Computers & Operations Research 136 (2021) 105478

Contents lists available at ScienceDirect

Computers and Operations Research

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

Multiple allocation tree of hubs location problem for non-complete networks

Betül Kayışoğlu, İbrahim Akgün *

Abdullah Gül University, Faculty of Engineering, Department of Industrial Engineering, Kayseri, Turkey

ARTICLE INFO

Keywords:
Hub location problem
Multiple allocation
Tree of hubs location problem
Benders decomposition
Benders-type heuristic

ABSTRACT

We study the *Multiple Allocation Tree of Hubs Location Problem* where a tree topology is required among the hubs and transportation cost of sending flows between OD pairs is minimized. Unlike most studies in the literature that assume a complete network with costs satisfying the triangle inequality to formulate the problem, we define the problem on non-complete networks and develop a modeling approach that does not require any specific cost and network structure. The proposed approach may provide more flexibility in modeling several characteristics of real-life hub networks. Moreover, the approach may produce better solutions than the classical approach, which may result from the differences in the selected hubs, the flow routes between origin–destination points, and the assignment of non-hub nodes to hub nodes. We solve the proposed model using CPLEX-based branch-and-bound algorithm and Gurobi-based branch-and-bound algorithm with Norel heuristic and develop Benders decomposition-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We conduct computational experiments using problem instances defined on non-complete networks with up to 500 nodes. The results indicate that the Benders-type heuristics are especially effective in finding good feasible solutions for large instances.

1. Introduction

Hubs act as aggregation, distribution, switching, and sorting centers in telecommunication, transportation, and computer networks where commodities (e.g., data, packages, etc.) are sent between many origin–destination (OD) pairs. In these networks, instead of sending flows directly between each OD pair, the flows are sent through hub facilities in at most three movements: *collection* from the origin to a hub, *transfer* between hubs, and *distribution* from the last hub to the destination (transfer movement may be skipped for some OD pairs). The transfer of consolidated flow between hubs enables to capture the *economies of scale*. Moreover, advantages resulting from reducing setup costs, centralized commodity handling, and sorting operations are obtained.

A generic *Hub Location Problem* (HLP) is concerned with determining the locations of hubs, allocation of supply and demand points to hubs, and determining the routes between OD pairs such that total cost is minimized. The research on hub location addresses different types of problems, e.g., p-hub median, p-hub center, hub location problem with fixed costs, and hub covering. These problems are also categorized according to the allocation strategy as *single allocation problems*, i.e., all the incoming and outgoing traffic of each node is routed through a single hub, and *multiple allocation problems*, i.e., each origin (destination) node can be allocated to more than one hub to send (receive) flows. For a comprehensive review of the problems, see, for instance, Alumur and Kara (2008), Campbell and O'Kelly (2012), Farahani et al. (2013), Contreras and O'Kelly (2019), and Alumur (2019).

In this paper, we study the *Multiple Allocation Tree of Hubs Location Problem* (MATHLP) that imposes a *tree topology* requirement on the part of the network where *the transfer between hubs* is carried out, i.e., on the *backbone* or *hub-level network*, using multiple allocation strategy. As a motivating example, consider the problem of determining a public transportation network in a city where the nodes represent bus/rail stations and the arcs represent the roads/railways between bus/rail stations. The objective is to move the passengers from their origin stations to their destination stations such that the total transportation cost (e.g., travel time or operating cost) is minimized. There are different types of public transport networks, e.g., direct, trunk-and-feeder, radial, diameter, hybrid, etc. (Vuchic, 2007). In a trunk-and-feeder system, which is similar to the hub-and-spoke system, the demand on the feeder routes is served by small vehicles and combined on the trunk routes so that passengers from multiple feeder routes can all use a much larger

* Corresponding author.
E-mail address: ibrahim.akgun@agu.edu.tr (İ. Akgün).

https://doi.org/10.1016/j.cor.2021.105478
Received 26 May 2020; Received in revised form 30 May 2021; Accepted 11 July 2021
Available online 19 July 2021

Nomenclature

OD	Origin-destination
HLP	Hub location problem
MATHLP	Multiple Allocation Tree of Hubs Location Problem
RealN	Real-world network
MN	Modeled network
HN	Hub network
HLN	Hub-level network
AN	Access network
MATHLM	Multiple allocation tree of hubs location model
MP	Master problem at iteration h of BD algorithm
SP	Subproblem at iteration h of BD algorithm
Model SPD	Linear problem obtained by fixing integer variables in MATHLM at iteration h
Model SP	Subproblem obtained by taking the dual of SPD at iteration h
Model PRT	Model to find a pareto-optimal solution
Model MPM	Master problem with disaggregated cuts at iteration h
Model RelMP	Relaxed master problem with a single cut
Model MCMP	Relaxed master problem with multiple cuts
SPTree	Rooted spanning tree formulation
BDHEUR1	Benders-Type Heuristic 1 employing strong cut generation
BDHEUR2	Benders-Type Heuristic 2 employing strong cut generation and cut disaggregation

trunk vehicle. The operating costs of large vehicles per passenger are lower than those of smaller vehicles. The trunk network, i.e., the backbone network, has mostly a tree structure, e.g., a tram network and/or a road network with high-capacity buses operating with higher frequency along *rapid transit corridors* or paths. The stations on the backbone network that are incident to feeder lines are transfer points (hub nodes) where the passengers change line. In a trunk-and-feeder system, a demand point is served by a single line and most passengers need to make two transfers, which is not desired by the passengers. In this regard, a hybrid system where direct travel service between some OD pairs is made possible by allowing some bus lines to go beyond the trunk route as necessary. In such a case, these bus lines are required to pass through a transfer point on the backbone network for the passengers that need to change line. Moreover, more than one bus line may be planned to serve the same demand point (bus station) so that passengers can prefer the line closer to their destinations, i.e., multiple allocation. Most of the cities in Turkey operate such a hybrid public transport system. A tree-like backbone network is especially preferred in cities where the metro/tram system is still in its infancy. The metro/tram network is complemented by high-capacity buses operated on rapid transit corridors. In the context of public transportation, MATHLP allows us to determine the physical network including the tree-like trunk routes on which the flow of passengers is achieved with the minimum cost. The resulting network and passenger loads on the links may be used to determine bus lines and their frequencies in accordance with the public transport service planning process (e.g., Ceder, 2007).

Tree topology on the hub-level network has been used or suggested for applications in railway transportation, telecommunication networks, electricity and water distribution networks, and pipeline transportation, where the connectivity between hubs is required but the *setup costs* for inter-hub links are significant. For example, a backbone network with tree topology is required in designing the high-speed train network in Spain with the stations being hubs (Contreras et al., 2010); private data networks, metropolitan area networks, and community antenna television network systems in a hierarchical structure with concentrators (that aggregate and forward data packets) being hubs (Klincewicz, 1998); gas pipeline network with valve sets being hubs, which receive gas produced in wells through production pipes and transfer it to a station via gathering pipes (Zhou et al., 2019); electricity power distribution networks with distribution substations being hubs (Sabattin et al., 2018); urban and public transport network with transfer points between cities and towns being hubs (Zhong et al., 2018).

Most hub location problems are known to be NP-hard (e.g., Carello et al., 2004; Alumur and Kara, 2008; Contreras and O'Kelly, 2019). MATHLP is NP-hard as well. When the hub locations and the allocation of supply and demand points to hubs are fixed in MATHLP, the problem of finding a tree spanning the hub nodes is equivalent to the Optimum Communication Spanning Tree Problem, which is NP-hard (Johnson et al., 1978; Contreras et al, 2010).

We will use the network terminology and definitions given by Akgün and Tansel (2018) to further discussion about MATHLP. They define five different types of networks: (1) *Real-world network* (RealN): The physical network, e.g., road and rail networks, in which hub system will operate. (2) *Modeled network* (MN): The network used as an input in developing a model for the problem. MN is not necessarily the same as RealN but may be obtained from RealN through preprocessing. (3) *Hub network* (HN): The sub-network of MN that consists of the hub nodes, non-hub nodes, and the arcs on the service routes between OD pairs. (4) *Hub-level network* (HLN): The subnetwork of HN consisting of the *hub nodes* and the *hub arcs* connecting them. (5) *Access network* (AN): The sub-network of HN consisting of the hub nodes, non-hub nodes, and *access arcs* that connect non-hub origin and destination nodes to hub nodes.

Using the given terminology, MATHLP requires HLN to have a tree topology. Most studies in the literature impose a complete HLN topology but do not discuss whether this is a result of the nature of the application or the assumptions regarding the network or data structure. It occurs that there are three main assumptions in most HLP models (For a detailed discussion, see, e.g., Contreras and O'Kelly, 2019; Akgün and Tansel, 2018): (1) *MN is a complete network with arc distances satisfying the triangle inequality*, (2) *transportation costs on all hub arcs are discounted by a constant factor independent of the actual amount of flow on the arcs*, i.e., collection and distribution are more costly, and (3) *all flows are routed via a set of hubs*, i.e., no direct flows between non-hub nodes. A model with these three assumptions and a cost minimization objective and without any topological requirements produces a solution where the flows between an OD pair visit at most two hubs. In other words, a route between an OD pair in an HN consists of at most three arcs, namely, *collection* (*access*), *transfer* (*hub*), and *distribution* (*access*) arcs. Given non-zero flows between all OD pairs, the resulting HLN is a complete network, i.e., all hubs are fully interconnected by hub arcs.

Considering the complete HLN topology to be restrictive, several researchers have studied incomplete HLN topologies. Some of these studies (e.g., Alumur et al., 2009; Nickel et al., 2001; Mohri and Akbarzadeh, 2018; Calık et al., 2009; Alumur and Kara, 2009; Yoon and Current, 2008; and Alumur et al., 2012; Martins de Sá et al., 2018a; Martins de Sá et al., 2018b) require HLN only to be connected and do not impose any specific HLN topology. Some studies (e.g., Campbell et al., 2005a; Campbell et al., 2005b; Campbell, 2010) do not even require HLN to be connected. The studies that impose a particular HLN topology other than a tree structure are Labbé and Yaman (2008) and Yaman (2008) that study a star HLN, Martins de Sá et al. (2015a), Martins de Sá et al. (2015b) that address a line HLN, and Lee et al. (1993) and Contreras et al. (2017) that investigate a cycle HLN.

To our knowledge, all studies that require a tree HLN address the single allocation version of the problem. Contreras et al. (2010) propose a mixed integer programming (MIP) formulation and present several families of valid inequalities and an exact separation procedure to strengthen the proposed formulation. Contreras et al. (2009) develop a new formulation that yields tighter LP bounds than that in Contreras et al. (2010) and propose an algorithm employing the Lagrangean dual, subgradient optimization, and a simple heuristic. Martins de Sa et al.

(2013) propose a Benders Decomposition (BD) approach to solve the model developed by Contreras et al. (2009). Sedehzadeh et al. (2016) address a multi-objective and multi-modal problem with uncertain input data allowing hubs to have different capacity levels and to be connected with different transportation modes. Blanco and Marin (2019) offer two MIP models for upgrading nodes, which implies a decrease in the cost of traversing arcs connecting those upgraded nodes. The largest problem instance solved in these studies is defined on networks with 100 nodes. The proposed models in these studies are hub location models with hub and/or arc fixed costs rather than p-hub median models.

We remark that the models that require a connected HLN may produce hub networks with a tree HLN depending on the data. For example, Martins de Sa et al. (2018b) obtain optimal solutions having a tree HLN for problem instances with sufficiently high fixed arc setup costs for a multiple allocation incomplete hub location problem with service time requirements.

Like most models developed for hub location problems, the models developed for problems requiring a tree HLN classically assume that the modeled network MN is a complete network with arc distances (costs) satisfying the triangle inequality. If the real-world network RealN is not complete or complete but its distances do not satisfy the triangle inequality (e.g., bus fares) as is the case for most real-life networks, a preprocessing on RealN is required to construct a complete MN by an algorithm (e.g., Floyd, 1962) that finds the shortest path lengths between all OD pairs in RealN. Thus, the resulting complete MN consists of the shortest path lengths in RealN and hence its distances satisfy the triangle inequality. That is, *an arc in a complete MN* (and hence in a hub network HN) *may actually correspond to a shortest path consisting of several arcs and not necessarily a single arc in RealN*. Most studies do not differentiate between RealN and MN and directly assume that a complete MN with arc distances satisfying the triangle inequality is given. Even though this approach has gained acceptance, this may cause several modeling and computational disadvantages. For example, when the triangle inequality is not satisfied, the models do not work correctly (e.g., Marín et al., 2006). Moreover, the shortest path may not be preferred or correct in some cases, e.g., communication networks.

Akgün and Tansel (2018) discuss these issues in detail and propose a problem setting and modeling framework that allows (non-complete or complete) RealN with any cost structure to be directly used as MN. They present the modeling framework in the context of the p-hub median problem defined on non-complete networks (RealN) and show how to extend it to handle different hub location problems. The approach provides flexibility in modeling several characteristics of real-life hub networks, e.g., the interactions between location and routing decisions, arcs with different costs and capacities, different topology and service level requirements.

Similar advantages exist for MATHLP as well. In this regard, we propose a new MIP model for MATHLP that is built upon the problem setting adopted by Akgün and Tansel (2018). We show through examples that the proposed modeling approach may produce better solutions than the classical approach, which may result from the differences in the selected hubs, the flow routes between origin–destination points, and the assignment of non-hub nodes to hub nodes. The proposed model is defined on non-complete networks but can also be used with complete networks. We solve the model by the CPLEX-based branch-and-bound algorithm and Gurobi-based branch-and-bound algorithm with Norel heuristic and develop a BD-based heuristic algorithms. We conduct computational tests to assess the performance of the proposed heuristics using instances defined on different networks with the number of nodes changing from 81 to 500. The results indicate that the BD-based heuristic algorithm can find good solutions within the allocated time for the test instances.

The rest of the paper is organized as follows: We discuss how the hub network and hence the total cost differs when different modeling approaches are used with different assumptions in Section 2. We define the problem and present the details of the MIP model for MATHLP in Section 3. We give the BD-based heuristic approach for the proposed formulation in Section 4. We present the computational studies in Section 5 and conclude the paper in Section 6.

2. Comparison of the hub networks for different modeling approaches

In this section, we investigate how the hub network topology and the total cost change depending on the modeling approach used under different assumptions. We compare two modeling approaches: (1) *The classical approach*: Modeled network MN is complete and its distances satisfy the triangle inequality. (2) *The proposed approach*: MN is the same as the real-world network RealN that may be complete or non-complete. For comparison purposes, we use two different types of networks: a *7-node complete network* whose distances do not satisfy the triangle inequality and a *30-node non-complete network*. The complete network is the network used by Marín et al. (2006) to show that some hub location models do not work correctly when the triangle inequality is not satisfied. The distance matrix of the complete network and other parameters used are given in Appendix A. The non-complete network given in Fig. 1 is the network that consists of 30 cities in Turkey as the nodes and the roads between neighboring cities as the arcs. The distances are direct distances between the neighboring cities. We assume that a discount factor of 0.7 is applied to the hub arc costs.

To represent the classical approach, we extend the model of Ernst and Krishnamoorthy (1998) developed for multiple allocation hub location problem by adding necessary constraints to achieve a tree HLN. The resulting classical model given in Appendix B produces a hub network having a tree HLN by minimizing total transportation cost and allowing multiple allocation. We remark that the classical model needs as MN a complete network whose distances satisfy the triangle inequality to work correctly. In this regard, we apply the Floyd's Algorithm (Floyd, 1962) to the non-complete network to find all-pairs shortest path distances and construct a complete network in order to obtain a solution for the non-complete network using the classical model. *To represent the proposed approach*, we use our proposed model for MATHLP, whose details are given in Section 3. The proposed model can use any type of network, i.e., complete or non-complete, as MN. In this regard, the proposed model uses directly the non-complete network and complete network as MN.

Our first goal is to show that the classical model does not work correctly when a complete network whose distances do not satisfy the triangle inequality is used as MN. We solve both the proposed model and the classical model to optimality using the complete network given in Fig. 2 (a) and setting $p = 3$. Fig. 2(b) and (c) indicate hub networks for the proposed model and the classical model on the original network, respectively. Filled circles and empty circles represent hub nodes and non-hub nodes, respectively. Solid lines and dashed lines indicate hub arcs and access arcs, respectively. Fig. 2(b) and (c) show that both models produce an HLN with a tree structure. However, the selected hubs, the tree structures, and the resulting objective function values are different. In Fig. 2(b), HLN consists of nodes 2 through 6 with nodes 2, 4, and 6 being the hub nodes and nodes 3 and 5 being the non-hub nodes. The tree in Fig. 2(b) consists of non-hub nodes 3 and 5 as intermediate nodes between hub nodes, which may be possible when the triangle inequality is not satisfied (e.g., Marín et al., 2006). We can think of nodes 3 and 5 as transshipment points in HLN. These nodes receive service from their adjacent hub nodes. Specifically, non-hub node 3 is assigned to hub nodes 2 and 4 while non-hub node 5 is assigned to hub nodes 4 and 6. Accordingly, dashed lines between hub nodes represent collection and distribution flows between non-hub nodes 3 and 5 and their adjacent hub nodes. Non-hub nodes 1 and 7 are assigned to a single hub. In Fig. 2(c), HLN consists of the nodes 2, 3, and 6 as the hub nodes. All non-hub nodes receive service from a single hub. The optimal objective function values for the proposed model and the classical model are 88 and 1266, respectively. This difference results from the fact that

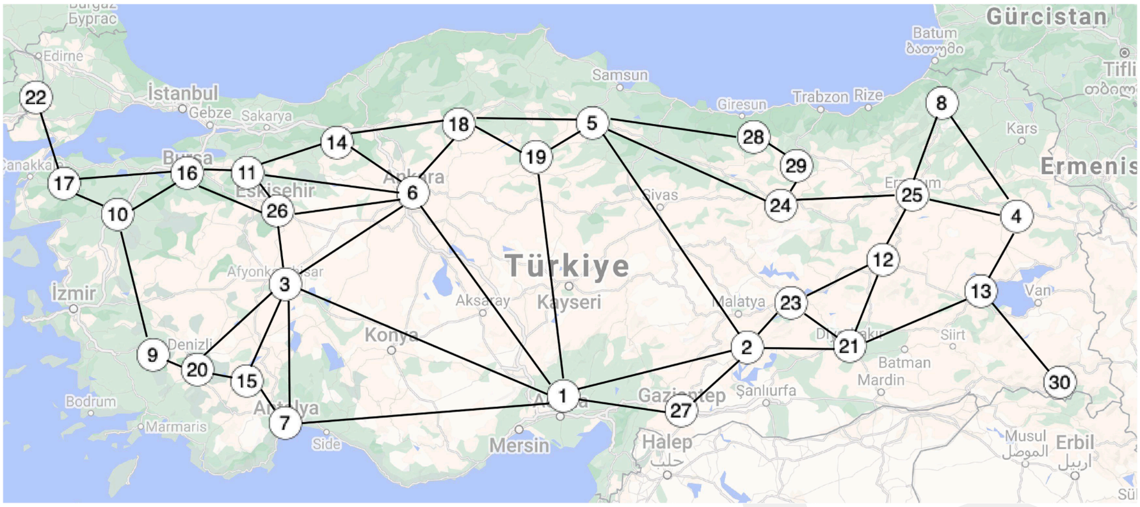

Fig. 1. Non-complete transportation network consisting of 30 cities of Turkey.

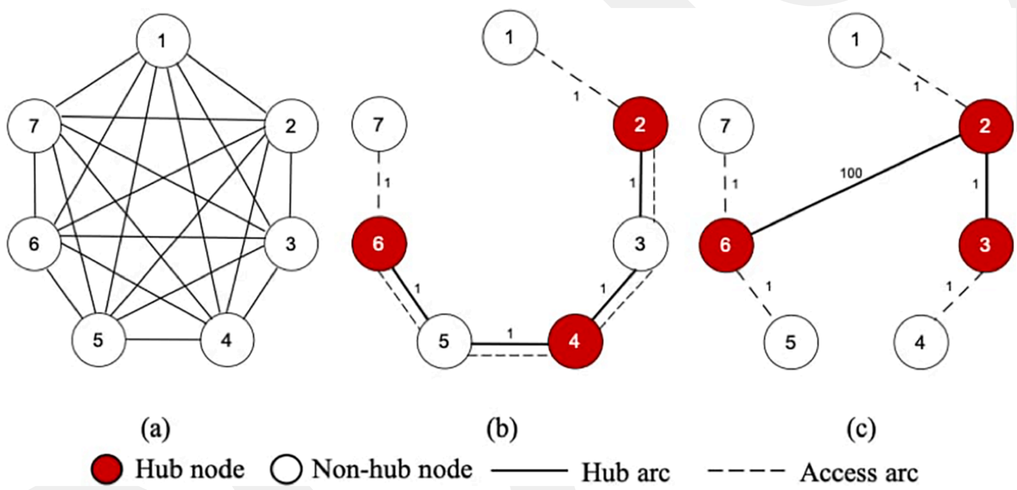

Fig. 2. Tree structures obtained using a complete network whose distances do not satisfy the triangle inequality. (a), (b), and (c) represent the original complete network, the resulting hub network using the proposed model, and the resulting the hub network using the classical model.

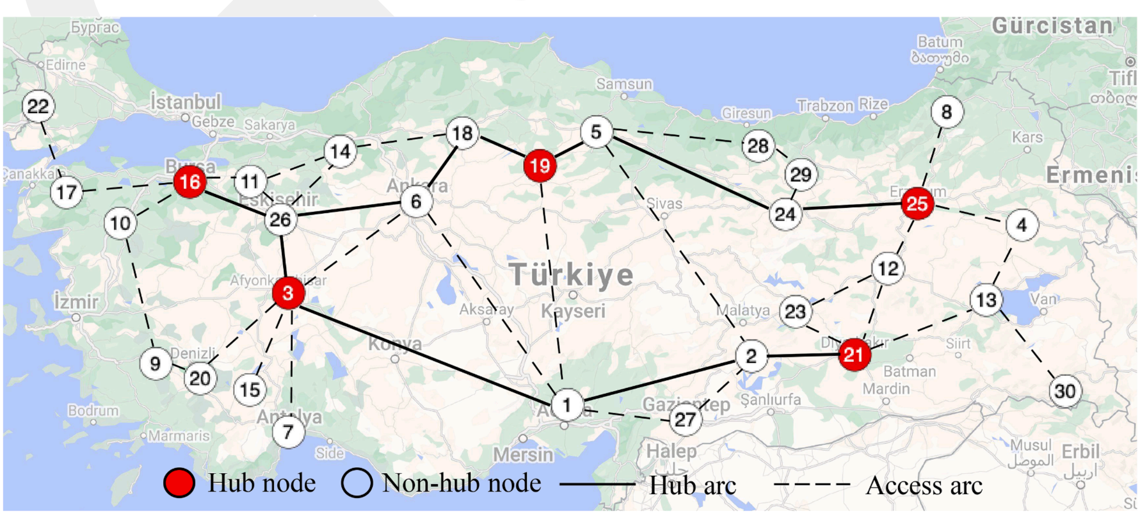

Fig. 3. Optimal hub network obtained with the proposed model (H = 30, p=3).

the proposed model is allowed to find and use routes with less cost than that of direct arcs between nodes to send flows while the classical model uses only direct arcs. For example, to send flow from node 2 to node 6, the classical model uses direct arc (2,6) with a cost of 100 while the proposed model uses the path consisting of the arcs (2,3), (3,4), (4,5), and (5,6) with a cost of 4.

Our second goal is to show that the proposed model may find a better solution than the classical model even when the triangle inequality is satisfied. We will consider *two cases*: (1) The proposed model and the classical model find different flow routes even though their optimal hub set and the tree structure are the same. (2) The proposed model and the classical model find different optimal hub sets and hence different hub networks. For both cases, we solve the models to optimality using appropriate versions of the non-complete network in Fig. 1 (non-complete version for the proposed model and complete version for the classical model). For *Case 1*, we allow all 30 nodes to be selected as hub nodes and set $p = 5$. Figs. 3 and 4 indicate the resulting hub networks for the proposed model and the classical model, respectively. Nodes 3, 16, 19, 21, and 25 are selected as the hub nodes by both models. Hub network in Fig. 3 directly gives the routing information on the real-world network RealN. For example, the route between hubs 3 and 21 is (3,1)-(1,2)-(1,21) with nodes 1 and 2 being transshipment points in HLN.

In Fig. 4, however, hub nodes 3 and 21 are connected by a single arc as expected and postprocessing is required to determine that the shortest-path arc (3,21) corresponds to the route consisting of the arcs (3,1)-(1,2)-(2,21) in RealN. Nevertheless, this should not be interpreted as that the routes between nodes are always the same in both models. Consider the routes between hub nodes 3, 16, and 19. In Fig. 4, these nodes are connected to each other by direct arcs (16,19) and (19,3). Any flow sent from node 16 to node 3 follows the route consisting of the arcs (16,19) and (19,3). The direct arcs (16,19) and (19,3) in Fig. 4 correspond to the routes (16,26)-(26,6)-(6,18)-(18,19) and (19,18)-(18,6)-(6,26)-(26,3) in RealN as shown in Fig. 3, respectively. This means that flow sent from 16 to 3 covers the route (26,6)-(6,18)-(18,19) twice, one going from 16 to 19 and one going from 19 to 3. In Fig. 3, however, flow from node 16 to node 3 follows the route (16, 26)-(26,3). That is, even though the tree structures of both models seem to be the same in RealN (after postprocessing the solution of the classical model), the resulting flows are different. This also affects the assignments of origin and destination nodes to the hubs. As a result, the optimal objective function values for the proposed model and the classical model are 14,279,772 and 14,297,370, respectively.

For *Case 2*, we allow 20 nodes indexed between 1 and 20 to be selected as hub nodes and set $p = 8$. Figs. 5 and 6 indicate the resulting hub networks for the proposed model and the classical model, respectively. The resulting hub networks are different because different optimal hub sets (2–6, 8, 12, and 16 for the proposed model and 1, 2, 4, 5, 6, 12, 16, and 20 for the classical model) are selected. The optimal objective function values for the proposed model and the classical model are 13,762,410 and 13,795,031, respectively.

We remark that the results in Case 1 and Case 2 are valid when we address MATHLP. When we relax the tree-HLN requirement, both approaches find solutions that are the same in RealN as long as arc distances (costs) satisfy the triangle inequality. This is also true for the single allocation version.

To sum up, using the proposed approach in addressing MATHLP may allow to obtain better solutions that may result from different hub network topologies, assignments, flows, and costs. The approach may also provide more flexibility in modeling several real-life issues directly, e.g., arc capacities or disruptions.

3. Problem definition and mathematical formulation

We define *Multiple Allocation Tree of Hubs Location Problem* (MATHLP) based on the modeling framework given by Akgün and Tansel (2018). Let $G = (N, E)$ be an undirected and connected network representing RealN with $N = \{1, \cdots, n\}$ and E being the node set and edge set, respectively. N consists of supply/origin nodes S, demand/destination nodes D, and transshipment nodes T. The nodes in S generate a positive flow w_{ij} for at least one node $j \in D$. The same node can be the element of both S and D.

Let $G^* = (N^*, E^*)$ represent the subnetwork of G that can be used for inter-hub transportation (e.g., potential links for rapid transit corridors). E^* is the set of edges that can be used as *hub arcs* for some reason, e.g., they have high capacities or more appropriate for construction. N^* is the set of nodes that are incident to E^*. We define $H \subseteq N^*$ as the set of nodes that can be hubs.

Let l_{ij} denote the length of edge $\{i, j\}$ with $l_{ij} = l_{ji}$. The χ_{ij}, α_{ij}, and δ_{ij} are the cost of moving one unit of flow per unit length along the edge $\{i, j\}$ for *collection*, *transfer*, and *distribution*, respectively, with $\alpha_{ij} \le \chi_{ij}$ and $\alpha_{ij} \le \delta_{ij}$ to achieve economies of scale.

MATHLP aims to (1) select p nodes from hub set H, (2) determine the service routes between OD pairs that visit at least one hub node, (3) connect all hubs through a tree structure and require all flows to use this tree structure by using a multiple allocation strategy such that total transportation cost is minimized.

We formulate MATHLP using a three-layer network $G_0 = (N_0, A_0)$ as the modeled network MN where the first, second, and third layers

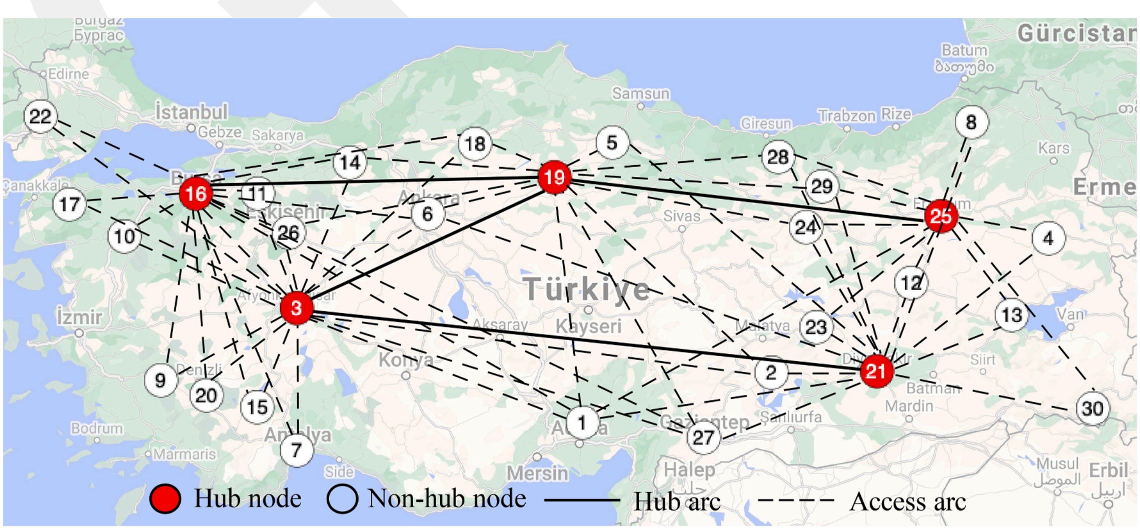

Fig. 4. Optimal hub network obtained with the classical model ($H = 30$, p=3).

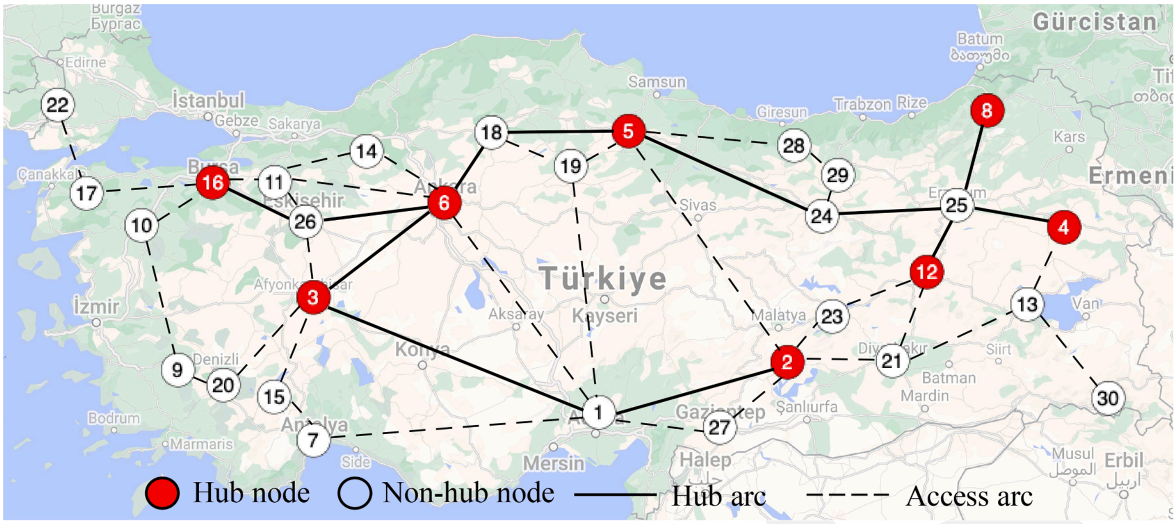

Fig. 5. Optimal hub network obtained with the proposed model ($H = 20$, p=8).

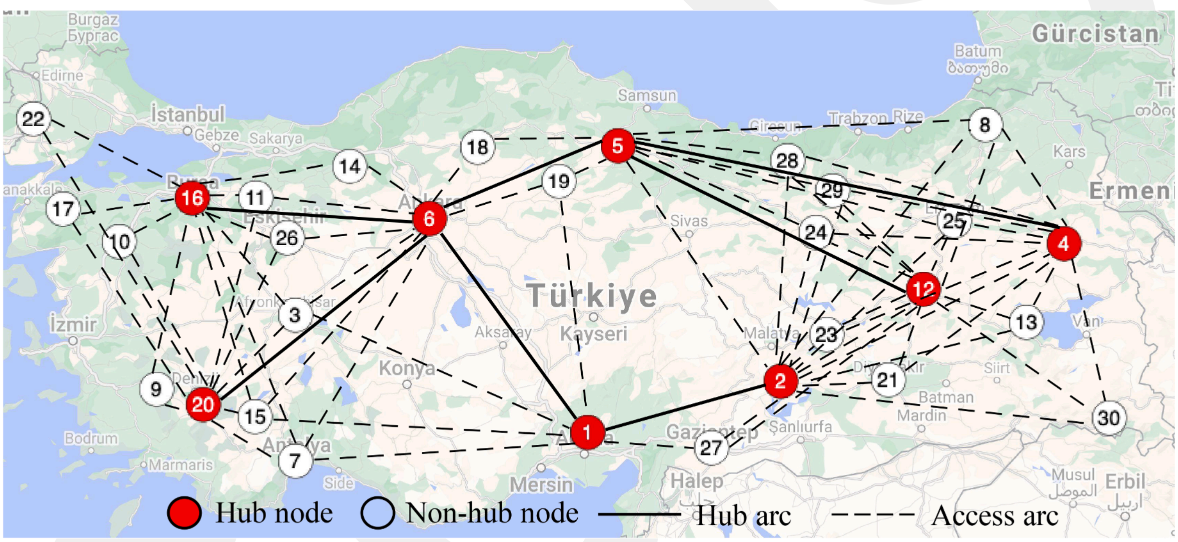

Fig. 6. Optimal hub network obtained with the classical model ($H = 20$, p=8).

represent the *collection/supply*, *transfer/hub*, and *distribution/demand* layers, respectively. To construct G_0, we use the directed version of $G = (N,E)$, $G^{'} = (N,A)$, which is obtained by replacing each edge $\{i,j\} \in E$ with a pair of directed arcs (i,j) and (j,i) such that $l_{ij} = l_{ji}$.

The supply layer network $G_1 = (N_1, A_1)$ and the distribution layer network $G_3 = (N_3, A_3)$ are copies of $G^{'} = (N,A)$ while the hub layer network $G_2 = (N_2, A_2)$ is the subnetwork of G' that corresponds to $G^{*} = (N^{*}, E^{*})$ with $N_m = \{m1, m2, \cdots, mn\}$ and $A_m = \{(mi, mj) : (i,j) \in A\}$ where $m = 1,2,3$. To exemplify, node 3 in RealN G is represented as 13, 23, and 33 in G_1, G_2, and G_3, respectively. G_1 and G_2 are connected by arcs of the form $A_{12} = \{(1i, 2i) : i \in H\}$ while G_2 and G_3 are connected by arcs of the form $A_{23} = \{(2i, 3i) : i \in H\}$ Thus, $N_0 = \bigcup_{m=1}^{3} N_m$ and $A_0 = \bigcup_{m=1}^{3} A_m \cup A_{12} \cup A_{23}$. Fig. 7 shows a three-layer MN constructed using the structure of RealN G where $E^{*} = \{\{2,3\},\{2,4\},\{3,4\},\{4,5\}\}$, $N^{*} = \{2,3,4,5\}$, $H = \{3,4,5\}$, and $S = D = \{1,2,3,4,5\}$.

We formulate MATHLP as a multicommodity flow problem with side constraints in G_0. The flows w_{ij} with $i \in S$ and $j \in D$ are sent from $1i \in N_1$ to $3j \in N_3$ through G_0. We associate with each node $i \in S$ a different commodity. $W_i = \sum_{j \in D} w_{ij}$ is the total supply of commodity i at node $1i$. We use the parameter $b_{\beta k}$ to represent the amount of supply/demand of commodity k at node $\beta \in N_0$. $b_{\beta\beta} = \sum_{j \in D} w_{\beta j}$ for $\beta = 1i$ and $i \in S$, $b_{\beta k} = -w_{k\beta} = -w_{kj}$ for $\beta = 3j$ and $j \in D$, and $b_{\beta k} = 0$ for all other nodes and $k \in S$. $F_\beta^{out}\left(F_\beta^{in}\right)$ is the forward (inward) star of a node $\beta \in (N_1 \cup N_2 \cup N_3)$.

We also associate with each arc (i,j) a unit cost c_{ij}, where l_{ij} is the length of arc (i,j), as follows:

$$c_{ij} = \begin{cases} \chi_{ij} \times l_{ij} & for\ (1i, 1j),\ (i,j) \in A \\ \alpha_{ij} \times l_{ij} & for\ (2i, 2j),\ (i,j) \in A^{*} \\ \delta_{ij} \times l_{ij} & for\ (3i, 3j),\ (i,j) \in A \\ 0 & for\ (1i, 2i)\ or\ (2i, 3i),\ i \in H \end{cases}$$

We define the following decision variables: (1) x_{ijk} is the amount of flow of commodity $k \in S$ in arc (i,j), (2) y_{2i} is a binary variable that takes on the value of 1 when a hub is located at node $i \in H$ and 0 otherwise, (3) s_{ij} is a binary variable that takes on the value of 1 when arc (i,j) belongs to the tree structure in G_2 and 0 otherwise, and (4) t_{ij} is a flow variable used to construct the tree structure in G_2 and represents the amount of fictitious flow in arc (i,j).

We construct the tree structure in G_2 by using a rooted spanning tree formulation based on single-commodity flows t_{ij}. We define a node $\theta \in N_2$ as the root/supply node from which one unit of fictitious flow is sent

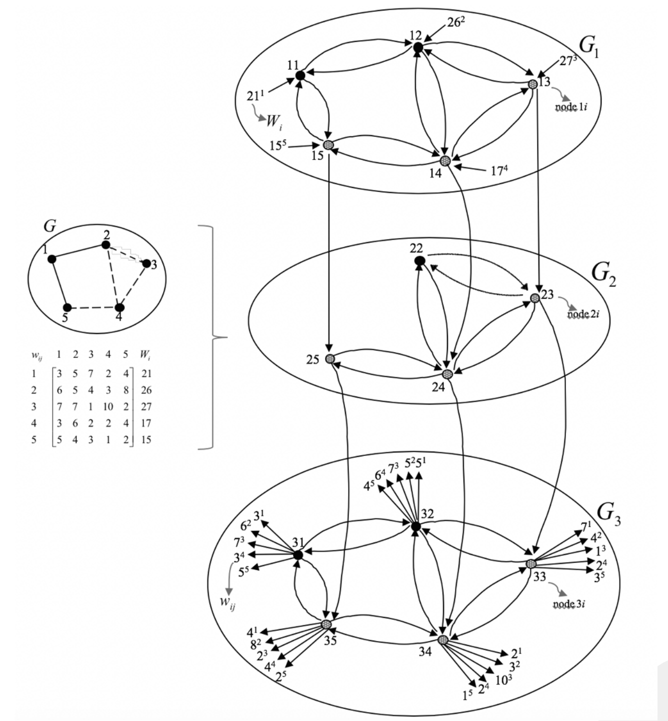

Fig. 7. Three-layer MN G_0 constructed from RealN G (Adapted from Akgün and Tansel, 2018).

to each other node $i \in (N_2 - \theta)$, i.e., a total of $|N_2 - 1|$ units of flow is sent from θ. The arcs with positive flows t_{ij} are selected as the arcs of the tree by the variables s_{ij}.

We present the sets, indices, parameters, and decision variables used in the formulation of the problem in Table 1. With these definitions, the proposed model, MATHLM, is given below:

Model MATHLM: Multiple allocation tree of hubs location model

$$Z^* = \mathrm{Min} \sum_{k \in S} \sum_{(i,j) \in} A_0 c_{ij} x_{ijk} \tag{1}$$

s.t.

$$\sum_{j \in F^{out}_\beta} x_{ijk} - \sum_{j \in F^{in}_\beta} x_{jik} = b_{\beta k} \quad \beta \in \left(N_1 \cup N_2 \cup N_3 \right), k \in S \tag{2}$$

$$\sum_{i \in H} y_{2i} = p \tag{3}$$

$$x_{(1i,2i)k} \le W_k y_{2i} \quad i \in H, k \in S \tag{4}$$

$$x_{(2i,3i)k} \le W_k y_{2i} \quad i \in H, k \in S \tag{5}$$

$$\sum_{j \in F^{out}_\theta} t_{\theta j} = |N_2 - 1| \tag{6}$$

$$\sum_{j \in F^{out}_\beta} t_{\beta j} - \sum_{j \in F^{in}_\beta} t_{j\beta} = -1 \quad \beta \in (N_2 - \theta) \tag{7}$$

$$\sum_{j \in F^{in}_\beta} s_{j\beta} = 1 \quad \beta \in (N_2 - \theta) \tag{8}$$

$$\sum_{j \in (F^{in}_\theta \cap N_2)} s_{j\theta} = 0 \tag{9}$$

$$t_{ij} \le |N_2 - 1| s_{ij} \quad (i,j) \in A_2 \tag{10}$$

Table 1
Sets, indices, parameters, and decision variables of the proposed model.

Sets, Indices, and Parameters	
G=(N,E)	Undirected real-world network with node set N and edge set E, N $= S \cup D \cup T$ where S, D, and T are the set of supply, demand, and transshipment nodes
$G^* = (N^*, E^*)$	Subnetwork of G that can be used for inter-hub transportation, E^* is the set of edges that can be used as hub arcs, N^* is the set of nodes that are incident to E^*
$G' = (N, A)$	Directed version of G=(N,E) obtained by replacing each edge $\{i, j\} \in E$ with a pair of directed arcs (i, j) and (j, i)
H	Set of nodes that can be hubs with $H \subseteq N^*$
$G_0 = (N_0, A_0)$	Three-layer modeled network, $G_0 = G_1 \bigcup G_2 \bigcup G_3$ with G_1, G_2, and G_3 representing the supply (first), hub (second), and distribution (third) layers of the network
$G_1 = (N_1, A_1)$	Supply layer network with $G_1 = G'$
$G_2 = (N_2, A_2)$	Hub layer network constructed from subnetwork of G' that corresponds to $G^* = (N^*, E^*)$
$G_3 = (N_3, A_3)$i,	Distribution layer network with $G_2 = G'$
$j \in N$	Nodes in the network G=(N,E)
1i	Nodes in the supply layer G_1
2i	Nodes in the hub layer G_2
3i	Nodes in the distribution layer G_3
(1i,1j)	Arcs in the supply layer
(2i,2j)	Arcs in the hub layer
(3i,3j)	Arcs in the distribution layer
A_{12}	Arcs connecting G_1 and G_2 with $A_{12} = \{(1i, 2i) : i \in H\}$
A_{23}	Arcs connecting G_2 and G_3 with $A_{23} = \{(2i, 3i) : i \in H\}$
$i, j \in N$	Nodes in the network G=(N,E)
k	Commodity type indicating the origin node of the flow, $k = i \in S$
l_{ij}	Length of arc(i,j)
χ_{ij}, α_{ij}, δ_{ij}	Cost of moving one unit of flow per unit length along arc (i,j) for supply, hub, and distribution layers, respectively
c_{ij}	Cost of arc (i,j) with $c_{ij} = l_{ij}\chi_{ij}$, $c_{ij} = l_{ij}\alpha_{ij}$, and $c_{ij} = l_{ij}\delta_{ij}$ for supply, hub, and distribution layers, respectively
w_{ij}	Flow to be sent from $i \in S$ to $j \in D$
W_i	Total supply of commodity i, $W_i = \sum_{j \in D} w_{ij}$
$b_{\beta k}$	Amount of supply/demand of commodity k at node $\beta \in N_0$
$F^{out}_\beta \left(F^{in}_\beta\right)$	Forward (inward) star of a node $\beta \in (N_1 \cup N_2 \cup N_3)$.
Decision Variables	
x_{ijk}	Amount of flow of commodity k in arc (i,j)
y_{2i}	1, if a hub is located at node $i \in H$ and 0, otherwise
s_{ij}	1, if arc (i,j) belongs to the tree structure in G_2 and 0, otherwise
t_{ij}	Flow variable used to construct the tree structure in G_2 and represents the amount of fictitious flow in arc (i,j)

$$x_{ijk} \le W_k (s_{ij} + s_{ji}) \quad (i,j) \in A_2, k \in S \tag{11}$$

$$t_{ij} \ge 0, s_{ij} \in \{0,1\} \quad (i,j) \in A_2 \tag{12}$$

$$x_{ijk} \ge 0 \quad (i,j) \in A_0, k \in S \tag{13}$$

$$y_{2i} \in \{0,1\} \quad i \in H \tag{14}$$

Objective function (1) minimizes the total transportation cost. Constraints (2) are the flow balance constraints for all the nodes in each layer of the network G_0 and commodities. Constraint (3) requires p hubs be selected. Constraints (4) and (5) ensure that the flow between layers is possible only through arcs $(1i, 2i)$ and $(2i, 3i)$ if a hub is located at node i, i.e, $y_{2i} = 1$. Constraints (6)-(10) construct a spanning tree in G_0. Constraint (6) sends $|N_2 - 1|$ units of fictitious flow from root node θ. Constraints (7) are flow-balance constraints that ensure all nodes in $N_2 - \theta$ receive one unit of fictitious flow. Constraints (8) require that there be exactly one incoming arc to each node $i \in (N_2 - \theta)$. Constraint (9) ensures that there is no incoming arc to root node θ from the nodes in N_2. Constraints (10) require an arc with a positive fictitious flow to be selected as an arc of the spanning tree in the hub layer. Constraints (11) allow commodity flows only on the arcs of the spanning tree and hence

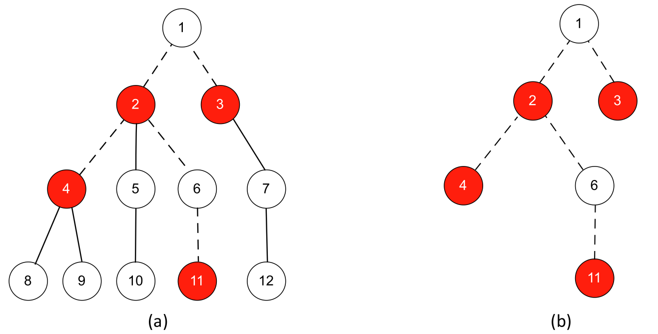

Fig. 8. Illustration of the tree spanning all nodes in the hub layer (a) and the resulting tree structure connecting the hubs (b).

the arcs with positive commodity flows connect all hubs in a tree structure. Fig. 8(a) illustrates an example of a tree spanning all nodes in the hub layer constructed by positive fictitious flows. Accordingly, Fig. 8 (b) gives the resulting tree structure connecting the hubs. Nodes 1 and 6 are the non-hub nodes serving as transition nodes to connect the hubs through a tree. Constraints (12)-(14) define the decision variables.

4. Proposed solution methodology

MATHLM is difficult to solve using standard optimization software. Computational studies indicate that CPLEX-based and Gurobi-based branch-and-bound algorithm can find optimal or near-optimal solutions for problem instances defined on 81-node network with a run time of 24 h. However, for larger size networks, they cannot find a solution or the resulting optimality gaps increase up to 96%. This has led us to develop a solution methodology based on Benders Decomposition (BD) (Benders, 1962), which has been used successfully in solving several variants of hub location problems including single allocation tree of hubs location problem (e.g., Camargo et al., 2008a; Camargo et al., 2008b; Camargo et al., 2011; Contreras et al., 2011; Martins de Sa et al., 2013; Camargo et al., 2017; Martins de Sá et al., 2018a; Ghaffarinasab and Kara, 2019; Mokhtar et al., 2019; and Taherkhani et al., 2020). However, we have encountered the convergence problem in the application of BD algorithm even for small-size problems. We have tried CPLEX's automatic BD and encountered the same situation. We have incorporated several acceleration strategies, namely, strong cut generation, cut disaggregation, and a combination of two strategies, in order to improve the convergence of the BD algorithm. However, these strategies have not been successful as well. For this reason, we have developed Benders-type heuristics. Of several heuristics, *the heuristic based on BD with strong cut generation and the heuristic based on BD with combined strong cut generation and cut disaggregation* have produced much better results than the other ones for all instances. In this regard, we present the development of these heuristics in this section. Computational studies indicate that the proposed heuristics are efficient and can find solutions for large-size problems with up to 500 nodes.

The BD approach partitions a difficult optimization problem into two simpler problems: an integer problem, named as the master problem (MP), and a linear problem, named as the subproblem (SP). The algorithm solves MP and SP iteratively and adds a new constraint known as Benders cut obtained from SP to the MP. The algorithm is terminated when the optimal objective function values of MP and SP are equal to each other.

One main challenge arising in the application of the BD algorithm is the need to solve difficult integer MPs in large size problems. As the number of iterations increases in this type of problems, the number of cuts added to the MP also increases, which makes the MP more difficult to solve and hence the convergence of the BD algorithm impossible due to time and memory limitations. This has led the researchers to develop algorithms referred to as Benders-type heuristics. In most Benders-type heuristics, researchers use (meta)heuristic algorithms or relaxations to solve MPs. Boschetti and Maniezzo (2009) state that the BD algorithm provides a rich framework for developing heuristics since it uses dual information to reduce search space, verifies solution quality, and obtains multiple starting points for local search. They solve both MP and SP using Lagrangean relaxation. Poojari and Beasley (2009), Lai et al. (2010), and Lai et al. (2012) solve the MP using the genetic algorithm. Jiang et al. (2009) solve the MP using tabu search. Pacqueau et al. (2012) solve the linear relaxation of MP and then fix the values of some variables to their upper/lower bounds after applying round-off heuristics.

In the following, we give the BD algorithm, strong cut generation, cut disaggregation, and finally the Benders-type heuristics where a relaxed version of MP obtained after removing some complicating constraints is solved.

4.1. Benders decomposition for MATHLM

As stated before, the BD algorithm decomposes the original problem into two simpler problems, namely, MP and SP, and solves MP and SP iteratively until their optimal objective function values are equal or a stopping criterion is reached. MP is a relaxed version of the original problem and involves the set of integer variables and associated constraints. SP is a linear program that is obtained by taking the dual of the problem formulated by fixing the values of integer variables in MATHLM.

Let y and s represent the vector of integer variables $y_{2i}, i \in H$, and $s_{ij}, (i, j) \in A_2$, respectively. Let SPD represent the linear problem obtained by fixing the values of integer variables y and s in MATHLM to y^h and s^h, respectively, at iteration h. When y and s are fixed, the resulting SPD consists of (15)-(20) and is a linear routing problem that finds the routes between OD pairs because the hubs and hub arcs in the tree structure are fixed. We find SP by taking the dual of SPD with the dual variables e_{ik}, f_{ik}, g_{ik}, and tt_{ijk} defined for constraints (16) through (19), respectively. The resulting SP consists of (21)-(28).

Model SPD: Linear problem at iteration h

$$\min \sum_{k \in S} \sum_{(i,j) \in} A_0 c_{ij} x_{ijk} \tag{15}$$

s.t.

$$\sum_{j \in F_i^{out}} x_{ijk} - \sum_{j \in F_i^{in}} x_{jik} = b_{ik} \quad i \in \left(N_1 \cup N_2 \cup N_3 \right), k \in S \tag{16}$$

$$x_{(1i,2i)k} \le W_k y_i^h \quad i \in H, k \in S \tag{17}$$

$$x_{(2i,3i)k} \le W_k y_i^h \quad i \in H, k \in S \tag{18}$$

$$x_{ijk} \le W_k (s_{ij}^h + s_{ji}^h) \quad (i,j) \in A_2 \tag{19}$$

$$x_{ijk} \ge 0 \quad (i,j) \in A, k \in S \tag{20}$$

Model SP: Subproblem at iteration h

$$Max \sum_{k \in S} \sum_{i \in N_0} b_{ik} e_{ik} + \sum_{k \in S} \sum_{i \in N_0} W_k y_i^h f_{ik} + \sum_{k \in S} \sum_{i \in N_0} W_k y_i^h g_{ik} + \sum_{k \in S} \sum_{(i,j) \in A_2} W_k \left(s_{ij}^h + s_{ji}^h \right) tt_{ijk} \tag{21}$$

$$\text{s.t.} \quad e_{ik} - e_{jk} \le l_{ij} \quad (i,j) \in (A_1 \cup A_3), k \in S \tag{22}$$

$$e_{ik} - e_{jk} + tt_{ijk} \le l_{ij} \quad (i,j) \in A_2, k \in S \tag{23}$$

$$e_{ik} - e_{jk} + f_{jk} \leq 0 \quad (i,j) \in A_{12}, k \in S \tag{24}$$

$$e_{ik} - e_{jk} + g_{ik} \leq 0 \quad (i,j) \in A_{23}, k \in S \tag{25}$$

$$f_{ik}, g_{ik} \leq 0 \quad i \in H, k \in S \tag{26}$$

$$e_{ik} \ \text{free} \quad i \in (N_1 \cup N_2 \cup N_3), k \in S \tag{27}$$

$$tt_{ijk} \leq 0 \quad (i,j) \in A_2, k \in S \tag{28}$$

We formulate MP by using the constraints associated with integer variables in MATHLM and adding Benders *optimality* cuts. A Benders optimality cut (29) can be derived from the objective function (21) of SP at iteration h. In (29), e^h_{ik}, f^h_{ik}, g^h_{ik}, and tt^h_{ijk} are the optimal values of the dual variables in SP at iteration h and η is the under-estimator for the total cost. The resulting MP consists of (30)-(39).

$$\eta \geq \sum_{k\in S}\sum_{i\in N_0} b_{ik}e^h_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k f^h_{ik} y_i + \sum_{k\in S}\sum_{i\in N_0} W_k g^h_{ik} y_i + \sum_{i\in N_2}\sum_{j\in N_2}\sum_{k\in S} W_k tt^h_{ijk}(s_{ij} + s_{ji}) \tag{29}$$

Model MP: Master problem at iteration h

$$\text{Min} \ \eta \tag{30}$$

s.t.

$$\eta \geq \sum_{k\in S}\sum_{i\in N_0} b_{ik}e^h_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k f^h_{ik} y_i + \sum_{k\in S}\sum_{i\in N_0} W_k g^h_{ik} y_i + \sum_{i\in N_2}\sum_{j\in N_2}\sum_{k\in S} W_k tt^h_{ijk}(s_{ji} + s_{ji}) \tag{31}$$

$$\sum_{i\in H} y_{2i} = p \tag{32}$$

$$\sum_{j\in F^{out}_\theta} t_{\theta j} = |N_2 - 1| \tag{33}$$

$$\sum_{j\in F^{out}_\beta} t_{\beta j} - \sum_{j\in F^{in}_\beta} t_{j\beta} = -1 \quad \beta \in (N_2 - \theta) \tag{34}$$

$$\sum_{j\in F^{in}_\beta} s_{j\beta} = 1 \quad \beta \in (N_2 - \theta) \tag{35}$$

$$\sum_{j\in (F^{in}_\theta \bigcap N_2)} s_{j\theta} = 0 \tag{36}$$

$$t_{ij} \leq |N_2 - 1| s_{ij} \quad (i,j) \in A_2 \tag{37}$$

$$t_{ij} \geq 0, s_{ji} \in \{0,1\} \quad (i,j) \in A_2 \tag{38}$$

$$y_{2i} \in \{0,1\} \quad i \in H \tag{39}$$

MP determines the locations of p hubs and connects them in a tree structure through constraints (32)-(37) at each iteration. Given $\boldsymbol{y}^h$ and $\boldsymbol{s}^h$ at iteration h, SPD is mainly a network flow problem. Because supply and demand quantities are equal and there are no capacity constraints in SPD, it is always feasible. Moreover, the objective function value of SPD is bounded because transportation costs are non-negative and finite, which means that SP, the dual of SPD, is feasible and has a bounded objective function value. Thus, at each iteration of the Benders algorithm, we obtain a feasible solution. This is why we do not need to generate and add Benders *feasibility* cuts but only Benders optimality cuts to MP. Otherwise, we would have to add feasibility cuts as well.

MP is a relaxation of the original problem and hence its objective function value provides a lower bound to that of MATHLM. We improve this bound at each iteration by adding Benders optimality cut (29). By construction, the objective function value of SP provides an upper bound on the objective function value of MATHLM.

4.2. Acceleration strategies for the BD algorithm

The performance of the BD Algorithm is mainly determined by the number of iterations and the time required to complete each iteration (e.g., Rahmaniani et al., 2017). The number of iterations may be high if the improvement rate of the LB is low, which results from weak Benders cuts. This also adversely affects the solution time of MP and memory requirements because the higher the number of cuts added to MP, the more difficult it becomes to solve MP. In some cases, solving SP may take excessive time too. All of these may cause poor convergence of the algorithm as we have experienced.

There are several strategies employed in the literature to accelerate the progress of the BD Algorithm. In this study, we employ *generating strong cuts and disaggregating the Benders cuts together*.

4.2.1. Generating strong cuts

Magnanti and Wong (1981) suggest an approach to generate *stronger* optimality cuts based on the determination of a *core point*.

To formalize the approach, let C_a and C_b represent two different cuts generated using (29) from two different solutions ($\boldsymbol{e}^a$,$\boldsymbol{f}^a$,$\boldsymbol{g}^a$,$\boldsymbol{tt}^a$) and ($\boldsymbol{e}^b$, $\boldsymbol{f}^b$,$\boldsymbol{g}^b$,$\boldsymbol{tt}^b$), respectively. Then, C_a is stronger than or dominates C_b if the right-hand-side value of C_a is greater than or equal to that of C_b with a strict inequality for at least one point $(\boldsymbol{y},\boldsymbol{s})$. That is, the cut that gives a better bound is a dominating cut. A cut is *pareto-optimal* if it is not dominated by any other cuts. Accordingly, the solution $(\boldsymbol{e},\boldsymbol{f},\boldsymbol{g},\boldsymbol{tt})$ is pareto-optimal if the cut defined by $(\boldsymbol{e},\boldsymbol{f},\boldsymbol{g},\boldsymbol{tt})$ is pareto optimal.

A pareto optimal solution $(\boldsymbol{e},\boldsymbol{f},\boldsymbol{g},\boldsymbol{tt})$ can be obtained by solving an optimization problem that finds a pareto-optimal point among alternative optimal solutions using a *core point*. A core point $(\boldsymbol{y}^0,\boldsymbol{s}^0)$ is a point in the relative interior of the convex hull of $\boldsymbol{y} \in Y$ and $\boldsymbol{s} \in \overline{S}$.

To define the optimization problem to solve, let $(\boldsymbol{y}^h,\boldsymbol{s}^h)$ be the optimal solution of MP and z^*_{SPh} be the optimal objective function value of SP at iteration h. The optimal solution $(\boldsymbol{e}^0,\boldsymbol{f}^0,\boldsymbol{g}^0,\boldsymbol{tt}^0)$ to the optimization problem PRT comprised of (40)-(44) is a pareto-optimal solution. The Benders cut obtained from the objective function (40) is a pareto-optimal cut. This cut is closest to the chosen core point $(\boldsymbol{y}^0,\boldsymbol{s}^0)$.

Model PRT: Model to find a Pareto-Optimal solution

$$\text{Max} \sum_{k\in S}\sum_{i\in N_0} b_{ik}e_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k y^0_i f_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k y^0_i g_{ik} + \sum_{i\in N_2}\sum_{j\in N_2}\sum_{k\in S} W_k \left(s^0_{ij} + s^0_{ji}\right) tt_{ijk} \tag{40}$$

s.t.

$$\sum_{k\in S}\sum_{i\in N_0} b_{ik}e_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k y^h_i f_{ik} + \sum_{k\in S}\sum_{i\in N_0} W_k y^h_i g_{ik} + \sum_{i\in N_2}\sum_{j\in N_2}\sum_{k\in S} W_k \left(s^h_{ij} + s^h_{ji}\right) tt_{ijk} = z^*_{SPh} \tag{41}$$

$$f_{ik}, g_{ik} \leq 0 \quad i \in H, k \in S \tag{42}$$

$$e_{ik} \ \text{free} \quad i \in (N_1 \cup N_2 \cup N_3), k \in S \tag{43}$$

$$tt_{ijk} \leq 0 \quad (i,j) \in A_2, k \in S \tag{44}$$

The approach is based on the fact when SP has multiple optimal solutions, different cuts with different strengths may be defined. PRT generates the strongest cut possible. A pareto-optimal cut may be added at each iteration or periodically considering the tradeoff between additional computational burden and the reduction in the number of iterations. In this study, we add pareto-optimal cuts in each iteration. In the Benders Algorithm with pareto-optimal cuts, PRT is solved after solving SP and the cut generated using a given core point is added to MP.

Finding a core point may be a challenge for some problems (e.g., Martins de Sa et al., 2013). Mercier et al. (2005) state that using a core point that is not in the interior of the convex hull does not preclude finding a valid Benders cut. However, the further the core point is from the interior of the convex hull, the weaker the Benders cuts that are generated by this method. This demonstrates the importance of finding a good representative core point. Mercier et al. (2005) state that values close to 0 or 1 generate stronger cuts for different problem types. In this study, we conduct computational experiments by setting $y$and s to 0 and 1 to identify a good core point. Computational results indicate that setting $\boldsymbol{y} = 1$ and $s = 0$ yields stronger cuts.

4.2.2. Disaggregating the Benders cuts and adding multiple cuts

Another strategy to improve the progress of BD Algorithm is to add multiple cuts instead of just one cut at each iteration. We can achieve this by disaggregating Benders cuts (29) as proposed by Birge and Louveaux (1988). Disaggregation is possible because SP can be decomposed into smaller problems based on commodity type k. This allows us to add $|S|$ cuts simultaneously. The resulting MP, **MPM**, is comprised of (32)-(39) and (45)-(46).

Model MPM: Master problem with disaggregated cuts at iteration h

$$Min \sum_{k \in S} \eta_k \tag{45}$$

s.t.

In addition to Constraints (32)-(39)

$$\eta_k \geq \sum_{i \in N_0} b_{ik} e_{ik}^h + \sum_{i \in N_0} W_k f_{ik}^h y_i + \sum_{i \in N_0} W_k g_{ik}^h y_i + \sum_{i \in N_2} \sum_{j \in N_2} W_k tt_{ijk}^h (s_{ij} + s_{ji}) \quad \forall k \in S \tag{46}$$

In the Benders Algorithm with multiple cuts, MPM rather than MP is solved.

It may be possible to disaggregate (29) further, e.g., based on commodity type k and node i. However, the number of cuts in this case is $3n^2$ and solving MPM even for small-size problems becomes computationally very expensive. This is why we prefer to disaggregate based on k.

4.2.3. Combining strong cut generation and multiple cuts

We use the strategies explained in 4.2.1 and 4.2.2 simultaneously. In doing that, we first find the pareto-optimal cut and then disaggregate this pareto-optimal cut into multiple cuts. This requires solving PRT to find a pareto-optimal solution after solving SP and then disaggregating the cut as given in MPM.

4.3. Benders-Type heuristic approach

Computational experiments with the BD-based algorithms with or without acceleration strategies indicate that they are not promising to be used to solve large-scale problem instances. We observe that the progress of the algorithms is limited due to difficulty in solving the master problems. This leads us to develop two Benders-Type heuristics that facilitate the solution of the master problems and keeping the rest of the algorithms essentially the same.

Our approach is based on obtaining (y^h, s^h) at iteration h in two steps: (1) determining the hubs to locate by solving a relaxed master problem and (2) finding the tree structure connecting the located hubs by solving a rooted spanning tree formulation. We adopt this approach because we observe that the constraints (33)-(37) that ensure the tree structure in the master problems increase the solution time of the master problems significantly or make them almost impossible to solve for large-size problems.

The master problems that need to be solved in the heuristic algorithms can be stated by eliminating the variables s^h and associated terms from the formulations. We define two relaxed master problems, **RelMP** to be used while adding a single cut and **MCMP** to be used while adding multiple cuts.

Model RelMP: Relaxed master problem with a single cut

$$\text{Min } \eta \tag{47}$$

s.t.

$$\eta \geq \sum_{k \in S} \sum_{i \in N_0} b_{ik} e_{ik}^h + \sum_{k \in S} \sum_{i \in N_0} W_k f_{ik}^h y_i + \sum_{k \in S} \sum_{i \in N_0} W_k g_{ik}^h y_i \tag{48}$$

$$\sum_{i \in H} y_{2i} = p \tag{49}$$

$$y_{2i} \in \{0, 1\} \quad \forall i \in H \tag{50}$$

Model MCMP: Relaxed master problem with multiple cuts

$$Min \sum_{k \in S} \eta_k \tag{51}$$

s.t.

In addition to (49)-(50)

$$\eta_k \geq \sum_{i \in N_0} b_{ik} e_{ik}^h + \sum_{i \in N_0} W_k f_{ik}^h y_i + \sum_{i \in N_0} W_k g_{ik}^h y_i \ \forall k \in S \tag{52}$$

RelMP and MCMP solved at iteration h determine the values of $\boldsymbol{y}^h$ where p of the variables are 1 and the remaining are zero. To determine the tree structure among the given hub locations $\boldsymbol{i}$ with $y_{2i} = 1$, we solve a rooted spanning tree formulation, **SPTree,** based on single commodity flows t_{ij}, which is given in Appendix C. For the arcs with $t_{ij} > 0$, we set $s_{ij} = 1$. The solution (y^h, s^h) is then fed into SP.

Given the relaxed master problems (RelMP and MCMP) and SPTree, we define two Benders-Type Heuristic algorithms, **BDHEUR1** and **BDHEUR2**, that are mainly different in the acceleration strategies employed and the master problem solved. In BDHEUR1, we use only *generating strong cuts*. In the application of the algorithm, we first determine a pareto-optimal cut as explained in Section 4.2.2 and then add it to RelMP. In BDHEUR2, we use two strategies together, *generating strong cuts* and *disaggregating Benders cuts*. In the application of the algorithm, we determine a pareto-optimal cut, disaggregate this cut into multiple cuts as explained in Section 4.2.3, and add them to MCMP. We outline the steps of BDHEUR1 below. The steps of BDHEUR2 are the same as BDHEUR1 except that we solve MCMP instead of RelMP, i.e., replace RelMP with MCMP. In the application of the algorithms, we solve SP and SPTree to optimality and RelMP/MCMP until an optimality gap of 10% is achieved. Moreover, for the test problems on PMED400 and PMED500, we set a time limit of 1 h and 5 h, respectively, for the Model PRT that finds a pareto-optimal cut because we observe that strong cuts are obtained within that time limit.

In the algorithm, UB and z_{SP}^* represent the upper bound and the optimal objective function value for SP, respectively.

Algorithm BDHEUR1: Benders-Type heuristic 1 employing strong cut Generation.

Step 1: (Initialization)
Set y^h and s^h to an initial feasible integer solution.
Set time limit.
Set UB $= +\infty$
Set $h = 0$
Find a core point (y^0, s^0) for strong cut generation
Step 2: Solve SP (21)-(28)
Step 3: Set UB $= \min(\text{UB}, z_{SP}^*)$
Step 4: Solve PRT (40)-(44) to choose a strong cut
Step 5: Add cut(s) to RelMP (47)-(50)
Step 6: Solve RelMP (47)-(50) to get y
Step 7: Solve SPTree (53)-(57) to get s
Step 8: Set $h = h + 1$
Step 9: Set $y^h = y$ and $s^h = s$
Step 11: If elapsed time > time limit, stop. Otherwise, go to Step 2

5. Computational experiments

We conduct computational experiments to test the performance of the proposed model and solution methodologies. Specifically, we observe the performance of MATHLM using *CPLEX*, *Gurobi*, *Gurobi with Norel Heuristic*, and *LocalSolver* and Benders-type heuristics. We prefer the aforementioned solvers and algorithms because they are known to be effective for difficult MIP problems. In the application of Gurobi with Norel heuristic, which may be useful for models where the root relaxation is quite expensive, the Norel heuristic first tries to find a high-quality feasible solution in the allocated time and then Gurobi implements the branch and bound algorithm with the feasible solution found by the Norel heuristic (Gurobi, 2021). Localsolver is an innovative optimization solver combining exact and heuristic techniques and finds high-quality solutions for large-scale optimization problems (LocalSolver, 2021). However, LocalSolver cannot find feasible solutions even for small-size instances. Gurobi with Norel heuristic performs much better than Gurobi for all instances in finding feasible solutions. Because of this, we only present the results obtained by CPLEX and Gurobi with Norel Heuristic against the results obtained by Benders-type heuristics.

We define test problems on TR81, PMED200, PMED300, PMED400, and PMED500 networks. TR81 is defined by Akgün and Tansel (2018) and the non-complete transportation network of Turkey including all 81 cities of Turkey. The edges on TR81 are defined only between adjacent cities. The length of the edges on TR81 are assumed to be the direct distances from the high-way transportation network of Turkey. PMED200 through PMED500 are the non-complete networks used for the *p*-median problem instances (e.g., Beasley 1990) with the numbers indicating the number of nodes. Different test problems are created on the networks by changing $|H|$ and p where $E^* = E$, $N^* = N$ and $H \subseteq N^*$. For all test problems, w_{ij} is uniformly distributed with the interval (10,30). For all arcs, χ_{ij} and δ_{ij} are taken as 1 whereas α_{ij} is taken as 0.7. In all problems, $S = D = N$.

We code the models and the algorithms using GAMS and conduct the experiments on a PC with 3.6 GHz Intel Core i7-7700CPU processor and 32 GB of RAM for TR81, PMED200, PMED300 instances and on a server with Intel® Xeon® CPU E5-2683 V4 @ 2.1 GHz 64 core processor and 256 GB of RAM for PMED400 and PMED500 instances due to memory requirements. The runtime for the solvers and the algorithms is set to 24 h (86,400 secs). In using Gurobi with Norel heuristic, we assign 12 h for the Norel heuristic and 12 h for the branch and bound algorithm because we obtain high-quality feasible solutions with this setting.

In the tables, we present (1) the runtime (T) in CPU secs, the lower bound (LB), the objective function value of the best integer solution at the end of runtime (BP) obtained by CPLEX or Gurobi with Norel heuristic, and the relative optimality gap (Gap%) between LB and BP for MATHLM and (2) the best integer solution achieved either from CPLEX or Gurobi with Norel heuristic (BP*); the number of iterations (# of iters), LB, and UB achieved by the heuristic algorithms.

5.1. Computational experiments for MATHLM using CPLEX and Gurobi with Norel heuristic

Table 2 gives the results obtained solving MATHLM by CPLEX and Gurobi with Norel heuristic. In solving MATHLM using CPLEX, we use two different parameter settings that differ only in the value of *mipemhasis* parameter. The mipemphasis parameter value tells CPLEX what the balance between finding better feasible solutions and proving optimality should be in solving a model. We use two values of mipemphasis parameter, namely, “balance feasibility and optimality” and “feasibility” because our focus is to find better feasible solutions. With regard to

Table 2
Test results for MATHLM using CPLEX and Gurobi with Norel heuristic.

						CPLEX			Gurobi with Norel Heuristic			
Pr. Id	Network	|N|	|H|	p	T (secs)	LB	BP	Gap (%)	LB	BP	Gap (%)	BP*
1	TR81	81	30	3	84,840	*113,261,890*	**113,261,890**	0.0	113,120,288	**113,261,890**	0.0	113,261,890
2		81	30	5	86,400	100,804,000	**103,530,000**	2.7	*102,044,941*	**103,530,000**	1.4	103,530,000
3		81	50	3	86,400	*107,115,000*	**112,944,470**	5.4	105,087,303	**112,944,470**	7.0	112,944,470
4		81	50	5	86,400	*96,404,800*	103,014,691	6.9	95,881,444	**102,883,188**	6.8	102,883,188
5		81	50	8	86,400	*91,322,400*	97,216,421	6.5	91,015,050	**96,098,414**	5.3	96,098,414
6		81	50	10	86,400	*89,545,200*	93,852,241	4.8	88,953,013	**93,793,657**	5.2	93,793,657
7		81	60	3	86,400	*102,576,000*	**112,944,470**	10.1	100,724,739	**112,944,470**	10.8	112,944,470
8		81	60	5	86,400	*95,209,200*	103,426,356	8.6	94,446,443	**102,940,709**	8.3	102,940,709
9		81	60	8	86,400	89,687,000	96,185,436	7.2	*90,368,120*	**96,098,414**	6.0	96,098,414
10		81	60	10	86,400	*89,205,300*	94,098,044	5.5	88,302,192	**93,793,657**	5.9	93,793,657
11		81	81	3	86,400	*99,773,300*	**113,067,656**	13.3	98,948,604	**113,067,656**	12.5	113,067,656
12		81	81	5	86,400	*93,627,400*	102,909,588	9.9	93,297,541	**102,883,188**	9.3	102,883,188
13		81	81	8	86,400	*89,710,800*	98,401,597	9.7	89,309,497	**96,184,127**	7.1	96,184,127
14		81	81	10	86,400	86,910,600	94,461,872	8.7	*87,454,873*	**93,710,445**	6.7	93,710,445
15	PMED200	200	200	3	86,400	69,962,197	83,921,015	20.0	*70,242,011*	**82,844,187**	15.2	82,844,187
16		200	200	5	86,400	*67,681,170*	81,743,444	20.8	67,065,627	**77,314,463**	13.2	77,314,463
17		200	200	8	86,400	*65,226,836*	74,061,217	13.5	64,502,322	**72,475,694**	11.0	72,475,694
18		200	200	10	86,400	*63,947,068*	72,858,377	13.9	63,309,299	**71,176,399**	11.0	71,176,399
19	PMED300	300	300	3	86,400	100,542,251	152,379,779	34.0	*102,105,607*	**129,512,902**	21.2	129,512,902
20		300	300	5	86,400	93,990,900	148,400,000	36.3	*99,812,107*	**126,641,742**	21.2	126,641,742
21		300	300	8	86,400	*97,915,345*	118,263,997	17.2	97,631,794	**117,762,104**	17.1	117,762,104
22		300	300	10	86,400	96,342,738	**111,688,878**	13.7	*97,046,000*	112,496,864	13.7	111,688,878
23	PMED400	400	400	3	86,400	*129,405,098*	2,521,891,782	94.9	126,492,232	**188,065,072**	32.0	188,065,072
24		400	400	5	86,400	*126,975,841*	**159,334,212**	20.3	125,777,210	217,002,369	42.0	159,334,212
25		400	400	8	86,400	*125,823,824*	191,544,185	34.3	124,213,039	**185,954,774**	33.0	185,954,774
26		400	400	10	86,400	*125,305,399*	**152,776,681**	18.0	123,575,853	164,174,581	24.0	152,776,681
27	PMED500	500	500	3	86,400	*172,189,807*	3,954,137,863	95.7	169,677,900	**288,398,427**	41.2	288,398,427
28		500	500	5	86,400	*172,544,439*	3,920,051,544	95.6	163,650,171	**267,833,441**	38.8	267,833,441
29		500	500	8	86,400	*171,012,235*	3,952,102,853	95.7	159,240,544	**238,740,589**	33.3	238,740,589
30		500	500	10	86,400	*169,673,118*	**4,094,682,459**	95.9	160,733,186	no solution	–	4,094,682,459

finding BP values, no setting dominates the other one. In this regard, we present the results of the setting under which better BP value is obtained for each instance in Table 2.

In Table 2, **bold** and *italic* values indicate the same or better **BP** and *LB* values for each instance. **Gurobi with Norel heuristic** (*CPLEX*) mostly finds better **BP** (*LB*) values than CPLEX (Gurobi with Norel heuristic); however, it cannot find a solution for Problem 30. As the problem size increases, the optimality gaps increases considerably. The last column indicates the best BP values, BP*.

5.2. Computational experiments with the Benders-type heuristics

We initiate the algorithms with an initial solution $(\boldsymbol{y}, \boldsymbol{s})$ where $\boldsymbol{y}$ is found by setting its first p elements to 1 and the remaining to 0 and $\boldsymbol{s}$ is found by solving SPTree. We use a core point with $\boldsymbol{y}^0 = 1$ and $\boldsymbol{s}^0 = 0$ since computational results indicate that this core point yields stronger cuts.

Table 3 presents the results. In the table, we give GapHeur (%) defined as $100\times(UB - BP^*)/BP^*$ in order to compare the solutions of the heuristics to BP*, the best solution found by either CPLEX or Gurobi with Norel heuristic. A *positive (negative)* value indicates that the UB achieved by the heuristic algorithm is *worse (better)* than BP*. Instances with **bold UB values** are the ones for which a heuristic can either find the same solution or a better solution than CPLEX or Gurobi with Norel heuristic.

Italic (normal or ***bold)*** UB values are used to show the heuristic that produces a better UB than the other heuristic. BDHEUR1 produces better UB values for 8 problems (4, 6, 9, 10, 11, 24–26) while BDHEUR2 produces better UB values for 19 problems (2, 5, 8, 12–23, 27–30). For the remaining problems, they find the same solutions. UB values found by BDHEUR1 (BDHEUR2) are on the average 0.39% (5.2%) better than those of BDHEUR2 (BDHEUR1). Considering these results, we can conclude that BDHEUR2 performs better than BDHEUR1. In this regard, we will continue our analysis with BDHEUR2.

For TR81 instances, GapHeur values change from 0% to 3% with an average of 1.6%. The heuristic can find a solution equivalent to BP* for three instances (problems 1, 3, and 7) out of 14 instances. For the remaining instances for which BP* values are better, GapHeur values change from 0.1% to 3%.

Table 3
Test results for the instances using BDHEUR (T = 24 h).

Pr. Id.	Network	|N|	|H|	p	BP*	BDHEUR1			BDHEUR2		
						# of Iters	UB	GapHeur (%)	# of Iters	UB	GapHeur (%)
1	TR81	81	30	3	113,261,890	4287	**113,261,890**	**0.0**	854	**113,261,890**	**0.0**
2	TR81	81	30	5	103,530,000	2670	105,345,943	1.8	520	*105,027,039*	1.4
3	TR81	81	50	3	112,944,470	3018	**112,944,479**	**0.0**	368	**112,944,479**	**0.0**
4	TR81	81	50	5	102,883,188	1415	*105,242,582*	2.3	514	*105,297,298*	2.3
5	TR81	81	50	8	96,098,414	1229	98,468,128	2.5	636	*98,340,058*	2.3
6	TR81	81	50	10	93,793,657	1175	*95,893,977*	2.2	1326	96,563,871	3.0
7	TR81	81	60	3	112,944,470	2539	**112,944,479**	**0.0**	284	**112,944,479**	**0.0**
8	TR81	81	60	5	102,940,709	1422	105,242,582	2.2	446	*104,985,323*	2.0
9	TR81	81	60	8	96,098,414	815	*98,911,891*	2.9	574	98,970,843	3.0
10	TR81	81	60	10	93,793,657	709	*95,947,980*	2.3	1048	96,141,575	2.5
11	TR81	81	81	3	113,067,656	1878	***112,944,479***	***−0.1***	245	113,203,327	0.1
12	TR81	81	81	5	102,883,188	659	105,692,069	2.7	115	*104,534,769*	1.6
13	TR81	81	81	8	96,184,127	555	98,815,936	2.7	547	*98,015,933*	1.9
14	TR81	81	81	10	93,710,445	499	96,494,930	3.0	1114	*96,046,631*	2.5
							Max	3.0		Max	3.0
							Min	−0.1		Min	0.0
							Average	1.8		Average	1.6
15	PMED200	200	200	3	82,844,187	335	86,751,752	4.7	199	***82,579,184***	**−0.3**
16	PMED200	200	200	5	77,314,463	180	84,192,026	8.9	232	***77,297,928***	**−0.6**
17	PMED200	200	200	8	72,475,694	132	79,995,083	10.4	200	*73,794,169*	1.8
18	PMED200	200	200	10	71,176,399	135	78,755,379	10.6	165	*71,904,326*	1.0
							Max	10.6		Max	1.8
							Min	4.7		Min	−0.3
							Average	8.7		Average	0.6
19	PMED300	300	300	3	129,512,902	30	146,804,453	13.4	35	*134,514,203*	3.9
20	PMED300	300	300	5	126,641,742	34	137,386,549	8.5	36	*130,470,514*	3.0
21	PMED300	300	300	8	117,762,104	34	132,241,629	12.3	40	***115,380,696***	**−2.0**
22	PMED300	300	300	10	111,688,878	34	128,240,795	14.8	42	*112,950,607*	1.1
							Max	14.8		Max	3.9
							Min	8.5		Min	−2.0
							Average	12.2		Average	1.5
23	PMED400	400	400	3	188,065,072	23	177,360,845	**−5.7**	23	***165,976,456***	**−11.7**
24	PMED400	400	400	5	159,334,212	23	***156,768,212***	**−1.6**	23	157,830,551	**−0.9**
25	PMED400	400	400	8	185,954,774	23	***154,057,167***	**−17.2**	23	154,424,692	**−17.0**
26	PMED400	400	400	10	152,776,681	23	***151,006,447***	**−1.2**	23	152,523,259	**−0.2**
							Max	**−1.2**		Max	**−0.2**
							Min	**−17.2**		Min	**−17.0**
							Average	**−6.4**		Average	**−7.5**
27	PMED500	500	500	3	288,398,427	5	260,076,970	**−9.8**	5	***242,867,355***	**−15.8**
28	PMED500	500	500	5	267,833,441	5	249,145,118	**−7.0**	5	***242,398,659***	**−9.5**
29	PMED500	500	500	8	238,740,589	5	234,735,216	**−1.7**	5	***221,405,699***	**−7.3**
30	PMED500	500	500	10	4,094,682,459	5	231,288,249	**−94.4**	5	***214,659,947***	**−94.8**
							Max	−1.7		Max	−7.3
							Min	−94.4		Min	−94.8
							Average	−28.2		Average	−31.8

For PMED200 instances, GapHeur values range from −0.3% to 1.8% with an average of 0.6%. The heuristic can find a better solution for one instance (problem 15) and the same solution for one instance (problem 16) out of 4 instances. For PMED300 instances, GapHeur values change from −2% to 3.9% with an average of 1.5%. The heuristic can find a better solution for one instance (problem 21) out of 4 instances. For PMED400 instances, GapHeur values change from −17% to −0.2% with an average of −7.5%. For PMED500 instances, GapHeur values change from −94.8% to −7.3% with an average of −31.8%. The heuristic can find a better solution for all instances.

The results show that, as the network size gets larger, CPLEX or Gurobi with Norel heuristic find a solution with high optimality gaps. On the other hand, the heuristics can find solutions either close to or better than those found by CPLEX or Gurobi with Norel heuristic, i.e., Benders-type heuristic algorithms are effective in finding good solutions.

6. Conclusion

In this study, we have studied the *Multiple Allocation Tree of Hubs Location Problem* where the hub-level network is required to have a tree topology and transportation cost of sending flows between OD pairs is minimized. Most studies in the literature assume a complete network with costs satisfying the triangle inequality to formulate the problem. If the underlying real-life network is not complete or complete but its distances do not satisfy the triangle inequality, a preprocessing on the underlying network is implemented to construct a complete network whose costs satisfy the triangle inequality.

Unlike the previous studies, we have defined the problem on non-complete networks and developed a modeling approach that does not require any specific cost and network structure. The modeling approach allows us to use the structure of the real physical network directly in the formulation of the problem. We have shown that the proposed modeling approach may produce better solutions than a modeling approach that uses a complete network structure whose costs satisfy the triangle inequality, which may result from the differences in the selection of the hubs, the flow routes between hubs, and the assignments of non-hub nodes to hub nodes. The proposed approach may also provide more flexibility in modeling several characteristics real-life hub networks, e.g., the interactions between location and routing decisions, arcs with different costs and capacities, different topology and service level requirements.

In the study, we have solved the proposed model by CPLEX-based branch-and-bound algorithm and Gurobi-based branch-and-bound algorithm with Norel heuristic and developed BD-based heuristic algorithms using two acceleration strategies, namely, strong cut generation and cut disaggregation. We have conducted computational experiments using networks with up to 500 nodes. As the network size gets larger, the resulting optimality gaps are high for the solutions found by CPLEX or Gurobi with Norel heuristic. On the other hand, the heuristic can find solutions either close to or better than those found by CPLEX and Gurobi with Norel heuristic for all instances, i.e., Benders-type heuristic algorithms are effective in finding good solutions.

In the future, we may incorporate other acceleration strategies not considered in this study, e.g., reduction of the model size and selection of good initial cuts, to improve the progress of exact Benders algorithms or Benders-type heuristics. We may develop hybrid algorithms utilizing metaheuristics and Benders decomposition to improve the effectiveness of the heuristic algorithms. A problem specific branch-and-bound algorithm may be developed as well.

CRediT authorship contribution statement

Betül Kayışoğlu: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft. **İbrahim Akgün:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

This research was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK Grant No: 114M363) and the Research Fund of the Abdullah Gül University (Grant No: FDK-2018-123).

The authors are grateful to Editor, Associate Editor and anonymous referees for providing constructive feedback that has helped improve in major ways the presentation of the material in the paper.

Appendix A. Distance Matrix for a Network Not Satisfying the Triangle Inequality (Marín et al., 2006)

$$(d_{ij}) = \begin{pmatrix} 0 & 1 & 100 & 100 & 100 & 100 & 100 \\ 1 & 0 & 1 & 100 & 100 & 100 & 100 \\ 100 & 1 & 0 & 1 & 100 & 100 & 100 \\ 100 & 100 & 1 & 0 & 1 & 100 & 100 \\ 100 & 100 & 100 & 1 & 0 & 1 & 100 \\ 100 & 100 & 100 & 100 & 1 & 0 & 1 \\ 100 & 100 & 100 & 100 & 100 & 1 & 0 \end{pmatrix}$$

Appendix B. The Model for the Classical Approach

We extend the model of Ernst and Krishnamoorthy (1998) developed for multiple allocation HLP by adding necessary constraints to achieve the tree-like HLN. The resulting model produces a hub network having a tree HLN by minimizing total transportation cost and allowing multiple allocation.

Suppose that there is a complete network with n nodes. Node i generates a positive annual flow w_{ij} for at least one node $j \in N - \{i\}$. The total amount of flow originating from node i is $O_i = \sum_j w_{ij}$ and the total amount of flow sent to node j is $D_j = \sum_i w_{ij}$. Let c_{ij} denote the transportation cost of a unit of flow between i and j and α represent the discount factor for hub-to-hub journeys. Ernst and Krishnamoorthy (1998) define X_{ilj} as the flow of commodity i flowing from hub l to node j, Z_{ik} as the flow from node i to hub k, Y_{ikl} as the total amount of flow of commodity i that is routed between hubs k and l. H_k takes the value 1 if node k is a hub, 0 otherwise. In addition, we define yy_{km} that takes the value 1 when arc (k, m) links two hubs and 0 otherwise in order to define a tree-like hub level network.

Model CAM: The Model for the Classical Approach

$$\text{Min}\sum_{i}\left[\sum_{k}c_{ik}Z_{ik}+\sum_{k}\sum_{l}c_{kl}Y_{ikl}+\sum_{l}\sum_{j}\alpha c_{lj}X_{ilj}\right] \quad (53)$$

subject to:

$$\sum_{k}H_k = p \quad (54)$$

$$\sum_{k}Z_{ik} = O_i \quad \text{i} \in \text{N} \quad (55)$$

$$\sum_{l}X_{ilj} = W_{ij} \quad i,j \in \text{N} \quad (56)$$

$$\sum_{l}Y_{ikl}+\sum_{j}X_{ikj}-\sum_{l}Y_{ilk}-Z_{ik} = 0 \quad i,k \in \text{N} \quad (57)$$

$$Z_{ik} \leq O_iH_k \quad i,k \in \text{N} \quad (58)$$

$$\sum_{i}X_{ilj} \leq D_jH_i \quad l,j \in \text{N} \quad (59)$$

$$yy_{kl} \leq H_k \quad k,l \in \text{N} \quad (60)$$

$$yy_{kl} \leq H_l \quad k,l \in \text{N} \quad (61)$$

$$\sum_{k}\sum_{l}yy_{kl} = p-1 \quad (62)$$

$$Y_{ilk}+Y_{ikl} \leq O_iyy_{kl} \quad i,k,l \in \text{N} \quad (63)$$

$$H_k \leq \sum_{l}yy_{kl}+\sum_{l}yy_{lk} \quad k \in \text{N} \quad (64)$$

$$H_k \in \{0,1\} \quad k \in \text{N} \quad (65)$$

$$Y_{ikl},X_{ilj},Z_{ik} \geq 0 \quad i,j,k,l \in \text{N} \quad (66)$$

$$yy_{kl} \in \{0,1\} \quad k,l \in \text{N} \quad (67)$$

Objective function (53) together with constraints (54)-(60) and (65)-(66) constitute the formulation of Ernst and Krishnamoorthy (1998). Objective function (53) minimizes the total transportation cost. Constraint (54) locates p hubs. Constraints (55) and (56) satisfy the supply and demand requirements, respectively. Constraints (57) are the flow balance constraints. Constraints (58) and (59) ensure that flow incoming to and going from a hub node is possible only if that node is chosen as a hub. Constraints (65) and (66) define the decision variables.

Constraints (60)-(64) and (67) ensure that the hubs are connected through a tree structure. Constraints (60) and (61) require that an arc (k,m) be chosen to form the tree structure between hubs if and only if both k and m are chosen as hubs. Constraint (62) limits the number of arcs connecting hub nodes to $p-1$. Constraints (63) allow flows between hubs only in arcs selected as a part of the tree structure. Constraints (64) guarantee that a selected hub must be connected by an arc which is a part of the tree structure. Constraints (67) define the new binary variables.

Appendix C. Rooted Spanning Tree Formulation

We define one of the hub locations, say node θ, as the root/supply node from which one unit of fictitious flow is sent to other hub locations, i.e., a total of $p-1$ units of flow is sent from θ.

Model SPTree: Rooted Spanning Tree Formulation Restricted to Hub Locations

$$\text{minZ}^* = \sum_{k\in S}\sum_{(i,j)\in A_0}c_{ij}t_{ij} \quad (68)$$

s.t.

$$\sum_{j\in F_\beta^{Out}}t_{\theta j} = p-1 \quad (69)$$

$$\sum_{j\in F_\beta^{Out}}t_{\beta j}-\sum_{j\in F_\beta^{in}}t_{j\beta} = -1 \quad \beta \in (H-\theta) \quad (70)$$

$$\sum_{j\in F_\beta^{out}}t_{\beta j}-\sum_{j\in F_\beta^{in}}t_{j\beta} = 0 \quad \beta \in (N_2-H) \quad (71)$$

$$t_{ij} \geq 0 \quad (i,j) \in A_2 \quad (72)$$

References

Akgün, İ., Tansel, B.Ç., 2018. p-hub median problem for non-complete networks. Comput. Oper. Res. 95 (2018), 56–72.

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

Alumur, S.A., Kara, B.Y., Karasan, O.E., 2009. The design of single allocation incomplete hub networks. Transp. Res. part B 43 (10), 936–951.

Alumur, S., Kara, B.Y., 2009. A hub covering network design problem for cargo applications in Turkey. Journal of the Operational Research Society 60 (10), 1349–1359.

Alumur, S., Kara, B.Y., Karasan, O.E., 2012. Multimodal hub location and hub network design. Omega 40 (2012), 927–939.

Alumur, S., 2019. Hub location and related models. International Series in Operations Research and Management Science 2019 (281), 237–252.

Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. J. Oper. Res. Soc. 41 (11), 1069–1072.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4 (1), 238–252.

Birge, J.R., Louveaux, F.V., 1988. A multicut algorithm for two-stage stochastic linear programs. European Journal of Operations Research 34 (3), 384–392.

Blanco, V., Marin, A., 2019. Upgrading nodes in tree-shaped hub location. Comput. Oper. Res. 102 (2019), 75–90.

Boschetti, M., Maniezzo, V., 2009. Benders Decomposition, lagrangean relaxation and metaheuristic design. Journal of Heuristics 15 (3), 283–312.

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 (12), 3088–3096.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005a. Hub arc location problems: part I - introduction and results. Manag. Sci. 51 (10), 1540–1555.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005b. Hub arc location problems: part II - formulations and optimal algorithms. Manag. Sci. 51 (10), 1556–1571.

Campbell, J.F., 2010. Designing hub networks with connected and isolated hubs. HICSS Proc. 1–10.

Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transportation Science 46 (2), 153–169.

Carello, G., Della Croce, F., Ghirardi, M., Tadei, R., 2004. Solving the hub location problem in telecommunication network design: A local search approach. Networks 44 (2), 94–105.

Ceder, A., 2007. Public Transit Planning and Operation: Theory, Modeling, and Practice. 1st Edition, Butterworth-Heinemann.

Contreras, I., Fernández, E., Marín, A., 2009. Tight bounds from a path based formulation for the tree of hub location problem. Comput. Oper. Res. 36 (12), 3117–3127.

Contreras, I., Fernández, E., Marín, A., 2010. The tree-of-hubs location problem: A comparison of formulations. Eur. J. Oper. Res. 202 (2), 390–400.

Contreras, I., Cordeau, J.-F., Laporte, G., 2011. Benders Decomposition for Large-Scale Uncapacitated Hub Location. Oper. Res. 59 (6), 1477–1490.

Contreras, I., Tanash, M., Vidyarthi, N., 2017. Exact and heuristic approaches for the cycle hub location problem Ann. Oper Res 258 (2), 655–677.

Contreras, I., O'Kelly, M., 2019. Hub location problems. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science (2nd). Springer.

Camargo, R.S., Miranda Jr, G., Luna, H.P., 2008a. Benders decomposition for the uncapacitated multiple allocation hub location problem. Comput. Oper. Res. 35, 1047–1064.

Camargo, R.S., Miranda Jr, G., Luna, H.P., 2008b. Benders decomposition for hub location problems with economies of scale, Transp. Sci. 43, 86-97.

Camargo, R.S., Miranda Jr, G., Ferreira, R.P.M., 2011. A hybrid Outer-Approximation/ Benders Decomposition algorithm for the single allocation hub location problem under congestion. Operations Research Letters 39 (2011), 329–337.

Camargo, R.S., de Miranda, G., 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, Applied Mathematical Modelling, vol. 51, C, pp. 274–301, 2017.

Ernst, A.T., Krishnamoorthy, M., 1998. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. Eur. J. Oper. Res. 104 (1), 100–112.

Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E., 2013. Hub location problems: A review of models, classification, solution techniques, and applications. Comput. Ind. Eng. 64 (4), 1096–1109.

Floyd, R.W., 1962. Algorithm 97: Shortest Path, Communications of the Association for Computing. Machinery 5 (6), 345. https://doi.org/10.1145/367766.368168.

Ghaffarinasab, N., Kara, B.Y., 2019. Benders Decomposition Algorithms for Two Variants of the Single Allocation Hub Location Problem. Networks and Spatial Economics 19 (1), 83–108.

Gurobi, 2021. https://www.gurobi.com/documentation/9.1/refman/norelheurwork.html, accessed 01.05.2021.

Jiang, W., Tang, L., Xue, S., 2009. A hybrid algorithm of tabu search and Benders decomposition for multi-product production distribution network design. Proceedings of the IEEE International Conference on Automation and Logistics. ICAL'09. Shenyang China, pp. 79-84.

Johnson, D. S., Lenstra J. K., and Rinnooy Kan A. H. G., 1978. The complexity of the network design problem. Networks 8 (4), 279–285.

Klincewicz, John G, 1998. Hub location in backbone/tributary network design: a review. Loc Sci 6 (1-4), 307–335.

Labbé, M., Yaman, H., 2008, Solving the hub location problem in a start-start network. Networks51:19–33.

Lai, M.C., Sohn, H.S., Tseng, T.L., Chiang, C., 2010. A hybrid algorithm for capacitated plan location problem. Expert Syst. Appl. 37 (12), 8599–8605.

Lai, M.C., Sohn, H.S., Tseng, T.L., Bricker, D.L., 2012. A hybrid Benders/genetic algorithm for vehicle routing and scheduling problem. International Journal of Industrial Engineering 19 (1), 33–46.

Lee, Chang-ho, Ro, Hyung-bong, Tcha, Dong-wan, 1993. Topological design of a two-level network with ring-star configuration. Comput Oper Res 20 (6), 625–637.

Localsolver, 2021. https://www.localsolver.com/product.html, accessed 01.05.2021.

Magnanti, T.L., Wong, R.T., 1981. Accelerating benders decomposition: algorithmic enhancement and model selection criteria. Oper. Res. 29 (3), 464–484.

Marín, Alfredo, Cánovas, Lázaro, Landete, Mercedes, 2006. New formulations for the uncapacitated multiple allocation hub location problem. Eur. J. Oper. Res. 172 (1), 274–292.

Martins de Sá, E.M., De Camargo, R.S., De Miranda, G., 2013. An improved Benders decomposition algorithm for the tree of hubs location problem, Eur. J. Oper. Res. 226, 185–202.

Martins de Sá, E., Contreras, I., Cordeau, J.-F., Saraiva de Camargo, R., de Miranda, G., 2015a. The hub line location problem. Transp. Sci. 49 (3), 500–518.

Martins de Sá, Elisangela, Contreras, Ivan, Cordeau, Jean-François, 2015b. Exact and heuristic algorithms for the design of hub networks with multiple lines. Eur. J. Oper. Res. 246 (1), 186–198.

Martins de Sá, E.M., Morabito, R., de Camargo, R.S., 2018a. Benders decomposition applied to a robust multiple allocation incomplete hub location problem. Comput. Oper. Res. 89 (2018), 31–50.

Martins de Sá, Elisangela, Morabito, Reinaldo, de Camargo, Ricardo Saraiva, 2018b. Efficient Benders decomposition algorithms for the robust multiple allocation incomplete hub location problem with service time requirements. Expert Syst Appl 93, 50–61.

Mercier, Anne, Cordeau, Jean-François, Soumis, François, 2005. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. Comput. Oper. Res. 32 (6), 1451–1476.

Mohri, Seyed Sina, Akbarzadeh, Meisam, 2018. Incomplete hub location model for designing van-taxi networks. Transp. Res. Rec. 2672 (8), 619–628.

Mokhtar, H., Krishnamoorthy, M., Ernst, A.T., 2019. The 2-allocation *p*-hub median problem and a modified Benders decomposition method for solving hub location problems. Comput. Oper. Res. 104 (2019), 375–393.

Nickel, S., Schobel, A., Sonneborn, T., 2001. Hub Location problems in urban traffic networks. In: Niittymaki J, Pursula M, editors. Mathematics methods and optimization in transportation systems Kluwer Academic Publishers; 2001. p. 1–12. [Chapter 1].

Pacqueau, R., Francois, S., Le Nguyen, H., 2012. A fast and accurate algorithm for stochastic integer programming, applied to stochastic shift scheduling. Publication G-2012-29,Groupe d'etudes et de recherché en analyse de decisions (GERAD), Universite de Montreal, Montreal, QC, Canada.

Poojari, C.A., Beasley, J.E., 2009. Improving benders decomposition using a genetic algorithm. Eur. J. Oper. Res. 199 (2009), 89–97.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders Decomposition Algorithm: A Literature Review. European Journal of Operational Research Volume 259, Issue 3, 16 June 2017, Pages 801-817.

Sabattin, J., Fuertas, G., Alfaro, M., Quezada, L., Vargas, M., 2018. Optimization of large electric power distribution using a parallel genetic algorithm with dandelion strategy. Turkish Journal of Electrical Engineering and Computer Sciences. 26 (5), 1–13.

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

Taherkhani, Gita, Alumur, Sibel A., Hosseini, Mojtaba, 2020. Benders Decomposition for the Profit Maximizing Capacitated Hub Location Problem with Multiple Demand Classes. Transportation Science 54 (6), 1446–1470.

Vuchic, V.R., 2007. Urban Transit Systems and Technology. John Wiley & Sons Inc, Hoboken, New Jersey.

Yaman, Hande, 2008. Star p-hub median problem with modular arc capacities. Comput Oper Res 35 (9), 3009–3019.

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. Journal of the Operational Research Society 59 (1), 80–89.

Zhong, W., Juan, Z., Zong, F., Su, H., 2018. Hierarchical hub location model and hybrid algorithm for integration of urban and rural public transport. International Journal of Distributed Sensor Networks. April 2018.

Zhou, J., Peng, J., Liang, G., Deng, T., 2019. Layout optimization of tree-tree gas pipeline network. J Petrol Sci Eng, 173 (2019), pp. 666-680.s.