem with up to 40 and 150 nodes, respectively. Zhang et al. (2023) proposed a heuristic algorithm and presented computational results for instances with up to 60 nodes. We emphasize that this is the first article to solve large PMHNDP instances, containing up to 500 nodes. The heuristic algorithm based on the ILS procedure, proposed in this study, is different from the VNS presented by Zhang et al. (2023). There are several methodological differences between the algorithms. For example, the construction of the initial solution, the evaluation of the obtained solution, and the perturbation procedure used to prevent the search from getting stuck in local minima. Furthermore, although both studies use similar neighborhood structures, they are not identical.

3. Problem definition and characterization

The PMHNDP addressed in this paper aims to maximize the total profit of the hub network. This value is given by the total revenue obtained by serving the demand of the selected pairs of nodes, minus the sum of the network design costs and the transportation costs. Network design costs refer to the fixed costs of installing hubs and hub arcs, while transportation costs refer to the associated costs for routing the demand flow between the origin and destination pairs. As an advantage of a hub network, it is also assumed that a constant discount factor is applied to the transportation cost in the hub arcs.

It is important to highlight that modeling economies of scale using a constant factor is, in general, an over-simplification of reality, by assuming, for instance, that economies of scale do not depend on the amount of flow. However, problems considering flow-dependent economies of scale are much more challenging to be tackled, often leading to a nonlinear problem (Alumur et al., 2021). Therefore, we assume the constant discount factor as also done by Taherkhani and Alumur (2018), Oliveira et al. (2022), and Zhang et al. (2023). We refer the reader to O'Kelly and Bryan (1998), Klincewicz (2002), de Camargo et al. (2009) and Alumur et al. (2021) for studies addressing hub location problems considering flow-dependent economies of scale.

The problem considered here has the following characteristics: *(i)* a non-hub node can be allocated to more than one hub; *(ii)* hub nodes and hub arcs are uncapacitated; *(iii)* the inter-hub network can be incomplete; *(iv)* direct connections between non-hub nodes are not allowed; *(v)* the number of hub nodes and hub arcs is determined by the model; *(vi)* there are fixed costs for the installation of hubs and hub arcs; and *(vii)* some origin and destination pairs can be not served, where the solution of the problem will define which origin and destination pairs will be served, through the selection of those pairs that generate profit.

To model the problem, consider the following parameters. Let N be the set of nodes that exchange flows and that are also potential candidates to be hubs. The quantity of demand to be routed from node $i \in N$ to node $j \in N$ is denoted by w_{ij}. For all $i, j \in N$, r_{ij} represents the revenue obtained by serving one unit of demand from the origin node i to the destination node j. The unitary transportation cost on the arc connecting nodes $i \in N$ and $j \in N$ is denoted by c_{ij}. The fixed cost of installing a hub at node $k \in N$ is given by s_k, and the fixed cost of installing an arc between hubs k and m is denoted by g_{km}, $k, m \in N$. Finally, we denote the constant discount factor on the transportation carried out through connections between hubs by α $(0 \leq \alpha < 1)$.

Also, consider the following set of variables: *(i)* binary variables h_k indicate whether a hub is located at node $k \in N$; *(ii)* binary variables z_{km} indicate whether the hub arc that allows flow from hub $k \in N$ to hub $m \in N$ is selected to be installed; *(iii)* flow variables a_{ijk} represent the fraction of demand between nodes $i \in N$ and $j \in N$ that is served through a path where the first hub is $k \in N$; *(iv)* flow variables b_{ijm} represent the fraction of demand between nodes $i \in N$ and $j \in N$ that is served through a path in which the last hub is $m \in N$; *(v)* flow variables x_{ijkm} determine the fraction of demand w_{ij} that is routed in the hub arc connecting hubs $k \in N$ and $m \in N$.

Thus, a formulation for the PMHNDP, proposed by Oliveira et al. (2022), is given by:

$$\max \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} r_{ij}w_{ij}a_{ijk} - \Bigg[\sum_{i\in N}\sum_{j\in N}\sum_{k\in N} c_{ik}w_{ij}a_{ijk} + \sum_{i\in N}\sum_{j\in N}\sum_{m\in N} c_{mj}w_{ij}b_{ijm} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha c_{km}w_{ij}x_{ijkm} + \sum_{k\in N} s_k h_k + \sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} g_{km}z_{km} \Bigg] \tag{1}$$

$$\text{subject to } \sum_{k\in N} a_{ijk} \leq 1 \qquad i,j \in N \tag{2}$$

$$\sum_{m\in N} b_{ijm} \leq 1 \qquad i,j \in N \tag{3}$$

$$a_{ijk} + \sum_{\substack{m\in N\\ m\neq k}} x_{ijmk} = b_{ijk} + \sum_{\substack{m\in N\\ m\neq k}} x_{ijkm} \qquad i,j,k \in N \tag{4}$$

$$a_{ijk} \leq h_k \qquad i,j,k \in N \tag{5}$$

$$b_{ijm} \leq h_m \qquad i,j,m \in N \tag{6}$$

$$x_{ijkm} \leq z_{km} \qquad i,j,k,m \in N, k \neq m \tag{7}$$

$$z_{km} \leq h_k \qquad k,m \in N, k \neq m \tag{8}$$

$$z_{km} \leq h_m \qquad k,m \in N, k \neq m \tag{9}$$

$$x_{ijkm} \geq 0 \qquad i,j,k,m \in N, k \neq m \tag{10}$$

$$a_{ijk} \geq 0 \qquad i,j,k \in N \tag{11}$$

$$b_{ijm} \geq 0 \qquad i,j,m \in N \tag{12}$$

$$h_k \in \{0,1\} \qquad k \in N \tag{13}$$

$$z_{km} \in \{0,1\} \qquad k,m \in N, k \neq m. \tag{14}$$

The objective function (1) expresses the total profit of the hub network, which is obtained by the difference between the total revenue obtained from meeting the demand (the first term) and the total cost, represented by the terms in square brackets. The first three terms in parentheses represent transportation costs, composed of access, distribution, and transfer costs on the network. The last two terms in the parentheses correspond to the fixed costs of installing hubs and hub arcs, respectively. Constraints (2) and (3) guarantee that, for each origin and destination pair, the fraction of demand routed through the hub network is less than or equal to one. The set of constraints (4) refers to the flow conservation equations. Constraints (5) and (6) ensure that the demand of each origin and destination pair can only be served through installed hubs. Constraints (7) ensure that demand flow can only be routed through installed hub arcs. Constraints (8) and (9) indicate that a hub arc can only be used if its two end nodes are selected to be hubs. Constraints (10)–(14) represent the domain of the decision variables.

4. Description of the proposed algorithms

In order to tackle large instances of the PMHNDP, this paper proposes two heuristic algorithms based on local search: Iterated Local Search (ILS) and Enhanced Iterated Local Search (E-ILS). Both algorithms use the Random Variable Neighborhood Descent (RVND) method for the local search, which performs a systematic exchanges of neighborhood structures. These algorithms differ from each other in the way in which the exploration of the solution space is done. While ILS changes the search location at each iteration without improving, E-ILS remains in the same region for a longer time, thus performing a greater intensification.

This section is organized as follows. Section 4.1 presents how a solution is represented and evaluated. A simple method for generating an initial solution is presented in Section 4.2. In Section 4.3, the neighborhood structures used to explore the solution space are detailed. Sections 4.4 and 4.5 detail the local search method and the perturbation procedure, respectively, applied in the algorithms. Finally, Sections 4.6 and 4.7 present the ILS and E-ILS algorithms applied to the problem.

4.1. Representation and evaluation of the solution

A solution for the PMHNDP is represented by $s = (H, Z)$, where H is the set of installed hubs and Z is the set of installed hub arcs. Fig. 2 shows a hub network and the representation of the solution associated with this network.

Note that, in the solution representation, it is not necessary to indicate the allocation of non-hub nodes to hubs, because the demand flows selected to be served must be routed through the shortest path between each pair of demand nodes. In addition to simplifying the representation, this fact also allows the evaluation of a solution to the problem. Given the sets of installed hubs and hub arcs, the solution evaluation can be performed by calculating the shortest path between all origin and destination pairs that generate profit by meeting their demand.

The objective value of a solution s, denoted by $f(s)$, can be determined as follows. Let C_{ij} be the minimum transportation cost to route a flow unit from the origin $i \in N$ to the destination $j \in N$. Let P_{ij} be the unit profit associated with the pair $i,\ j \in N$, which can be calculated as

$$P_{ij} = \max\{0, r_{ij} - C_{ij}\}. \tag{15}$$

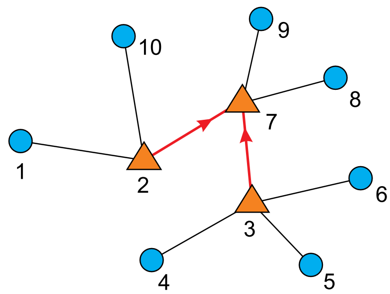
Thus, $f(s)$ is given by:

$$f(s) = \sum_{i\in N}\sum_{j\in N} w_{ij}P_{ij} - \sum_{k\in H} s_k - \sum_{(k,m)\in Z} g_{km}. \tag{16}$$

The value of C_{ij} can be determined using the Floyd–Warshall algorithm (Floyd, 1962) as follows. Let $Z^0 \subset N \times N$ be the set of inactive hub arcs. Instead of applying the Floyd–Warshall algorithm to a network composed of all nodes, the algorithm is applied to determine the lowest transportation costs between all pairs of hubs in the inter-hub network, assuming that the transportation cost in arcs in the set Z^0 is large enough to not be used in any route. Thus, letting C^{FW}_{km} be the lowest unit transportation cost between the pairs of hubs $k, m \in H$, C_{ij} is determined by:

$$C_{ij} = \min_{k,m\in H} \left\{ c_{ik} + C^{FW}_{km} + c_{mj} \right\}. \tag{17}$$

Algorithm 1 outlines this procedure.



(a) Hub network.

$$s = (\{2,3,7\}, \{(2,7),(3,7)\})$$

(b) Solution representation.

Fig. 2. Example of representation of a solution for the PMHNDP.

Algorithm 1 Minimum transportation cost between all pairs of nodes

Input: $N, H, Z^0, \alpha, c_{ij}$
Output: C_{ij}

for all $(k,m) \in H \times H$ **do**
  if $(k,m) \in Z^0$ **then**
    $C^{FW}_{km} \leftarrow \infty$
  else
    $C^{FW}_{km} \leftarrow \alpha c_{km}$
  end if
end for
for all $k \in H, (i,j) \in H \times H$ **do**
  $C^{FW}_{ij} \leftarrow \min\{C^{FW}_{ij}, C^{FW}_{ik} + C^{FW}_{kj}\}$
end for
for all $(i,j) \in N \times N$ **do**
  $C_{ij} \leftarrow \min_{k,m \in H} \left\{c_{ik} + C^{FW}_{km} + c_{mj}\right\}$
end for
return C_{ij}

Algorithm 2 Initial solution: best solution with a single hub

Input: $N, r_{ij}, w_{ij}, c_{ij}, s_k$
Output: Initial solution s^0, Initial solution value ϕ

1: $s^0 \leftarrow (\emptyset, \emptyset)$
2: $\phi \leftarrow 0$
3: $\phi_{\text{aux}} \leftarrow 0$
4: **for all** $k \in N$ **do**
5:   $\phi_{\text{aux}} \leftarrow -s_k$
6:   **for all** $(i,j) \in N \times N$ **do**
7:     $l_{ij} \leftarrow (r_{ij} - c_{ik} - c_{kj})w_{ij}$
8:     **if** $l_{ij} > 0$ **then**
9:       $\phi_{\text{aux}} \leftarrow \phi_{\text{aux}} + l_{ij}$
10:     **end if**
11:   **end for**
12:   **if** $\phi_{\text{aux}} \geq \phi$ **then**
13:     $\phi \leftarrow \phi_{\text{aux}}$
14:     $s^0 \leftarrow (\{k\}, \emptyset)$
15:   **end if**
16: **end for**
17: **return** s^0, ϕ

4.2. Initial solution

An initial solution for the PMHNDP was generated considering the best solution obtained with the installation of a single hub in the network. This procedure is described in Algorithm 2, which has as input the parameters of the problem, except the fixed cost of installing hub arcs (g_{km}) and the constant discount factor (α) since only one hub will be installed. This algorithm returns the initial solution s^0 and its objective value, denoted by ϕ. First, the initial solution is initialized with the empty set, and the objective value of the solution and an auxiliary variable are initialized to zero (lines 1–3). In the main loop of the algorithm (lines 4–16), the total profit is calculated iteratively considering the installation of each node as the only hub in the network, and the best solution value found is then stored. First, the auxiliary variable receives the fixed cost of installing the current node as the hub (line 5). After that, for each pair (i,j), the profit obtained from meeting its demand is calculated (lines 6–11). If this profit is positive, then its value is added to the auxiliary variable. As we are interested in maximizing the profit value, the origin and destination pair of nodes that generate profits less than or equal to zero is not considered, and therefore will not be served. After this procedure, it is verified if the profit obtained by taking the current node as the hub node is better than the best value previously registered (lines 12–15). If this happens, the best solution found and its value are updated. Finally, the algorithm returns the initial solution and its value (line 17).

4.3. Neighborhood structures

In this work, the solution space is explored by applying six types of moves from a solution $s = (H, Z)$. Each move gives rise to a neighborhood structure $\mathcal{N}_i, i = 1, \ldots, 6$. To evaluate the complexity of each neighborhood structure, that is, the number of neighbors of a given solution s, consider that $H^0 \subset N$ and $Z^0 \subset N \times N$ denote the sets of inactive hubs and inactive hub arcs, respectively. Thus, the neighborhood structures are detailed below.

- $\mathcal{N}_1(s)$: set of solutions that can be obtained from solution s by installing a new hub. The number of solutions in this neighborhood is $|H^0|$.
- $\mathcal{N}_2(s)$: set of solutions that can be obtained from solution s by removing an installed hub. In this case, the hub arcs incident on that hub are also removed. The number of solutions in this neighborhood is $|H|$.
- $\mathcal{N}_3(s)$: set of solutions that can be obtained from solution s by installing a hub arc connecting installed hubs. The number of solutions in this neighborhood is $|Z^0|$.
- $\mathcal{N}_4(s)$: set of solutions that can be obtained from solution s by removing a hub arc. The number of solutions in this neighborhood is $|Z|$.
- $\mathcal{N}_5(s)$: set of solutions that can be obtained from solution s by installing a new hub and connecting it to all the other hubs in the network. The number of solutions in this neighborhood is $|H^0|$.
- $\mathcal{N}_6(s)$: set of solutions that can be obtained from solution s by exchanging an installed hub for an uninstalled one. In this case, hub arcs incident on the removed hub are also removed. The number of solutions in this neighborhood is $|H| \times |H^0|$.

4.4. Local search

The proposed algorithms employ a local search procedure based on a variant of the Variable Neighborhood Descent (VND) algorithm (Hansen et al., 2017), which performs a systematic exchange of the neighborhood structures presented in Section 4.3, using the best improvement strategy. Given an ordered set of neighborhood structures, the local search procedure is performed as follows. From the first neighborhood structure, whenever a local search in the current neighborhood does not improve the best-known solution, a local search in the next neighborhood structure is performed. In addition, when a better solution is found, the algorithm restarts in the first neighborhood, according to the predetermined order.

In this work, the local search uses the Random Variable Neighborhood Descent (RVND) method (Souza et al., 2010; Subramanian et al., 2010). In RVND, presented in Algorithm 3, the exploration order of the neighborhoods is generated randomly at the beginning of the algorithm. This strategy differs from VND, in which this order is previously established, generally according to the order of complexity of the neighborhoods.

The advantage of using RVND instead of the classic VND version lies mainly in the fact that the randomness of the order of the neighborhoods can be favorable in solving problem instances since they present different characteristics; thus, different orders may be more convenient.

Algorithm 3 *Random Variable Neighborhood Descent* - RVND

Input: Solution s, Set of neighborhoods $\mathcal{N}$
Output: Refined solution s

$\mathcal{NR} \leftarrow \mathcal{N}$ in random order
$i \leftarrow 1$
while $i \leq |\mathcal{N}|$ **do**
    Find the best neighbor $s' \in \mathcal{NR}_i(s)$
    if $f(s') > f(s)$ **then**
        $s \leftarrow s'$
        $i \leftarrow 1$
    else
        $i \leftarrow i + 1$
    end if
end while
return s

4.5. Perturbation

Let k be the level of perturbation in a given iteration. The perturbation procedure carried out in the current solution consists of selecting successively and randomly k neighbor solutions using the neighborhood structure $\mathcal{N}_6$. Algorithm 4 details this procedure. Initially, a copy of the current solution is made in the solution s', which will store the perturbed solution at the end of the procedure (line 1), and the counter i of the number of perturbations is started with the value 1 (line 2). The perturbation in the solution s' is performed as follows. While the counter is less than or equal to the quantity k of perturbations, a solution s'' in the neighborhood $\mathcal{N}_6(s')$ is randomly selected (line 4). Then, the perturbed solution and the counter of the number of perturbations are updated (lines 5–6). The procedure ends by returning the perturbed solution (line 8).

4.6. Iterated Local Search (ILS)

Iterated Local Search (ILS) (Lourenço et al., 2001, 2003) is a simple metaheuristic that explores the solution space through perturbations in locally optimal solutions. Instead of exploring the entire solution space, the main idea of this method is to perform the search in a smaller subspace, considering only solutions that are locally optimal.

Algorithm 4 Perturbation

Input: Current solution s, Neighborhood structure $\mathcal{N}_6$, Perturbation level k
Output: Perturbed solution s'

1: $s' \leftarrow s$
2: $i \leftarrow 1$
3: **while** $i \leq k$ **do**
4:     Randomly select a neighboring solution $s'' \in \mathcal{N}_6(s')$
5:     $s' \leftarrow s''$
6:     $i \leftarrow i + 1$
7: **end while**
8: **return** s'

The perturbation procedure applied by the method, which allows obtaining a locally optimal solution from another locally optimal solution, needs to be strong enough to allow the local search to explore different regions of the solution space and weak enough to avoid random restarts. In this way, ILS combines two very important elements: intensification and diversification. Intensification consists of properly exploring a particular region of the solution space, while diversification involves altering the search region. These elements are usually obtained according to the intensity of the perturbation applied to the solutions so that weaker perturbations allow a region to be better explored, and stronger perturbations favor the investigation of distinct regions.

The ILS metaheuristic has already been successfully applied to solve hub location problems. Guan et al. (2018) proposes an ILS algorithm to handle the uncapacitated single allocation hub location problem. To tackle a hub location problem under congestion, Karimi-Mamaghan et al. (2020) proposed a learning-based ILS. This algorithm incorporates machine learning techniques with the ILS heuristic, which is a very promising research area (Karimi-Mamaghan et al., 2022).

Algorithm 5 outlines the method named ILS-RVND, applied to the PMHNDP. The algorithm takes as input the maximum number of iterations without improvement, denoted by $iterMax$. In the first step of the algorithm, the level of perturbation, given by k, and the number of iterations, denoted by $iter$, are initialized (lines 2 and 1). Then, an initial solution s^0 is generated using Algorithm 2. After that, the algorithm performs a local search on s^0 using the RVND, outlined in Algorithm 3, obtaining a refined solution s (line 4). The remainder of the algorithm is composed of a repetition structure to improve the value of the current solution until reaching the maximum number of iterations without improvement, given by $iterMax$ (lines 5–16), where the number of iterations is incremented by one unit in line 6. At each iteration, the perturbation procedure, presented in Algorithm 4, is applied to the current solution, obtaining a solution s' (line 7). Then, a solution s'' is achieved by applying the RVND to s' (line 8). After that, it is checked if the solution s'' is better than the solution s (line 9). If this is true, then the solution s is updated to s'' and the number of iterations and perturbation level counters are reset (lines 10 to 12). Otherwise, the perturbation level k is increased (lines 13 to 15). The algorithm ends by returning the best solution obtained during this procedure.

4.7. Enhanced Iterated Local Search (E-ILS)

The Enhanced Iterated Local Search (E-ILS) metaheuristic (Reinsma et al., 2018) is an adaptation of the Iterated Local Search (ILS) method. The difference between them refers to how the perturbation level is updated. There is an increase in the perturbation level in ILS whenever there is no improvement in the current solution. However, this ILS strategy can lead to a loss of solution quality due to the abrupt change in the search region since the perturbation occurs at random. On the other hand, in E-ILS, the perturbation level is only increased after a few unsuccessful attempts at improvement. This E-ILS strategy allows a

Algorithm 5 ILS-RVND

Input: *iterMax*
Output: s

$k \leftarrow 1$
$iter \leftarrow 0$
$s^0 \leftarrow$ InitialSolution()   ▷ According to Algorithm 2
$s \leftarrow$ RVND($s^0, \mathcal{N}$)   ▷ According to Algorithm 3
while $iter \le iterMax$ **do**
    $iter \leftarrow iter + 1$
    $s' \leftarrow$ Perturbation($s, \mathcal{N}_6, k$)   ▷ According to Algorithm 4
    $s'' \leftarrow$ RVND($s', \mathcal{N}$)   ▷ According to Algorithm 3
    if $f(s'') > f(s)$ **then**
        $s \leftarrow s''$
        $iter \leftarrow 0$
        $k \leftarrow 1$
    else
        $k \leftarrow k + 1$
    end if
end while
return s

Algorithm 6 E-ILS-RVND

Input: *iterMax*, *timesMax*
Output: s

1: $k \leftarrow 1$
2: $iter \leftarrow 0$
3: $ntimes \leftarrow 1$
4: $s^0 \leftarrow$ InitialSolution()   ▷ According to Algorithm 2
5: $s \leftarrow$ RVND($s^0, \mathcal{N}$)   ▷ According to Algorithm 3
6: **while** $iter \le iterMax$ **do**
7:     $iter \leftarrow iter + 1$
8:     $s' \leftarrow$ Perturbation($s, \mathcal{N}_6, k$)   ▷ According to Algorithm 4
9:     $s'' \leftarrow$ RVND($s', \mathcal{N}$)   ▷ According to Algorithm 3
10:     **if** $f(s'') > f(s)$ **then**
11:         $s \leftarrow s''$
12:         $iter \leftarrow 0$
13:         $k \leftarrow 1$
14:         $ntimes \leftarrow 1$
15:     **else**
16:         **if** $ntimes \ge timesMax$ **then**
17:             $k \leftarrow k + 1$
18:             $ntimes \leftarrow 1$
19:         **else**
20:             $ntimes \leftarrow ntimes + 1$
21:         **end if**
22:     **end if**
23: **end while**
24: **return** s

better investigation to be carried out in a certain region of the solution space, enabling a more precise intensification during the algorithm's execution.

The E-ILS-RVND method, implemented for solving the PMHNDP, is presented in Algorithm 6. This algorithm has, as inputs, the maximum number of iterations without improvement, given by $iterMax$, and the maximum number of times in the same level of perturbation, given by $timesMax$. Initially, the number of iterations without improvement, denoted by $iter$, the perturbation level, denoted by k, and the number of times within the same perturbation level, denoted by $ntimes$, are initialized (lines 1–3). After that, an initial solution s^0 is generated using Algorithm 2 (line 4). Then, a local search is performed on the initial solution, using RVND (Algorithm 3), and the returned solution is then stored in s. Later, the algorithm enters a loop, with the purpose of improving the value of the current solution until the number of iterations without improvement is equal to $iterMax$ (lines 6–23). At the beginning of the loop, the iteration count without improvement is incremented (line 7). After that, a solution s' is obtained through the perturbation procedure applied to the solution s, as described in Algorithm 4 (line 8). Then, a local search is performed on the solution s', and the refined solution is stored in s'' (line 9). If the solution s'' is better than solution s, then the solution s is updated to s'', and the counter of iterations without improvement, the perturbation level, and the number of times in the same level of perturbation are reset (lines 10–14). If the objective value of s'' is not better than the objective value of the solution s, then it is checked if the number of times in the same level of perturbation reached its maximum value (lines 16–21). If this occurs, then the perturbation level is increased by one unit, and the number of times within the same perturbation level is reset. Otherwise, $ntimes$ is updated. At the end of the procedure, the algorithm returns the best-obtained solution s.

5. Computational experiments

This section presents the results of the computational experiments carried out using the proposed algorithms, ILS-RVND and E-ILS-RVND, applied to the PMHNDP. The algorithms were implemented in the C++ language and executed on a computer Intel Core i7-7500U, 2.70 GHz, with 16 GB of RAM, using the Linux environment. The parameters of the heuristic algorithms were calibrated by the IRACE tool (Iterated Racing for Automatic Algorithm Configuration) (López-Ibañez et al., 2016). The IRACE tool provides an implementation of the iterated F-race algorithm (Birattari et al., 2010) and other variants of the algorithm, which include, for instance, mechanisms to avoid premature convergence. This study uses the default version of the IRACE.

The computational experiments were performed using two benchmark instances for hub location problems: (i) the *Australian Post* (AP) data set, which includes instances with up to 200 nodes, introduced by Ernst and Krishnamoorthy (1996), and (ii) the set of instances proposed by Contreras et al. (2011a), which have instances with up to 500 nodes and will be denoted here as the Group 2 data set. Sections 5.1 and 5.2 detail and report the experiments using the AP and the Group 2 data set, respectively.

Although the comparison of the results for the algorithms proposed in this work with the results for the algorithm presented by Zhang et al. (2023) is not fair, since the authors implemented their algorithm using another programming language (python), and ran the experiments on a machine with a different configuration, and used only small instances of the CAB data set, Appendix C presents the results of the comparison between the results of the two studies.

5.1. Tests with AP instances

This subsection presents the results of the computational experiments using the AP instances. Initially, Section 5.1.1 details this data set and the values of the model parameters not defined in the data set. Section 5.1.2 presents the results of the calibration of the parameters of the algorithms. Section 5.1.3 presents an analysis of the neighborhood structures. To compare the performance of the algorithms, experiments were performed in two steps. In step 1, tests were performed with instances with 40, 50, 75, and 100 nodes. Meanwhile, step 2 contemplated instances with 125, 150, 175, and 200 nodes. Section 5.1.4 presents and discusses the results of step 1. Furthermore, Section 5.1.5 presents the results of a statistical analysis of the performance of the algorithms based on experiments from step 1. Finally, Section 5.1.6 shows the results of the computational experiments regarding step 2.

Table 3
Set of instances from AP data set submitted to IRACE.

AP - 40			AP - 50			AP - 75			AP – 100		
Inst	α	Revenue	Inst	α	Revenue	Inst	α	Revenue	Inst	α	Revenue
40L	0.2	20	50L	0.6	20	75L	0.2	20	100L	0.6	20
40L	0.4	30	50L	0.4	30	75L	0.6	30	100L	0.2	30
40L	0.8	50	50L	0.2	50	75L	0.8	50	100L	0.4	50
40T	0.4	20	50T	0.2	20	75T	0.8	20	100T	0.4	20
40T	0.8	30	50T	0.6	30	75T	0.4	30	100T	0.2	30
40T	0.2	50	50T	0.8	50	75T	0.6	50	100T	0.8	50

5.1.1. AP data set

The *Australian Post* (AP) data set, introduced by Ernst and Krishnamoorthy (1996), has instances with up to 200 nodes and provides the distance and demand between each pair of nodes. The demand matrix (w_{ij}) is not symmetric and the distances between nodes are taken as the transportation costs (c_{ij}). The revenue (r_{ij}) between each origin and destination pair is considered independent of the location of the pair of nodes. Based on Taherkhani and Alumur (2018), three values were considered: 20, 30 and 50. The AP data set also provides two sets of values for fixed hub installation costs (s_k), called loose (L) and tight (T). It was assumed that the fixed cost of installing hubs arcs (g_{km}) corresponds to 10% of the average of all fixed costs of installing hubs, as suggested in Taherkhani and Alumur (2018). Collection and distribution costs per unit of demand were not considered and it is considered the discount factor $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$.

Instances with 40, 50, 100, and 200 nodes are the same as those originally available in the AP data set. The other instances, with 75, 125, 150, and 175 nodes, were generated according to the procedure indicated by Oliveira et al. (2022).

5.1.2. Calibration of the parameters for the AP instances

The parameter values of the two implemented algorithms were calibrated by the IRACE tool (Iterated Racing for Automatic Algorithm Configuration) (López-Ibañez et al., 2016). Table 3 shows the instances used, from the AP data set, in the execution of IRACE.

For both algorithms, the maximum number of iterations without improvement $(iterMax)$ was calibrated and, additionally, for E-ILS-RVND, the parameter that determines the number of times at the same level of perturbation $(timesMax)$ was also calibrated. For the IRACE execution, the following ranges of values were considered for the parameters: $iterMax \in \{1, 2, 3, 4, 5\}$ and $timesMax \in \{2, 3, 4, 5, 6\}$. For ILS-RVND, it was returned $iterMax = 1$, while the values returned for E-ILS-RVND were $iterMax = 4$ and $timesMax = 3$. These parameter values were adopted for the two algorithms in all computational experiments.

5.1.3. Analysis of the neighborhood structures

A set of computational experiments was carried out to analyze the impact of the neighborhood structures, described in Section 4.3, in the heuristic. These experiments consisted of removing each neighborhood structure from the executions of the E-ILS-RVND algorithm, keeping the other five neighborhoods. The version of the algorithm considering all neighborhood structures is denoted by N+all and the version of the algorithm without the neighborhood structure $\mathcal{N}_i$ is denoted by N-i, $i = 1, \dots, 6$. The experiments used the AP instances with 40, 50, 75, and 100 nodes, excluding those instances used in the calibration of the parameters (indicated in Table 3). For each instance, were carried out 10 runs of the algorithm, using the values of the parameters indicated by IRACE in Section 5.1.2.

Fig. 3 presents the number of times that an optimal value was returned by each version of the algorithm. Note that, for all cases, the greatest number of optimal values was achieved when all six neighborhood structures are used. Furthermore, the removal of the neighborhood $\mathcal{N}_3$ reduced the number of optimal values achieved for instances with sizes 40 and 50. On the other hand, the removal of the neighborhood $\mathcal{N}_1$ reduced the number of optimal values, for instances with 75 and 100 nodes.

To evaluate the impact of the neighborhood structures on the quality of the solutions achieved, the average of the gaps between the optimal value and the average objective values of the solutions provided by the E-ILS-RVND algorithm, denoted by Avg_{gaps}, was calculated, which is given by

$$Avg_{gaps} = 100 \times \left(\frac{1}{q \times r} \sum_{i=1}^{q} \sum_{j=1}^{r} \frac{opt_{f(i)} - f(i,j)}{opt_{f(i)}} \right),$$

where q denotes the number of instances, r denotes the number of executions of the algorithm, $f(i,j)$ denotes the objective value of the best solution found for the instance i in the execution j, and $opt_{f(i)}$ denotes the objective value of the optimal solution for the instance i. In our case, we have $q = 18$ and $r = 10$. The results, in percentage, are shown in Fig. 4.

Similar to the results previously presented, the best average gaps were attained when running the algorithm with all neighborhood structures. Note that, for the set of instances with 100 nodes, the smallest and the largest value of the average gap were obtained when considering all neighborhood structures and removing the neighborhood $\mathcal{N}_1$, respectively. Because of the presented results, all neighborhood structures are used in the following computational experiments.

5.1.4. Results of step 1

In Step 1, computational experiments were performed with instances of 40, 50, 75, and 100 nodes. These experiments were carried out to obtain a preliminary analysis of the behavior and performance of the proposed methods, as well as to compare the results achieved with those presented in the literature. The instances used to calibrate the parameters of the algorithms, indicated in Table 3, were discarded from these experiments.

The results of these computational experiments are presented in Tables 4 and 5. Table 4 shows the results obtained with instances with 40 and 50 nodes, while Table 5 shows the results for instances of size 75 and 100. In these tables, the first column indicates the tested instances, represented in the format $nF - \alpha - r$, where n represents the number of nodes, F represents the kind of fixed cost for installing hubs (L for loose and T for tight), α represents the discount factor on hub arcs and r represents the revenue. The optimal value and the best runtime obtained in Oliveira et al. (2022) are presented, where the instances were solved using an exact algorithm based on the Benders decomposition method that makes insertions of non-dominated cuts in a branch-and-cut structure. The next columns of the tables show, for each of the proposed algorithms, the best value, the average value, the gap associated with the best value (Δ_{best}), the gap associated with the average value (Δ_{avg}) and the average runtime, in seconds, recorded in 30 executions of the algorithms. The gap associated with the best value and the gap associated with the average value were calculated by:

$$\Delta_{best} = 100 \times \frac{\text{Optimal value} - \text{Best value}}{\text{Optimal value}} \tag{18}$$

and

$$\Delta_{avg} = 100 \times \frac{\text{Optimal value} - \text{Average value}}{\text{Optimal value}}. \tag{19}$$

These values represent, respectively, the percentage deviations of the best objective value and the average objective value obtained for each instance relative to the optimal objective value. Thus, the closer to zero these values are, the better the results achieved by the algorithms.

Table 4 indicates that the proposed heuristic algorithms achieved good results for instances with 40 and 50 nodes. The optimal value was obtained for almost all instances. ILS-RVND did not attain the optimal solution in four instances, while E-ILS-RVND did not reach the optimal solution in only one. The average values were close to or equal to the optimal value for most cases. This fact is reflected in the low values recorded for the average gap of both algorithms, except for some

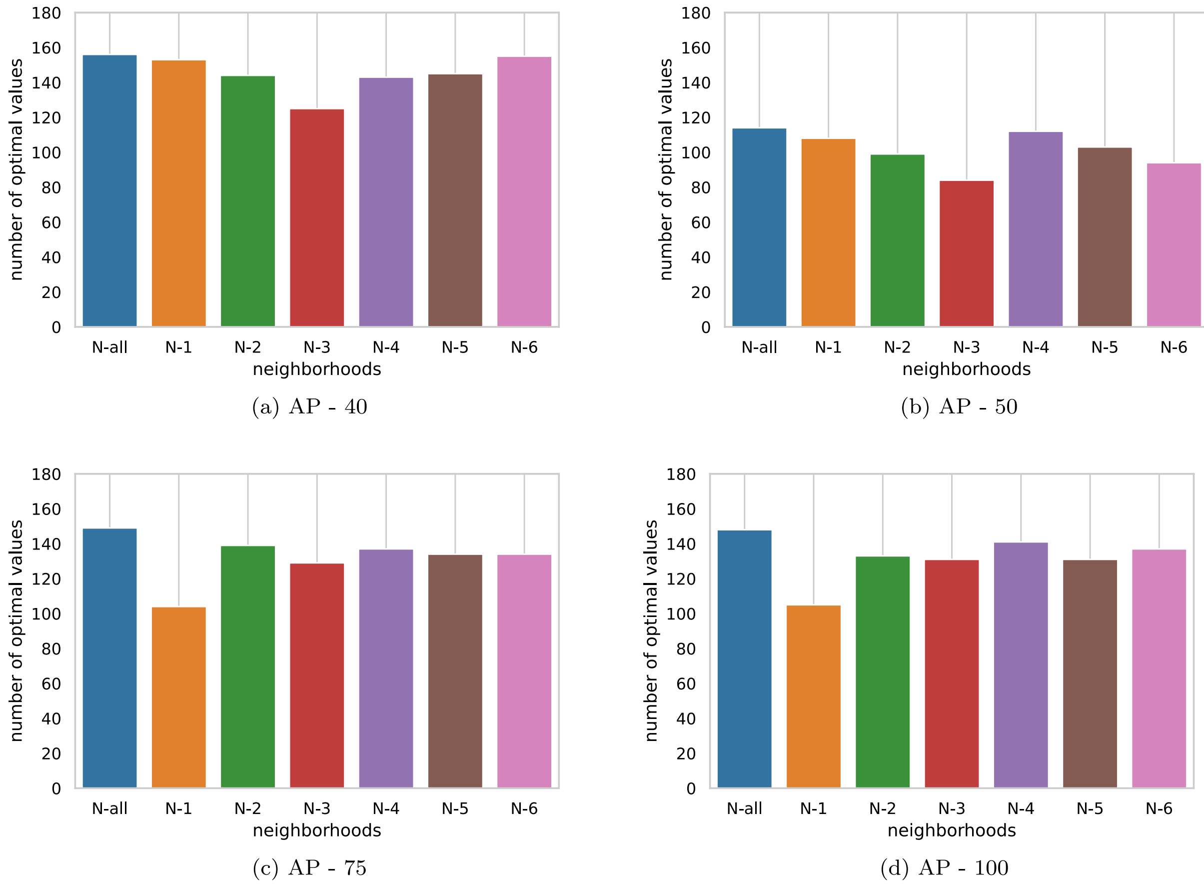


Fig. 3. Number of optimal values in 10 runs of the E-ILS-RVND algorithm, considering all neighborhood structures (N+all) and removing the ith neighborhood $(N-i)$, $i = 1, \ldots, 6$.

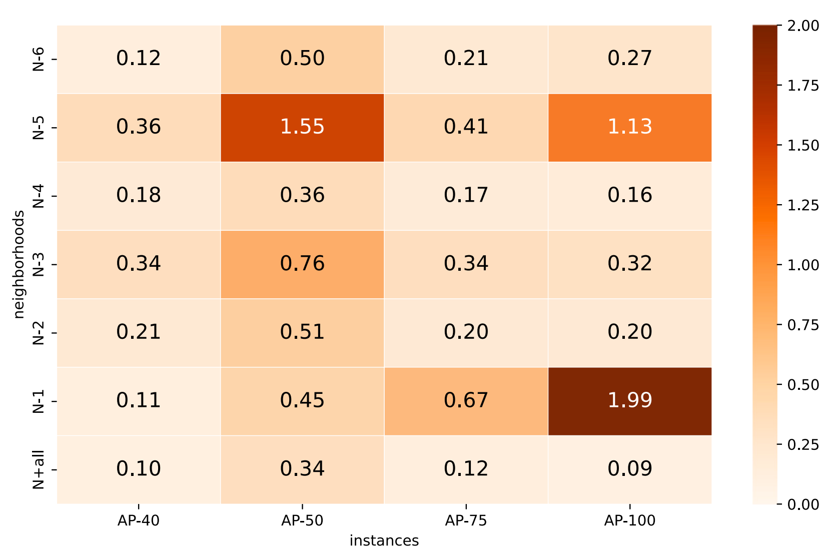


Fig. 4. $Avg_{gaps}(\%)$ for the versions of the E-ILS-RVND algorithm with all neighborhood structures (N+all) and removing the ith neighborhood $(N-i)$, $i = 1, \ldots, 6$.

instances that have small values for α. In general, the instances with lower values of α are more challenging once a greater discount on hub arcs encourages the installation of more hubs and arcs between hubs in the network. We also see in this table that E-ILS-RVND had better values for the average gap than ILS-RVND. Regarding computational time, the values recorded by the proposed algorithms are less than the ones recorded by the exact method.

The results in Table 5 show that the proposed algorithms also achieved good performance for instances with 75 and 100 nodes. Regarding the quality of the solution produced, E-ILS-RVND attained the optimal value of all tested instances, and ILS-RVND only did not reach the optimum in two instances (100L-0.2-50 and 75L-0.4-50). Good average values were obtained, as the small average gaps highlight. As seen in Table 4, E-ILS-RVND recorded lower values for the average gap

Table 4
Results for AP instances with 40 and 50 nodes — 30 runs.

Instances	Benders		ILS-RVND					E-ILS-RVND				
	Optimal value	Time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)
40L-0.4-20	14,635.61	34.67	14,635.61	14,543.83	0.00	0.63	0.45	14,635.61	14,593.00	0.00	0.29	2.65
40L-0.6-20	14,099.19	20.11	14,099.19	14,099.19	0.00	0.00	0.17	14,099.19	14,099.19	0.00	0.00	0.63
40L-0.8-20	14,099.19	12.22	14,099.19	14,099.19	0.00	0.00	0.15	14,099.19	14,099.19	0.00	0.00	0.59
40T-0.2-20	6,404.58	35.10	6,404.58	6,404.58	0.00	0.00	0.18	6,404.58	6,404.58	0.00	0.00	0.60
40T-0.6-20	6,404.58	16.54	6,404.58	6,404.58	0.00	0.00	0.15	6,404.58	6,404.58	0.00	0.00	0.53
40T-0.8-20	6,404.58	15.17	6,404.58	6,404.58	0.00	0.00	0.15	6,404.58	6,404.58	0.00	0.00	0.46
40L-0.2-30	52,299.43	106.70	52,299.43	51,688.43	0.00	1.17	4.62	52,299.43	52,021.38	0.00	0.53	14.45
40L-0.6-30	44,213.55	40.06	44,213.55	44,213.55	0.00	0.00	1.83	44,213.55	44,213.55	0.00	0.00	6.29
40L-0.8-30	42,990.07	27.32	42,990.07	42,990.07	0.00	0.00	1.32	42,990.07	42,981.96	0.00	0.02	4.26
40T-0.2-30	36,135.48	55.02	36,135.48	35,993.61	0.00	0.39	0.76	36,135.48	36,135.48	0.00	0.00	2.54
40T-0.4-30	33,918.82	39.19	33,918.82	33,918.82	0.00	0.00	0.55	33,918.82	33,918.82	0.00	0.00	2.00
40T-0.6-30	32,709.49	30.78	32,709.49	32,709.49	0.00	0.00	0.58	32,709.49	32,709.49	0.00	0.00	1.74
40L-0.2-50	130,906.36	74.92	130,240.82	129,053.68	0.51	1.42	5.47	130,747.04	129,880.77	0.12	0.78	37.55
40L-0.4-50	124,056.22	97.30	124,056.22	123,690.01	0.00	0.30	5.32	124,056.22	123,955.55	0.00	0.08	19.66
40L-0.6-50	120,056.17	49.27	120,056.17	119,646.85	0.00	0.34	4.60	120,056.17	119,900.24	0.00	0.13	12.76
40T-0.4-50	107,847.14	55.77	107,847.14	107,838.95	0.00	0.01	0.96	107,847.14	107,845.35	0.00	0.00	3.60
40T-0.6-50	105,857.00	49.38	105,857.00	105,857.00	0.00	0.00	0.61	105,857.00	105,857.00	0.00	0.00	2.05
40T-0.8-50	105,603.50	32.36	105,603.50	105,603.50	0.00	0.00	0.56	105,603.50	105,603.50	0.00	0.00	1.74
50L-0.2-20	17,666.95	148.99	17,666.95	17,383.42	0.00	1.60	3.63	17,666.95	17,425.21	0.00	1.37	13.67
50L-0.4-20	14,588.82	87.25	14,588.82	14,588.82	0.00	0.00	1.30	14,588.82	14,588.82	0.00	0.00	3.47
50L-0.8-20	13,920.55	29.77	13,920.55	13,920.55	0.00	0.00	0.30	13,920.55	13,920.55	0.00	0.00	1.12
50T-0.4-20	8,001.47	56.35	8,001.47	8,001.47	0.00	0.00	0.29	8,001.47	8,001.47	0.00	0.00	0.93
50T-0.6-20	8,001.47	40.85	8,001.47	8,001.47	0.00	0.00	0.31	8,001.47	8,001.47	0.00	0.00	0.93
50T-0.8-20	8,001.47	37.27	8,001.47	8,001.47	0.00	0.00	0.29	8,001.47	8,001.47	0.00	0.00	0.92
50L-0.2-30	53,200.19	334.89	53,066.89	51,896.89	0.25	2.45	16.06	53,200.19	52,854.48	0.00	0.65	62.84
50L-0.6-30	43,763.67	121.95	43,763.67	43,357.24	0.00	0.93	8.59	43,763.67	43,577.47	0.00	0.43	24.74
50L-0.8-30	42,566.95	227.77	42,566.95	42,566.95	0.00	0.00	6.13	42,566.95	42,566.95	0.00	0.00	15.28
50T-0.2-30	34,513.57	219.72	34,513.57	33,225.54	0.00	3.73	1.96	34,513.57	34,162.25	0.00	1.02	7.07
50T-0.4-30	31,911.87	114.27	31,911.87	31,681.42	0.00	0.72	1.36	31,911.87	31,750.56	0.00	0.51	4.57
50T-0.8-30	31,220.52	58.82	31,220.52	31,220.52	0.00	0.00	1.25	31,220.52	31,220.52	0.00	0.00	3.61
50L-0.4-50	124,770.82	265.10	124,770.82	124,134.13	0.00	0.51	19.35	124,770.82	124,267.84	0.00	0.40	61.06
50L-0.6-50	119,972.77	445.13	119,902.39	119,440.87	0.06	0.44	16.49	119,972.77	119,707.89	0.00	0.22	53.70
50L-0.8-50	118,299.45	82.92	118,299.45	118,214.17	0.00	0.07	8.90	118,299.45	118,287.27	0.00	0.01	21.00
50T-0.2-50	110,962.98	289.18	110,858.78	109,713.87	0.09	1.13	3.85	110,962.98	110,226.98	0.00	0.66	17.47
50T-0.4-50	106,451.33	165.83	106,451.33	106,285.12	0.00	0.16	1.71	106,451.33	106,439.17	0.00	0.01	6.04
50T-0.6-50	105,260.70	115.27	105,260.70	104,824.69	0.00	0.41	1.43	105,260.70	104,933.70	0.00	0.31	5.25
Average	–	100.92	–	–	0.03	0.46	3.38	–	–	0.00	0.21	11.60

than ILS-RVND. For these instances, the computational times obtained by the heuristic algorithms are much lower than the ones shown by the Benders decomposition method, mainly for instances with a greater economy of scale. For example, the average runtime recorded by ILS-RVND is 145 times less than that obtained with the exact method.

Fig. 5 presents a graphical visualization of the computational times spent by the Benders decomposition method and by the developed algorithms. The logarithmic scale was used because of the difference in the order of magnitude of the obtained values. As discussed earlier, this figure shows that, in all analyzed cases, the heuristic algorithms recorded significantly lower runtimes compared to the exact method and that this difference tends to increase with the size of the instances. ILS-RVND had the shortest computational times. This behavior was already expected due to the result of the parameter calibration indicated by IRACE ($iterMax = 1$) and the fact that this algorithm is simpler than E-ILS-RVND. Fig. 5 also shows that the instances with the lowest discount factors in the arcs hub are the ones that consume more processing time.

As discussed by Oliveira et al. (2022), for the PMHNDP, tight cost AP instances are less challenging than loose cost instances. Note that the variation of α does not influence the optimal values of these instances since arcs between hubs are not installed in the network in these cases. Tables 4 and 5 also show that, in general, heuristic algorithms spend less time solving these instances.

Therefore, the results showed in Tables 4 and 5 suggest that the two proposed algorithms have satisfactory performance and are very promising since they could generate good solutions in acceptable computational times (according to the dimension of the instances). Note also that, for 58% of the instances present in these tables, both algorithms reached the optimal values in all executions.

To complement the analysis of the proposed algorithms, we investigate their convergence in some instances of the AP data set. For this analysis, we chose one instance from each size, in which the E-ILS-RVND recorded the worst values for the average gaps reported in Tables 4 and 5. Hence, the instances selected were 40L-0.2-50, 50L-0.2-20, 75L-0.4-30, and 100L-0.2-20. The results are reported in Fig. 6, which shows the gaps relative to the optimal value of the objective function over 0–1 normalized runtimes, where 0 is the start of the algorithms and 1 is the termination. For each instance, the results obtained for 30 runs of the algorithms are shown. This figure allows us to conclude that the E-ILS-RVND algorithm has a faster and more pronounced convergence than the ILS-RVND for all analyzed instances.

In order to complement the tests carried out in this section, Appendix A presents results with the ILS-RVND algorithm considering $iterMax = 4$, that is, the same value used for the E-ILS-RVND. In turn, Appendix B presents the results for the PMHNDP version in which direct connections between non-hub nodes are allowed.

Table 5
Results for AP instances with 75 and 100 nodes — 30 runs.

Instances	Benders		ILS-RVND					E-ILS-RVND				
	Optimal value	Time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)
75L-0.4-20	15,004.99	1,348.26	15,004.99	14,864.97	0.00	0.93	5.25	15,004.99	14,930.70	0.00	0.50	21.24
75L-0.6-20	14,058.46	732.49	14,058.46	14,058.46	0.00	0.00	1.03	14,058.46	14,058.46	0.00	0.00	4.41
75L-0.8-20	14,058.46	577.37	14,058.46	14,058.46	0.00	0.00	1.06	14,058.46	14,058.46	0.00	0.00	4.17
75T-0.2-20	1,144.13	931.39	1,144.13	1,144.13	0.00	0.00	0.83	1,144.13	1,144.13	0.00	0.00	2.91
75T-0.4-20	1,144.13	627.04	1,144.13	1,144.13	0.00	0.00	0.85	1,144.13	1,144.13	0.00	0.00	2.91
75T-0.6-20	1,144.13	545.76	1,144.13	1,144.13	0.00	0.00	0.81	1,144.13	1,144.13	0.00	0.00	2.84
75L-0.2-30	53,570.37	3,516.02	53,570.37	53,075.54	0.00	0.92	42.25	53,570.37	53,283.50	0.00	0.54	115.87
75L-0.4-30	48,097.70	1,988.41	48,097.70	47,590.01	0.00	1.06	31.17	48,097.70	47,708.07	0.00	0.81	112.47
75L-0.8-30	43,645.17	932.62	43,645.17	43,645.17	0.00	0.00	21.60	43,645.17	43,645.17	0.00	0.00	51.52
75T-0.2-30	25,999.62	2,298.20	25,999.62	25,999.62	0.00	0.00	2.72	25,999.62	25,999.62	0.00	0.00	8.91
75T-0.6-30	25,999.62	872.35	25,999.62	25,999.62	0.00	0.00	2.90	25,999.62	25,999.62	0.00	0.00	8.00
75T-0.8-30	25,999.62	644.32	25,999.62	25,999.62	0.00	0.00	2.73	25,999.62	25,999.62	0.00	0.00	8.27
75L-0.2-50	131,831.98	3,464.41	131,831.98	130,596.85	0.00	0.94	72.01	131,831.98	130,771.82	0.00	0.80	193.11
75L-0.4-50	125,133.91	2,366.14	125,097.05	124,581.66	0.03	0.44	45.30	125,133.91	124,852.08	0.00	0.23	194.81
75L-0.6-50	120,693.62	2,571.40	120,693.62	120,605.87	0.00	0.07	46.10	120,693.62	120,638.19	0.00	0.05	126.84
75T-0.2-50	99,186.32	2,374.03	99,186.32	99,186.32	0.00	0.00	6.73	99,186.32	99,186.32	0.00	0.00	19.69
75T-0.4-50	99,186.32	1,561.35	99,186.32	99,186.32	0.00	0.00	5.92	99,186.32	99,186.32	0.00	0.00	18.58
75T-0.8-50	99,186.32	818.08	99,186.32	99,186.32	0.00	0.00	5.39	99,186.32	99,186.32	0.00	0.00	15.85
100L-0.2-20	17,616.73	9,427.82	17,616.73	16,904.36	0.00	4.04	31.70	17,616.73	17,295.80	0.00	1.82	156.61
100L-0.4-20	14,211.00	6,299.13	14,211.00	14,211.00	0.00	0.00	9.23	14,211.00	14,211.00	0.00	0.00	38.13
100L-0.8-20	13,603.42	1,874.36	13,603.42	13,603.42	0.00	0.00	3.50	13,603.42	13,603.42	0.00	0.00	9.94
100T-0.2-20	2,116.57	3,470.80	2,116.57	2,116.57	0.00	0.00	2.41	2,116.57	2,116.57	0.00	0.00	6.79
100T-0.6-20	2,116.57	1,781.70	2,116.57	2,116.57	0.00	0.00	2.26	2,116.57	2,116.57	0.00	0.00	7.08
100T-0.8-20	2,116.57	1,610.26	2,116.57	2,116.57	0.00	0.00	2.12	2,116.57	2,116.57	0.00	0.00	6.59
100L-0.4-30	47,173.46	6,411.47	47,173.46	46,775.31	0.00	0.84	67.55	47,173.46	47,077.29	0.00	0.20	265.53
100L-0.6-30	43,747.51	4,758.21	43,747.51	43,728.59	0.00	0.04	63.05	43,747.51	43,741.20	0.00	0.01	182.42
100L-0.8-30	42,790.52	3,290.31	42,790.52	42,790.52	0.00	0.00	38.12	42,790.52	42,790.52	0.00	0.00	101.23
100T-0.4-30	25,271.32	3,435.93	25,271.32	25,271.32	0.00	0.00	5.96	25,271.32	25,271.32	0.00	0.00	19.78
100T-0.6-30	25,271.32	2,457.55	25,271.32	25,271.32	0.00	0.00	6.33	25,271.32	25,271.32	0.00	0.00	21.10
100T-0.8-30	25,271.32	2,059.14	25,271.32	25,271.32	0.00	0.00	6.97	25,271.32	25,271.32	0.00	0.00	17.59
100L-0.2-50	130,437.81	21,249.75	130,421.44	128,476.56	0.01	1.50	125.42	130,437.81	129,730.18	0.00	0.54	580.39
100L-0.6-50	119,319.31	6,389.96	119,319.31	119,226.65	0.00	0.08	84.75	119,319.31	119,268.86	0.00	0.04	259.29
100L-0.8-50	117,676.01	3,889.53	117,676.01	117,674.98	0.00	0.00	57.93	117,676.01	117,673.95	0.00	0.00	139.02
100T-0.2-50	98,113.77	8,729.69	98,113.77	98,113.77	0.00	0.00	15.15	98,113.77	98,113.77	0.00	0.00	49.99
100T-0.4-50	98,113.77	4,295.65	98,113.77	98,113.77	0.00	0.00	16.33	98,113.77	98,113.77	0.00	0.00	40.51
100T-0.6-50	98,113.77	3,667.19	98,113.77	98,113.77	0.00	0.00	14.63	98,113.77	98,113.77	0.00	0.00	41.91
Average	–	3,424.11	–	–	0.00	0.30	23.56	–	–	0.00	0.15	79.34

Table 6
Shapiro–Wilk normality test with 95% confidence using average gap values.

	$\Delta_{avg}^{\mathrm{ILS}}$	$\Delta_{avg}^{\mathrm{E\text{-}ILS}}$	$\Delta_{avg}^{\mathrm{ILS}} - \Delta_{avg}^{\mathrm{E\text{-}ILS}}$
p-values	1.951e−13	8.051e−13	1.071e−14

5.1.5. Statistical comparison of the proposed algorithms

To compare the proposed algorithms, statistical tests were performed using the average gap values (Δ_{avg}) obtained for instances belonging to set 1, with 40, 50, 75, and 100 nodes. Fig. 7 shows the boxplot graph of these values, reported in Tables 4 and 5. Note that the distribution of the values of Δ_{avg} is asymmetric for both algorithms and that ILS-RVND has greater variability than E-ILS-RVND.

Table 6 shows the p-values of the Shapiro–Wilk normality test (Shapiro and Wilk, 1965), with 95% confidence, applied to the average gaps obtained by each algorithm and also for the difference of these values. According to the results in this table, the hypothesis of data normality can be rejected.

Therefore, to analyze these data, the Paired Wilcoxon Signed-Rank Test (Wilcoxon, 1945) was applied, using the procedure proposed by Pratt (1959) to deal with the significant amount of zeros present in our data sets. A confidence level of 95% was considered and the hypotheses tested were:

$$\begin{cases} H_0: & \text{the median of the average gaps of ILS-RVND is equal to} \\ & \text{the median of the average gaps of E-ILS-RVND.} \\ H_1: & \text{the median of the average gaps of ILS-RVND is greater than} \\ & \text{the median of the average gaps of E-ILS-RVND.} \end{cases} \tag{20}$$

The p-value obtained as a result of this test was equal to p = 2.794e−09. This result means that the null hypothesis was rejected by the hypothesis test and, therefore, we can say that there is statistical evidence, with 95% confidence, that the median value of the gaps of ILS-RVND is greater than the median value of the gaps of E-ILS-RVND. Thus, we conclude that, for the parameters provided by IRACE, there is a significant difference between the implemented algorithms concerning the average gaps and that the lowest values of this statistic, and consequently, the best average values, were obtained by E-ILS-RVND.

5.1.6. Results of step 2

In this step, computational experiments were carried out using the AP instances with 125, 150, 175, and 200 nodes. It used the same values of revenue, installation costs of hubs and arcs between hubs, and the constant discount factor indicated in Section 5.1.1. For the parameters

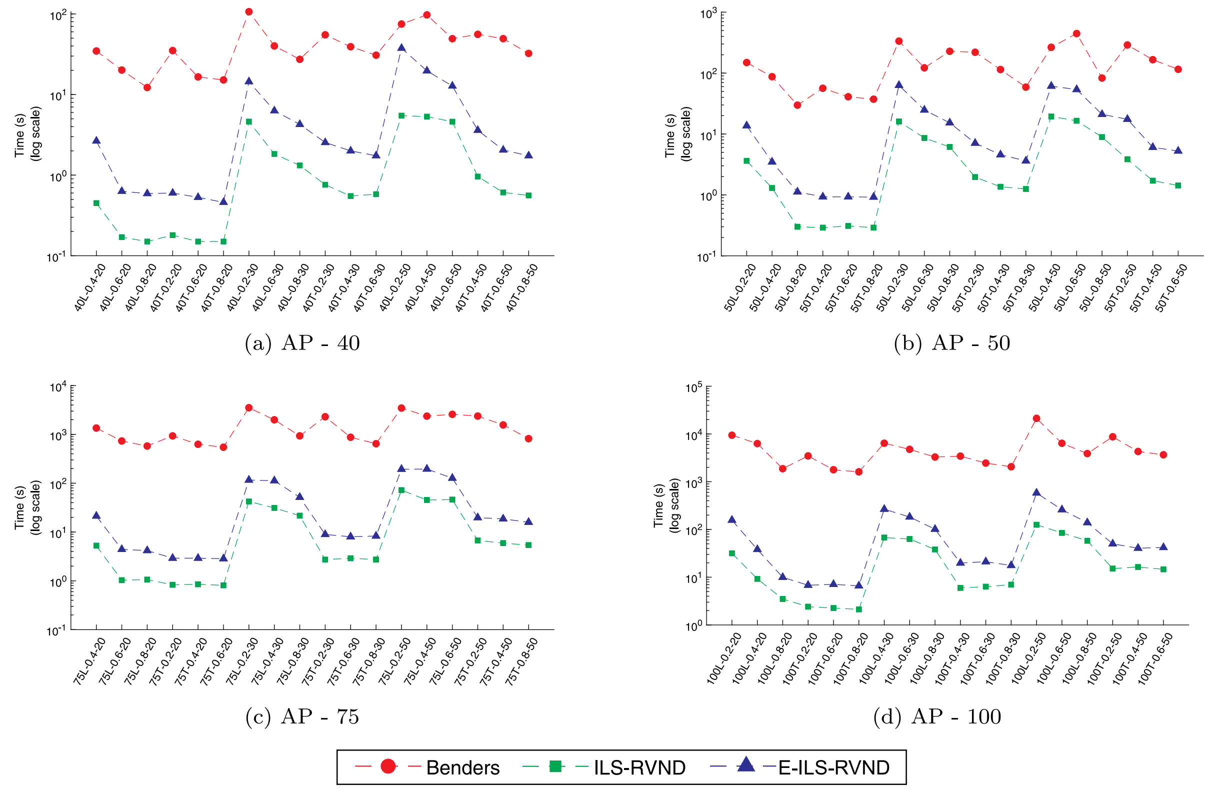


Fig. 5. Time spent by the Benders decomposition method and average times recorded by the proposed heuristic algorithms with AP instances — 40 to 100 nodes.

of the algorithms, the values obtained in the calibration performed in Section 5.1.2 were used. These tests were carried out to analyze the performance of the proposed algorithms when tackling larger instances. The results of this step, referring to a single execution of the algorithms, are given in Tables 7 and 8.

Table 7 shows the results for instances with 125 and 150 nodes. As before, the results obtained with the Benders decomposition, reported in Oliveira et al. (2022), are presented for the instances that the Benders decomposition algorithm was able to solve. For both algorithms proposed here, we reported the objective function value for the solution obtained by the algorithm (Value obtained), the relative gap between the value reached by the algorithm, the optimal objective function value registered by the exact method (Δ), calculated by:

$$\Delta = 100 \times \frac{\text{Optimal value} - \text{Value obtained by the algorithm}}{\text{Optimal value}}, \tag{21}$$

and the CPU time spent for the tested instances (Time).

The results in Tables 7(a) and 7(b) show that the two developed algorithms were able to tackle all the tested instances, while the exact method did not solve 9 of them once it exceeded the available memory of 16 GB. Regarding the quality of the solutions produced, we see that the gap relative to the optimal solution (calculated only for the instances solved by the Benders algorithm) was equal to zero for most cases. Thus, the algorithms obtained the optimal value, with a single execution, for most of these instances. For the 125L-0.6-50 instance, both ILS-RVND and E-ILS-RVND found better results than those achieved by the exact method due to the optimality gap tolerance. Additionally, we see that the computational times spent by the heuristic algorithms are much better than the computational times recorded by the exact method, especially for the more complex instances. Again, the ILS-RVND computational times were lower than the E-ILS-RVND computational times. On the other hand, the values of Δ obtained by E-ILS-RVND were better than those obtained with ILS-RVND.

Table 8 reports the results for the computational experiments using instances with 175 and 200 nodes. The objective function values of the solution obtained with a single execution of the algorithms (Value obtained) and the processing time spent to solve each instance (Time) are presented. We emphasize that the largest instance size solved in the literature so far for the PMHNDP, with the characteristics considered in this work, is 150 nodes and, for this reason, there are no reference values to perform a comparison with the results here presented, as done previously. Tables 8(a) and 8(b) show that E-ILS-RVND obtained solutions better than or equal to those found by ILS-RVND for most instances. This slightly better performance is in agreement with the observations made in Section 5.1.5 concerning the quality of the solutions provided by the developed algorithms. In addition, the E-ILS version also registered the highest computational times.

5.2. Computational experiments using Group 2 data set

For evaluating the efficiency and limitations of the proposed algorithms, computational tests were performed with the second group of instances, which contains instances with up to 500 nodes. This group of instances, proposed by Contreras et al. (2011a), consists of three sets of instances, called Sets I, II, and III. The composition of these sets considers three distinct levels of the amount of demand flows originating in the network nodes: low level (LL), medium level (ML), and high level (HL). The total flow originated at LL, ML, and HL nodes belonging to the ranges $[1, 10]$, $[10, 100]$, and $[100, 1000]$, respectively. Set I is composed of 2% of HL nodes, 38% of ML nodes, and 60% of LL nodes. Set II has 30% from HL nodes, 35% from ML nodes, and 35% from LL nodes. Finally, in Set III, the number of HL, ML, and LL nodes is 0%, 1%, and 99% of the total number of nodes, respectively. Instances

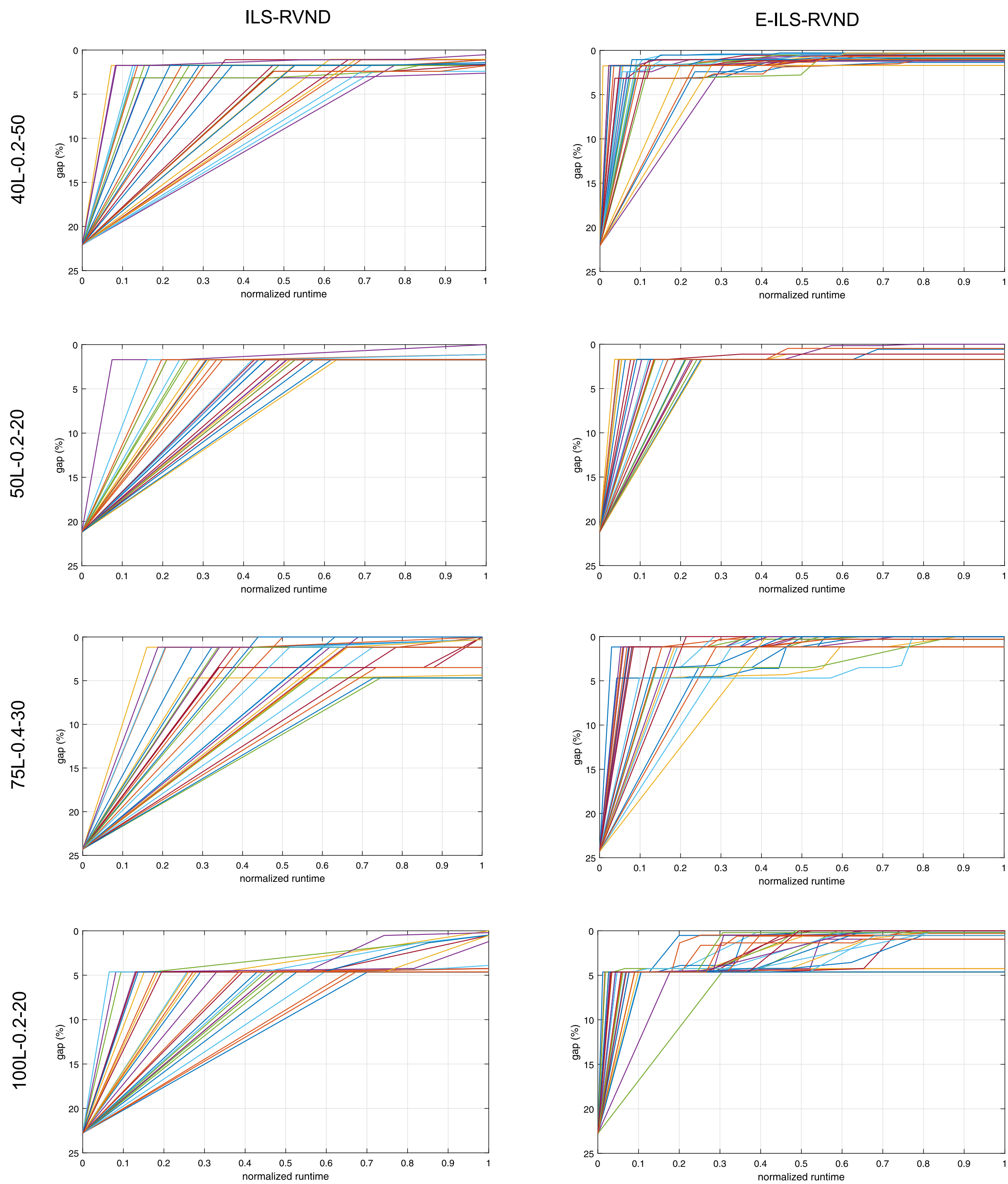


Fig. 6. Convergence of the proposed algorithms in some instances of the AP data set.

of Set I with 50, 100, 200, 300, 400, and 500 nodes were tested, and, from Sets II and III, instances of size 50, 100, 150, and 200 were tested.

These data sets provide the amount of demand between the origin and destination pairs (w_{ij}), the transportation cost in an arc connecting a pair of nodes (c_{ij}), and the fixed costs of installing hubs (s_k). For revenue, the values $r_{ij} \in \{2, 3, 5\}$ were considered. As defined for the AP instances, the fixed costs for installing hub arcs (g_{km}) were taken to be 10% of the average of all fixed costs of installing hubs, and the constant discount factor (α) equal to 0.2, 0.4, 0.6, and 0.8.

5.2.1. Calibration of parameters using instances of Group 2

Since the characteristics of the Group 2 data set are different from those presented by the AP instances, a new calibration of the parameters of the algorithms was performed by IRACE, considering instances of this new group. Table 9 shows the instances used in the calibration. Note that only instances from Set I were submitted to IRACE, as we are more interested in evaluating the performance of metaheuristics on the largest available instances.

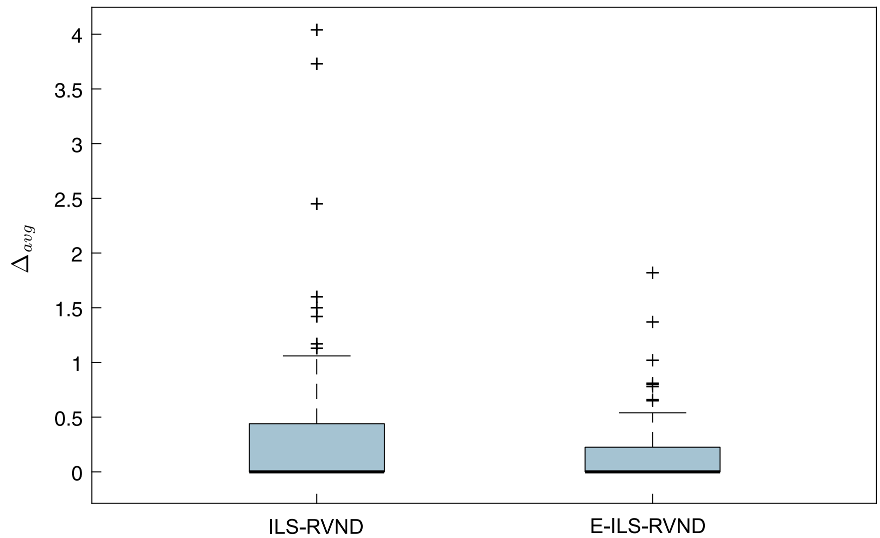


Fig. 7. Boxplot of the average gaps of AP instances with 40, 50, 75 e 100 nodes.

For the IRACE execution, the following value ranges were considered for the parameters: $iterMax \in \{1, 2, 3, 4, 5\}$ and $timesMax \in \{2, 3, 4, 5, 6\}$. IRACE returned $iterMax = 1$ for the ILS-RVND algorithm, and $iterMax = 4$ and $timesMax = 2$ for the E-ILS-RVND algorithm.

5.2.2. Results for instances of Group 2 data set

The results of the computational experiments performed with the instances of Group 2 are presented in Tables 10, 11, and 12, which show the values obtained and the runtime of each algorithm for the instances belonging to Sets I, II, and III, respectively. These values refer to a single execution of the algorithms. These tables also show the relative deviation between the objective function values of the solutions obtained by the algorithms, given by:

$$\Delta_{val} = 100 \times \frac{V_{E-ILS} - V_{ILS}}{V_{E-ILS}}, \tag{22}$$

where V_{E-ILS} and V_{ILS} are the objective function values for the best solution obtained by E-ILS-RVND and ILS-RVND, respectively. As previously done, the instances used in the parameter calibration were excluded.

In Table 10, we notice that the proposed algorithms presented a similar behavior pattern for all sizes of analyzed instances. The values obtained by them were relatively close for most cases, as can be seen by the small values of Δ_{val}. For some instances, solutions having the same objective function value were obtained. E-ILS-RVND obtained the best solutions for most instances. However, there were cases where ILS-RVND had a more satisfactory performance than E-ILS-RVND, generating negative deviation values (highlighted in bold). As expected, ILS-RVND recorded better computational times. Furthermore, the difference between the computational times spent by the algorithms becomes more expressive as the instance size increases.

The results of the tests performed with the instances of Sets II and III are shown, respectively, in Tables 11 and 12. Analogous remarks regarding the performance of the algorithms in solving the instances of Set I can be made for these two sets. The objective function values of the best solutions found by the algorithms for each instance are very close. The best solutions were achieved by E-ILS-RVND, except for those values corresponding to negative Δ_{val} (highlighted in bold). The shortest runtimes were obtained by ILS-RVND.

The experiments with the Group 2 data set, which include large-scale instances, show that the two proposed algorithms have a good computational performance and were able to generate solutions for all tested instances in reasonable computational times. The results obtained are in accordance with the statistical analysis performed previously, in which the E-ILS-RVND stood out, generating the best values for the objective function.

Finally, it is worth comparing the results obtained between the AP instances and the Group 2 data set. In the AP data set, several instances had coincident objective values (notably those with tight cost and some with α equal to 0.6 and 0.8). This particularity is not observed for instances of Group 2, which suggests that instances from this group are more challenging for PMHNDP than instances belonging to the AP data set. In fact, these data sets present significant differences regarding the amount of flow that is sent between the network nodes and the fixed costs of installation of the hubs.

6. Conclusions

This paper presented two heuristic algorithms based on local search and systematic exchanges of neighborhood structures to deal with the uncapacitated hub network design problem with an incomplete inter-hub network, focusing on profit maximization and also comparing the results with those presented in the literature so far. Both are ILS-based algorithms. The first, named ILS-RVND, is a classic version of the ILS metaheuristic that uses RVND as the local search method. The second, named E-ILS-RVND, is an enhanced version of ILS and performs greater intensification. Computational experiments were performed on two reference data sets from the hub location literature. An analysis of the neighborhood structures used in the proposed algorithms was also carried out.

The results of the computational experiments showed that the two proposed algorithms have a good performance and were able to solve instances with up to 500 nodes. For the set of instances that have optimal values indicated in the literature by an exact method (including instances with up to 150 nodes), it was found that both algorithms reached optimality for most cases in much shorter times. Notably, this difference in times is more expressive in the resolution of more complex instances and significantly increases with the number of nodes in the network.

Table 7
Results for AP instances with 125 and 150 nodes in one execution of heuristic algorithms.

(a) AP - 125

Instances	Benders		ILS-RVND			E-ILS-RVND		
	Optimal value	Time (s)	Value obtained	Δ(%)	Time (s)	Value obtained	Δ(%)	Time (s)
125L-0.2-20	19,751.55	26,482.87	19,659.82	0.46	113.05	19,751.55	0.00	735.44
125L-0.4-20	16,272.93	12,272.01	16,272.93	0.00	23.34	16,272.93	0.00	601.82
125L-0.6-20	15,028.67	8,180.95	15,028.67	0.00	20.20	15,028.67	0.00	77.65
125L-0.8-20	14,966.38	5,693.27	14,966.38	0.00	24.19	14,966.38	0.00	88.84
125T-0.2-20	7,169.77	8,600.38	7,169.77	0.00	12.52	7,169.77	0.00	17.97
125T-0.4-20	7,169.77	6,907.87	7,169.77	0.00	4.83	7,169.77	0.00	11.31
125T-0.6-20	7,169.77	4,863.38	7,169.77	0.00	5.02	7,169.77	0.00	14.77
125T-0.8-20	7,169.77	4,306.39	7,169.77	0.00	11.87	7,169.77	0.00	14.57
125L-0.2-30	54,503.08	53,161.89	54,503.08	0.00	289.63	54,503.08	0.00	1,078.72
125L-0.4-30	49,301.78	21,207.46	48,996.37	0.62	103.80	49,301.78	0.00	469.95
125L-0.6-30	45,861.48	14,901.48	45,606.25	0.56	114.13	45,828.04	0.07	234.34
125L-0.8-30	44,820.26	7,325.15	44,820.26	0.00	83.20	44,820.26	0.00	175.02
125T-0.2-30	29,586.11	10,352.58	29,586.11	0.00	4.56	29,586.11	0.00	17.85
125T-0.4-30	29,586.11	8,333.74	29,586.11	0.00	3.56	29,586.11	0.00	8.43
125T-0.6-30	29,586.11	6,955.91	29,586.11	0.00	3.92	29,586.11	0.00	16.01
125T-0.8-30	29,586.11	5,645.32	29,586.11	0.00	5.75	29,586.11	0.00	14.53
125L-0.2-50	–	mem	129,067.23	–	162.69	132,339.32	–	861.93
125L-0.4-50	126,103.54	28,117.28	124,072.92	1.61	246.25	125,672.24	0.34	381.05
125L-0.6-50	120,985.82	17,584.78	121,371.50	−0.32	404.60	121,250.65	−0.22	408.82
125L-0.8-50	119,696.69	10,761.85	119,696.69	0.00	61.21	119,696.69	0.00	376.70
125T-0.2-50	93,251.38	16,541.83	93,251.38	0.00	3.97	93,251.38	0.00	14.27
125T-0.4-50	93,251.38	11,164.57	93,251.38	0.00	3.60	93,251.38	0.00	21.08
125T-0.6-50	93,251.38	7,742.55	93,251.38	0.00	10.53	93,251.38	0.00	16.11
125T-0.8-50	93,251.38	6,421.03	93,251.38	0.00	3.01	93,251.38	0.00	12.47
Average	–	13,196.72	–	0.13	71.64	–	0.01	236.24

(b) AP - 150

Instances	Benders		ILS-RVND			E-ILS-RVND		
	Optimal value	Time (s)	Value obtained	Δ(%)	Time (s)	Value obtained	Δ(%)	Time (s)
150L-0.2-20	–	mem	18,739.29	–	285.22	18,973.30	–	729.78
150L-0.4-20	15,557.64	30,078.62	15,243.01	2.02	19.55	15,557.64	0.00	194.00
150L-0.6-20	14,446.68	29,523.28	14,446.68	0.00	29.05	14,446.68	0.00	138.36
150L-0.8-20	14,434.49	13,859.46	14,434.49	0.00	16.76	14,434.49	0.00	57.00
150T-0.2-20	10,913.35	33,412.04	10,913.35	0.00	16.62	10,913.35	0.00	54.92
150T-0.4-20	10,913.35	21,305.05	10,913.35	0.00	23.89	10,913.35	0.00	83.73
150T-0.6-20	10,913.35	16,679.75	10,913.35	0.00	15.34	10,913.35	0.00	71.19
150T-0.8-20	10,913.35	15,581.67	10,913.35	0.00	23.77	10,913.35	0.00	172.65
150L-0.2-30	–	mem	53,534.30	–	493.97	53,534.30	–	1,323.67
150L-0.4-30	–	mem	48,341.55	–	423.38	48,416.84	–	1,505.64
150L-0.6-30	–	43,851.14	45,095.07	–	182.24	45,095.07	–	901.81
150L-0.8-30	43,931.25	26,252.31	43,931.25	0.00	105.01	43,931.25	0.00	294.23
150T-0.2-30	–	64,565.21	38,270.84	–	36.77	38,270.84	–	118.20
150T-0.4-30	38,270.84	25,887.18	38,270.84	0.00	37.04	38,270.84	0.00	155.21
150T-0.6-30	38,270.84	17,370.94	38,270.84	0.00	42.16	38,270.84	0.00	270.40
150T-0.8-30	38,270.84	14,019.36	38,270.84	0.00	45.15	38,270.84	0.00	104.52
150L-0.2-50	–	mem	131,483.79	–	322.46	131,798.37	–	2,774.00
150L-0.4-50	–	mem	124,640.80	–	266.50	125,363.36	–	1,950.75
150L-0.6-50	–	mem	120,622.81	–	101.43	120,720.20	–	869.58
150L-0.8-50	118,924.32	29,908.91	118,924.32	0.00	102.62	118,924.32	0.00	314.40
150T-0.2-50	–	mem	112,647.50	–	46.21	112,647.50	–	261.60
150T-0.4-50	111,248.30	35,371.05	111,248.30	0.00	62.88	111,248.30	0.00	253.51
150T-0.6-50	111,248.30	25,300.32	111,248.30	0.00	96.22	111,248.30	0.00	293.66
150T-0.8-50	111,248.30	17,870.28	111,248.30	0.00	80.86	111,248.30	0.00	205.51
Average	–	27,108.03	–	0.13	119.80	–	0.00	545.76

Note. mem, 16 GB memory exceeded with exact method.

Table 8
Results for the AP instances with 175 and 200 nodes in one execution of heuristic algorithms.

(a) AP - 175

Instances	ILS-RVND		E-ILS-RVND	
	Value obtained	Time (s)	Value obtained	Time (s)
175L-0.2-20	19,167.00	259.14	19,210.26	1,276.93
175L-0.4-20	15,657.97	278.95	15,144.71	117.81
175L-0.6-20	14,551.17	12.91	14,551.17	44.10
175L-0.8-20	14,551.17	29.92	14,551.17	51.17
175T-0.2-20	14,054.94	92.60	14,054.94	341.65
175T-0.4-20	14,054.94	83.84	14,054.94	543.05
175T-0.6-20	14,054.94	89.37	14,054.94	263.83
175T-0.8-20	14,054.94	159.46	14,054.94	238.73
175L-0.2-30	53,748.27	381.50	53,589.90	1,250.67
175L-0.4-30	48,493.12	372.78	48,493.12	698.45
175L-0.6-30	45,583.63	113.52	45,583.63	1,088.18
175L-0.8-30	44,655.02	412.45	44,655.02	952.29
175T-0.2-30	42,592.04	271.12	42,592.04	259.47
175T-0.4-30	42,592.04	91.71	42,592.04	186.74
175T-0.6-30	42,592.04	86.88	42,592.04	221.34
175T-0.8-30	42,592.04	137.67	42,592.04	207.10
175L-0.2-50	130,316.46	794.28	131,500.97	4,964.69
175L-0.4-50	124,929.44	555.34	125,049.72	1,806.91
175L-0.6-50	119,295.45	351.52	121,181.71	2,158.86
175L-0.8-50	119,621.18	218.36	119,621.18	804.21
175T-0.2-50	117,120.33	307.13	117,120.33	387.83
175T-0.4-50	116,629.93	274.11	116,629.93	665.10
175T-0.6-50	116,629.93	288.14	116,629.93	762.75
175T-0.8-50	116,629.93	261.31	116,629.93	735.57
Average	–	246.83	–	834.48

(b) AP - 200

Instances	ILS-RVND		E-ILS-RVND	
	Value obtained	Time (s)	Value obtained	Time (s)
200L-0.2-20	17,821.16	623.40	17,821.16	969.42
200L-0.4-20	14,046.07	44.28	14,046.07	110.22
200L-0.6-20	13,529.70	37.66	13,529.70	86.63
200L-0.8-20	13,529.70	13.53	13,529.70	45.33
200T-0.2-20	7,392.64	39.48	7,392.64	167.61
200T-0.4-20	7,392.64	35.58	7,392.64	250.60
200T-0.6-20	7,392.64	39.36	7,392.64	209.99
200T-0.8-20	7,392.64	37.10	7,392.64	257.03
200L-0.2-30	52,407.80	1,521.70	51,673.85	2,169.97
200L-0.4-30	47,196.18	384.77	47,342.11	3,458.44
200L-0.6-30	43,723.12	308.19	44,199.47	2,211.10
200L-0.8-30	43,210.86	327.22	43,210.86	586.86
200T-0.2-30	35,317.26	435.97	35,317.26	949.72
200T-0.4-30	35,317.26	300.23	35,317.26	810.99
200T-0.6-30	35,317.26	246.19	35,317.26	559.84
200T-0.8-30	35,317.26	267.67	35,317.26	694.00
200L-0.2-50	128,770.70	565.30	130,289.39	7,211.59
200L-0.4-50	123,258.31	993.45	123,693.40	4,064.11
200L-0.6-50	119,353.97	156.95	119,353.97	921.04
200L-0.8-50	118,105.97	364.34	118,105.97	1,141.47
200T-0.2-50	109,833.88	442.85	109,833.88	890.81
200T-0.4-50	109,676.37	442.92	109,676.37	935.93
200T-0.6-50	109,676.37	357.82	109,676.37	1,545.97
200T-0.8-50	109,676.37	904.77	109,676.37	1,449.05
Average	–	370.45	–	1,320.74

Table 9
Set of instances from Group 2 data set submitted to IRACE.

Instances	α	Revenue
50-I	0.2	2
50-I	0.4	3
50-I	0.8	5
100-I	0.2	2
100-I	0.6	3
100-I	0.4	5

Considering the parameters indicated in calibration with IRACE, E-ILS-RVND generally obtained better solutions than ILS-RVND, as indicated by the presented statistical analysis. In addition, as showed in Fig. 6, it proved to be faster in the convergence tests. On the other hand, ILS-RVND achieved the shortest computational times.

An additional set of experiments was performed to run the ILS-RVND algorithm with the same number of iterations indicated by IRACE for the E-ILS-RVND algorithm. According to the results of this experiment, presented in Appendix A, there was an improvement in the average quality of the obtained solutions provided by this algorithm, despite increasing the runtime. However, as the PMHNDP is a strategic planning problem, the most significant criterion for selecting a solution procedure is to obtain good-quality solutions. Therefore, considering that, in a real application, a greater number of iterations can be performed to improve the quality of the solutions, we concluded that both algorithms are equally promising for solving instances of the PMHNDP that cannot be solved via exact methods. The analysis of the neighborhood structures demonstrated that the use together of the six neighborhoods, as implemented in the proposed algorithms, allowed them to obtain better results.

In future works, a heuristic hybrid algorithm can be developed, combining one of the heuristic algorithms with an exact method. Another promising research direction is to incorporate machine learning techniques with the proposed algorithms to make them more efficient.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES), the Minas Gerais State Research Foundation (FAPEMIG), the National Council of Technological and Scientific Development (CNPq), as well as the Federal Center of Technological Education of Minas Gerais (CEFET-MG), the Federal Institute Catarinense (IFC) and the Federal University of Ouro Preto (UFOP) for supporting the development of the present study.

Appendix A. Additional tests with ILS-RVND

In order to complement the analysis between the proposed algorithms, we performed additional experiments with the ILS-RVND algorithm, considering the maximum number of iterations equal to 4 ($iterMax = 4$), which is the same value used for the algorithm E-ILS-RVND (indicated in parameter calibration with IRACE). The results for 30 runs are shown in Tables A.13 and A.14.

According to the two tables, we can conclude that by increasing the maximum number of iterations of the ILS-RVND, it presented values for the mean gaps lower than those obtained by the E-ILS-RVND algorithm. On the other hand, the times spent by E-ILS-RVND to solve the instances were, in general, better than the times recorded by ILS-RVND.

Appendix B. Supplementary tests allowing direct connections between non-hub nodes

In this section, we present the results for supplementary computational experiments, considering the PMHNDP version in which direct connections between non-hub nodes are allowed. To present a formulation for the problem in this case, consider the flow variables $e_{ij} \geq 0$,

Table 10
Results for instances of Set I with one execution of the heuristic algorithms.

(a) 50 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
50I-0.4-2	77,217.69	12.50	77,732.47	26.11	0.66
50I-0.6-2	75,114.36	9.44	75,309.05	35.56	0.26
50I-0.8-2	74,523.89	2.44	74,523.89	7.94	0.00
50I-0.2-3	137,623.85	3.62	138,232.96	91.80	0.44
50I-0.6-3	131,793.36	9.23	131,690.02	10.08	**−0.08**
50I-0.8-3	131,008.19	2.02	131,128.35	29.06	0.09
50I-0.2-5	250,906.19	30.14	251,181.08	48.05	0.11
50I-0.4-5	246,669.11	10.23	247,185.38	28.90	0.21
50I-0.6-5	244,292.56	7.43	244,761.96	43.12	0.19
Average	–	9.67	–	35.62	0.21

(b) 100 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
100I-0.4-2	790,670.17	96.63	802,917.07	217.16	1.53
100I-0.6-2	763,608.46	23.26	767,157.51	54.24	0.46
100I-0.8-2	745,225.66	26.45	744,299.04	33.10	**−0.12**
100I-0.2-3	1,423,583.81	148.74	1,423,583.81	255.96	0.00
100I-0.4-3	1,373,862.17	59.13	1,377,298.93	581.56	0.25
100I-0.8-3	1,318,680.89	48.65	1,321,552.28	114.17	0.22
100I-0.2-5	2,572,347.52	66.58	2,572,347.52	324.51	0.00
100I-0.6-5	2,490,303.08	98.75	2,490,303.08	130.72	0.00
100I-0.8-5	2,467,444.60	11.91	2,468,371.23	78.70	0.04
Average	–	64.46	–	198.90	0.26

(c) 200 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
200I-0.2-2	4,631,395.48	1,529.50	4,666,588.74	5,018.50	0.75
200I-0.4-2	4,421,528.73	707.08	4,422,849.27	4,981.46	0.03
200I-0.6-2	4,298,026.62	821.67	4,298,026.62	2,072.66	0.00
200I-0.8-2	4,243,732.28	272.03	4,243,732.28	714.44	0.00
200I-0.2-3	7,720,479.41	1,149.81	7,751,553.83	9,681.17	0.40
200I-0.4-3	7,515,014.17	830.14	7,536,319.93	5,664.98	0.28
200I-0.6-3	7,409,976.78	1,520.70	7,399,117.69	1,229.63	**−0.15**
200I-0.8-3	7,355,682.44	591.09	7,355,682.44	927.59	0.00
200I-0.2-5	13,909,304.25	939.88	13,999,838.74	5,683.88	0.65
200I-0.4-5	13,746,009.64	1,705.43	13,752,003.98	7,757.97	0.04
200I-0.6-5	13,622,747.37	597.73	13,633,877.10	2,492.49	0.08
200I-0.8-5	13,579,582.75	256.31	13,579,582.75	1,203.51	0.00
Average	–	910.11	–	3,952.36	0.17

(d) 300 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
300I-0.2-2	3,526,706.83	1,509.78	3,524,133.20	8,780.41	**−0.07**
300I-0.4-2	3,353,065.27	960.52	3,377,016.49	16,822.64	0.71
300I-0.6-2	3,272,038.48	3,787.93	3,277,996.23	8,675.06	0.18
300I-0.8-2	3,241,783.16	596.07	3,241,783.16	2,786.03	0.00
300I-0.2-3	5,985,630.44	6,375.69	6,007,718.34	25,574.22	0.37
300I-0.4-3	5,829,035.93	1,710.54	5,855,920.97	23,611.59	0.46
300I-0.6-3	5,731,890.56	2,727.01	5,735,910.27	11,872.46	0.07
300I-0.8-3	5,710,595.42	712.84	5,710,595.42	10,149.97	0.00
300I-0.2-5	10,933,143.60	6,475.64	10,936,100.20	15,935.69	0.03
300I-0.4-5	10,778,553.38	2,060.72	10,771,288.78	17,978.79	**−0.07**
300I-0.6-5	10,668,308.95	1,207.74	10,677,969.98	7,973.00	0.09
300I-0.8-5	10,643,731.41	4,286.06	10,648,219.93	5,029.38	0.04
Average	–	2,700.88	–	12,932.44	0.15

(e) 400 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
400I-0.2-2	10,989,150.28	50,772.17	11,086,081.53	88,276.80	0.87
400I-0.4-2	10,393,134.66	24,761.86	10,452,267.96	34,680.14	0.57
400I-0.6-2	10,135,089.81	23,951.83	10,142,039.10	15,850.21	0.07
400I-0.8-2	10,026,350.29	4,844.62	10,026,350.29	8,737.47	0.00
400I-0.2-3	18,502,620.72	15,377.89	18,469,380.75	50,390.81	**−0.18**
400I-0.4-3	17,838,839.26	2,116.92	17,983,702.96	79,472.15	0.81
400I-0.6-3	17,614,319.97	19,848.56	17,635,370.90	29,278.07	0.12
400I-0.8-3	17,550,611.66	2,903.55	17,529,294.36	10,064.29	**−0.12**
400I-0.2-5	33,515,907.53	16,914.60	33,584,583.49	177,312.48	0.20
400I-0.4-5	32,973,773.25	17,349.76	32,994,369.21	30,014.38	0.06
400I-0.6-5	32,608,910.67	9,712.87	32,650,871.30	26,434.39	0.13
400I-0.8-5	32,556,499.79	4,766.40	32,556,499.79	9,751.93	0.00
Average	–	16,110.09	–	46,688.59	0.21

(f) 500 nodes — Set I

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
500I-0.2-2	18,188,071.64	13,011.14	18,272,245.67	34,574.73	0.46
500I-0.4-2	17,462,617.99	32,533.31	17,449,106.51	32,974.28	**−0.08**
500I-0.6-2	16,867,449.69	14,447.20	16,890,954.22	37,708.11	0.14
500I-0.8-2	16,714,075.60	4,167.47	16,745,516.47	22,799.63	0.19
500I-0.2-3	30,587,171.24	25,850.80	30,654,613.37	71,533.96	0.22
500I-0.4-3	29,836,179.86	36,600.85	29,804,132.93	50,084.30	**−0.11**
500I-0.6-3	29,253,593.32	11,410.05	29,273,321.92	46,675.58	0.07
500I-0.8-3	29,127,884.17	7,334.69	29,127,884.17	19,586.55	0.00
500I-0.2-5	55,419,348.77	3,564.65	55,505,871.34	122,510.53	0.16
500I-0.4-5	54,596,209.60	26,141.31	54,596,209.60	60,603.05	0.00
500I-0.6-5	54,014,552.79	19,699.53	54,058,234.26	110,677.72	0.08
500I-0.8-5	53,892,619.56	10,684.33	53,892,619.56	18,532.87	0.00
Average	–	17,120.44	–	52,355.11	0.09

which represent the fraction of the demand that is routed through the direct connection between non-hub nodes with origin $i \in N$ and destination $j \in N$. A formulation for this version of the problem is given by

$$\max \sum_{i\in N}\sum_{j\in N} w_{ij}\left[\sum_{k\in N}(r_{ij}-c_{ik})a_{ijk} + (r_{ij}-c_{ij})e_{ij} - \sum_{m\in N} c_{mj} b_{ijm} - \sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha c_{km} x_{ijkm}\right] - \sum_{k\in N} s_k h_k - \sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} g_{km} z_{km} \tag{B.1}$$

$$\text{s. t.} \quad (4)\text{–}(14) \tag{B.2}$$

$$\sum_{k\in N} a_{ijk} + e_{ij} \leq 1 \qquad \forall\, i,j \in N \tag{B.3}$$

$$\sum_{m\in N} b_{ijm} + e_{ij} \leq 1 \qquad \forall\, i,j \in N \tag{B.4}$$

$$e_{ij} + h_i \leq 1 \qquad \forall\, i,j \in N \tag{B.5}$$

$$e_{ij} + h_j \leq 1 \qquad \forall\, i,j \in N \tag{B.6}$$

$$e_{ij} \geq 0 \qquad \forall\, i,j \in N. \tag{B.7}$$

The objective function (B.1) minimizes the profit of the hub network, given by the difference between the revenue (obtained from

Table 11
Results for instances of Set II with one execution of the heuristic algorithms.

(a) 50 nodes — Set II

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
50II-0.2-2	1,078,493.97	15.90	1,096,305.43	142.03	1.62
50II-0.4-2	1,044,987.43	10.18	1,043,044.74	21.37	**−0.19**
50II-0.6-2	1,010,804.75	22.66	1,011,266.51	44.23	0.05
50II-0.8-2	997,795.36	9.58	997,795.36	19.40	0.00
50II-0.2-3	1,820,504.31	125.74	1,823,909.08	210.25	0.19
50II-0.4-3	1,769,953.82	35.24	1,766,775.85	101.03	**−0.18**
50II-0.6-3	1,734,535.86	27.18	1,734,997.62	35.12	0.03
50II-0.8-3	1,721,526.47	6.98	1,721,526.47	12.68	0.00
50II-0.2-5	3,270,816.56	77.16	3,267,966.53	93.45	**−0.09**
50II-0.4-5	3,216,180.76	14.38	3,217,416.05	67.02	0.04
50II-0.6-5	3,178,256.69	8.13	3,181,998.08	50.90	0.12
50II-0.8-5	3,168,988.70	6.14	3,168,988.70	12.65	0.00
Average	–	29.94	–	67.51	0.13

(b) 100 nodes — Set II

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
100II-0.2-2	6,032,852.76	276.26	6,062,719.60	1,458.89	0.49
100II-0.4-2	5,700,582.26	287.03	5,731,602.73	531.37	0.54
100II-0.6-2	5,513,735.77	143.62	5,503,613.22	288.17	**−0.18**
100II-0.8-2	5,410,472.43	41.46	5,410,472.43	133.43	0.00
100II-0.2-3	10,115,922.05	419.81	10,153,544.95	471.66	0.37
100II-0.4-3	9,790,236.13	192.09	9,857,119.23	671.12	0.68
100II-0.6-3	9,589,165.47	43.62	9,618,581.28	464.94	0.31
100II-0.8-3	9,520,971.77	52.79	9,520,971.77	111.78	0.00
100II-0.2-5	18,378,814.61	164.11	18,404,782.60	1,011.73	0.14
100II-0.4-5	18,031,300.81	82.64	18,051,863.15	223.52	0.11
100II-0.6-5	17,821,888.65	139.92	17,835,111.24	582.99	0.07
100II-0.8-5	17,722,352.97	97.61	17,741,970.45	268.55	0.11
Average	–	161.75	–	518.18	0.22

(c) 150 nodes — Set II

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
150II-0.2-2	12,993,821.36	438.22	13,143,130.27	1,106.36	1.14
150II-0.4-2	12,395,244.13	260.83	12,512,091.88	1,307.95	0.93
150II-0.6-2	11,959,400.21	266.65	11,959,400.21	147.94	0.00
150II-0.8-2	11,747,054.03	110.24	11,766,006.72	163.22	0.16
150II-0.2-3	22,039,402.89	755.68	22,157,343.46	2,837.30	0.53
150II-0.4-3	21,519,169.05	1,543.64	21,511,651.27	1,618.10	**−0.03**
150II-0.6-3	20,931,032.34	164.82	21,049,480.90	1,087.10	0.56
150II-0.8-3	20,833,281.45	402.06	20,833,281.45	626.38	0.00
150II-0.2-5	40,122,708.87	477.42	40,228,677.37	2,947.76	0.26
150II-0.4-5	39,383,521.04	216.89	39,567,597.54	1,102.17	0.47
150II-0.6-5	39,032,042.94	177.78	39,097,909.39	751.96	0.17
150II-0.8-5	38,819,696.76	113.87	38,881,709.94	903.16	0.16
Average	–	410.68	–	1,216.62	0.36

(d) 200 nodes — Set II

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
200II-0.2-2	27,259,702.02	703.09	27,482,530.69	7,053.16	0.81
200II-0.4-2	26,215,802.33	1,717.40	26,234,968.77	5,831.13	0.07
200II-0.6-2	25,408,802.54	623.38	25,543,771.95	2,102.02	0.53
200II-0.8-2	25,335,140.22	831.89	25,335,140.22	1,427.59	0.00
200II-0.2-3	46,052,260.84	1,177.19	46,087,937.66	1,603.46	0.08
200II-0.4-3	44,874,854.65	635.49	45,028,459.52	14,027.97	0.34
200II-0.6-3	44,260,313.73	871.73	44,215,871.55	4,852.86	**−0.10**
200II-0.8-3	44,094,798.10	540.85	44,094,798.10	1,434.80	0.00
200II-0.2-5	83,639,643.44	4,908.87	83,738,733.15	3,459.27	0.12
200II-0.4-5	82,365,303.87	1,647.57	82,542,766.58	4,133.30	0.21
200II-0.6-5	81,785,397.85	2,101.31	81,730,036.45	2,804.91	**−0.07**
200II-0.8-5	81,610,893.36	464.84	81,614,113.85	881.48	0.00
Average	–	410.68	–	4,134.33	0.17

pairs served through the hubs and also from those served through direct connections) and the costs associated with the network design. Constraints (B.3) and (B.4) ensure that the demand flow between a given origin and destination pair will be sent through hubs or direct connections. The set of constraints (B.5) and (B.6) guarantee that the demand between a pair of nodes will be served through a direct connection only when these nodes are not hubs installed in the network. Finally, the constraints (B.7) represent the domain of the variables e_{ij}.

The computational experiments were performed with the E-ILS-RVND algorithm, applied to solve AP instances of 25 nodes. The algorithm uses the same parameters presented in Section 5.1.2. The evaluation of the objective function requires the profit obtained by the pairs of nodes served through direct connections. Hence, the value of the minimum transportation cost to route a flow unit between the pair (i, j) (Eq. (17)) was replaced by

$$C_{ij} = \min \left\{ c_{ij}, \min_{k,m \in H} \left\{ c_{ik} + C^{FW}_{km} + c_{mj} \right\} \right\}. \tag{B.8}$$

Table B.15 presents the results of this set of experiments. The first part of the table shows the results with CPLEX. The column "Total pairs served' shows, in percentage, the total amount of pairs of nodes that were served, while the column "Pairs served - DC' displays the percentage of pairs that were served through direct connections. Then, the nodes selected to be hubs and the runtime, in seconds, are informed. The second part of the table presents the results for 30 executions of the E-ILS-RVND.

The results from Table B.15 illustrate some impacts in solving the problem by allowing direct connections between non-hub nodes. Note that for instances with cost Tight and revenue equal to 20, no hub is installed in the network, and consequently demand from serviced pairs is sent exclusively through direct connections. For these cases, the trade-off between cost and revenue showed that it is more profitable to consider a network without hubs and hub arcs. Also, note that for these instances the lowest percentage of served pairs was obtained (30.88%).

As expected, for each type of fixed cost for installing hubs and discount factor in the hub arcs, the increase in revenue causes an increase in the number of pairs served and also in the value of the objective function. Note that even with the higher values for revenue (30 and 50), the number of pairs served through direct connections is much lower than the number of pairs served with hubs. For example, for the instance 25L-0.2-50, we have 100% of node pairs served, with only 11% for direct connections.

Regarding the performance of the E-ILS-RVND algorithm applied to the problem with direct connections, Table B.15 shows that it obtained a good performance for most instances, except for those instances in which no hub is installed on the network. This is because the implemented algorithm admits solutions that necessarily have at least one hub installed in the network. During the execution of the algorithm, if all hubs are removed from the network, a node is randomly chosen and installed as a hub. This mechanism was adopted considering the problem originally discussed in this study, where direct connections were not allowed. In addition, this mechanism contributes to the success of

Table 12
Results for instances of Set III with one execution of the heuristic algorithms.

(a) 50 nodes — Set III

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
50III-0.2-2	206,070.12	54.53	204,814.51	108.80	**−0.61**
50III-0.4-2	195,110.04	24.23	194,643.67	30.84	**−0.24**
50III-0.6-2	186,661.14	3.03	187,467.18	25.07	0.43
50III-0.8-2	185,035.52	4.65	185,237.56	28.02	0.11
50III-0.2-3	344,245.94	33.31	344,245.94	125.87	0.00
50III-0.4-3	333,285.86	32.42	332,290.02	31.02	**−0.30**
50III-0.6-3	324,882.00	6.16	325,653.38	57.37	0.24
50III-0.8-3	323,413.39	7.94	323,413.39	34.36	0.00
50III-0.2-5	620,597.59	22.33	620,597.59	90.41	0.00
50III-0.4-5	608,641.66	6.02	609,637.50	113.92	0.16
50III-0.6-5	601,994.65	4.97	602,025.73	119.05	0.01
50III-0.8-5	599,765.03	8.29	599,562.99	16.90	**−0.03**
Average	–	17.32	–	65.14	−0.02

(b) 100 nodes — Set III

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
100III-0.2-2	821,985.68	190.02	821,985.68	764.54	0.00
100III-0.4-2	766,259.28	40.87	778,776.32	853.59	1.61
100III-0.6-2	744,751.66	102.20	744,751.66	398.01	0.00
100III-0.8-2	736,015.82	42.63	737,607.81	85.90	0.22
100III-0.2-3	1,384,128.42	169.92	1,384,128.42	344.65	0.00
100III-0.4-3	1,339,705.40	255.55	1,340,919.06	2,097.45	0.09
100III-0.6-3	1,308,783.64	98.69	1,310,031.95	430.14	0.10
100III-0.8-3	1,300,454.66	237.65	1,300,454.66	191.14	0.00
100III-0.2-5	2,501,885.36	230.74	2,508,413.88	380.90	0.26
100III-0.4-5	2,463,321.44	170.61	2,465,204.52	592.98	0.08
100III-0.6-5	2,428,706.43	96.73	2,433,069.11	432.40	0.18
100III-0.8-5	2,424,036.00	43.10	2,422,444.01	99.13	**−0.07**
Average	–	139.89	–	555.90	0.20

(c) 150 nodes — Set III

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
150III-0.2-2	1,816,108.28	198.68	1,837,952.93	1,435.73	1.19
150III-0.4-2	1,742,566.28	199.00	1,741,313.53	802.50	**−0.07**
150III-0.6-2	1,691,635.18	406.03	1,688,475.83	496.41	**−0.19**
150III-0.8-2	1,672,855.05	165.68	1,672,855.05	386.76	0.00
150III-0.2-3	3,105,575.58	191.46	3,104,569.82	5,193.47	**−0.03**
150III-0.4-3	3,009,834.15	492.85	3,005,651.07	1,005.43	**−0.14**
150III-0.6-3	2,938,302.93	695.78	2,951,480.04	1,194.76	0.45
150III-0.8-3	2,935,859.26	157.95	2,935,859.26	980.58	0.00
150III-0.2-5	5,605,120.90	186.33	5,635,617.46	4,395.90	0.54
150III-0.4-5	5,504,566.90	150.13	5,537,871.05	3,130.22	0.60
150III-0.6-5	5,469,178.02	142.88	5,476,281.28	2,127.86	0.13
150III-0.8-5	5,461,867.67	355.42	5,461,867.67	512.64	0.00
Average	–	278.52	–	1,805.19	0.21

(d) 200 nodes — Set III

Instances	ILS-RVND		E-ILS-RVND		Δ_{val}
	Value obtained	Time (s)	Value obtained	Time (s)	
200III-0.2-2	3,287,020.50	3,891.22	3,296,392.35	5,347.91	0.28
200III-0.4-2	3,115,130.24	1,388.61	3,128,868.02	4,515.90	0.44
200III-0.6-2	3,027,751.46	2,238.30	3,038,468.03	3,228.81	0.35
200III-0.8-2	3,015,236.04	677.83	3,015,236.04	1,445.20	0.00
200III-0.2-3	5,528,795.05	3,428.22	5,559,956.49	10,774.30	0.56
200III-0.4-3	5,356,800.18	548.25	5,375,750.42	2,331.35	0.35
200III-0.6-3	5,275,152.19	286.75	5,276,216.22	3,563.94	0.02
200III-0.8-3	5,271,114.82	1,306.36	5,271,114.82	674.14	0.00
200III-0.2-5	10,027,239.89	2,960.32	10,036,025.67	5,887.79	0.09
200III-0.4-5	9,898,139.41	2,384.27	9,896,954.40	29,780.17	**−0.01**
200III-0.6-5	9,794,379.70	2,065.62	9,806,104.38	3,180.40	0.12
200III-0.8-5	9,782,872.38	376.98	9,782,872.38	893.87	0.00
Average	–	1,796.06	–	5,968.65	0.18

the perturbation procedure, since it replaces an installed hub with a non-installed one. Finally, it is noted that the runtimes of the proposed algorithm are significantly smaller than the runtimes of CPLEX.

Appendix C. Comparison with the literature

We present in this section a comparison between the E-ILS-RVND algorithm and the results indicated by Zhang et al. (2023), obtained with a VNS algorithm with CAB data set. It is important to highlight that this comparison has some limitations since Zhang et al. (2023) ran the experiments on a machine with a different configuration from the machine used in this work and they implemented the algorithm using another programming language (python). Furthermore, Zhang et al. (2023) ran computational experiments using only instances from the CAB data set, which are smaller-sized instances compared to the set of instances from AP data set and the set of instances proposed by Contreras et al. (2011a).

The tests were performed with the CAB data set with 25 nodes. The parameter settings were the same parameters used by Zhang et al. (2023). For the revenue, three values were considered $r_{ij} \in \{1,000; 1,500, 2,000\}$, respectively referred to as low revenue, medium revenue, and high revenue. Hub installation costs were assumed to be $s_k \in \{50; 100; 150\}$, also referred to as low (L), medium (M), and high (H) costs, respectively. The cost for installing hub arcs was taken to be $0.10 \cdot s_k$, for each type of cost considered. The values of α were $0.2, 0.4, 0.6$, and 0.8.

Since this study and the study developed by Zhang et al. (2023) used different machines, we adjusted the runtime presented by Zhang et al. (2023) using the PassmarMark software, available at https://www.cpubenchmark.net/singleCompare.php, which provides the CPU Mark of each machine. The adjusted time is obtained by multiplying the runtime presented by Zhang et al. (2023) by an adjustment factor given by the ratio between the CPU Mark value of the computer that they used by the CPU Mark value of the computer used in our computational experiments. We found an adjustment factor of 0.896.

Table C.16 shows the results indicated by Zhang et al. (2023), including the adjusted runtimes, and those obtained with E-ILS-RVND, referring to 20 runs. In Zhang et al. (2023), instead of presenting the mean values, as was done in other sections of this article, the authors present the results of the median values for the gaps. This table presents the median gap (Δ_{med}), the average runtime, and the adjusted average runtime for the VNS of Zhang et al. (2023). For the E-ILS-RVND, this table presents the gap associated with the best value found (Δ_{best}), the median gap (Δ_{med}), and the average runtime. The values in this table show that, although the median gaps of the E-ILS-RVND are not all zero, as obtained by the VNS, it reached the optimal value in all cases. In addition, the runtimes recorded by the E-ILS-RVND were notably lower compared to the other method.

Table A.13
Results for AP instances with 40 and 50 nodes (30 runs).

Instances	Optimal value	ILS-RVND ($iterMax = 4$)					E-ILS-RVND				
		Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)
40L-0.4-20	14,635.61	14,635.61	14,573.33	0.00	0.43	2.79	14,635.61	14,593.00	0.00	0.29	2.65
40L-0.6-20	14,099.19	14,099.19	14,099.19	0.00	0.00	0.67	14,099.19	14,099.19	0.00	0.00	0.63
40L-0.8-20	14,099.19	14,099.19	14,099.19	0.00	0.00	0.57	14,099.19	14,099.19	0.00	0.00	0.59
40T-0.2-20	6,404.58	6,404.58	6,404.58	0.00	0.00	0.65	6,404.58	6,404.58	0.00	0.00	0.60
40T-0.6-20	6,404.58	6,404.58	6,404.58	0.00	0.00	0.49	6,404.58	6,404.58	0.00	0.00	0.53
40T-0.8-20	6,404.58	6,404.58	6,404.58	0.00	0.00	0.49	6,404.58	6,404.58	0.00	0.00	0.46
40L-0.2-30	52,299.43	52,299.43	52,134.67	0.00	0.32	20.03	52,299.43	52,021.38	0.00	0.53	14.45
40L-0.6-30	44,213.55	44,213.55	44,213.55	0.00	0.00	7.84	44,213.55	44,213.55	0.00	0.00	6.29
40L-0.8-30	42,990.07	42,990.07	42,990.07	0.00	0.00	6.58	42,990.07	42,981.96	0.00	0.02	4.26
40T-0.2-30	36,135.48	36,135.48	36,135.48	0.00	0.00	3.14	36,135.48	36,135.48	0.00	0.00	2.54
40T-0.4-30	33,918.82	33,918.82	33,918.82	0.00	0.00	2.67	33,918.82	33,918.82	0.00	0.00	2.00
40T-0.6-30	32,709.49	32,709.49	32,709.49	0.00	0.00	1.91	32,709.49	32,709.49	0.00	0.00	1.74
40L-0.2-50	130,906.36	130,906.36	129,817.67	0.00	0.83	42.16	130,747.04	129,880.77	0.12	0.78	37.55
40L-0.4-50	124,056.22	124,056.22	123,896.66	0.00	0.13	25.86	124,056.22	123,955.55	0.00	0.08	19.66
40L-0.6-50	120,056.17	120,056.17	119,997.31	0.00	0.05	20.13	120,056.17	119,900.24	0.00	0.13	12.76
40T-0.4-50	107,847.14	107,847.14	107,847.14	0.00	0.00	3.36	107,847.14	107,845.35	0.00	0.00	3.60
40T-0.6-50	105,857.00	105,857.00	105,857.00	0.00	0.00	2.48	105,857.00	105,857.00	0.00	0.00	2.05
40T-0.8-50	105,603.50	105,603.50	105,603.50	0.00	0.00	2.48	105,603.50	105,603.50	0.00	0.00	1.74
50L-0.2-20	17,666.95	17,366.52	17,366.52	1.70	1.70	10.36	17,666.95	17,425.21	0.00	1.37	13.67
50L-0.4-20	14,588.82	14,588.82	14,588.82	0.00	0.00	4.26	14,588.82	14,588.82	0.00	0.00	3.47
50L-0.8-20	13,920.55	13,920.55	13,920.55	0.00	0.00	1.16	13,920.55	13,920.55	0.00	0.00	1.12
50T-0.4-20	8,001.47	8,001.47	8,001.47	0.00	0.00	1.07	8,001.47	8,001.47	0.00	0.00	0.93
50T-0.6-20	8,001.47	8,001.47	8,001.47	0.00	0.00	0.95	8,001.47	8,001.47	0.00	0.00	0.93
50T-0.8-20	8,001.47	8,001.47	8,001.47	0.00	0.00	1.03	8,001.47	8,001.47	0.00	0.00	0.92
50L-0.2-30	53,200.19	53,200.19	52,731.78	0.00	0.88	69.71	53,200.19	52,854.48	0.00	0.65	62.84
50L-0.6-30	43,763.67	43,763.67	43,700.59	0.00	0.14	47.56	43,763.67	43,577.47	0.00	0.43	24.74
50L-0.8-30	42,566.95	42,566.95	42,566.95	0.00	0.00	22.06	42,566.95	42,566.95	0.00	0.00	15.28
50T-0.2-30	34,513.57	34,513.57	34,080.34	0.00	1.26	9.60	34,513.57	34,162.25	0.00	1.02	7.07
50T-0.4-30	31,911.87	31,911.87	31,842.74	0.00	0.22	5.18	31,911.87	31,750.56	0.00	0.51	4.57
50T-0.8-30	31,220.52	31,220.52	31,220.52	0.00	0.00	3.93	31,220.52	31,220.52	0.00	0.00	3.61
50L-0.4-50	124,770.82	124,770.82	124,520.43	0.00	0.20	81.15	124,770.82	124,267.84	0.00	0.40	61.06
50L-0.6-50	119,972.77	119,972.77	119,829.88	0.00	0.12	83.82	119,972.77	119,707.89	0.00	0.22	53.70
50L-0.8-50	118,299.45	118,299.45	118,299.45	0.00	0.00	29.41	118,299.45	118,287.27	0.00	0.01	21.00
50T-0.2-50	110,962.98	110,858.78	110,580.05	0.09	0.35	28.39	110,962.98	110,226.98	0.00	0.66	17.47
50T-0.4-50	106,451.33	106,451.33	106,439.17	0.00	0.01	7.94	106,451.33	106,439.17	0.00	0.01	6.04
50T-0.6-50	105,260.70	105,260.70	105,011.55	0.00	0.24	7.53	105,260.70	104,933.70	0.00	0.31	5.25
Average	–	–	–	0.05	0.19	15.54	–	–	0.00	0.21	11.60

Table A.14
Results for AP instances with 75 and 100 nodes (30 runs).

Instances	Optimal value	ILS-RVND ($iterMax = 4$)					E-ILS-RVND				
		Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)
75L-0.4-20	15,004.99	15,004.99	14,948.20	0.00	0.38	26.17	15,004.99	14,930.70	0.00	0.50	21.24
75L-0.6-20	14,058.46	14,058.46	14,058.46	0.00	0.00	4.07	14,058.46	14,058.46	0.00	0.00	4.41
75L-0.8-20	14,058.46	14,058.46	14,058.46	0.00	0.00	4.13	14,058.46	14,058.46	0.00	0.00	4.17
75T-0.2-20	1,144.13	1,144.13	1,144.13	0.00	0.00	3.26	1,144.13	1,144.13	0.00	0.00	2.91
75T-0.4-20	1,144.13	1,144.13	1,144.13	0.00	0.00	2.86	1,144.13	1,144.13	0.00	0.00	2.91
75T-0.6-20	1,144.13	1,144.13	1,144.13	0.00	0.00	3.01	1,144.13	1,144.13	0.00	0.00	2.84
75L-0.2-30	53,570.37	53,570.37	53,289.72	0.00	0.52	178.32	53,570.37	53,283.50	0.00	0.54	115.87
75L-0.4-30	48,097.70	48,097.70	48,024.68	0.00	0.15	188.45	48,097.70	47,708.07	0.00	0.81	112.47
75L-0.8-30	43,645.17	43,645.17	43,645.17	0.00	0.00	80.36	43,645.17	43,645.17	0.00	0.00	51.52
75T-0.2-30	25,999.62	25,999.62	25,999.62	0.00	0.00	10.76	25,999.62	25,999.62	0.00	0.00	8.91
75T-0.6-30	25,999.62	25,999.62	25,999.62	0.00	0.00	11.30	25,999.62	25,999.62	0.00	0.00	8.00
75T-0.8-30	25,999.62	25,999.62	25,999.62	0.00	0.00	10.96	25,999.62	25,999.62	0.00	0.00	8.27
75L-0.2-50	131,831.98	131,831.98	131,015.40	0.00	0.62	280.48	131,831.98	130,771.82	0.00	0.80	193.11
75L-0.4-50	125,133.91	125,133.91	124,975.94	0.00	0.13	246.75	125,133.91	124,852.08	0.00	0.23	194.81
75L-0.6-50	120,693.62	120,693.62	120,647.48	0.00	0.04	172.75	120,693.62	120,638.19	0.00	0.05	126.84
75T-0.2-50	99,186.32	99,186.32	99,186.32	0.00	0.00	24.01	99,186.32	99,186.32	0.00	0.00	19.69
75T-0.4-50	99,186.32	99,186.32	99,186.32	0.00	0.00	23.73	99,186.32	99,186.32	0.00	0.00	18.58
75T-0.8-50	99,186.32	99,186.32	99,186.32	0.00	0.00	23.35	99,186.32	99,186.32	0.00	0.00	15.85
100L-0.2-20	17,616.73	17,616.73	17,333.28	0.00	1.61	204.87	17,616.73	17,295.80	0.00	1.82	156.61
100L-0.4-20	14,211.00	14,211.00	14,211.00	0.00	0.00	42.89	14,211.00	14,211.00	0.00	0.00	38.13
100L-0.8-20	13,603.42	13,603.42	13,603.42	0.00	0.00	9.92	13,603.42	13,603.42	0.00	0.00	9.94
100T-0.2-20	2,116.57	2,116.57	2,116.57	0.00	0.00	6.25	2,116.57	2,116.57	0.00	0.00	6.79
100T-0.6-20	2,116.57	2,116.57	2,116.57	0.00	0.00	6.59	2,116.57	2,116.57	0.00	0.00	7.08
100T-0.8-20	2,116.57	2,116.57	2,116.57	0.00	0.00	8.33	2,116.57	2,116.57	0.00	0.00	6.59
100L-0.4-30	47,173.46	47,173.46	47,083.85	0.00	0.19	333.71	47,173.46	47,077.29	0.00	0.20	265.53
100L-0.6-30	43,747.51	43,747.51	43,744.36	0.00	0.01	241.59	43,747.51	43,741.20	0.00	0.01	182.42
100L-0.8-30	42,790.52	42,790.52	42,790.52	0.00	0.00	123.83	42,790.52	42,790.52	0.00	0.00	101.23
100T-0.4-30	25,271.32	25,271.32	25,271.32	0.00	0.00	22.74	25,271.32	25,271.32	0.00	0.00	19.78
100T-0.6-30	25,271.32	25,271.32	25,271.32	0.00	0.00	28.93	25,271.32	25,271.32	0.00	0.00	21.10
100T-0.8-30	25,271.32	25,271.32	25,271.32	0.00	0.00	24.79	25,271.32	25,271.32	0.00	0.00	17.59
100L-0.2-50	130,437.81	130,437.81	129,382.70	0.00	0.81	497.93	130,437.81	129,730.18	0.00	0.54	580.39
100L-0.6-50	119,319.31	119,319.31	119,283.14	0.00	0.03	346.62	119,319.31	119,268.86	0.00	0.04	259.29
100L-0.8-50	117,676.01	117,676.01	117,674.98	0.00	0.00	197.42	117,676.01	117,673.95	0.00	0.00	139.02
100T-0.2-50	98,113.77	98,113.77	98,113.77	0.00	0.00	53.27	98,113.77	98,113.77	0.00	0.00	49.99
100T-0.4-50	98,113.77	98,113.77	98,113.77	0.00	0.00	58.27	98,113.77	98,113.77	0.00	0.00	40.51
100T-0.6-50	98,113.77	98,113.77	98,113.77	0.00	0.00	66.09	98,113.77	98,113.77	0.00	0.00	41.91
Average	–	–	–	0.00	0.12	99.13	–	–	0.00	0.15	79.34

Table B.15
Results for AP instances with 25 nodes allowing direct connections.

Instances	CPLEX					E-ILS-RVND (30 runs)				
	Optimal value	Total pairs served (%)	Pairs served - DC (%)	Hubs	Time (s)	Best value	Average value	Δ_{best} (%)	Δ_{avg} (%)	Average time (s)
25L-0.2-20	22,957.90	57.76	14.72	8, 15, 18	4,856.34	22,957.90	22,919.79	0.00	0.17	0.89
25L-0.4-20	21,500.18	38.88	21.12	18	998.46	21,500.18	21,500.18	0.00	0.00	0.17
25L-0.6-20	21,500.18	38.88	21.12	18	248.01	21,500.18	21,500.18	0.00	0.00	0.16
25L-0.8-20	21,500.18	38.88	21.12	18	47.36	21,500.18	21,500.18	0.00	0.00	0.14
25T-0.2-20	20,791.17	30.88	30.88	–	1,431.53	18,862.83	18,862.83	9.27	9.27	0.10
25T-0.4-20	20,791.17	30.88	30.88	–	262.59	18,862.83	18,862.83	9.27	9.27	0.11
25T-0.6-20	20,791.17	30.88	30.88	–	37.21	18,862.83	18,862.83	9.27	9.27	0.11
25T-0.8-20	20,791.17	30.88	30.88	–	29.21	18,862.83	18,862.83	9.27	9.27	0.11
25L-0.2-30	60,058.61	99.36	11.04	2, 5, 8, 15, 16,18	5,060.12	59,894.50	59,502.97	0.27	0.93	5.12
25L-0.4-30	53,674.25	88.00	14.88	2, 8, 14, 18	2,568.79	53,674.25	53,501.89	0.00	0.32	1.67
25L-0.6-30	51,372.13	75.36	22.24	8,18	654.81	51,372.13	51,372.13	0.00	0.00	0.44
25L-0.8-30	50,544.37	65.76	32.00	18	270.61	50,544.37	50,544.37	0.00	0.00	0.17
25T-0.2-30	47,099.69	70.56	32.96	13	7,402.95	47,099.69	47,099.69	0.00	0.00	0.20
25T-0.4-30	47,099.69	70.56	32.96	13	1,977.48	47,099.69	47,099.69	0.00	0.00	0.19
25T-0.6-30	47,099.69	70.56	32.96	13	691.68	47,099.69	47,099.69	0.00	0.00	0.17
25T-0.8-30	47,099.69	70.56	32.96	13	175.41	47,099.69	47,099.69	0.00	0.00	0.15
25L-0.2-50	139,622.88	100.00	11.04	2, 5, 8, 15, 16,18	5,352.58	139,380.16	138,470.32	0.17	0.83	5.96
25L-0.4-50	131,997.39	99.68	15.84	2, 8, 14, 18	3,615.80	131,997.39	131,997.39	0.00	0.00	2.80
25L-0.6-50	128,096.89	98.72	24.16	8,18	1,133.26	128,096.89	128,096.89	0.00	0.00	0.71
25L-0.8-50	126,482.64	98.40	24.16	8,18	554.55	126,482.64	126,482.64	0.00	0.00	0.47
25T-0.2-50	123,489.06	99.20	21.44	8, 14, 24	15,539.26	123,489.06	122,855.28	0.00	0.51	0.79
25T-0.4-50	121,961.45	98.08	38.72	13	3,409.59	121,961.45	121,942.97	0.00	0.02	0.16
25T-0.6-50	121,961.45	98.08	38.72	13	1,120.00	121,961.45	121,924.49	0.00	0.03	0.17
25T-0.8-50	121,961.45	98.08	38.72	13	387.24	121,961.45	121,924.49	0.00	0.03	0.16
Average	–	–	–	–	2,409.37	–	–	1.56	1.66	0.88

Table C.16
Results for the instances with 25 nodes from the CAB data set considering 20 runs.

Costs	α	Optimal value	VNS - Zhang et al. (2023)			E-ILS-RVND		
			Δ_{med} (%)	Average time (s)	Adjus. time (s)	Δ_{best} (%)	Δ_{med} (%)	Average time (s)
High revenue								
L	0.2	1,162.92	0.00	250.24	224.22	0.00	0.00	10.62
	0.4	1,008.46	0.00	179.21	160.57	0.00	0.99	9.69
	0.6	898.24	0.00	115.65	103.62	0.00	0.00	5.73
	0.8	839.40	0.00	80.50	72.13	0.00	0.01	2.79
M	0.2	911.27	0.00	66.07	59.20	0.00	0.35	2.87
	0.4	803.73	0.00	32.35	28.99	0.00	0.52	2.46
	0.6	717.73	0.00	30.15	27.01	0.00	0.18	2.33
	0.8	690.90	0.00	28.91	25.90	0.00	0.00	0.52
H	0.2	738.08	0.00	28.35	25.40	0.00	0.73	2.00
	0.4	633.73	0.00	26.53	23.77	0.00	0.42	1.79
	0.6	599.18	0.00	36.64	32.83	0.00	0.00	0.22
	0.8	599.18	0.00	34.56	30.97	0.00	0.00	0.23
Medium revenue								
L	0.2	665.79	0.00	140.74	126.10	0.00	2.54	7.78
	0.4	520.25	0.00	74.89	67.10	0.00	0.11	5.28
	0.6	439.14	0.00	48.80	43.72	0.00	0.00	1.10
	0.8	424.73	0.00	59.04	52.90	0.00	0.00	0.73
M	0.2	426.89	0.00	58.24	52.18	0.00	0.00	2.03
	0.4	348.40	0.00	25.47	22.82	0.00	0.00	0.73
	0.6	327.82	0.00	24.73	22.16	0.00	0.00	0.73
	0.8	324.73	0.00	40.03	35.87	0.00	0.00	0.42
H	0.2	266.41	0.00	19.41	17.39	0.00	1.71	0.43
	0.4	259.89	0.00	26.28	23.55	0.00	0.00	0.22
	0.6	259.89	0.00	15.64	14.01	0.00	0.00	0.19
	0.8	259.89	0.00	12.33	11.05	0.00	0.00	0.21
Low revenue								
L	0.2	197.97	0.00	60.58	54.28	0.00	2.77	2.45
	0.4	156.90	0.00	20.35	18.23	0.00	0.00	0.48
	0.6	141.69	0.00	22.11	19.81	0.00	0.00	0.54
	0.8	132.16	0.00	37.03	33.18	0.00	0.00	0.49
M	0.2	69.02	0.00	22.04	19.75	0.00	0.00	0.39
	0.4	65.28	0.00	8.75	7.84	0.00	0.00	0.17
	0.6	65.28	0.00	8.07	7.23	0.00	0.00	0.16
	0.8	65.28	0.00	7.91	7.09	0.00	0.00	0.15
H	0.2	15.28	0.00	10.45	9.36	0.00	0.00	0.20
	0.4	15.28	0.00	9.14	8.19	0.00	0.00	0.17
	0.6	15.28	0.00	9.83	8.81	0.00	0.00	0.16
	0.8	15.28	0.00	11.30	10.12	0.00	0.00	0.15

References

Alibeyg, A., Contreras, I., Fernández, E., 2016. Hub network design problems with profits. Transp. Res. Part E: Logist. Transp. Rev. 96, 40–59. http://dx.doi.org/10.1016/j.tre.2016.09.008.

Alibeyg, A., Contreras, I., Fernández, E., 2017. Exact solution of hub network design problems with profits. European J. Oper. Res. 1–15. http://dx.doi.org/10.1016/j.ejor.2017.09.024.

Alumur, S.A., Campbell, J.F., Contreras, I., Kara, B.Y., Marianov, V., O'Kelly, M.E., 2021. Perspectives on modeling hub location problems. European J. Oper. Res. 291, 1–17. http://dx.doi.org/10.1016/j.ejor.2020.09.039.

Alumur, S., Kara, B.Y., 2008. Network hub location problems: The state of the art. European J. Oper. Res. 190, 1–21. http://dx.doi.org/10.1016/j.ejor.2007.06.008.

Alumur, S., Kara, B.Y., 2009. A hub covering network design problem for cargo applications in Turkey. J. Oper. Res. Soc. 60, 1349–1359. http://dx.doi.org/10.2307/40295695.

Alumur, S.A., Kara, B.Y., Karasan, O.E., 2009. The design of single allocation incomplete hub networks. Transp. Res. B 43, 0–951. http://dx.doi.org/10.1016/j.trb.2009.04.004.

Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T., 2010. F-race and iterated F-race: An overview. In: Experimental Methods for the Analysis of Optimization Algorithms. Springer, pp. 311–336. http://dx.doi.org/10.1007/978-3-642-02538-9_13.

Calık, 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. http://dx.doi.org/10.1016/j.cor.2008.11.023.

Camargo, R.S., Miranda, G., 2012. Single allocation hub location problem under congestion: Network owner and user perspectives. Expert Syst. Appl. 39, 3385–3391. http://dx.doi.org/10.1016/j.eswa.2011.09.026.

Camargo, R.S., Miranda, G.d., O'Kelly, M.E., Campbell, J.F., 2017. Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. Appl. Math. Model. 274–301. http://dx.doi.org/10.1016/j.apm.2017.06.035.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005a. Hub arc location problems: Part I: Introduction and results. Manage. Sci. 51, 1540–1555. http://dx.doi.org/10.2307/20110442.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005b. Hub arc location problems: Part II: Formulations and optimal algorithms. Manage. Sci. 51, 1556–1571. http://dx.doi.org/10.2307/20110443.

Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transp. Sci. 46, 153–169. http://dx.doi.org/10.1287/trsc.1120.0410.

Contreras, I., 2015. Hub location problems. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. Springer International Publishing, pp. 311–344. http://dx.doi.org/10.1007/978-3-319-13111-5_12.

Contreras, I., Cordeau, J.-F., Laporte, G., 2011a. Benders decomposition for large-scale uncapacitated hub location. Oper. Res. 59, 1477–1490. http://dx.doi.org/10.1287/opre.1110.0965.

Contreras, I., Díaz, J.A., Fernández, E., 2009. Lagrangean relaxation for the capacitated hub location problem with single assignment. OR Spectrum 31, 483–505. http://dx.doi.org/10.1007/s00291-008-0159-y.

Contreras, I., Díaz, J.A., Fernández, E., 2011b. Branch and price for large-scale capacitated hub location problems with single assignment. INFORMS J. Comput. 23, 41–55. http://dx.doi.org/10.1287/ijoc.1100.0391.

Čvokić, D.D., Stanimirović, Z., 2020. A single allocation hub location and pricing problem. Comput. Appl. Math. 39 (1), 1–24. http://dx.doi.org/10.1007/s40314-019-1025-z.

Dai, W., Zhang, J., Sun, X., Wandelt, S., 2019. HUBBI: Iterative network design for incomplete hub location problems. Comput. Oper. Res. 104, 394–414. http://dx.doi.org/10.1016/j.cor.2018.09.011.

Davari, S., Zarandi, M.H.F., Turksen, I.B., 2013. The incomplete hub-covering location problem considering imprecise location of demands. Sci. Iran. 20, 983–991. http://dx.doi.org/10.1016/j.scient.2013.04.010.

de Camargo, R.S., de Miranda, Jr., G., Luna, H.P.L., 2009. Benders decomposition for hub location problems with economies of scale. Transp. Sci. 43 (1), 86–97. http://dx.doi.org/10.1287/trsc.1080.0233.

Ernst, A.T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Locat. Sci. 4, 0–154. http://dx.doi.org/10.1016/s0966-8349(96)00011-3.

Fazel Zarandi, M.H., Davari, S., Haddad Sisakht, A., 2015. An empirical comparison of simulated annealing and iterated local search for the hierarchical single allocation hub median location problem. Sci. Iran. 22 (3), 1203–1217.

Floyd, R.W., 1962. Algorithm 97: Shortest path. Commun. ACM 5, 345. http://dx.doi.org/10.1145/367766.368168.

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

Ghaffarinasab, N., 2020. A tabu search heuristic for the bi-objective star hub location problem. Int. J. Manag. Sci. Eng. Manag. 15 (3), 213–225. http://dx.doi.org/10.1080/17509653.2019.1709992.

Guan, J., Lin, G., Feng, H.-B., 2018. A multi-start iterated local search algorithm for the uncapacitated single allocation hub location problem. Appl. Soft Comput. 73, 230–241. http://dx.doi.org/10.1016/j.asoc.2018.08.035.

Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S., 2017. Variable neighborhood search: Basics and variants. EURO J. Comput. Optim. 5, 423–454. http://dx.doi.org/10.1007/s13675-016-0075-x.

Hoff, A., Peiró, J., Corberán, Á., Martí, R., 2017. Heuristics for the capacitated modular hub location problem. Comput. Oper. Res. 86, 94–109. http://dx.doi.org/10.1016/j.cor.2017.05.004.

Huo, J.-Z., Hou, Y.-T., Chu, F., He, J.-K., 2019. A combined average-case and worst-case analysis for an integrated hub location and revenue management problem. Discrete Dyn. Nat. Soc. 1–13. http://dx.doi.org/10.1155/2019/8651728.

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A.M., Talbi, E.-G., 2022. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. European J. Oper. Res. 296 (2), 393–422. http://dx.doi.org/10.1016/j.ejor.2021.04.032, URL https://www.sciencedirect.com/science/article/pii/S0377221721003623.

Karimi-Mamaghan, M., Mohammadi, M., Pirayesh, A., Karimi-Mamaghan, A.M., Irani, H., 2020. Hub-and-spoke network design under congestion: A learning based metaheuristic. Transp. Res. Part E: Logist. Transp. Rev. 142, 102069. http://dx.doi.org/10.1016/j.tre.2020.102069, URL https://www.sciencedirect.com/science/article/pii/S1366554520307201.

Klincewicz, J.G., 2002. Enumeration and search procedures for a hub location problem with economies of scale. Ann. Oper. Res. 110 (1), 107–122. http://dx.doi.org/10.1023/A:1020715517162.

Kratica, J., Milanović, M., Stanimirović, Z., Tošić, D., 2011. An evolutionary-based approach for solving a capacitated hub location problem. Appl. Soft Comput. 11 (2), 1858–1866. http://dx.doi.org/10.1016/j.asoc.2010.05.035.

Lin, C.-C., Lee, S.-C., 2018. Hub network design problem with profit optimization for time-definite LTL freight transportation. Transp. Res. Part E: Logist. Transp. Rev. 114, 104–120. http://dx.doi.org/10.1016/j.tre.2018.03.007.

López-Ibañez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Oper. Res. Perspect. 3, 43–58. http://dx.doi.org/10.1016/j.orp.2016.09.002.

Lourenço, H., Martin, O., Stützle, T., 2001. A beginner's introduction to iterated local search. In: Proceedings of the 4th Metaheuristic International Conference. Porto, Portugal.

Lourenço, H., Martin, O., Stützle, T., 2003. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics, Vol. 57. International Series in Operations Research & Management Science, Boston, MA, pp. 321–353. http://dx.doi.org/10.1007/0-306-48056-5_11.

Lür-Villagra, A., Marianov, V., 2013. A competitive hub location and pricing problem. European J. Oper. Res. 231 (3), 734–744. http://dx.doi.org/10.1016/j.ejor.2013.06.006.

Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., Jamiri, A., 2020. A multi-modal competitive hub location pricing problem with customer loyalty and elastic demand. Comput. Oper. Res. 123, 105048. http://dx.doi.org/10.1016/j.cor.2020.105048.

Martins de Sá, E., Morabito, R., de Camargo, R.S., 2018a. Benders decomposition applied to a robust multiple allocation incomplete hub location problem. Comput. Oper. Res. 89, 31–50. http://dx.doi.org/10.1016/j.cor.2017.08.001.

Martins de Sá, E., Morabito, R., de Camargo, R.S., 2018b. Efficient benders decomposition algorithms for the robust multiple allocation incomplete hub location problem with service time requirements. Expert Syst. Appl. 93, 50–61. http://dx.doi.org/10.1016/j.eswa.2017.10.005.

Neamatian Monemi, R., Gelareh, S., Hanafi, S., Maculan, N., 2017. A co-opetitive framework for the hub location problems in transportation networks. Optimization 66 (12), 2089–2106. http://dx.doi.org/10.1080/02331934.2017.1295045.

Nickel, S., Schöbel, A., Sonneborn, T., 2001. Hub location problems in urban traffic networks. In: Pursula, M., Niittymäki, J. (Eds.), Mathematical Methods on Optimization in Transportation Systems. Springer, pp. 95–107. http://dx.doi.org/10.1007/978-1-4757-3357-0_6.

O'Kelly, M.E., Bryan, D., 1998. Hub location with flow economies of scale. Transp. Res. B 32 (8), 605–616. http://dx.doi.org/10.1016/S0191-2615(98)00021-6.

O'Kelly, M.E., Miller, H.J., 1994. The hub network design problem. J. Transp. Geogr. 2, 31–40. http://dx.doi.org/10.1016/0966-6923(94)90032-9.

Oliveira, F.A., de Sá, E.M., de Souza, S.R., 2022. Benders decomposition applied to profit maximizing hub location problem with incomplete hub network. Comput. Oper. Res. 142, 105715. http://dx.doi.org/10.1016/j.cor.2022.105715.

Öztürk, C., Tuzkaya, G., Bulkan, S., 2021. Centrality based solution approaches for median-type incomplete hub location problems. Comput. Ind. Eng. 156, 107275. http://dx.doi.org/10.1016/j.cie.2021.107275.

Pratt, J.W., 1959. Remarks on zeros and ties in the wilcoxon signed rank procedures. J. Amer. Statist. Assoc. 54 (287), 655–667. http://dx.doi.org/10.2307/2282543.

Reinsma, J.A., Penna, P.H.V., Souza, M.J.F., 2018. Um algoritmo simples e eficiente para resolução do problema do caixeiro viajante generalizado (in Portuguese). In: Anais Do 50^{O} Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro, Brazil, Available at https://proceedings.science/sbpo/papers/um-algoritmo-simples-e-eficiente-para-resolucao-do-problema-do-caixeiro-viajante-generalizado.

Rodríguez-Martí, I., Salazar-González, J.-J., Yaman, H., 2014. A branch-and-cut algorithm for the hub location and routing problem. Comput. Oper. Res. 50, 161–174. http://dx.doi.org/10.1016/j.cor.2014.04.014.

Rodríguez-Martín, I., Salazar-González, J.-J., 2006. An iterated local search heuristic for a capacitated hub location problem. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Hybrid Metaheuristics. Springer, pp. 70–81. http://dx.doi.org/10.1007/11890584_6.

Salehi, M., Tikani, H., 2020. Using revenue management technique to allocate the capacity in reliable hub network design under uncertain air passenger traffic. J. Ind. Eng. Manag. Stud. 7 (2), 139–164. http://dx.doi.org/10.22116/jiems.2020.195853.1293.

Shang, X., Yang, K., Jia, B., Gao, Z., Ji, H., 2021. Heuristic algorithms for the bi-objective hierarchical multimodal hub location problem in cargo delivery systems. Appl. Math. Model. 91, 412–437. http://dx.doi.org/10.1016/j.apm.2020.09.057.

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52 (3-4), 591–611. http://dx.doi.org/10.1093/biomet/52.3-4.591.

Souza, M., Coelho, I., Ribas, S., Santos, H., Merschmann, L., 2010. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. European J. Oper. Res. 207 (2), 1041–1051. http://dx.doi.org/10.1016/j.ejor.2010.05.031.

Subramanian, A., Drummond, L., Bentes, C., Ochi, L., Farias, R., 2010. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Comput. Oper. Res. 37 (11), 1899–1911. http://dx.doi.org/10.1016/j.cor.2009.10.011, Metaheuristics for Logistics and Vehicle Routing.

Taherkhani, G., Alumur, S.A., 2018. Profit maximizing hub location problems. Omega 86, 1–15. http://dx.doi.org/10.1016/j.omega.2018.05.016.

Taherkhani, G., Alumur, S.A., Hosseini, S.M., 2020. Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes. Transp. Sci. 54, 1446–1470. http://dx.doi.org/10.1287/trsc.2020.1003.

Tikani, H., Honarvar, M., Mehrjerdi, Y.Z., 2018. Developing an integrated hub location and revenue management model considering multi-classes of customers in the airline industry. Comput. Appl. Math. 37 (3), 3334–3364. http://dx.doi.org/10.1007/s40314-017-0512-3.

Tiwari, R., Jayaswal, S., Sinha, A., 2021. Alternate solution approaches for competitive hub location problems. European J. Oper. Res. 290 (1), 68–80. http://dx.doi.org/10.1016/j.ejor.2020.07.018.

Todosijević, R., Urošević, D., Mladenović, N., Hanafi, S., 2017. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. Optim. Lett. 11, 1109–1121. http://dx.doi.org/10.1007/s11590-015-0867-6.

Wandelt, S., Dai, W., Zhang, J., Sun, X., 2022. Toward a reference experimental benchmark for solving hub location problems. Transp. Sci. 56 (2), 543–564.

Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biom. Bull. 1, 80–83. http://dx.doi.org/10.2307/3001968.

Xu, Y., Dai, W., Sun, X., Wandelt, S., 2017. Improved benders decomposition for capacitated hub location problem with incomplete hub networks. In: 2017 IEEE Symposium Series on Computational Intelligence. SSCI, pp. 1–8. http://dx.doi.org/10.1109/SSCI.2017.8285341.

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. http://dx.doi.org/10.2307/4622884.

Zhang, C., Sun, X., Dai, W., Wandelt, S., 2023. Solving hub location problems with profits using variable neighborhood search. Transp. Res. Rec. 2677 (1), 1675–1695. http://dx.doi.org/10.1177/03611981221105501.