

Al Athamneh, Raed et al.

Article

Variable neighborhood search algorithm for the single assignment incomplete hub location problem with modular capacities and direct connections

Operations Research Perspectives

Provided in Cooperation with:

Elsevier

Suggested Citation: Al Athamneh, Raed et al. (2023) : Variable neighborhood search algorithm for the single assignment incomplete hub location problem with modular capacities and direct connections, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 11, pp. 1-11,
<https://doi.org/10.1016/j.orp.2023.100286>

This Version is available at:

<https://hdl.handle.net/10419/325771>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

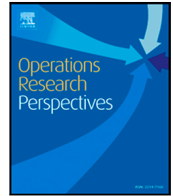
Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc-nd/4.0/>



Variable Neighborhood Search Algorithm for the Single Assignment Incomplete Hub Location Problem with Modular Capacities and Direct Connections

Raed AL Athamneh^a, Moayad Tanash^{a,*}, Dania Bani Hani^b, Mustafa Rawshdeh^a, Abdallah Alawin^a, Zaid Albataineh^b

^a Department of Industrial Engineering, Faculty of Engineering, The Hashemite University, Zarqa, 13133, Jordan

^b Hijawi Faculty for Engineering Technology, Yarmouk University, Irbid, 21163, Jordan

ARTICLE INFO

Keywords:

Hub-and-Spoke network
Hub location
Flow dependent costs
Direct connection
Heuristic algorithms
Integer programming
Network loading problem

ABSTRACT

In distribution systems such as airlines and express package delivery, the use of hub-and-spoke networks is common, and flow consolidation at hub facilities is essential for cost reduction. While a constant discount factor is typically used to model cost reduction in interhub links, this paper explores an extension of the incomplete hub location problem with modular capacity that enables direct connections between non-hub nodes. The modified approach, called MHLPCD, aims to locate a set of hub facilities, connect each non-hub node to a hub, and activate hub facility links, access arc links, and direct links between non-hub nodes to minimize network costs. The MHLPCD integrates link activation decisions into the decision-making process and utilizes modular arc costs to model the flow dependence of transportation costs in all arcs. To solve the problem, the paper presents a mixed-integer mathematical programming formulation and heuristic algorithm based on a greedy randomized adaptive search and variable neighborhood search approach. The proposed algorithm produces high-quality solutions, as demonstrated through computational experiments on benchmark instances with up to 40 nodes. Furthermore, a sensitivity analysis of the optimal network structure indicates that increasing the discount factor, by varying hub and access arc capacities as well as the associated variable costs, results in fewer hubs being established and more direct shipments between non-hub nodes being permitted.

1. Introduction

Hub location problems (HLPs) are optimization problems that are known to be challenging and classified as NP-hard. These problems are crucial for network design and planning in transportation systems, such as airlines and freight transport. Hub-and-spoke architecture is utilized in these systems to transport commodities, including passengers and goods, between different origins and destinations. Instead of directly connecting each origin/destination (O/D) pair, hub-and-spoke networks serve demand through fewer links. Hub facilities consolidate flows from multiple origins, transfer them via the hub-level network, and distribute them to their final destinations. By utilizing fewer network links, economies of scale can be applied to transportation costs, reducing setup costs and centralizing the handling and sorting of commodities.

When addressing HLPs, the decision-making process involves considering both the location and arc selection. The former deals with identifying the optimal places to locate the hub facilities, while the

latter focuses on determining the network's structure by establishing connections between the origins, destinations, and hubs, forming a routing framework within the hub-and-spoke network. The main objective is to reduce the overall cost of flow by determining the optimal location for the hubs and constructing the hub network in the most cost-efficient manner [1–6].

After [7] influential work, hub location problems have garnered significant research attention. In line with discrete facility location problems, the literature has explored several versions of uncapacitated hub location, p-hub median, hub covering, and p-hub center problems. Various factors, such as imposing specific topological structures, the pattern of assignment of nodes to hubs (single versus multiple), or the existence of capacity limitations on hub nodes or arcs, have led to many variations in these problems. Nonetheless, the majority of HLPs rely on four key assumptions:

1. Commodities must traverse a hub network; therefore, the O/D path must contain at least one hub node.

* Corresponding author.

E-mail address: moayad@hu.edu.jo (M. Tanash).

2. There are no setup costs for hub arcs; thus, hubs are linked at no additional expense.
3. Commodities cannot be sent directly between non-hub nodes.
4. Triangle inequality is satisfied by the distances between nodes.

The first two assumptions influence the O/D paths on the solution network, which can have one or two hubs. When combined with the second and third assumptions, a fully interconnected hub network is created, which has a complete backbone network. Such fully interconnected networks provide more efficient routes and enable a constant discount factor ($0 < \alpha < 1$) to be applied to all inter-hub transportation costs.

The assumptions underlying classical HLPs have attractive theoretical properties and have led to the development of mathematical models that exploit the network's structure [8–13], as well as advanced solution algorithms that can handle large instances [14–17]. However, these assumptions may lead to unreasonable solutions. For instance, applying a constant discount factor to all inter-hub arcs, irrespective of their traversed volume, oversimplifies the costs and may not capture the economies of scale resulting from the consolidation of flows on hub arcs. Additionally, assuming complete interconnection between hub nodes may result in hub arcs carrying significantly less flow than access arcs, even though only the transportation costs of hub arcs are discounted. Furthermore, hub arcs may transport varying quantities of flow using the same discount factor. Thus, the classical HLPs assumptions of complete interconnection, flow-independent costs, and the prohibition of direct connections can lead to an incorrect calculation of the network's total transportation cost, as well as an inappropriate selection of the optimal number of hub nodes and allocation pattern for O/D nodes to hubs.

This paper investigates the incomplete hub location problem with modular capacity and direct connections (MHLPCD), which extends the work of Tanash et al. [18] by introducing direct links between non-hub nodes. The MHLPCD aims to determine a set of hub facilities and assign each non-hub node to a single hub while ensuring connectivity between O/D nodes by activating hub arcs, access arcs, and direct links. Unlike traditional hub location problems, the MHLPCD does not assume a fully interconnected hub network or a specific topological structure and instead incorporates hub network design into its decision-making process. Furthermore, the MHLPCD permits direct connections between non-hub nodes, allowing for more flexibility in hub network design and making it suitable for scenarios where traditional HLP assumptions do not apply. Our study highlights the potential benefits of this intriguing variant of the HLP problem.

A variety of modeling strategies have been developed to account for flow-independent economies of scale. O'Kelly and Bryan [19] developed a model in which the discount factor on hub arcs is stated as a function of flow. Later formulations, such as those by Bryan [20], Kliniewicz [21], and De Camargo et al. [22], represented economies of scale as flow-dependent functions on the inter-hub arcs of a fully interconnected hub network. Horner and O'Kelly [23] presented a nonlinear cost function that rewards economies of scale in all network arcs, which is based on the link performance function. In contrast, [24] established a network design model that emphasizes the architecture of the network over the location of hub facilities, with the discount factor only applying to arcs with flows greater than a set threshold. Racunica and Wynter [25] introduced a nonlinear concave cost hub location model that determines the lowest cost location for intermodal freight hubs but only models the discounted flow-dependent cost for access arcs. Cunha and Silva [26] constructed a hub-and-spoke network using a nonlinear cost function, where the discount factor on hub arcs fluctuates based on the amount of freight transported between hubs. Other studies, such as those by Kimms [27], Tanash et al. [18], Tanash [28], O'Kelly et al. [29], and Fard and Alfandari [30], used linear cost functions that combine variable transportation costs with fixed costs for activating arcs. Yaman [31] discussed modular links in the context

of a star-star network, and Tanash et al. [18], Tanash [28] used flow-dependent modular arc costs in a modular hub location problem. Fard and Alfandari [30] analyzed the stepwise cost function with its linear approximation, and Lüer-Villagra et al. [32] presented a hub-and-spoke network model with piecewise linear cost functions in each arc. Other studies focused on the design of specific topological structures, such as star-star networks [31,33], tree-star networks [34,35], cycle-star networks [36], and hub line networks [34,37], to reduce overall costs.

The MHLPCD provides valuable managerial insights for optimizing hub network configurations. By strategically considering direct connections between non-hub nodes, decision-makers can determine the optimal placement of hub facilities, resulting in efficient transportation and logistics operations. Solving the MHLPCD informs crucial decisions on hub location selection, capacity planning, and route optimization, leading to improved resource allocation, reduced transportation costs, enhanced service quality, and increased customer satisfaction. Furthermore, it enables managers to identify and address bottlenecks in existing hub networks, streamlining operations and gaining a competitive edge.

The MHLPCD has diverse applications across various industries and sectors. In transportation and logistics, it optimizes operations by designing efficient routes, determining optimal hub locations, and improving supply chain management. Strategic hub placement and direct connections between non-hub nodes enhance the speed, reliability, and cost-effectiveness of transportation networks, resulting in improved efficiency and better meeting customer demands.

In e-commerce and last-mile delivery, incomplete modular hub networks with direct services play a crucial role in optimizing the distribution of goods. By strategically locating hubs and establishing direct connections, companies improve delivery efficiency, timeliness, and cost-effectiveness, leading to faster delivery times, reduced shipping costs, and increased customer satisfaction.

In the aviation industry, incomplete modular hub networks with direct services bring significant benefits for optimizing airline and airport operations. Strategic hub airport placement and direct connections between non-hub airports improve connectivity, flight schedules, and passenger transfer times, resulting in improved operational efficiency, enhanced passenger convenience, and better utilization of airport infrastructure. This approach also finds applications in urban planning and infrastructure development, where incomplete modular hub networks with direct services are utilized to optimize public transportation systems, traffic flow, and congestion. By incorporating these networks into city planning, cities can effectively manage their transportation systems, leading to the creation of more sustainable and livable urban environments.

The failure to acknowledge the significance of direct service in transportation systems has been a notable oversight in previous research endeavors. This lack of recognition has resulted in additional transportation costs due to obligatory transfers at hub facilities, compelling goods or passengers to take unnecessary detours through intermediate hubs, leading to increased expenses and potential delays. Disregarding the potential of direct service disregards a crucial factor that has a profound impact on the efficiency and cost-effectiveness of transportation systems.

To optimize these systems, it is essential to recognize the indispensability of point-to-point direct service, minimize unnecessary transfers at hubs, and establish streamlined and cost-effective flows of goods or passengers. Several studies have addressed the inclusion of direct flows between non-hub nodes in HLPs. For example, [38] explored the capacitated hub and spoke network design problem with capacity constraints on hub-routed traffic. Mahmutogullari and Kara [39] expanded the scope of HLPs by introducing direct flows between non-hub nodes and developed mathematical models for various problem formulations. Martins de Sá et al. [34] presented the hub line location problem, which allows flows to be routed either through hub arcs or directly between origin and destination nodes based on the shortest

travel time. De Camargo et al. [40] extensively investigated the incomplete hub location network design problem, taking hop constraints into account and proposing enhanced formulations for the p-hub location problem variant.

Furthermore, Dai et al. [41] introduced HUBBI, a heuristic-based method for p-hub location problems with incomplete hub networks. HUBBI allows for direct links between spoke nodes, enabling specific node pairs to bypass the hubs, and considers fixed and variable costs associated with fulfilling travel demands. de Sá et al. [42] investigated the multiple allocation incomplete hub location problem, incorporating direct connections between non-hub nodes. Taherkhani and Alumur [43] proposed novel mathematical formulations for each allocation strategy, aiming to maximize profit by serving demand nodes and allowing direct connections between non-hub nodes.

These studies highlight the importance of recognizing the significance of direct service in transportation systems and the potential benefits it brings. By incorporating direct connections and optimizing hub network designs, transportation systems can achieve improved efficiency, reduced costs, and enhanced profitability.

The inclusion of direct connections between non-hub nodes in the context of a modular incomplete hub network with capacity restrictions has received limited research attention. Figueiredo et al. [44] presented a linear integer programming model that extends the p-median problem and focuses on air passenger transportation. Their multiple allocation model aims to identify the optimal and cost-effective approach for connecting O/D pairs in a transportation network, considering the incorporation of direct services between the origin and destination nodes. Hoff et al. [45] developed a heuristic approach to solve the capacitated hub location problem with modular link capacities, considering both with and without direct connections. However, their models only consider a fully interconnected hub network. Fard and Alfandari [30] analyzed the potential for direct shipments within a fully interconnected hub-and-spoke network with modular capacity, introducing stepwise and generalized linear cost functions in a mixed-integer linear programming formulation to account for economies of scale.

Osorio-Mora et al. [46] studied the complete hub network design in a capacitated multimodal multi-commodity scenario that allows direct shipments between non-hub nodes. Shang et al. [47] focused on the stochastic version of the multi-modal hub location problem, incorporating a direct link strategy and multiple capacity levels in cargo delivery systems. They developed a memetic algorithm (MA) to efficiently generate high-quality solutions. However, their model assumes a complete hub network. In a 2021 study, [48] introduced a model addressing the multi-modal competitive hub location pricing problem, considering a transportation system design for a company entering a market with elastic demand. They took into account a connected backbone network that allows direct services between non-hub nodes. The company's goal was to strategically position its hubs and offer competitive pricing to attract customers, considering variations in customer loyalty across different nodes.

Dai et al. [49] proposed a model for the air/rail multi-modal, multiple allocation hub location problem under uncertain travel demands. Their approach involved constructing a fully interconnected hub-and-spoke network integrating air and railway transportation modes, allocating hub links exclusively for air transportation, while access links could utilize both modes. Passengers in cities with rail stations and airports had the flexibility to switch between modes, considering transit costs and times. The model also considered capacity considerations for various types of air links, including hub links, access links (collection and distribution), and direct links.

Our study stands out from previous research due to its comprehensive integration of all the aforementioned features. In contrast to existing literature, our model specifically focuses on the design of an incomplete hub location problem. This problem involves assigning modular capacities to all network arcs and accommodating direct

point-to-point demands. This unique combination of features creates a highly intricate problem, as our model encompasses two levels of difficulty. Firstly, we address the location allocation problem, which is widely recognized for its NP-Hard complexity. Secondly, we tackle the resulting network loading problem, which is also acknowledged as an NP-hard problem. The incorporation of these two challenging aspects in our study sets it apart as a novel and ambitious endeavor in the field. We strive to push the boundaries of current research by seeking more effective solutions to the problem at hand. To the best of our knowledge, this problem was initially introduced by Mirzaghafour [50]. However, their formulation posed significant challenges due to the large number of decision variables and constraints involved. This made it difficult to solve even for small-sized problems using general solvers.

The primary focus of this paper is on the incomplete hub location problem, which entails designing an efficient hub network with incomplete connections and direct services. To tackle this problem, we propose a novel mixed integer programming formulation known as the single assignment problem with modular capacities and direct connections (MHLPCD). Our formulation takes into consideration the transportation costs and flow dependence of all network arcs by utilizing modular arc costs, which provide a more accurate representation of the dynamic nature of real-world scenarios. Additionally, our formulation enables the creation of an incomplete hub network that incorporates direct connections, offering greater flexibility and practicality compared to previous approaches.

Addressing the MHLPCD poses a complex optimization challenge involving decisions related to location, allocation, and network design under capacity restrictions. To obtain high-quality and feasible solutions, we introduce a novel metaheuristic algorithm that employs greedy randomized adaptive procedures. This algorithm serves as the foundation of our approach and is further enhanced through a variable neighborhood search aimed at refining the solutions obtained during the initial greedy phase. Furthermore, we develop a greedy heuristic and local search phase integrated into each iteration of the metaheuristic algorithm to tackle the network loading problem (NLP), which remains an NP-hard optimization problem even when location allocation decisions are predetermined. By incorporating the NLP within our iterative process, our objective is to enhance the overall performance and effectiveness of the metaheuristic algorithm.

In this paper, we conducted a thorough sensitivity analysis to examine the influence of direct connections on the optimal structure of the network. Our main objective was to assess how the inclusion of direct connections impacts the overall costs and efficiency of the hub network. The analysis yielded insightful results, demonstrating that incorporating additional direct connections between non-hub nodes and reducing the number of hub facilities led to a significant reduction in overall costs compared to the MHL case without direct connections. This highlights the importance of considering direct connections during the design phase of hub networks, as it presents a valuable opportunity for substantial cost savings and improved operational efficiency.

The implications of our findings are highly valuable for professionals in the field. By recognizing the cost-reducing benefits of direct connections and strategically optimizing the number and placement of hub facilities, practitioners can design hub networks that are both cost-efficient and operationally effective. This aspect of our research addresses a crucial gap in the existing literature by providing practical insights into the impact of direct connections on the design and optimization of networks.

To provide strong evidence for our claims, we conducted extensive computational experiments to evaluate the effectiveness of our solution algorithm. These experiments involved applying our algorithm to forty instances of the AP dataset. The consistent results demonstrated the algorithm's ability to generate high-quality and feasible solutions within an acceptable CPU time. This empirical evidence further reinforces the effectiveness and practicality of our proposed approach, showcasing

the performance and reliability of our algorithm in solving real-world instances of the incomplete hub location problem.

The remainder of this paper is organized as follows: Section 2 formally defines the problem and presents the mixed integer programming formulation, Section 3 describes the proposed heuristic, Section 4 presents the computational results and analysis, and Section 5 provides the conclusions.

2. Mathematical formulation

Consider a complete graph $G = (N, A)$ with nodes $N = 1, 2, \dots, n$ representing candidate hub locations and arcs A representing the flow between the nodes. Let W_{ij} denote the amount of flow between nodes $i \in N$ and $j \in N$, and let $O_i = \sum_j W_{ij}$ denote the total flow originating from node $i \in N$. For each node $i \in N$, f_i is the setup cost for locating a hub facility. The distance between nodes i and j is represented by $d_{ij} \geq 0$, which need not be symmetric or satisfy the triangle inequality.

The transportation cost is calculated using the method described in the works of Tanash et al. [18]. This calculation involves determining the necessary number of hub connections to guide the flow along inter-hub arcs, access arcs, and direct arcs. For each type of arc, the calculation determines the required number of facility links for efficient flow routing. The transportation cost for a hub arc (k, m) with a capacity of B is determined by the formula $c_{km} = l_c + bd_{km}$, where l_c represents the fixed costs and b represents the variable cost proportional to the travel distance d_{km} . Similarly, the transportation cost for an access arc (i, k) with a capacity of H is given by $q_{ik} = l_q + pd_{ik}$, where l_q and p denote the fixed and variable costs associated with the arc. Lastly, the transportation cost for a direct arc (i, j) with a capacity of R is calculated as $r_{ij} = l_r + ad_{ij}$, where l_r and a represent the fixed and variable costs.

To account for economies of scale when consolidating flows at hub facilities and utilizing more efficient paths between hubs, we make the assumption that the unit transportation cost on hub arcs is lower than the unit transportation cost on access arcs, and the unit transportation cost on access arcs is lower than the unit transportation cost on direct arcs. In other words, we have $\frac{c_{km}}{B} < \frac{q_{ik}}{H} < \frac{r_{ij}}{R}$, where $B > H > R$, $b > p > a$, and $l_c > l_q > l_r$.

The MHLPCD involves selecting a set of hub facilities, establishing a hub-and-spoke network with facility links, and optimizing flow routing to minimize overall setup and transportation costs. Each non-hub node is assigned to exactly one hub facility, with the option to allow direct connections between non-hub nodes. The network topology is not predetermined, but a fixed activation cost is considered for hub arcs, enabling the selection of the optimal hub-level layout. These assumptions make the MHLPCD a highly challenging problem, with the activation of facility links being NP-hard due to its equivalence to the network loading problem. The decision variables for the MHLPCD include determining which non-hub nodes are assigned to which hubs and which flows should travel directly. The decision variables used in the model are as follows:

For each $i \in N, k \in N$, we define binary variables

$$z_{ik} = \begin{cases} 1 & \text{if node } i \text{ is allocated to hub facility } k, \\ 0 & \text{otherwise.} \end{cases}$$

Note that, when $z_{kk} = 1$, node k is selected as a hub and assigned to itself.

$$M_{ij} = \begin{cases} 1 & \text{if node } i \text{ receives flow directly from non hub node } j, \\ 0 & \text{otherwise.} \end{cases}$$

For each pair of nodes $k \in N$ and $m \in N$, we define the following network design variables:

$$\begin{aligned} A_{km} &= \text{number of facility links between non-hub nodes } k \text{ and } m. \\ y_{km} &= \text{number of facility links between hub nodes } k \text{ and } m. \\ v_{km}^1 &= \text{number of facility links between non-hub node } k \text{ and hub node } m. \end{aligned}$$

v_{km}^2 = number of facility links between hub node m and non hub node k .

For each $i \in N, k \in N$ and $m \in N$, we define the following flow variables:

x_{imk} = the amount of flow from hub m to hub k originating from node i .

g_{imj} = the amount of flow from hub m to node j originating from node i .

h_{ik} = the amount of flow from node i to hub k .

Using these sets of variables, the MHLPCD can be formulated as follows:

$$\begin{aligned} \text{minimize } & \sum_{k \in N} F_k z_{kk} + \sum_{i \in N} \sum_{k \in N} q_{ik}(v_{ik}^1 + v_{ik}^2) + \sum_{m \in N} \sum_{k \in N} c_{km} y_{km} \\ & + \sum_{i \in N} \sum_{j \in N} r_{ij} A_{ij} \\ \text{subject to } & \sum_{k \in N} z_{ik} = 1 \quad i \in N \quad (1) \\ & z_{ik} \leq z_{kk} \quad i, k \in N \quad (2) \end{aligned}$$

$$\sum_{k \in N} h_{ik} = \sum_{j \in N} (1 - M_{ij}) W_{ij} \quad \forall i \in N \quad (3)$$

$$h_{ik} \leq O_i z_{ik} \quad \forall i \in N, k \in N \quad (4)$$

$$\sum_{m \in N} g_{imj} = W_{ij} (1 - M_{ij}) \quad \forall i \in N, j \in N \quad (5)$$

$$g_{imj} \leq W_{ij} z_{jm} \quad \forall i \in N, j \in N, m \in N \quad (6)$$

$$h_{ik} \leq H v_{ik}^1 \quad \forall i \in N, k \in N \quad (7)$$

$$\sum_{i \in N} g_{ikj} \leq H v_{jk}^2 \quad \forall j \in N, k \in N \quad (8)$$

$$\sum_{i \in N} x_{ikm} \leq B y_{km} \quad \forall k, m \in N \quad (9)$$

$$y_{km} \leq Q z_{mm} \quad k, m \in N \quad (10)$$

$$y_{km} \leq Q z_{kk} \quad k, m \in N \quad (11)$$

$$M_{ij} \leq 1 - z_{ii} \quad \forall i, j \in N \quad (12)$$

$$M_{ij} \leq 1 - z_{jj} \quad \forall i, j \in N \quad (13)$$

$$M_{ij} W_{ij} \leq R A_{ij} \quad \forall i, j \in N \quad (14)$$

$$\sum_{j \in N} g_{ikj} + \sum_{m \in N} x_{ikm} - \sum_{m \in N} x_{imk} - h_{ik} = 0 \quad \forall i, k \in N \quad (15)$$

$$M_{ik}, z_{ik} \in \{0, 1\} \quad \forall i, k \in N \quad (16)$$

$$y_{km}, v_{km}^1, v_{km}^2 \in \mathbb{Z}^+ \quad \forall k, m \in N \quad (17)$$

$$x_{ikm}, g_{ikm}, h_{ik} \geq 0 \quad \forall i, k, m \in N. \quad (18)$$

The objective function aims to minimize the total costs associated with locating hub facilities, transportation between hubs, non-hub nodes and hubs, and direct flow between non-hub nodes. The sets of constraints (1) and (2) ensure that each node is allocated to a single hub facility. Constraints (3)–(6) ensure that all undirected flows from and to each node must pass through a hub facility. Capacity Constraints (7)–(9) limit the amount of flow that can be sent on each hub and access arc, respectively. Constraints (10) and (11) guarantee that hub arc (k, m) is active only if both k and m are hub nodes. The upper limit on the number of hub arcs that can be used between each node k and m is represented by Q . This limit is computed as $Q = \sum_{i \in N, j \in N} W_{ij} / H$. Constraints (12) and (13) ensure that a direct connection is established only if i and j are not hub nodes. Constraints (14) represent the capacity constraints on the direct connection arcs. Constraints (15) are the flow balance equations. Finally, constraints (16)–(18) are the integrality and non-negativity constraints.

3. Solution algorithm

The proposed method in this work is a metaheuristic approach that combines the Greedy Randomized Adaptive Search Procedure (GRASP) with the Variable Neighborhood Search (VNS) to address combinatorial optimization and global optimization problems. Initially, GRASP is used to generate starting solutions, which are then improved upon using VNS. Brimberg [51] introduced VNS, which systematically explores a set of k neighbors of the current solution and switches to a new one only if an improvement is found. VNS conducts a local search within a neighborhood \aleph_1 to find the local optimum solution and then progresses to neighborhoods $\aleph_2, \dots, \aleph_k$ until the optimal (best) solution is obtained. The method switches back to \aleph_1 if it improves the best-known solution.

In this study, solutions are represented as a set of hub nodes Ω , hub arcs E , an assignment mapping a , and a direct arc connecting non-hub nodes r . The solution is denoted as $S = (\Omega, E, a, M)$, where $\Omega(i) = 1$ if node $i \in N$ is chosen as a hub, $E(i, j) : A \rightarrow Z^+$ denotes the number of facility connections on hub arcs (i, j) , $a(j) = m$ if node $j \in N$ is assigned to hub node $m \in \Omega$, and $M : r \rightarrow \theta$ indicates the existence of a direct link between non-hub nodes. After determining location, allocation decisions, and direct services arcs, the number of facility links with modular capacity H on access arc, and the number of facility links with modular capacity R on direct connections can be estimated as $v_{ik}^1 = \lceil \frac{(O_i - \sum_{j \in N} W_{ij} M_{ij})}{H} \rceil$, $v_{ik}^2 = \lceil \frac{(D_i - \sum_{j \in N} M_{ij} W_{ij})}{H} \rceil$, and $A_{ij} = \lceil \frac{W_{ij} M_{ij}}{R} \rceil$.

3.1. Initial solution

GRASP is a problem-solving technique used to optimize combinatorial problems, as described in Festa and Resende [52]. The method involves a construction phase and a local search phase. To apply this approach to the MHLPCD, we first randomly select a subset of candidate nodes to serve as hubs. Then, we allocate the remaining nodes to the nearest open hub and activate a set of direct connection arcs between non-hub nodes. We also activate a set of hub arcs associated with open hub nodes to transmit demands between O/D nodes in the most cost-effective manner. In the local search phase, we systematically explore a set of neighborhoods using the VNS approach to enhance the initial solution.

The construction phase consists of four steps: selection of hub sites, assigning nodes to open hubs, activating a set of hub arcs, and selecting a set of direct links. We represent a partial solution as $S = (\Omega, E, a, M)$, where $\Omega(i) = NULL$, $E(e) = NULL$, $a(j) = NULL$, and $M(k) = NULL$. For each potential hub location $i \in N \setminus \Omega$, we compute a greedy function π_i based on outgoing flow O_i , destination flow D_i , and the cost of locating a hub facility at i . The greedy function is computed as:

$$\pi_i = \frac{O_i + D_i}{f_i} \quad (19)$$

To achieve a balance between quality and diversity, we employ a partially randomized greedy procedure. In each iteration, we randomly select a hub element from the Restricted Candidate List (RCL). The RCL is updated during each iteration and contains the best candidate nodes based on the greedy function, sorted in descending order. We set $\pi_{min} = \min\{\pi_i : i \in N\}$ and $\pi_{max} = \max\{\pi_i : i \in N\}$, and define the RCL as $i : \pi_i < \pi_{min} + \alpha(\pi_{max} - \pi_{min})$, where $0 \leq \alpha \leq 1$.

Once a hub is located at a candidate node i , we modify the existing solution to allocate demand more effectively and identify direct connections between non-hub nodes. We begin by assigning each demand node to the nearest hub, ensuring that all demands are routed through the hub network.

Next, we identify direct connection links between non-hub nodes. To do this, we order the flow W_{ij} between every pair of non-hub nodes in decreasing order and construct the set M step by step. We add the direct link with the highest W_{ij} first and evaluate the objective function. If adding the new direct link improves the objective function, the arc is

permanently included in M . Otherwise, we discard the candidate and move on to the next one.

This iterative process allows us to systematically explore the solution space and improve the solution by identifying direct connections between non-hub nodes. By incorporating these direct links, we can reduce costs and enhance the efficiency of the transportation network.

Activating facility links in the hub-level network involves solving the computationally challenging network loading problem. To tackle this challenge, we propose a greedy heuristic that efficiently finds a solution.

Our heuristic algorithm begins with an empty problem graph and progressively adds edges with capacities that allow for the simultaneous transmission of all O/D demands at the lowest cost. The algorithm then iteratively searches for the shortest paths with enough arc capacities to route the flow within the current graph. We use the Floyd-Warshall algorithm [53] to compute the shortest path between hub nodes.

At first, we calculate the minimum number of facility links with capacity B required to directly send flows between each pair of hubs $k \in \Omega$ and $m \in \Omega$. This calculation is crucial in determining the total cost of activating the facility links. The minimum number of facility links with capacity B required for direct flow between each pair of hubs $k \in \Omega$ and $m \in \Omega$ is computed as:

$$Y_{km} = \lfloor \frac{\sum_{i \in N} \sum_{j \in N} W_{ij} z_{ik} z_{jm}}{B} \rfloor : a(i) = k, a(j) = m \text{ and } k \neq m, \quad (20)$$

To efficiently activate the facility links in the hub-level network and achieve the desired flow routing at minimum cost, we employ a computationally efficient algorithm suitable for large-scale transportation networks. The algorithm routes flow and updates network arcs with the necessary capacities according to the following steps:

1. For each pair of hubs $k \in \Omega$ and $m \in \Omega$ where $Y_{km} \geq 1$, we implement Y_{km} facility links with capacity B on the hub arc (k, m) and adjust the remaining unused flow as follows:

$$unsend\ flow_{ij} = \sum_{i \in N} \sum_{j \in N} W_{ij} z_{ik} z_{jm} - B Y_{km} \quad (21)$$

2. Sort the unsend flow between each pair of hubs k and m in decreasing order, and then verify one of the following cases:

Case 1: The shortest path has enough residual capacity to route the largest unsend flow. In this case, we only update the unused capacity and unsend flow then go to Step2.

Case 2: On the shortest path, the residual capacity is insufficient. To ensure that it has enough capacity to route the unsend flow, we add facility links with capacity B to the shortest route and make the necessary adjustments then go to Step2.

To ensure the effectiveness and optimality of our solution in terms of minimum cost while satisfying flow requirements, we incorporate a simple local search during the flow routing phase.

The local search algorithm focuses on the current set of hub arcs and their modular capacities, utilizing two strategies: opening/closing hub arcs and closing hub arcs. By employing the best improvement strategy, we consistently move towards better solutions.

The opening/closing hub arcs strategy involves closing a hub arc and opening a new one on a different path, enabling the rerouting of flow through the network. This strategy helps us discover improved solutions that meet flow requirements at a lower cost. The rerouted flow is directed through the new hub arc with the lower cost.

Alternatively, the close hub arc strategy involves closing a hub arc and rerouting the flow through the hub network. This strategy optimizes the solution and reduces the overall transportation network cost. The rerouted flow is then directed through the remaining hub arcs with the lowest cost.

The local search algorithm is repeated until no further improvement can be achieved, indicating that the best possible solution for the given transportation network has been reached. This approach enables us to

efficiently activate facility links on the hub-level network and achieve the desired flow routing at the minimum cost.

Algorithm 1 outlines the constructive phase, providing an efficient and effective method for activating facility links on the hub-level network and achieving the desired flow routing with minimum cost.

Algorithm 1: Constructive Phase

```

 $H \leftarrow \phi, t \leftarrow 1, N^0 \leftarrow N, \min \leftarrow \infty$ 
while ( $\text{totalcost} < \min$ ) do
    Evaluate  $\pi_i^t$  for all  $i \in N^t$ 
     $RCL = \{i : \pi_i^t \leq \pi_{\min}^t + \alpha(\pi_{\max}^t - \pi_{\min}^t)\}$ 
    Select randomly  $i^* \in RCL$ 
     $H \leftarrow H \cup i^*$ 
     $N^t \leftarrow N^{t-1} \setminus \{i^*\}$ 
    Assign each node  $j \in N^t$  to its closest hub in  $H$ 
    add direct services
    installing capacities on hub arcs
    evaluate total cost
     $t \leftarrow t + 1$ 
end while

```

3.2. Variable neighborhood search

The objective of the local search phase is to improve the initial solutions obtained during the construction phase. By utilizing our implementation of the VNS algorithm, we explore four neighborhood types for the MHLPCD, with traditional shift-and-swap neighborhoods being the first type. The former examines all solutions that differ from the current solution by reassigning a single node, while the latter aims to enhance the existing solution by exchanging the assignments of two nodes. Let $S = (\Omega, E, a, M)$, is the current solution then

$$\aleph_1(s) = \{\bar{s} = (\Omega, E, \bar{a}, M) : \exists! j \in N, \bar{a}(j) \neq a(j)\}$$

and,

$$\aleph_2(s) = \{\bar{s} = (\Omega, E, \bar{a}, M) : \exists!(i_1, j_1), \bar{j}_1 = a(j_2), \bar{j}_2 = a(j_1)\}$$

The shift neighborhood, denoted as $\aleph_1(s)$, is the set of all solutions generated by reassigning a single node from the current solution. It involves systematically examining all pairs (i, j) , where $a(j) \neq i$, indicating that node j is not currently assigned to node i . This exploration allows for a comprehensive evaluation of the effects that changing the assignment of a single node has on the resulting solutions and their overall quality.

The swap neighborhood, denoted as $\aleph_2(s)$, is the set of all solutions obtained by exchanging the assignments of two nodes from the current solution. It systematically explores the effects of swapping the assignments of all possible pairs of nodes in the solution. By considering these pairwise exchanges, the swap neighborhood enables the exploration of alternative solutions that have the potential to yield improved overall outcomes.

These neighborhood types, $\aleph_1(s)$ and $\aleph_2(s)$, provide distinct avenues for searching and improving the current solution during the local search phase. By systematically examining single node reassignments and pairwise node swaps, we can effectively explore a wide range of potential solutions and guide the search towards better solutions for the MHLPCD problem.

The second type of neighborhood structure introduces modifications to the set of open hubs in the solution. It encompasses three distinct neighborhoods, namely, opening a new hub, opening and closing hubs, and closing open hubs. Each neighborhood focuses on specific changes to the hub configuration.

The first neighborhood of the second type involves exploring a subset of feasible solutions obtained from the current solution by introducing a new hub facility. This exploration includes reassigning non-hub nodes to their closest open hub. Let us consider a current

solution represented by $S = (\Omega, E, a, M)$, where Ω represents the set of existing open hub nodes. Additionally, let $i \in N \setminus \Omega$ denote the candidate nodes that have the potential to become new hub locations. Formally, the neighborhood structure can be defined as follows:

$$\aleph_3(s) = \{\bar{s} = (\bar{\Omega}, \bar{E}, \bar{a}, M) : \bar{s} = \Omega \cup i, i \in N \setminus \Omega\}$$

To explore the $\aleph_3(s)$ neighborhood, we adopt a systematic approach that involves considering all nodes i from the set of nodes N that are not currently assigned as open hubs (i.e., $i \in N \setminus \Omega$). This neighborhood is specifically designed to investigate potential modifications to the existing set of open hubs by evaluating the consequences of reassigning these candidate nodes as open hubs. By taking into account all nodes outside the current set of open hubs, we can thoroughly explore a variety of configurations and assess their impact on the overall solution. This exploration enables us to analyze different possibilities for selecting new open hubs, which ultimately has the potential to enhance the quality of the solution.

In the context of the second neighborhood structure within the second type, our objective is to modify the existing set of open hub facilities. This is achieved by randomly selecting and closing an open hub, followed by the reassignment of non-hub nodes to their nearest open hub. We denote the current solution as $S = (\Omega, E, a, M)$, where Ω represents the set of open hub nodes and $m \in \Omega$ represents the candidate hub for closure. Our primary focus is to thoroughly evaluate the consequences of closing a hub and reallocating its associated nodes to the nearest open hub. Through this evaluation, our aim is to analyze the resulting effects on the overall solution and determine any potential enhancements that can be made. Formally, the neighborhood can be defined as follows:

$$\aleph_4(s) = \{\bar{s} = (\bar{\Omega}, \bar{E}, \bar{a}, M) : \bar{s} = \bar{\Omega} \setminus \{m\}, m \in \bar{s}\} \quad (22)$$

The final neighborhood structure of the second type involves systematically exchanging an open hub with a closed hub and reassigning all non-hub nodes to their nearest open hub. To explore this neighborhood, we consider all hub nodes in Ω as potential candidates for closure, and all nodes $i \in N \setminus \Omega$ as potential locations for new hubs. Let $S = (\Omega, E, a, M)$ represent the current solution. Within this structure, we have $m \in \Omega$ as the candidate hub nodes eligible for closure, and $i \in N \setminus \Omega$ as the candidate locations for new hubs. That is, the neighborhood can be defined as follows:

$$\aleph_5(s) = \{\bar{s} = (\bar{\Omega}, \bar{E}, \bar{a}, M) : \bar{s} = \bar{\Omega} \setminus \{m\} \cup \{i\}, m \in \bar{s}, i \in N \setminus \bar{\Omega}\} \quad (23)$$

To explore the $\aleph_5(s)$ neighborhood, we conduct a systematic analysis of all nodes i that are not currently open hubs (i.e., $i \in N \setminus \Omega$) and all nodes m that are currently open hubs (i.e., $m \in \Omega$). This analysis involves thoroughly examining the effects of potential hub closures and reassignments on the overall solution. By considering all nodes outside the current set of open hubs and all nodes within the set of open hubs, we ensure a comprehensive exploration of alternative configurations and their implications. This allows us to evaluate the potential benefits of hub exchanges and node reassignments and assess their impact on the quality of the solution. It is important to highlight that at each step of this neighborhood exploration, the network loading problem must be evaluated in order to assess the effects resulting from the hub exchanges and node reassignments.

The final neighborhood structure type pertains to the manipulation of the existing set of direct links within the solution. It encompasses three distinct neighborhoods: adding a new direct arc, removing an existing direct arc, and adding a new direct arc while simultaneously removing an existing one.

The neighborhood called 'add a new direct arc,' represented as $\aleph_6(s)$, is dedicated to exploring feasible solutions obtained by introducing a new direct arc between two nodes. This process involves considering all potential pairs of nodes that are currently not connected by a direct link. By adding such a link to the existing solution, we can assess its impact on the overall quality of the solution. Let the

current solution be denoted as $S = (\Omega, E, a, M)$, then we can define the neighborhood as follows:

$$\mathcal{N}_6(s) = \{\bar{s} = (\Omega, \bar{E}, \bar{a}, \bar{M}) : \bar{s} = \bar{M} \cup (i, j), (i, j) \notin s\} \quad (24)$$

By systematically exploring solutions within this neighborhood, we can evaluate the potential benefits of introducing new direct arcs and their impact on improving the overall quality of the solution.

The neighborhood named 'remove an existing direct arc' focuses on investigating the effects of removing a direct arc from the current solution. It involves a systematic analysis of all possible direct arcs within the solution, evaluating the resulting solutions obtained by removing each arc individually. This analysis enables us to assess the consequences of link removal on the overall quality of the solution. Formally, the neighborhood structure can be defined as follows:

$$\mathcal{N}_7(s) = \{\bar{s} = (\Omega, \bar{E}, \bar{a}, \bar{M}) : \bar{s} = \bar{M} \setminus (i, j), (i, j) \in s\} \quad (25)$$

The neighborhood called 'add a new direct arc while removing an existing one' combines the previous two neighborhoods by simultaneously adding a new direct arc between two nodes and removing an existing direct arc. This exploration involves considering potential pairs of nodes that are not currently connected by a direct arc, selecting an existing direct arc to be removed, and evaluating the resulting solution obtained by simultaneously adding the new direct arc and removing the selected one. In other words

$$\begin{aligned} \mathcal{N}_8(s) &= \{\bar{s} = (\Omega, \bar{E}, \bar{a}, \bar{M}) : \bar{s} \\ &= \bar{M} \setminus \{(i_1, j_1)\} \cup \{(i_2, j_2)\}, (i_1, j_1) \in s, (i_2, j_2) \notin s\} \end{aligned} \quad (26)$$

Through the systematic exploration of all types of neighborhoods, our main objective is to enhance the current solution by carefully manipulating its configuration. By thoroughly evaluating the effects of these manipulations, our aim is to identify alternative solution configurations that have the potential to significantly improve the overall quality of the solution. It is important to highlight that during the exploration of each neighborhood, the Network Loading Problem (NLP) must be solved to accurately assess the impact of each change on the solution's quality.

In our VNS approach, we employ a best improvement strategy, which involves iteratively selecting the most advantageous changes within each neighborhood to enhance the solution. By prioritizing changes that yield the greatest improvement in solution quality, we effectively guide the search process towards better solutions. This strategy enables us to systematically explore the search space and progressively converge towards an improved solution configuration.

It is worth emphasizing that the combination of comprehensive neighborhood exploration, careful evaluation of changes, and the best improvement strategy are key components of our approach. Through these principles, we aim to achieve significant enhancements in both the quality and performance of the solution.

4. Results

In this section, we compare and evaluate the performance of the formulation and VNS algorithm using computational experiments and investigate the effect of changing capacities on hub arcs and access arcs through sensitivity analysis. The formulation and solution algorithm were implemented in C++ and executed on an HP workstation equipped with an Intel Processor Intel(R) Core(TM) i7-1065G7 @ 1.30 GHz, 1501 Mhz, and 16 GB of RAM, running on a Windows 11 environment. The MIP formulation was solved using the Concert technology in CPLEX 12.5.2.

In designing our experiment, we utilized the well-known Australian post (AP) dataset, which is available from the OR library (<http://people.brunel.ac.uk/~mastijb/jeb/orlib/phubinfo.html>). The AP dataset includes postal traffic and Euclidean distances between 200 Australian districts. For our experiments, we selected problems with

Table 1

Input parameters.

Parameters	$\alpha = 0.2$			$\alpha = 0.4$			$\alpha = 0.63$		
name	L.1	L.2	L.3	L.4	L.5	L.6	L.7	L.8	L.9
B	750	750	600	400	650	500	200	300	400
H	100	100	100	100	150	100	100	150	200
b	600	450	600	800	650	600	500	500	500
p	400	300	500	500	350	300	400	400	400

node sizes of $|N| = \{10, 20, 25, 40\}$. Following the approach in Tanash et al. [18], Tanash [28], we generated nine configurations for each problem size, resulting in an equivalent discount factor α of 0.2, 0.4, and 0.63. Each instance consisted of a hub facility link capacity B with a variable cost b , and a facility link capacity on access arcs H with a variable cost p . In our experiments, we assumed the same facility link capacity for direct arcs as for access arcs. Table 1 illustrates the parameter settings for each of the nine generated configurations.

4.1. Computational experiments

In our computational experiments, we conducted a comprehensive assessment of our mathematical formulation and the VNS algorithm's performance on small-sized instances with 10 and 20 nodes. Initially, we solved the MIP model using CPLEX's default settings and compared the obtained results with those from the VNS algorithm. The detailed experimental results are presented in Table 2, which includes information on the number of nodes ($|N|$) and the instance name (Name) in the first two columns. The CPLEX section provides various metrics, such as the linear programming relaxation gap (%LP), the final gap (%GAP) at termination, the CPU time in seconds (CPU), the number of explored nodes in the enumeration tree (B&B), and the percentage deviation between the best-known solution obtained from either CPLEX or VNS and the upper bound. The VNS section showcases the results of our meta-heuristic approach described in Section 3, with %dev1 representing the deviation between the VNS solution and the best-known solution. The average CPU time in seconds is reported under the CPU column. To evaluate the quality and robustness of our approach, we executed the algorithm 15 times for each instance, selecting the best solution among these runs, enabling us to determine the current deviation (%dev1) from the optimal or best-known solution. Additionally, we introduce the column labeled %dev2 in the COMP section, which quantifies the degree of improvement in the solution by comparing the best-known solution of MHLPC to that of MHLPCD.

The optimal solution value, known as BESTSOLN, is obtained from either CPLEX or VNS. The LP relaxation value is represented by LP. The %LP gap is calculated as $(BESTSOLN - LP)/BESTSOLN \times 100$. The final percent gap, denoted as %GAP, is computed as $(BESTSOLN - LB)/BESTSOLN \times 100\%$. The percent deviation, or %dev, is calculated using the formula $(BESTSOLN - UB)/BESTSOLN \times 100\%$. Similarly, %dev1 is computed as $(BESTSOLN - VNSsolution)/BESTSOLN \times 100\%$.

Moreover, we have evaluated the impact of adding direct connections by comparing the MHLPCD to the MHLPC presented in Tanash et al. [18]. To measure the difference between the solutions, we have used the percent deviation, denoted as %dev2, which is calculated using the formula: $(MHLPCsolution - BESTSOLN)/MHLPCsolution \times 100\%$. Throughout the experiment, the maximum amount of CPU time is set to two hours. Instances that fail to solve to optimally within this time limit are marked with the label "Limit".

As shown in Table 2, CPLEX was able to optimally solve 13 out of 18 tested instances for both 10 and 20 nodes. For the unsolved instances, it provided optimality gaps ranging from 2.2% to 5.54%. Specifically, CPLEX solved all 10-node instances in an average time of 41.96 s. However, for the 20-node instances, CPLEX only achieved optimal solutions for 4 out of 9 instances, with optimality gaps ranging

Table 2
Comparison between CPLEX and VNS algorithm on a small size instances.

Instance		CPLEX					VNS		COMP
N	Name	%GAP	%LP	CPU	B&B	%dev	CPU	%dev1	%dev2
10	L1	0.00	3.72	23.66	1591	0.00	8.52	0.00	1.52
	L2	0.00	3.61	11.59	1244	0.00	11.87	0.00	1.39
	L3	0.00	4.08	141.52	6712	0.00	28.38	0.23	0.00
	L4	0.00	4.93	83.41	2980	0.00	30.56	0.00	3.49
	L5	0.00	3.85	8.09	684	0.00	10.26	0.00	2.74
	L6	0.00	5.16	42.24	2824	0.00	10.45	0.89	3.85
	L7	0.00	5.89	59.95	4303	0.00	18.63	0.33	2.94
	L8	0.00	4.00	5.84	900	0.00	5.68	0.90	5.96
	L9	0.00	0.48	1.31	13	0.00	3.27	0.00	13.42
AVG		0.00	3.97	41.96	2361.22	0.00	14.18	0.26	3.92
20	L1	0.00	4.47	5179.76	9595	0.00	389.30	0.00	0.93
	L2	0.00	4.25	2308.03	3918	0.00	188.40	0.25	0.85
	L3	2.21	5.34	Limit	13608	0.00	550.97	0.00	0.00
	L4	0.00	4.80	6923.66	14502	0.00	305.39	0.00	0.87
	L5	4.63	6.62	Limit	3957	0.00	8.32	0.31	1.24
	L6	3.21	5.59	Limit	3633	0.00	13.29	0.00	2.37
	L7	2.23	4.74	Limit	4769	0.00	311.25	0.24	1.99
	L8	2.47	7.08	Limit	13059	0.00	9.37	0.00	2.77
	L9	0.00	3.41	311.34	1232	0.00	3.26	0.91	1.62
AVG		1.64	5.15	3680.70	7585.89	0.00	197.73	0.19	1.40

from 2.21% to 5.54% (average of 1.64%) for the remaining unsolved instances, and an average CPU time of 36080.89 s.

The proposed VNS algorithm demonstrated its effectiveness by achieving optimal solutions for 5 out of 9 instances with 10 nodes. Additionally, it provided near-optimal solutions with small deviations ranging from 0.23% to 0.9% (average of 0.26%) while maintaining an average CPU time of 14.18 s for the remaining instances. Similarly, for the 20-node instances, the VNS algorithm achieved optimal solutions for 5 instances and successfully solved the remaining 4 instances with a percent deviation of less than 1% (average of 0.19%). These accomplishments were achieved within an average CPU time of 197.73 s. These notable results emphasize the efficiency of the algorithm and its ability to generate high-quality solutions within a reasonable computational time for the given problem instances.

Analyzing the “%dev2” column reveals an interesting observation: when direct connections are allowed, the total network costs decrease. This reduction can be primarily attributed to two factors: locating fewer hubs and enabling more direct connections between nodes. By minimizing the number of hubs, the overall infrastructure and operational costs associated with maintaining and operating them are reduced. Additionally, the inclusion of direct connections allows for more efficient point-to-point transportation, eliminating the need for intermediate hubs and reducing the overall travel distance and associated costs.

The extent of cost reduction varies across different instances of the problem. In the case of instance L9, the maximum reduction in total network costs is observed, with a significant decrease of 13.42% compared to the MHLPC solution without direct connections. This substantial reduction highlights the significant impact that incorporating direct connections can have on improving the cost-efficiency of the network.

These findings emphasize the potential advantages of integrating direct connections into hub location problems. By strategically optimizing the placement of hubs and leveraging direct connections, significant cost savings can be achieved within transportation networks. This information holds valuable insights for decision-makers and practitioners in the field, as it underscores the benefits of incorporating direct connections when designing and optimizing hub-based transportation systems.

Table 3 provides a comprehensive overview of the performance results obtained by our VNS algorithm on larger instances, specifically those with $|N| = 25$ and 40. The results demonstrate that the VNS

algorithm outperforms CPLEX in terms of both solution quality and CPU time. Within the given time limit, CPLEX was unable to solve any of the 25-node instances optimally. However, it was able to provide optimality gaps ranging from 3.64% to 12.60%, with an average of 8.41%. Furthermore, CPLEX failed to provide valid upper or lower bounds for any of the 40-node instances. In contrast, the VNS algorithm consistently produced superior results across all 9 tested instances of the 25-node problem. It is worth noting that the average running time of the VNS algorithm on the 25-node and 40-node instances was 482.14 s and 3282.63 s, respectively, while CPLEX required an average running time of 7200 s.

The MHLPCD exhibited a substantial enhancement in the overall cost when compared to the MHLPC without direct connections. This notable improvement was achieved by effectively utilizing a number of direct connections, which directly contributed to the establishment of more efficient transportation routes. The realized enhancements in total network costs ranged from 0.06% to 2.01%, emphasizing the significant cost savings that can be achieved for real-world transportation systems. These findings strongly emphasize the crucial significance of incorporating direct connections in hub-based transportation systems to enhance both cost-efficiency and network performance.

4.2. Sensitivity analysis

In this section, we conduct a comparison between the hub network structures obtained by solving the modular hub location problem (MHLPC) as presented in Tanash et al. [18] and the structures obtained by our MHLPCD problem when the target discount factor varies. Figs. 1 through 3 illustrate the optimal hub networks for 10-node instances extracted from the AP dataset, with target discount factors of 0.2, 0.4, and 0.6 achieved by adjusting capacity and variable cost. These figures also provide information on the number of facility links utilized on each hub arc and the average percentage of hub arc utilization.

Figs. 1(a) and 1(b) specifically depict the optimal network structure for a target discount factor of 0.2. As anticipated, when the target discount factor is set to 0.2, it becomes more cost-effective to route flow through a hub-and-spoke network, enabling transportation costs to leverage the aggregation of flow at hub facilities and capitalize on economies of scale. The number and placement of hubs, as well as the utilization of hub arcs to connect the backbone network, remain

Table 3
Comparison between CPLEX and VNS algorithm on a larger size instance.

Instance		CPLEX					VNS		COMP
N	name	%GAP	%LP	CPU	B&B	%dev	CPU	%dev1	%dev2
25	L.1	10.89	11.14	Limit	484	1.12	748.91	0.00	0.06
	L.2	8.04	8.35	Limit	640	1.81	649.58	0.00	1.65
	L.3	9.78	9.82	Limit	301	1.21	1643.94	0.00	0.38
	L.5	12.6	12.71	Limit	247	3.21	287.20	0.00	0.07
	L.6	5.55	5.93	Limit	688	0.16	56.99	0.00	0.59
	L.8	8.23	8.48	Limit	495	0.00	574.30	0.00	0.95
	L.9	7.89	8.81	Limit	1136	0.14	238.87	0.00	0.51
	L.10	5.79	5.92	Limit	644	0.83	116.26	0.00	2.01
	L.11	3.64	4.54	Limit	649	0.00	23.18	0.00	0.00
	AVG		8.41	Limit		0.94	482.14	0.00	0.69
40	L.1	**	**	**	**	**	3082.47	0.00	0.00
	L.2	**	**	**	**	**	16252.89	0.00	0.14
	L.3	**	**	**	**	**	218.16	0.00	0.00
	L.4	**	**	**	**	**	158.68	0.00	0.00
	L.5	**	**	**	**	**	2865.36	0.00	0.00
	L.6	**	**	**	**	**	2653.99	0.00	0.00
	L.7	**	**	**	**	**	856.36	0.00	0.27
	L.8	**	**	**	**	**	998.15	0.00	0.45
	L.9	**	**	**	**	**	2457.58	0.00	0.00
	AVG	**	**	Limit	**	**	3282.63	0.00	0.10

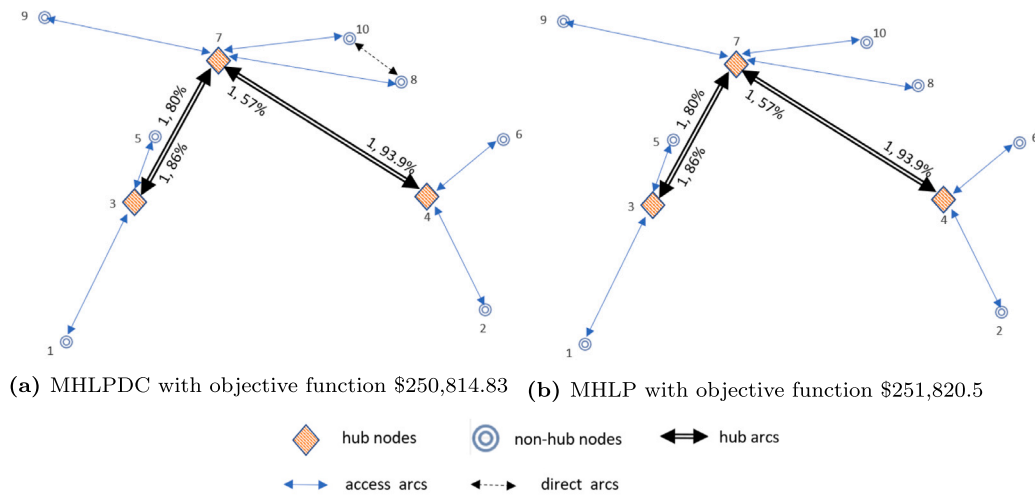


Fig. 1. Hub Networks structure with $\alpha = 0.2$ ($B = 750, H = 100, b = 450, p = 300$).

consistent between both models. However, the MHLPDC introduces two direct arcs between nodes 10 and 8, resulting in a 0.4 percent reduction in total network costs.

Figs. 2(a) and 2(b) illustrate the distinction between the optimal solution structures of the MHLPDC and MHLP when the target discount factor is reduced to 0.4. Fig. 2(a) demonstrates that the MHLPDC reduces total network costs by 3.7% in comparison to the MHLP by utilizing more direct connections and locating fewer hub facilities. The MHLP model places three hub facilities at locations 3, 4, and 7, connecting them with four hub arcs. In contrast, the MHLPDC has two hubs at locations 4 and 7, as well as six direct connections. With fewer hub facilities and more direct connections, the MHLPDC is able to maximize inter-hub arc utilization and attain the desired discount factor.

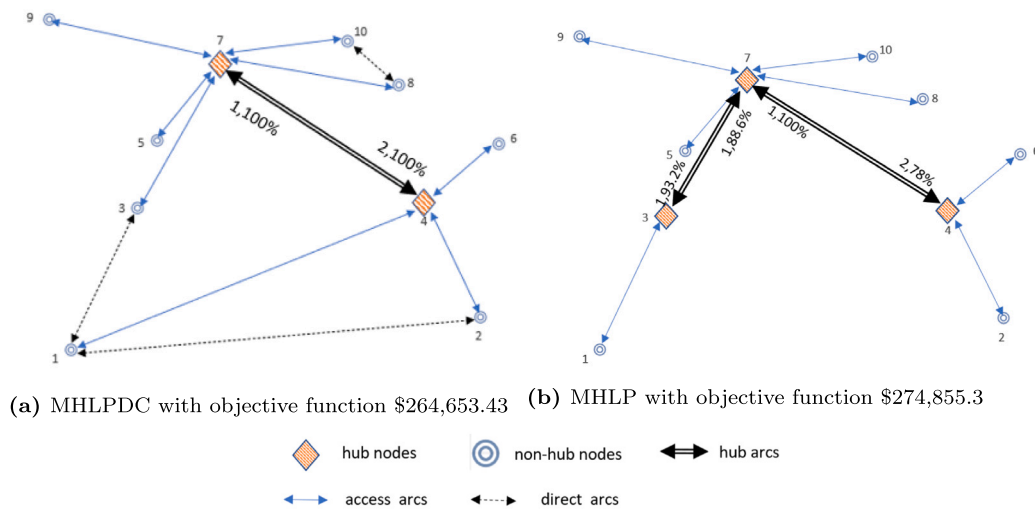
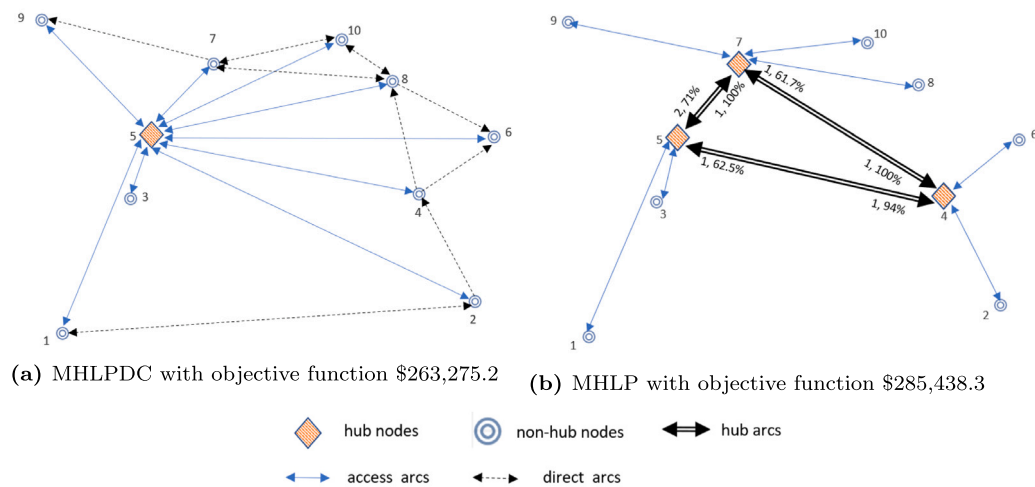
Figs. 3(a) and 3(b) illustrate the network structure after a further decrease in the target discount factor. When the discount factor is 0.6, as shown in Fig. 3(a), the MHLPDC locates a single hub facility at location 5 and permits more direct connections between non-hub nodes. The MHLP, on the other hand, has three hubs at locations 4, 5, and 7 and a network that is fully interconnected. In addition, the use of direct connections results in a 7.76 percent decrease in total network costs when compared to MHLP.

5. Conclusions and future research directions

This paper presents a mathematical formulation for addressing the incomplete single assignment hub location problem with modular capacities and direct connections. This problem extends the classic hub location problem while relaxing three critical assumptions. The first assumption considers economies of scale by accurately modeling the flow dependence of transportation costs using modular arc costs. The second assumption incorporates hub network design into the decision-making process, without assuming a specific topological structure. The final assumption allows for direct communication between O/D nodes, enhancing the model's realism.

To handle the complexity of the MHLPDC, a metaheuristic algorithm is proposed in this paper, which combines greedy randomized adaptive processes and variable neighborhood search techniques. The algorithm consists of two main phases. In the first phase, a greedy randomized adaptive procedure generates a set of initial solutions. The second phase utilizes variable neighborhood search techniques to refine these solutions.

To solve the network loading problem, which is an NP-hard optimization problem, the algorithm employs a greedy heuristic and a local search phase in each iteration of the metaheuristic algorithm. This

Fig. 2. Hub Networks structure with $\alpha = 0.4$ ($B = 500, H = 100, b = 600, p = 300$).Fig. 3. Hub Networks structure with $\alpha = 0.6$ ($B = 400, H = 200, b = 600, p = 500$).

phase involves selecting appropriate hub and arc capacities for the hubs and their corresponding arcs based on the current solution.

Computational experiments conducted on benchmark instances with 40 nodes demonstrate the efficacy of the proposed algorithm in generating high-quality solutions for the MHLPDC. However, the study also highlights several avenues for future research, including developing exact solution algorithms to optimally solve larger instances, improving the performance of the metaheuristic algorithm, and incorporating more realistic travel time constraints between O/D nodes.

Overall, this paper makes a significant contribution to the field of transportation planning by extending the classic hub location problem and presenting a novel metaheuristic algorithm for solving the MHLPDC. It provides a valuable framework for modeling and solving hub location problems with modular capacities and direct connections, which can find practical applications in various domains, such as parcel delivery systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] Alumur S, Kara BY. Network hub location problems: The state of the art. *European J Oper Res* 2008;190(1):1–21.
- [2] Campbell JF, O’Kelly ME. Twenty-five years of hub location research. *Transp Sci* 2012;46(2):153–69.
- [3] Farahani RZ, Hekmatfar M, Arabani AB, Nikbaksh E. Hub location problems: A review of models, classification, solution techniques, and applications. *Comput Ind Eng* 2013;64(4):1096–109.
- [4] Contreras I, O’Kelly M. Hub location problems. In: *Location science*. Springer; 2019, p. 327–63.
- [5] Alumur SA, Campbell JF, Contreras I, Kara BY, Marianov V, O’Kelly ME. Perspectives on modeling hub location problems. *European J Oper Res* 2021;291(1):1–17.
- [6] Wandelt S, Dai W, Zhang J, Sun X. Toward a reference experimental benchmark for solving hub location problems. *Transp Sci* 2022;56(2):543–64.
- [7] O’Kelly ME. The location of interacting hub facilities. *Transp Sci* 1986;20(2):92–106.
- [8] Campbell JF. Integer programming formulations of discrete hub location problems. *European J Oper Res* 1994;72(2):387–405.
- [9] Kara BY, Tansel BC. On the single-assignment p-hub center problem. *European J Oper Res* 2000;125(3):648–55.

- [10] Labbé M, Laporte G, Martin IR, González JJS. Locating median cycles in networks. *European J Oper Res* 2005;160(2):457–70.
- [11] Villa D, Montoya A, Herrera AM. The electric riverboat charging station location problem. *J Adv Transp* 2020;2020.
- [12] Xu W, Huang J, Qiu Y. Study on the optimization of hub-and-spoke logistics network regarding traffic congestion. *J Adv Transp* 2021;2021.
- [13] Zhou J, Zhao Y, Niu X. Two-phase optimization models for liner shipping network based on hub ports cooperation: from the perspective of supply-side reform in China. *J Adv Transp* 2021;2021.
- [14] Ernst AT, Krishnamoorthy M. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European J Oper Res* 1998;104(1):100–12.
- [15] Ernst AT, Hamacher H, Jiang H, Krishnamoorthy M, Woeginger G. Uncapacitated single and multiple allocation p-hub center problems. *Comput Oper Res* 2009;36(7):2230–41.
- [16] Contreras I, Díaz JA, Fernández E. Lagrangean relaxation for the capacitated hub location problem with single assignment. *OR spectrum* 2009;31(3):483–505.
- [17] Contreras I, Cordeau J-F, Laporte G. Exact solution of large-scale hub location problems with multiple capacity levels. *Transp Sci* 2012;46(4):439–59.
- [18] Tanash M, Contreras I, Vidyarthi N. An exact algorithm for the modular hub location problem with single assignments. *Comput Oper Res* 2017;85:32–44.
- [19] O'Kelly ME, Bryan D. Hub location with flow economies of scale. *Transp Res Part B: Methodol* 1998;32(8):605–16.
- [20] Bryan D. Extensions to the hub location problem: Formulations and numerical examples. *Geogr Anal* 1998;30(4):315–30.
- [21] Kliniewicz JG. Enumeration and search procedures for a hub location problem with economies of scale. *Ann Oper Res* 2002;110(1):107–22.
- [22] De Camargo RS, de Miranda Jr. G, Luna HPL. Benders decomposition for hub location problems with economies of scale. *Transp Sci* 2009;43(1):86–97.
- [23] Horner MW, O'Kelly ME. Embedding economies of scale concepts for hub network design. *J Transp Geogr* 2001;9(4):255–65.
- [24] Podnar H, Skorin-Kapov J, Skorin-Kapov D. Network cost minimization using threshold-based discounting. *European J Oper Res* 2002;137(2):371–86.
- [25] Racunica I, Wynter L. Optimal location of intermodal freight hubs. *Transp Res B* 2005;39(5):453–77.
- [26] Cunha CB, Silva MR. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *European J Oper Res* 2007;179(3):747–58.
- [27] Kimms A. Economies of scale in hub & spoke network design models: We have it all wrong. In: *Perspectives on operations research*. Springer; 2006, p. 293–317.
- [28] Tanash M. Hub network design and discrete location: Economies of scale, reliability and service level considerations (Ph.D. thesis), Concordia University; 2018.
- [29] O'Kelly ME, Campbell JF, De Camargo RS, de Miranda Jr. G. Multiple allocation hub location model with fixed arc costs. *Geogr Anal* 2015;47(1):73–96.
- [30] Fard MK, Alfandari L. Trade-offs between the stepwise cost function and its linear approximation for the modular hub location problem. *Comput Oper Res* 2019;104:358–74.
- [31] Yaman H. Star p-hub median problem with modular arc capacities. *Comput Oper Res* 2008;35(9):3009–19.
- [32] Lüer-Villagra A, Eiselt HA, Marianov V. A single allocation p-hub median problem with general piecewise-linear costs in arcs. *Comput Ind Eng* 2019;128:477–91.
- [33] Labbé M, Yaman H. Solving the hub location problem in a star-star network. *Networks: Int J* 2008;51(1):19–33.
- [34] Martins de Sá E, Contreras I, Cordeau JF, Saraiva De Camargo R, de Miranda G. The hub line location problem. *Transp Sci* 2015;49(3):500–18.
- [35] Contreras I, Fernández E, Marín A. The tree of hubs location problem. *European J Oper Res* 2010;202(2):390–400.
- [36] Contreras I, Tanash M, Vidyarthi N. Exact and heuristic approaches for the cycle hub location problem. *Ann Oper Res* 2017;258(2):655–77.
- [37] de Sá EM, Contreras I, Cordeau J-F. Exact and heuristic algorithms for the design of hub networks with multiple lines. *European J Oper Res* 2015;246(1):186–98.
- [38] Aykin T. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European J Oper Res* 1994;79(3):501–23.
- [39] Mahmutogullari Aİ, Kara BY. Hub location problem with allowed routing between nonhub nodes. *Geogr Anal* 2015;47(4):410–30.
- [40] De Camargo RS, de Miranda Jr. G, O'Kelly ME, Campbell JF. Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. *Appl Math Model* 2017;51:274–301.
- [41] Dai W, Zhang J, Sun X, Wandelt S. HUBBI: Iterative network design for incomplete hub location problems. *Comput Oper Res* 2019;104:394–414.
- [42] de Sá EM, Morabito R, De Camargo RS. Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Comput Oper Res* 2018;89:31–50.
- [43] Taherkhani G, Alumur SA. Profit maximizing hub location problems. *Omega* 2019;86:1–15.
- [44] Figueiredo RM, O'Kelly ME, Pizzolato ND. A two-stage hub location method for air transportation in Brazil. *Int Trans Oper Res* 2014;21(2):275–89.
- [45] Hoff A, Peiro J, Corberan A, Martí R. Heuristics for the capacitated modular hub location problem. *Comput Oper Res* 2017;86:94–109.
- [46] Osorio-Mora A, Núñez-Cerda F, Gatica G, Linfati R. Multimodal capacitated hub location problems with multi-commodities: An application in freight transport. *J Adv Transp* 2020;2020.
- [47] Shang X, Yang K, Jia B, Gao Z. The stochastic multi-modal hub location problem with direct link strategy and multiple capacity levels for cargo delivery systems. *Transportmetrica A: Transport Science* 2021;17(4):380–410.
- [48] Mahmoodjanloo M, Tavakkoli-Moghaddam R, Baboli A, Jamiri A. A multi-modal competitive hub location pricing problem with customer loyalty and elastic demand. *Comput Oper Res* 2020;123:105048.
- [49] Dai W, Wandelt S, Zhang J, Sun X. Capacitated air/rail hub location problem with uncertainty: A model, efficient solution algorithm, and case study. *IEEE Trans Intell Transp Syst* 2021;23(7):8451–66.
- [50] Mirzaghafour FS. Modular hub location problems (Ph.D. thesis), Concordia University; 2013.
- [51] Brimberg J. A variable neighborhood algorithm for solving the continuous location-allocation problem. *Stud Locat Anal* 1996;10:1–12.
- [52] Festa P, Resende MG. GRASP: basic components and enhancements. *Telecommun Syst* 2011;46(3):253–71.
- [53] Floyd RW. Algorithm 97: shortest path. *Commun ACM* 1962;5(6):345.