Computers and Operations Research 85 (2017) 32–44

Contents lists available at ScienceDirect

Computers and Operations Research

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

An exact algorithm for the modular hub location problem with single assignments

Moayad Tanash, Ivan Contreras*, Navneet Vidyarthi

Concordia University, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, QC H3G 1M8, Canada

ARTICLE INFO

Article history:
Received 11 March 2016
Revised 28 February 2017
Accepted 15 March 2017
Available online 23 March 2017

Keywords:
Hub location
Flow-dependent costs
Lagrangean relaxation

ABSTRACT

A key feature of hub-and-spoke networks is the consolidation of flows at hub facilities. The bundling of flows allows reduction in the transportation costs, which is frequently modeled using a constant discount factor that is applied to the flow cost associated with all interhub links. In this paper, we study the modular hub location problem, which explicitly models the flow-dependent transportation costs using modular arc costs. It neither assumes a full interconnection between hub nodes nor a particular topological structure, instead it considers link activation decisions as part of the design. We propose a branch-and-bound algorithm that uses a Lagrangean relaxation to obtain lower and upper bounds at the nodes of the enumeration tree. Numerical results are reported for benchmark instances with up to 75 nodes.

1. Introduction

Hub location problems (HLPs) lie at the heart of network design planning in logistics systems such as the trucking and airline industries. These systems frequently employ hub-and-spoke architectures to efficiently route commodities or passengers between many origins and destinations. Their key feature is the use of transshipment or consolidation points, typically called hubs, to connect a large number of origin/destination (O–D) pairs with only a small number of links. This strategy centralizes handling and sorting operations and reduces set-up costs; most importantly, it makes it possible to achieve economies of scale on routing costs through the consolidation of flows.

HLPs are a challenging class of *NP*–hard combinatorial optimization problems combining location and arc selection decisions. The location decision problem involves the selection of a set of nodes at which hub facilities can be located; the arc selection decision problem addresses the design of the hub network by choosing the links to connect origins, destinations, and hubs, establishing a framework for the routing of commodities through the network. Broadly speaking, the aim of HLPs is to determine the locations of the hubs and the design the hub network so as to minimize the total flow cost (see Alumur and Kara, 2008; Campbell and O'Kelly, 2012; Zanjirani Farahani et al., 2013; Contreras, 2015).

HLPs have received increasing attention since the seminal work of O'Kelly (1986). Analogous to the literature on discrete facility location problems, several classes of HLPs have been studied, including uncapacitated hub location, p-hub median, hub covering, and p-hub center problems. The various applications within each class give rise to variants that differ in terms of assumptions, such as the required topological structure, the allocation pattern of nodes to hubs, and the existence of capacity constraints on the hub nodes or arcs. Nonetheless, there are four common assumptions underlying most HLPs. The first assumption is that commodities have to be routed via a set of hubs, so O–D paths must include at least one hub node. The network induced by the solution of a hub location problem consists of two types of arcs: hub arcs connecting two hubs; and access arcs connecting O–D nodes to hubs. For some applications, in addition to enabling economies of scale, hub facilities may act as consolidation, sorting, and distribution centers. The second assumption is that the hubs can be fully interconnected with more efficient, higher volume pathways that allow a constant discount factor ($0 < \alpha < 1$) to be applied to all transportation costs associated with the commodities that are routed between any pair of hubs. Note that the discount factor is assumed to be independent of the amount of flow that is sent through hub arcs. The third assumption is that hub arcs incur no set-up costs, so hubs can be connected at no extra cost. The forth one is that distances between nodes satisfy the triangle inequality. As a result, the backbone network is typically a complete graph, i.e., the hubs are fully interconnected at no cost.

These assumptions and their implications simplify network design decisions as they are determined mainly by the allocation

* Corresponding author.
E-mail addresses: m_tanash@encs.concordia.ca (M. Tanash), icontrer@encs.concordia.ca (I. Contreras), navneetv@jmsb.concordia.ca (N. Vidyarthi).

http://dx.doi.org/10.1016/j.cor.2017.03.006
0305-0548/

pattern of O–D nodes to hub facilities. As a result, classical HLPs have a number of attractive theoretical features which have given rise to mathematical models that exploit the structure of the network (Alumur et al., 2012; Contreras and Fernández, 2014; Correia et al., 2014; 2010; Ernst and Krishnamoorthy, 1998a; Hamacher et al., 2004; Labbé and Yaman, 2004) and to sophisticated solution algorithms that are able to solve real-size instances (Contreras et al., 2011a; 2011c; Ernst and Krishnamoorthy, 1998b; Labbé et al., 2005; Martins de Sá et al., 2015b).

However, these assumptions may lead to unrealistic results. The independence of flow discounted costs is appropriate in applications in which the links between hubs are associated with faster transportation modes, but it can be an oversimplification in applications where the costs represent the economies of scale due to the bundling of flows on the hub arcs. For instance, full interconnection between hub nodes could lead to solutions in which hub arcs carry considerably less flow than access arcs, yet the transportation costs are discounted only on the hub arcs. It may also be the case that the amounts of flow that are routed on various hub arcs are different, yet the same discount factor is applied across the board. Under the assumption of flow-independent costs and the use of fully interconnected hubs, the overall transportation cost may be miscalculated, and the set of hub nodes selected and the corresponding allocation pattern of O–D nodes to hubs may be suboptimal.

In this paper, we study a *modular hub location problem* (MHLP) which considers explicitly the flow dependence of transportation costs based on modular arc costs. Thus, the total transportation cost is estimated not in terms of the per unit flow cost but in terms of the number of facility links used on each arc, eliminating the use of nonlinear functions and their linearizations to compute the discount factor for each hub arc. The cost is modeled using a stepwise function that determines, for each arc on the network, the total transportation cost as a function of the amount of flow routed through the arc. Our approach can be interpreted in terms of its ability to incorporate multiple capacity levels on the arcs. Another advantage is that it neither assumes a fully interconnected hub network nor a particular topological structure, instead it considers the design of the hub network as part of the decision process. Other variants of MHLP involving multiple assignments and direct connections were initially introduced in Mirzaghafour (2013).

The assumption of modular (or stepwise) transportation costs is consistent with applications in freight transportation and telecommunications networks. In the case of ground transportation, trucking companies send commodities (e.g., goods, express packages, ordinary mail) along hub arcs between break bulk terminals, and along access arcs between an end-of-line terminal and a break bulk terminal, using one or more trucks. The number and capacity of the trucks and the distance traveled can be used to obtain an accurate estimate of the transportation cost between terminals. Here, fixed costs represent the cost of leasing or buying a truck, whereas variable costs may represent the average fuel and labor costs for operating a truck to travel a given distance. The consolidation of flows at hubs allows trucking companies to use large line-haul trucks, typically fully loaded, between hub facilities. Local delivery trucks, typically partially loaded, are used between break bulk and end-of-line terminals to route commodities from origin to destination nodes. Even though both the fixed and variable costs for line-haul trucks are greater than those for local delivery trucks, the per unit transportation cost for hub arcs is lower than that for the access arcs because the trucks have larger capacities. An analogous situation is the use of regional and hub airports by air cargo companies to efficiently route commodities between many origins and destinations. The transportation cost between airports can be estimated based on the number and capacity of the cargo planes, together with the distance.

In the case of telecommunications networks, hub facilities correspond to electronic devices such as multiplexors, concentrators, servers, etc. Commodities correspond to data transmissions that are frequently routed over a variety of physical media (i.e., fiber optic cables, co-axial cables, or telephone lines). The number and capacity of these physical media can be used to provide an estimation of the transmission cost between pair of nodes. Moreover, the modular cost may also represent the usually large set-up cost of the communication links.

Several papers have already pointed out that the discount factor should be regarded as function of the flow volume (see, O'Kelly, 1998; O'Kelly and Bryan, 1998; Bryan and O'Kelly, 1999; Campbell, 2013). O'Kelly and Bryan (1998) were among the first to develop a hub location model that expresses the discount factor on hub arcs as a function of flow. It was later extended by Bryan (1998), Klincewicz (2002) and de Camargo et al. (2009). However, their models use a nonlinear cost function to compute the transportation costs on a hub arc as a function of its flow. This function is approximated by a piecewise linear function to obtain a linear integer programming formulation for the problem. Horner and O'Kelly (2001) proposed a nonlinear cost function based on link performance functions; it is designed to reward economies of scale in all arcs in the network. Podnar et al. (2002) formulated a network design model in which the discount factor applies only on arcs that have flows larger than a given threshold; however, the model focuses on the design of the network rather than on the location of the hub facilities. Racunica and Wynter (2005) introduced a nonlinear concave cost hub location model that determines the optimal location of intermodal freight hubs. The cost function models the flow-dependent discounted cost only on origin-to-hub and hub-to-destination legs.

Yoon and Current (2006) and O'Kelly et al. (2015) adopted a different approach for modeling economies of scale on all the arcs in a hub-and-spoke network. Rather than relying on a nonlinear cost function, they use linear cost functions which combine variable transportation costs for flows on arcs and fixed costs for activating those arcs. In these approaches, the use of fixed costs of arcs allows the link costs per unit of flow to decrease as the flow increases on that link, resulting in the economies of scale. Kimms (2006) presented three different models for hub location problems with fixed and variable costs. He introduced a model, similar to ours, in which the goal is to determine the optimal number of vehicles used on each arc to route flow through the fully interconnected network. However, unlike Kimms (2006), we do not assume a fully interconnected hub network. Cunha and Silva (2007) designed a hub-and-spoke network for a less-than-truckload trucking company in Brazil based on a nonlinear cost function that allows the discount factor on hub arcs to vary according to the total amount of freight between hubs. Campbell et al. (2005a); 2005b) study hub arc location problems, in which the goal is to locate a set of hub arcs, therefore, no longer considering a fully interconnected hub network. To some extent, this mitigates the limitations of flow-independent costs. Other studies consider hub location models focusing on the design of particular topological structures such as star-star networks (Labbé and Yaman, 2008), tree-star networks (Contreras et al., 2009b; Martins de Sá et al., 2013), cycle-star networks (Contreras et al., 2016), and hub line networks (Martins de Sá et al., 2015a; 2015b).

The MHLP is also related to other hub location models where capacities are considered at the arcs of the network. Sasaki and Fukushima (2003) study a capacitated multiple allocation HLP where capacity constraints are considered both on hub nodes and hub arcs. However, in their model, the flow between each O-D pair can go through at most one hub facility and hence there is no discount between hubs. Yaman and Carello (2005) introduce the capacitated single assignment hub location problem with modular link capacities (CHLP-ML) in which additional capacity constraints are considered on the incoming and outgoing flow at hubs. The CHLP-ML assumes hubs to be fully interconnected and does not consider O/D paths containing more than two hub nodes and one hub arc. Even though it is not explicitly mentioned in the paper, this model can be seen as one in which flow-dependent costs are considered. The authors present a quadratic mixed-inter programming formulation and compare different linearizations schemes based on the properties of the optimal solution. They also present a branch-and-cut algorithm and a tabu search heuristic for solving it. Corberán et al. (2016) propose a metaheuristic algorithm based on strategic oscillation for the CHLP-ML that improves on the results obtained in Yaman and Carello (2005). However, to the best of our knowledge, the best heuristic algorithm for the CHLP-ML is given in Hoff et al. (2016), where a heuristic based on adaptive memory programing is developed to solve it. Yaman (2008) present a hub location model for a star-star network with modular link capacities in which hub nodes are directly connected to a central node. Rastani et al. (2015) study a capacitated single-allocation HLP in which the capacities of hubs and hub links are parts of the decision process. The proposed model considers a flow-independent discount factor and assume hubs to be fully interconnected.

In this paper, we present two mixed integer programming (MIP) formulations for the MHLP. The first formulation uses flow variables to compute the flow through hub arcs, whereas the second formulation uses path variables to determine whether a specified hub arc lies on the path between a pair of nodes. We propose a Lagrangean relaxation for the path-based formulation (PF) of MHLP by relaxing the linking constraints of the location/allocation and routing variables. This makes it possible to decompose the Lagrangean function into two independent subproblems which can be solved efficiently. We also propose a heuristic algorithm to obtain feasible solutions. To prove optimality, we develop a branch-and-bound algorithm that uses the Lagrangean relaxation and a heuristic to obtain lower and upper bounds at the nodes of the enumeration tree.

The remainder of this paper is organized as follows. Section 2 formally defines the problem and presents the proposed formulations. In Section 3, we describe the proposed Lagrangean relaxation and study the structure of the subproblems and their solutions. Section 3.2 describes the primal heuristic algorithm. While in Section 4, we present a branch-and-bound algorithm. The computational results and analysis are presented in Section 5, followed by conclusions in Section 6.

2. Problem definition and formulation

Let $G = (N, A)$ be a complete digraph without loops, where $N = \{1, \dots, n\}$ is the set of nodes and A is the set of arcs. Let N also represent the set of potential locations, and let W_{ij} denote the amount of flow between nodes $i \in N$ and $j \in N$. Thus, $O_i = \sum_{j \in N} W_{ij}$ is the total flow originating at node $i \in N$, and $D_i = \sum_{j \in N} W_{ji}$ is the total flow destined to node $i \in N$. For each $i \in N$, f_i is the set-up cost for locating a hub facility. The distances between nodes i and j, $d_{ij} \geq 0$, are assumed to be neither symmetric nor satisfy the triangular inequality.

To estimate the transportation cost on both access and hub arcs, our model determines the number of facility links with a given capacity that will be needed to route the flow on these arcs. There is a fixed cost associated with these capacitated facility links as well as variable cost that depends on the amount of flow. Therefore, the transportation costs on arcs are modeled using a step-wise function. A hub arc $(k, m) \in A$ connects two different hub nodes k and m and has an associated transportation cost $c_{km} = l_c + b \times d_{km}$ for each facility link with capacity B used to route flow from k to m, where l_c and b represent the fixed and variable costs, respectively. Note that transportation costs of hub arcs model costs incurred when routing flow between two different hub facilities but not within the same facility. An access arc $(i, k) \in N \times N$ connects two nodes i and k, not necessarily different, and has an associated transportation cost $q_{ik} = l_q + p \times d_{ik}$ for each facility link with capacity R used to route flow from i to k, where l_q and p represent the fixed and variable costs, respectively. Note that access arcs model the transportation cost to collect and distribute flow between O-D nodes and hub facilities, even when an O-D node is selected to be a hub, i.e. $i = k$.

For each $i \in N$, let $v_i^1 = \lceil O_i/R \rceil$ denote the number of facility links required to route the flow originating from i directly to a hub, and let $v_i^2 = \lceil D_i/R \rceil$ denote the number of facility links required to route the flow from a hub directly to destination i. This means that, because of the single assignment assumption, the number of facility links required to connect each non-hub node to any hub node can be determined a priori and thus, it is not part of the decision process. The transportation costs on access arcs are thus flow independent because the number and actual utilization of the access arcs depend only on the flows O_i and D_i and the capacity R, which are parameters, and not on the assignment decisions. In order to account for the flow-dependent economies of scale when consolidating flows at hub facilities and using more efficient paths between hubs, we assume the following inequalities on hub arcs and access arcs: $B > R$, $b > p$ and $l_c > l_q$. This ensures that if facility links were fully loaded (or highly utilized), the unit transportation cost on hub arcs would be less than the unit flow cost on access arcs. That is, $\frac{c_{km}}{B} < \frac{q_{km}}{R}$. However, when links are only partially utilized it may happen that the unit transportation costs on hub arcs could be higher than those of access arcs.

Under these assumptions, the MHLP consists of locating a set of hub facilities, activating a set of hub arc facility links, allocating each node to exactly one hub, and determining the route of flows through the network such that the total setup and transportation cost is minimized. The model assumes a single assignment pattern of O-D nodes to hubs. As it is the case in other well-known hub location models with single assignments (Contreras et al., 2011c; Ernst and Krishnamoorthy, 1996), this assumption is consistent with applications in which outgoing and incoming flows of each non-hub node have to be processed by a single hub facility due to managerial or contractual reasons. However, an interesting feature of the MHLP is that it does not make any assumption on a particular topological structure to connect hub facilities. Instead, it considers a fixed set-up cost for the activation of hub arcs, allowing the model to select the most cost effective hub-level network structure. These features make the MHLP a very challenging problem to solve. Even if the location of hubs and the assignment of non-hub nodes to hubs are given, the remaining subproblem of activating facility links on the hub-level network is still *NP*-hard as it is equivalent to the well-known *network loading problem* (Magnanti et al., 1995).

In what follows, we present two MIP formulations for the MHLP based on the widely used path-based and flow-based formulations for classical HLPs (see, Contreras, 2015).

2.1. Path-based formulation

For each $i, k \in N$, we define binary variables z_{ik} equal to one if non-hub i is assigned to hub k. Note that, when $z_{kk} = 1$, node k is selected as a hub and assigned to itself. For each $(k, m) \in A$ we define integer variables y_{km} equal to the number of hub arcs between hub nodes k and m. For each $i, j, k, m \in N$, we also introduce continuous routing variables x_{ijkm} equal to the fraction of the flow originating from i and destined to j that is routed via hub arc (k, m). Using these sets of variables, the MHLP can be formulated as follows:

$$(PF)\text{minimize} \sum_{k\in N} f_k z_{kk} + \sum_{i\in N}\sum_{k\in N}\left(q_{ik}v_i^1 + q_{ki}v_i^2\right)z_{ik} + \sum_{(k,m)\in A} c_{km}y_{km}$$

$$\text{subject to} \sum_{k\in N} z_{ik} = 1 \quad i\in N \tag{1}$$

$$z_{ik} \le z_{kk} \quad i,k\in N \tag{2}$$

$$z_{ik} + \sum_{m\in N} x_{ijmk} = z_{jk} + \sum_{m\in N} x_{ijkm} \quad i,j,k\in N, i\neq j \tag{3}$$

$$\sum_{i\in N}\sum_{j\in N} W_{ij}x_{ijkm} \le By_{km} \quad (k,m)\in A \tag{4}$$

$$y_{km} \le Qz_{mm} \quad (k,m)\in A \tag{5}$$

$$y_{km} \le Qz_{kk} \quad (k,m)\in A \tag{6}$$

$$z_{ik}\in\{0,1\} \quad i,k\in N \tag{7}$$

$$y_{km}\in Z^+ \quad (k,m)\in A \tag{8}$$

$$0\le x_{ijkm}\le 1 \quad i,j,k,m\in N. \tag{9}$$

The objective function minimizes the sum of setup costs for locating hub facilities and the transportation cost on access and hub arcs. Note that when node i is assigned to hub k, the transportation costs of both access arcs (i, k) and (k, i) are considered to represent the collection and distribution cost, even when $i = k$. However, in the case of the transportation costs of hub arcs, these are considered only between two different hubs nodes k and m. Constraints (1) ensure that each non-hub node is assigned to exactly one hub. Constraints (2) ensure that each node is assigned to an open hub. Constraints (3) are the well-known flow conservation constraints, that are used to model O–D paths. Constraints (4) are capacity constraints that limit the amount of flow on each hub arc (k, m). Constraint (5) and (6) ensure that hub arc (k, m) is established only if k and m are hub nodes. Q is a sufficiently large number representing an upper bound on the number of hub arcs between hub nodes k and m. In this case, Q is set to $\lceil \Sigma_{i\in N}\Sigma_{j\in N}W_{ij}/B\rceil$. Constraints (7)–(9) are usual integrality and non-negativity constraints.

2.2. Flow-based formulation

For each $i \in N$ and $(k, m) \in A$, we define X_{ikm} equal to the amount of flow with origin i that traverse hub arc (k, m). We also use the z_{ik} and y_{km} variables for the location/allocation and network design decisions. The MHLP can then be formulated as follows:

$$(FF)\text{minimize} \sum_{k\in N} f_k z_{kk} + \sum_{i\in N}\sum_{k\in N}\left(q_{ik}v_i^1 + q_{ki}v_i^2\right)z_{ik} + \sum_{(k,m)\in A} c_{km}y_{km}$$

$$\text{subject to } (1)-(2),\ (5)-(8)$$

$$\sum_{j\in N} W_{ij}z_{jk} + \sum_{m\in N} X_{ikm} - \sum_{m\in N} X_{imk} - O_i z_{ik} = 0 \quad i,k\in N \tag{10}$$

$$\sum_{i\in N} X_{ikm} \le By_{km} \quad (k,m)\in A \tag{11}$$

$$X_{ikm}\ge 0 \quad i,k,m\in N. \tag{12}$$

Constraints (10) are the flow conservation constraints, whereas (11) are the capacity constraints.

3. Lagrangean relaxation

Lagrangean relaxation (LR) is a well-known decomposition technique that exploits the inherent structure of the problem to obtain dual bounds on the optimal solution value (see, Guignard, 2003). LR has been successfully applied to solve different variants of HLPs (An et al., 2015; Contreras et al., 2011b; 2009a). We now present a LR that is based on formulation PF for the MHLP. In the next section, we embed this relaxation into a branch-and-bound algorithm to obtain optimal solutions.

In the case of PF, if we relax constraints (3), (5), and (6), in a Lagrangean fashion, weighting their violations with multiplier vectors λ^1, $\lambda^2 \ge 0$, $\lambda^3 \ge 0$ of appropriate dimension, we obtain the following Lagrangean function:

$$\begin{aligned} L(\lambda^1,\lambda^2,\lambda^3) = \min &\sum_{k\in N} f_k z_{kk} + \sum_{i\in N}\sum_{k\in N} q_{ik}(v_i^1+v_i^2)z_{ik} + \sum_{(k,m)\in A} c_{km}y_{km} \\ &+ \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\lambda^1_{ijk}\Big(z_{ik} + \sum_{m\in N} x_{ijmk} - z_{jk} - \sum_{m\in N} x_{ijkm}\Big) \\ &+ \sum_{(k,m)\in A}\lambda^2_{km}(y_{km} - Qz_{mm}) + \sum_{(k,m)\in A}\lambda^3_{km}(y_{km} - Qz_{kk}) \\ &\text{s.t. } (1)-(2),\ (4),\ \text{and } (7)-(9). \end{aligned}$$

For a given value of the Lagrangean multipliers $(\lambda^1, \lambda^2, \lambda^3)$, the Lagrangean function $L(\lambda^1, \lambda^2, \lambda^3)$ can actually be decomposed into two independent subproblems: one in the space of z variables and the other in the space of (x, y) variables. The subproblem in the space of z variables is:

$$\begin{aligned} L_z(\lambda^1,\lambda^2,\lambda^3) = \min &\sum_{k\in N}\bar{F}_k z_{kk} + \sum_{i\in N}\sum_{k\in N}\bar{A}_{ik}z_{ik} \\ &\text{s.t. } (1),\ (2),\ (7), \end{aligned}$$

where the coefficients of the objective function are:

- $\bar{F}_k = f_k - \sum_{m\in N} Q\lambda^2_{mk} - \sum_{m\in N} Q\lambda^3_{km}$,
- $\bar{A}_{ik} = q_{ik}(v_i^1+v_i^2) + \sum_{j\in N}(\lambda^1_{ijk} - \lambda^1_{jik})$.

Observe that the $L_z(\lambda^1, \lambda^2, \lambda^3)$ can be evaluated by solving a classical *uncapacitated facility location problem* (UFLP) (Cornuejols et al., 1983). Even though this problem is known to be NP-hard, it can be solved in reasonable CPU times using ad-hoc solution algorithms.

The subproblem in the space of the (x, y) variables can be expressed as:

$$\begin{aligned} L_{x,y}(\lambda^1,\lambda^2,\lambda^3) = \min &\sum_{(k,m)\in A}\bar{R}_{km}y_{km} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{m\in N}\bar{M}_{ijkm}x_{ijkm} \\ &\text{s.t. } (4),\ (8),\ (9), \end{aligned}$$

where the coefficients of the objective function are:

- $\bar{R}_{km} = \lambda^2_{km} + \lambda^3_{km} + c_{km}$,
- $\bar{M}_{ijkm} = \lambda^1_{ijm} - \lambda^1_{ijk}$.

Given that each of the y_{km} variables appear in exactly one constraint, we can further decompose $L_{x,y}(\lambda^1, \lambda^2, \lambda^3)$ into several independent subproblems, one for each (k, m) pair, of the form:

$$\begin{aligned} L^{k,m}_{x,y}(\lambda^1,\lambda^2,\lambda^3) = \min\ &\bar{R}_{km}y_{km} + \sum_{i\in N}\sum_{j\in N}\bar{M}_{ijkm}x_{ijkm} \\ &\text{s.t. } (4),\ (8),\ (9). \end{aligned}$$

For a given candidate hub arc (k, m), the subproblem computes the optimal number of facility links to open and the commodities to be routed on this hub arc. These subproblems can be efficiently solved by iteratively setting y_{km} to a non-negative integer value and finding the optimal value for the associated x_{ijkm} variables. That is, upon fixing y_{km} the problem reduces to a continuous knapsack problem, which can be optimally solved with a greedy knapsack algorithm (Lawler, 1979). This algorithm first orders the x_{ijkm} variables so that

$$\frac{\bar{M}_{(s)km}}{W_{(s)}} \le \frac{\bar{M}_{(s+1)km}}{W_{(s+1)}},$$

for $s = 1, \dots, n^2 - n$, where $W_{(s)}$ denotes the demand flow of the s^{th} ordered node pair (i, j). Starting from $s = 1$, the algorithm adds the ordered items, i.e., $x_{(s)km} = 1$, one at a time to the knapsack and continues until the residual capacity is equal to zero or $M_{(s)km} > 0$. Note that only a fraction of the last considered item (denoted as r) may have been added, i.e., $x_{(r)km} = \left(By_{km} - \sum_{s=1}^{r-1} W(s)\right)/W(r) < 1$.

To determine the optimal value of y_{km}, the algorithms starts from $y_{km} = 1$ and evaluates the objective value by solving the corresponding continuous knapsack problem. If the objective value is strictly negative, y_{km} is increased by one to add B extra units of capacity to the knapsack so as to allow more $x_{(s)km}$ variables to take a positive value. The value of y_{km} is increased until the capacity increases to a point that all $x_{(s)km}$ can be set to one or whenever the next element to be added deteriorates the objective (i.e., $\bar{M}_{(s)km} \ge 0$). A value of $y_{km} = 0$ is selected as optimal whenever setting $y_{km} \ge 1$ yields strictly positive objective values.

3.1. Solving the lagrangean dual problem

In order to obtain the best possible lower bound, we solve the *Lagrangean Dual* problem, which is given by:

$$(LD) \quad L_D = \max_{\substack{\lambda^1 \\ \lambda^2, \lambda^3 \ge 0}} L(\lambda^1, \lambda^2, \lambda^3).$$

We apply the subgradient optimization method to solve problem *LD*. It is well known that the classical subgradient algorithm tends to suffer from slow convergence. To overcome this difficulty, we use a deflected subgradient algorithm. This algorithm uses a linear combination of the current subgradient direction s^t and the direction used in the previous iteration d^{t-1} to obtain the next direction of movement. That is, at every iteration t, $d^t = s^t + \theta^t d^{t-1}$. The efficiency of this method depends on selecting the deflected subgradient parameter θ^t (see for instance, Camerini et al., 1975; Brännlund, 1995). To this end, we use the following rule based on geometrical arguments (see Guta, 2003):

$$\theta^t = \begin{cases} -\pi \frac{s^t d^{t-1}}{\|d^{t-1}\|^2} & if\ s^t d^{t-1} < 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $0 \le \pi \le 2$. For a given vector $(\lambda^1, \lambda^2, \lambda^3)$, let $z(\lambda)$, $y(\lambda)$, and $x(\lambda)$ be the optimal solution to $L(\lambda^1, \lambda^2, \lambda^3)$. Thus, a subgradient of $L(\lambda^1, \lambda^2, \lambda^3)$ is given by

$$s(\lambda^1, \lambda^2, \lambda^3) = \Bigg(\Bigg(z_{ik}(\lambda) + \sum_{m \in N} x_{ijmk}(\lambda) - z_{jk}(\lambda) - \sum_{m \in N} x_{ijkm}(\lambda) \Bigg)_{(i,j,k)}, \\ (y_{km}(\lambda) - Qz_{kk}(\lambda))_{(k,m)}, \\ (y_{km}(\lambda) - Qz_{mm}(\lambda))_{(k,m)} \Bigg).$$

At each iteration t of the subgradient algorithm, the dual multipliers are updated as:

$$(\lambda^1, \lambda^2, \lambda^3)^{(t+1)} = (\lambda^1, \lambda^2, \lambda^3)^{(t)} + \delta^t \frac{\bar{\phi} - L((\lambda^1, \lambda^2, \lambda^3)^t)}{||\Gamma(\lambda^1, \lambda^2, \lambda^3)^t||^2} d^t,$$

where $\bar{\phi}$ denotes an upper bound on the optimal solution value and δ is a constant between 0 and 2.

3.2. Primal heuristic

We exploit the information generated at some iterations of the subgradient algorithm to construct feasible solutions. In what follows, solutions are represented by a set of hub nodes H, a set of hub arcs D, and an assignment mapping M. Solutions are designated in the form $s = (H, D, M)$, where H represents the set of selected sites at which hubs are located, i.e., $H(i) = 1$ if site $i \in N$ is chosen to be a hub, $D((i, j)) : A \to Z^+$ represents the number of facility links installed on hub arcs (i, j) and M: $N \to H$ is the assignment mapping, i.e., $M(j) = k$ if node $j \in N$ is assigned to hub $k \in H$.

The proposed heuristic constructs feasible solutions as follows. Let $\hat{z}^t$, $\hat{y}^t$ and $\hat{x}^t$ be the optimal solution to the Lagrangean subproblems $L_z(\lambda)$ and $L_{x,\,y}(\lambda)$ at a given iteration t of the subgradient algorithm. The optimal solution of the subproblem $L_z(\lambda^t)$ provides a set of hubs and an assignment mapping of non-hub nodes to hubs, that is $H = \{k : \hat{z}_{kk} = 1, k \in N\}$, and $M(i) = \bar{k}$ where $\hat{z}_{i\bar{k}} = 1$. Since $L_z(\lambda^t)$ and $L_{x,\,y}(\lambda^t)$ are solved independently, the solution obtained from the subproblem $L_{x,\,y}(\lambda^t)$ might not be feasible for the set H of hubs obtained in solving L_z. Therefore, once the location/allocation variables are fixed, the next step is to determine the number of facility links to be activated on each hub arc in order to route the flows at minimum cost. This subproblem is actually equivalent to solving a *network loading problem* (NLP) on an auxiliary network.

Let $\hat{G} = (\hat{H}, \hat{A})$ be a directed graph where $\hat{H} = \left\{k \in N : \hat{z}_{kk} = 1\right\}$ is the set of open hubs at iteration t and $\hat{A} = \{(k, m) \in A : k, m \in \hat{H}\}$ is the set of candidate hub arcs. For each pair $(k, m) \in \hat{H} \times \hat{H}$, let $w_{km} = \sum_{i \in O(k)} \sum_{j \in O(m)}$ denote the amount of flow that needs to be routed from k to m, where $O(k) = \left\{i \in N : \hat{z}_{ik} = 1\right\}$. Recall that c_{km} represents the (transportation) cost for using one facility link with capacity B on hub arc (k, m). Using the y_{km} and x_{ijkm} variables defined in Section 2, the NLP can be formulated as:

$$\text{minimize} \sum_{(k,m) \in \hat{A}} c_{km} y_{km}$$

$$\text{subject to} \sum_{i \in \hat{H}} \sum_{j \in \hat{H}} w_{ij} x_{ijkm} \le B y_{km} \quad (k, m) \in \hat{A}$$

$$\sum_{m \in \hat{H}} x_{ijkm} - \sum_{m \in \hat{H}} x_{jikm} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j. \\ 0 & \text{if } k \neq i, j. \end{cases} \qquad i, j, k \in \hat{H}$$

$$0 \le x_{ijkm} \le 1 \quad i, j, k, m \in \hat{H}$$

$$y_{ij} \in Z^+ \quad (i, j) \in \hat{A}.$$

Even though the NLP is known to be a NP-hard, for instances of reasonable size it can be solved efficiently using a general purpose solver. The output of the NLP is a set of hub arcs to open and the associated routing decisions for all demand flow of the MHLP. Thus, the optimal solution of the NLP provides a feasible solution to the MHLP. The overall Lagrangean relaxation algorithm is depicted in Algorithm 1.

This constructive phase of the heuristic is executed every time the subgradient algorithm improves the best known lower bound. Once the subgradient algorithm terminates, we apply a local search procedure on the best known solution obtained so far. This procedure iteratively explores two neighborhoods namely classical shift and swap neighborhoods.

In the shift neighborhood, a non-hub node i currently assigned to hub m, is reassigned to a different open hub k. In the swap neighborhood, two non-hub nodes i and j currently assigned to

Algorithm 1: Lagrangean relaxation heuristic.

Initialize $z_D \leftarrow -\infty$; $(\lambda^1, \lambda^2, \lambda^3)^0 \leftarrow 0$; $\delta^0 \leftarrow 2$; $\bar{\phi} \leftarrow \infty$; $t \leftarrow 0$; $\pi \leftarrow 1.5$
while (*Stopping criteria not satisfied*) **do**
  Solve the Lagrangean function $L((\lambda^1, \lambda^2, \lambda^3)^t)$
  if $(L((\lambda^1, \lambda^2, \lambda^3)^t)) > z_D)$ **then**
    $z_D \leftarrow L((\lambda^1, \lambda^2, \lambda^3)^t)$
    Apply constructive heuristic to obtain upper bound UB^t
    if $(UB^t < \bar{\phi})$ **then**
      $\bar{\phi} \leftarrow UB^t$
    end if
  end if
  Evaluate the subgradient $\gamma(\lambda^1, \lambda^2, \lambda^3)^t$
  if$(\gamma(\lambda^1, \lambda^2, \lambda^3)^t d^{t-1} < 0)$ **then**
    $\theta^t = -\pi\gamma(\lambda^1, \lambda^2, \lambda^3) d^{t-1} / \parallel d^{t-1} \parallel^2$
  else
    $\theta^t = 0$
  end if
  Obtain the direction $d^t = \gamma(\lambda^1, \lambda^2, \lambda^3)^t + \theta^t d^{t-1}$
  Calculate the step length $s^t \leftarrow \delta^t \frac{\bar{\phi} - L((\lambda^1, \lambda^2, \lambda^3)^t)}{||\gamma(\lambda^1, \lambda^2, \lambda^3)^t||^2}$,
  Set $(\lambda^1, \lambda^2, \lambda^3)^{(t+1)} \leftarrow (\lambda^1, \lambda^2, \lambda^3)^{(t)} + s^t d^t$
  Set $t \leftarrow t + 1$
end while

hubs m and k with $k \neq m$, respectively, are reallocated to k and m, respectively. Let $s = (H, A, M)$ be the current solution, then

$$\mathcal{N}_{shift}(s) = \{s' = (H, A, M') : \exists! j \in N,\ M'(j) \neq M(j)\},$$

and

$$\mathcal{N}_{swap}(s) = \{s' = (H, A, M') : \exists!(j_1, j_2),\ j_1' = M(j_2),\ j_2' = M(j_1),\ \forall j \neq j_1, j_2\}.$$

The local search procedure explores $\mathcal{N}_{shift}$ first until a local optimal solution is found. The algorithm then tries to improve the solution by exploring $\mathcal{N}_{swap}$. Each time the search improves the best known solution, the procedure starts with $\mathcal{N}_{shift}$. In both neighborhoods, a best improvement strategy is used. Note that in order to reoptimize the arc selection and routing decisions in each neighbor, we solve one NLP to optimality. An outline of the local search scheme is depicted in Algorithm 2.

Algorithm 2: Local search procedure.

stoppingcriteria $\leftarrow$ *false*
while(*stoppingcriteria* = *false*)**do**
  explore $\mathcal{N}_{shift}$
  if(solution not improved in $\mathcal{N}_{shift}$) **then**
    explore $\mathcal{N}_{swap}$
    if(solution has not been updated) **then**
      stoppingcriteria $\leftarrow$ *true*
    end-if
  end-if
end-while

4. Branch-and-bound algorithm

We describe a branch-and-bound algorithm for solving the MHLP to optimality. It uses the Lagrangean relaxation to obtain lower and upper bounds at every node of the enumeration tree. It is composed of three phases. In the first phase, the enumeration tree is created by branching on the location variables z_{kk}, producing terminal nodes in which all location variables have been fixed. The second phase proceeds from each unfathomed node, creating an enumeration tree by branching on the assignment variables z_{ik}. When all the location and allocation variables are fixed, the third phase finds the optimal link activation and routing decisions for each unfathomed node by solving an associated NLP.

Let $(\bar{z}, \bar{y}, \bar{x})$ be the best solution found at the end of the Lagrangean relaxation algorithm at any node of the tree. The branching strategy used in the first phase of the enumeration tree is as follows. If there are any unfixed location variables such that $\bar{z}_{kk} = 1$, we select among these the one with the largest reduced cost $\bar{F}_k$ and explore the branch with $\bar{z}_{kk} = 1$. We store the associated branch with $\bar{z}_{kk} = 0$ on a list of unexplored nodes for later. When there are no more location variables which have not been fixed in the tree such that $\bar{z}_{kk} = 1$, we branch on the remaining unfixed variables by selecting the one with the largest reduced cost and explore the branch with $\bar{z}_{kk} = 1$. The first phase is completed once all locational decisions have been fixed.

When some of the nodes of the first phase have not been fathomed, we continue with the second phase. In this phase, we select each of these unfathomed nodes from the previous phase, one at a time, in non-decreasing way with respect to their lower bounds obtained and branch on the assignment variables. During this phase, the tree is not binary. That is, the number of branches generated at a node of the tree when selecting a non-hub node i for branching is equal to the number of open hubs on its path. The non-hub nodes are selected to be explored in the order of decreasing values of the highest reduced cost associated with the $\bar{z}_{ik}$ variables.

When all nodes of the second phase have been explored, but a subset of terminal nodes (i.e., nodes of depth n) have not been eliminated we move to the third (and last) phase of the algorithm. Note that at this point all locational and assignment decisions have been fixed and thus, the resulting subproblems reduces to a NLP. For each of these unfathomed nodes, we solve an associated NLP to optimality. We explore the entire enumeration tree in a depth first search fashion. At each node of the tree, the dual multipliers are initialized using the dual solutions from its parent node.

5. Computational experiments

We run computational experiments to compare and analyze the performance of the formulations, the Lagrangean relaxation and the branch-and-bound algorithm. All formulations and algorithms have been coded in C++ and run on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40GHz and 24 GB of RAM under windows 7 environment. All MIP problems have been solved using Concert technology of CPLEX 12.5.1.

We generate a set of benchmark instances for the MHLP using the well known Australian post (AP) instances which can be downloaded from the OR library (see, Beasley, 1990). The AP data set consists of postal flow and Euclidean distances between 200 districts in an Australian city. In our experiments, we have selected problems with $|N|$ =10, 20, 25, 40, 50, 60, and 75 nodes, and disregarded the flows W_{ii} for each $i \in N$, i.e., $W_{ii} = 0$. For each problem size, we generated 9 instances for the MHLP. Each instance comprises a hub facility link capacity chosen from $B \in$ {200, 300, 400, 500, 600, 650, 750} with an associated variable cost in $b \in$ {450, 500, 600, 800}, and a facility link capacity on access arcs chosen from $R \in$ {100, 150, 200} with an associated variable cost in $p \in$ {300, 345, 400, 500}. The choice of the parameters in each generated instance is such that $\frac{c_{km}}{B} < \frac{q_{km}}{R}$ where $B > R$, $b > p$ and $l_c = l_q$=0, to guarantee that the unit transportation cost on hub arcs, when fully utilized, is smaller than the unit transportation cost on access arcs. That is, there is a potential discount factor associated with hub arcs due to consolidation and use of more efficient modes of transportation between hub nodes. We note that $\frac{b/B}{p/R}$ corresponds to the smallest discount factor that can be achieved on hub arcs when compared to the transportation cost of access arcs when fully utilized. In practice, some arcs might be underutilized and thus, the actual discount factor may be higher which in turn, may lead to a lower unit cost on access arcs than on some hub arcs.

In order to generate a variety of instances, we selected the values for the capacities and costs in such a way that we obtain different potential discount factors on hub arcs. In particular, we consider the following configurations:

(i) for $\frac{b/B}{p/R} = 0.2$:
- $L1$: $(B = 750,\ R = 100,\ b = 600,\ p = 400)$,
- $L2$: $(B = 750,\ R = 100,\ b = 450,\ p = 300)$,
- $L3$: $(B = 600,\ R = 100,\ b = 600,\ p = 500)$,

(ii) for $\frac{b/B}{p/R} = 0.4$:
- $L4$: $(B = 400,\ R = 100,\ b = 800,\ p = 500)$,
- $L5$: $(B = 650,\ R = 150,\ b = 600,\ p = 345)$,
- $L6$: $(B = 500,\ R = 100,\ b = 600,\ p = 300)$,

(iii) for $\frac{b/B}{p/R} = 0.63$:
- $L7$: $(B = 200,\ R = 100,\ b = 500,\ p = 400)$,
- $L8$: $(B = 300,\ R = 150,\ b = 500, p = 400)$,
- $L9$: $(B = 400,\ R = 200,\ b = 500,\ p = 400)$.

For each one of these nine configurations, we generate seven different instances, one for each size of network. Therefore, we generated a total of 63 instances.

In all the experiments, the subgradient algorithm terminates when one of the following criteria has been met: *i)* the difference between the upper and lower bound is below a given threshold value, i.e.$|\bar{\phi} - z^t_D| < \epsilon$, *ii)* the improvement on the lower bound after $t_{\max}$ consecutive iterations is below a threshold value ψ, *iii)* the maximum number of iterations $iter_{\max}$ has been reached.

After some tuning, we set the following parameters to: $\epsilon = 10^{-6}$, $\psi = 0.05$, and $t_{\max} = 150$. For the first stage in the branch-and-bound algorithm, we set the maximum number of subgradient iterations at the root node to $iter_{\max} = 4,000$ and to $iter_{\max} = 25$ for the rest of the nodes. The parameter δ has been reduced by 0.25 after 100 consecutive iterations without improvement in the lower bound. In the second stage, the maximum number of subgradient iterations has been fixed to $iter_{\max} = 300$ at the root node and $iter_{\max} = 25$ for the rest of the nodes.

5.1. Comparison of formulations and algorithm

The first set of computational experiments has been performed to compare the path-based formulation (*PF*) with the flow-based formulation (*FF*) when solved using CPLEX. Throughout experiments, we used the default settings of CPLEX. The detailed results of this comparison on a set of instances ranging from 10 to 40 nodes are reported in Table 1. The first column provides the number of nodes, *n*, and the instance name ($n - name$). The next set of columns reports the linear programming relaxation gap (%*LP*), the linear programming relaxation gap after adding CPLEX cuts (%LP_{cut}), the percent deviation between the final upper and lower bounds (%GAP), the CPU time in seconds (*CPU*), and the number of explored node in the enumeration tree (Nodes), for both formulations. The %*LP* gap has been computed as $(UB - LP)/UB \times 100$, where *UB* is the best upper bound (or the optimal solution value), and *LP* is the optimal value of the *LP* relaxation. The final percent gap %GAP has been evaluated as $(UB - LB)/UB \times 100$, where *UB* and *LB* denote the best upper and lower bounds obtained at termination, respectively. Throughout experiments, the maximum time limit is set to one day of *CPU* time. Instances that could not be solved to optimality within this time limit have been marked with the label "time".

As can be seen in Table 1, *PF* is able to optimally solve 17 out of the 36 instances within the time limit. The percent *LP* gap of *PF* ranges from 2.21 to 10.15. The column %LP_{cut} shows that the addition of CPLEX cuts has a significant impact on the improvement of the lower bound at the root node of the tree. Nevertheless, CPLEX is unable to solve the *LP* relaxation for all 40-node instances in one day of CPU time. In the case of the *FF*, CPLEX is able to solve 26 out of the 36 instances within the time limit. The %*LP* gaps for the instances that have been solved using *PF* are slightly better than that obtained in the *FF*. The percent *LP* gap of *FF* ranges from 3.36 to 11.48. However, given that there is a considerably smaller number of variables and constraints in *FF*, CPLEX is able to optimally solve all 25-node instances and one of the 40-node instances that the *PF* cannot solve. Moreover, *FF* was able to provide optimality gaps for the remaining unsolved 40-node instances.

In order to analyze the performance of our proposed exact algorithm, we conduct a second series of computational experiments using a set of instances ranging from 10 to 50 nodes. The results are summarized in Table 2. The first five columns have the same meaning as in Table 1. The next two columns under heading *LR* provide duality gap of the best lower bound obtained with Lagrangean relaxation with respect to the best known solution (%*LR*) and the *CPU* time in seconds needed to obtain both lower and upper bounds using Lagrangean relaxation (*CPU*). The results of the columns under heading *Branch and Bound* report: the final percent deviation at termination (%*Gap*), the *CPU* time in second (*CPU*), and the number of the explored nodes in the enumeration tree (Nodes).

The results in Table 2 show that by using *FF*, we were able to solve 26 out of the 45 problem instances to optimality using CPLEX (final percent gaps on the remaining instances range from 0.29 to 8.87). The exact algorithm, on the other hand, was able to confirm the optimality of the solutions obtained in 35 out of the 45 instances within the CPU time limit. For the remaining 10 unsolved instances, the final percent deviation is below 2.8. In all instances considered, the percent deviation of the *LR* is smaller than the one obtained with *FF* even after the addition of CPLEX cuts. As a result, the proposed algorithm produces significantly smaller enumeration trees and is much faster than CPLEX for all instances, except on the small size, 10-node instances. Moreover, our exact algorithm is able to optimally solve 9 instances that *FF* is unable to solve within the time limit. For the instances that have not been solved to optimality, our algorithm always provides much smaller percent gaps as compared to *FF*. We note that the percent of time taken by the algorithm for solving the UFLPs at every iteration and the NLPs during the local search and at the end of the enumeration tree never exceeds 5% of the total computational time for the larger instances with 40 and 50 nodes.

In order to further analyze the performance of our proposed algorithm, we have run a third series of computational experiments using 60-node and 75-node instances. The results are summarized in Table 3. The column $CPU_{LP_{cuts}}$ reports the computational time in seconds to solve the *LP* relaxation and to add CPLEX cuts whereas the other columns have the same meaning as in the previous tables.

As can be seen in Table 3, the lower bounds obtained from Lagrangean relaxation are significantly tighter than those obtained with *FF*. In particular, the *LP* gaps of *FF* range from 6% to 13%, whereas the *LP* gaps of the Lagrangean relaxation algorithm never exceed 5%. It is worth mentioning that both *FF* and our algorithm are unable to solve any of these 60 and 75 nodes instances within the time limit of one day of CPU due to the size and complexity of the problem. However, the final gap of our branch-and-bound algorithm is at most 3.05%.

Table 1
Comparison between path-based and flow-based formulations.

Instance	Path-based formulation (*PF*)					Flow-based Formulation (*FF*)				
	% LP	% LP_{cut}	%GAP	CPU	Nodes	% LP	% LP_{cut}	%GAP	CPU	Nodes
10-L1	7.75	2.03	0.00	39	374	8.61	3.84	0.00	< 5	935
10-L2	4.30	1.87	0.00	< 5	39	4.80	2.05	0.00	< 5	48
10-L3	9.06	2.58	0.00	142	2521	9.38	5.47	0.00	33	22,449
10-L4	10.15	4.35	0.00	1,187	23,408	11.48	6.96	0.00	129	43,548
10-L5	4.01	2.65	0.00	< 5	35	4.59	3.03	0.00	< 5	25
10-L6	5.85	3.10	0.00	23	452	6.44	3.28	0.00	< 5	317
10-L7	4.47	3.05	0.00	28	715	6.31	3.39	0.00	< 5	676
10-L8	3.61	1.54	0.00	< 5	52	4.50	2.13	0.00	< 5	69
10-L9	5.02	3.99	0.00	11	199	6.24	4.52	0.00	< 5	467
20-L1	5.72	2.16	1.07	86,400	3529	6.14	4.01	0.00	1027	20,977
20-L2	2.96	1.44	0.00	6543	603	3.36	2.10	0.00	67	1451
20-L3	8.22	2.85	2.74	86,400	3098	8.31	7.68	0.00	10492	679,516
20-L4	5.93	2.17	0.00	78,393	6547	6.68	4.04	0.00	1707	34,623
20-L5	3.48	1.69	0.00	1286	472	3.92	3.70	0.00	72	1485
20-L6	5.05	3.74	0.00	24,329	3276	5.24	5.18	0.00	133	5224
20-L7	3.35	2.80	0.00	24,049	3960	4.15	3.53	0.00	166	8135
20-L8	2.98	2.32	0.00	6055	575	3.80	3.05	0.00	101	2633
20-L9	2.97	2.26	0.00	3116	357	3.67	2.78	0.00	56	1091
25-L1	8.82	4.07	4.07	Time	0	9.02	6.29	0.00	74,023	312,521
25-L2	3.84	1.84	1.79	Time	3	4.36	3.57	0.00	3300	14,906
25-L3	9.15	3.97	3.97	Time	0	9.34	8.57	2.15	Time	242,779
25-L4	8.39	3.17	3.14	Time	200	8.82	8.69	0.29	Time	378,612
25-L5	3.94	2.30	2.14	Time	1330	4.68	4.34	0.00	1503	14,439
25-L6	4.11	2.89	2.89	Time	539	4.96	4.72	0.00	2180	24,962
25-L7	2.21	1.74	1.16	Time	1329	3.97	3.69	0.00	1928	17,138
25-L8	2.45	1.77	0.00	Time	873	3.39	3.05	0.00	560	4023
25-L9	3.36	2.15	2.00	Time	1038	4.61	3.94	0.00	646	4987
40-L1	n.a	n.a	n.a	Time	n.a	6.91	6.89	4.08	Time	167,604
40-L2	n.a	n.a	n.a	Time	n.a	3.52	3.23	0.00	51,740	37,187
40-L3	n.a	n.a	n.a	Time	n.a	7.43	7.35	5.14	Time	289,952
40-L4	n.a	n.a	n.a	Time	n.a	7.41	6.79	4.84	Time	148,985
40-L5	n.a	n.a	n.a	Time	n.a	5.12	4.88	2.63	Time	69,828
40-L6	n.a	n.a	n.a	Time	n.a	4.67	4.64	0.75	Time	52,983
40-L7	n.a	n.a	n.a	Time	n.a	4.21	4.12	0.86	Time	107,754
40-L8	n.a	n.a	n.a	Time	n.a	5.41	4.99	3.08	Time	82,524
40-L9	n.a	n.a	n.a	Time	n.a	5.73	4.71	3.55	Time	38,074

5.2. Sensitivity analysis

In the last part of the experiments, we perform a sensitivity analysis on some of the parameters of the problem to analyze the changes in optimal solution networks. In particular, Tables 4–7 show how optimal network configurations change depending on: (i) the capacity of hub arcs (B), (ii) the capacity of access arcs (R), (iii) the variable cost of access arcs (p), and (iv) the variable cost of hub arcs (b), respectively. The tables present the optimal network configuration – hub nodes, hub arcs, number of hub facility links (y_{km}), and the percent hub arc utilization (% Utilization). That is, the hub arc utilization measures how much of the available capacity is being used on each hub arc (k, m) and has been computed as $\sum_{i\in N}\sum_{j\in N} W_{ij}\bar{x}_{ijkm}/(B\bar{y}_{km})100$, where $(\bar{x}, \bar{y})$ denotes the optimal solution.

Table 4 illustrates the effect of changing the capacity of hub arcs (B) on optimal solution networks while the rest of the parameters remain fixed. We observe that at higher capacity levels of hub arcs, more hubs are opened. However, the MHLP tries to utilize hub arc facilities by activating fewer hub arcs and thus, resulting in a cost efficient hub-and-spoke network structure. For instance, with $B = 650$, MHLP opens six hubs at nodes 1, 3, 6, 8, 10, and 13 and connects them with only nine hub arcs. The average arc utilization is 75.7%. When B decreases to 325, the model closes hub 3 while the rest is open and increases the number of active hub arcs to 11. The average utilization increases to 86.86%. Upon decreasing B further to 206, only four hubs are opened at nodes 3, 5, 10 and 13 with eight hub arcs. Note that the average utilization increases to 88.5%. In all cases, nodes 10 and 13 are chosen as hubs. In addition, we note that larger O/D paths are used when hub arc capacities are higher. For example, the flow routed from node 16 to node 0 visits five hub facilities when $B = 650$, four hubs when $B = 325$, and only two when $B = 206$.

Table 5 shows the changes in network configuration when varying the capacities of access arcs R, while the rest of the parameters remain fixed. From Table 5, we observe that as the capacity of links on access arcs decreases, the MHLP locates more hub facilities as well as more hub arcs to route the flow between all $O-D$ nodes. For instance, when $R = 80$, the model opens two hubs at nodes 6 and 13 and activates only two hub arcs to route the flow between all O/D nodes. Moreover, the average hub arc utilization in the network is 86.45% which implies that higher flows through the hub arcs. Upon decreasing R to 50, the optimal solution recommends locating three fully interconnected hub facilities at nodes 5, 10, and 13. In this case, the average hub arc utilization is 89.35%. Further decreasing R to 25, the model opens six hubs at nodes 1, 3, 6, 8, 11, and 13 and activates 13 hub arcs to route the flow between all O/D nodes. Moreover, the average hub arc utilization in the network is 94.15% which implies that higher flows through the hub arcs. Furthermore, a lower capacity on access arcs leads to the selection of the most isolated node 3 as a hub given that it is the one with the highest amount of incoming/outgoing flow. It can also be seen that, as R decreases, the location of hub facilities tend to be closer to each other.

Table 6 illustrates the effects of varying the variable cost of hub arcs b on the optimal solution network while the capacities

Table 2
Results of branch-and-bound algorithm for small to medium-size instances.

Instance	Flow-based Formulation				LR		Branch and Bound		
	% LP_{cut}	%GAP	CPU	Nodes	%LR	CPU	%GAP	CPU	Nodes
10-L1	3.84	0.00	< 5	935	2.02	10	0.00	22	213
10-L2	2.05	0.00	< 5	48	1.86	6	0.00	9	42
10-L3	5.47	0.00	33	22,449	2.48	50	0.00	88	624
10-L4	6.96	0.00	129	43,548	4.52	22	0.00	337	4537
10-L5	3.03	0.00	< 5	25	2.20	12	0.00	19	87
10-L6	3.28	0.00	< 5	317	3.18	7	0.00	15	127
10-L7	3.39	0.00	< 5	676	3.61	29	0.00	80	714
10-L8	2.13	0.00	< 5	69	1.58	13	0.00	22	137
10-L9	4.52	0.00	< 5	467	4.03	13	0.00	21	180
20-L1	4.01	0.00	1027	20,977	1.79	35	0.00	102	1268
20-L2	2.10	0.00	67	1451	1.47	23	0.00	48	383
20-L3	7.68	0.00	10492	679,516	1.83	83	0.00	308	4624
20-L4	4.04	0.00	1707	34,623	2.05	47	0.00	241	4173
20-L5	3.70	0.00	72	1485	1.50	42	0.00	58	349
20-L6	5.18	0.00	133	5224	3.75	37	0.00	149	3102
20-L7	3.53	0.00	166	8135	3.22	79	0.00	356	5116
20-L8	3.05	0.00	101	2633	2.38	39	0.00	78	818
20-L9	2.78	0.00	56	1091	2.32	37	0.00	77	764
25-L1	6.29	0.00	74,023	312,521	1.57	129	0.00	697	9,155
25-L2	3.57	0.00	3300	14,906	0.95	87	0.00	167	700
25-L3	8.57	2.15	Time	242,779	2.01	145	0.00	887	11,632
25-L4	8.69	0.29	Time	378,612	2.58	450	0.00	3,163	41,232
25-L5	4.34	0.00	1,503	14,439	2.03	67	0.00	276	1957
25-L6	4.72	0.00	2,180	24,962	2.79	82	0.00	439	5147
25-L7	3.69	0.00	1,928	17,138	2.27	168	0.00	1,331	13,750
25-L8	3.05	0.00	560	4,023	1.55	82	0.00	289	3020
25-L9	3.94	0.00	646	4,987	1.77	101	0.00	249	2185
40-L1	6.89	4.08	Time	167,604	1.78	699	0.00	16,144	78,423
40-L2	3.23	0.00	51,740	37,187	0.64	437	0.00	865	1177
40-L3	7.35	5.14	Time	289,952	1.76	734	0.00	32,290	174,658
40-L4	6.79	4.84	Time	148,985	2.68	731	1.91	Time	290,062
40-L5	4.88	2.63	Time	69,828	1.79	135	0.00	4,153	15,151
40-L6	4.64	0.75	Time	52,983	2.48	497	0.00	24,200	101,864
40-L7	4.12	0.86	Time	107,754	2.85	786	0.00	40,284	152,099
40-L8	4.99	3.08	Time	82,524	2.35	681	0.00	59,693	323,662
40-L9	4.71	3.55	Time	38,074	1.87	578	0.00	78,105	431,532
50-L1	8.33	7.69	Time	62,875	2.79	1502	1.84	Time	239,520
50-L2	6.52	5.86	Time	21,077	2.35	1613	2.00	Time	250,642
50-L3	9.76	9.53	Time	125,427	2.41	1903	2.16	Time	166,954
50-L4	8.62	7.49	Time	154,566	2.56	3675	2.01	Time	190,193
50-L5	9.21	8.56	Time	14,841	4.45	1224	2.61	Time	210,039
50-L6	5.83	5.23	Time	30,198	2.23	1305	1.67	Time	226,644
50-L7	5.03	3.56	Time	105,850	3.34	2169	1.79	Time	173,901
50-L8	6.53	5.75	Time	76,346	3.41	1963	2.72	Time	176,399
50-L9	5.40	5.30	Time	9,144	1.87	1,319	0.12	Time	283,396

Table 3
Results of branch and bound algorithm for 60 and 75-node instances.

Instance	Flow-based Formulation		LR		Branch and Bound		
	%LP_{cuts}	$CPU_{LP_{cuts}}$	%LR	CPU	%GAP	CPU	Nodes
60-L1	8.76	4,419	2.00	3129	1.73	Time	118,013
60-L2	7.82	3,338	1.75	2523	1.53	Time	131,604
60-L3	9.78	2,408	2.29	3347	2.09	Time	21,369
60-L4	8.57	1877	2.95	4191	2.13	Time	15,657
60-L5	8.70	2682	1.90	2880	1.61	Time	130,091
60-L6	8.03	3392	2.69	3128	2.13	Time	98,955
60-L7	6.02	1553	4.33	4037	2.45	Time	14,426
60-L8	7.60	1920	3.69	3346	2.56	Time	15,460
60-L9	7.04	1543	2.60	4772	1.58	Time	93,931
75-L1	11.05	11,339	2.30	9678	2.21	Time	5721
75-L2	8.79	10,851	1.83	8219	1.52	Time	7680
75-L3	10.09	8390	2.00	15,620	0.86	Time	4940
75-L4	11.01	11,640	3.18	23,013	2.11	Time	2249
75-L5	12.64	11,727	2.80	9069	0.28	Time	4391
75-L6	10.85	13,972	3.55	8070	2.67	Time	4198
75-L7	8.25	11,235	4.96	13,183	3.05	Time	2488
75-L8	9.12	8639	3.55	13,160	1.70	Time	3883
75-L9	8.61	9888	2.39	10,451	0.44	Time	4959

Table 4
Effect of varying capacity of hub arcs on optimal solution networks with $n = 20$, $R = 110$, $p = 634$ and $b = 750$.

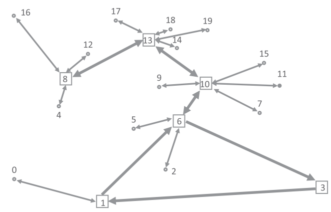
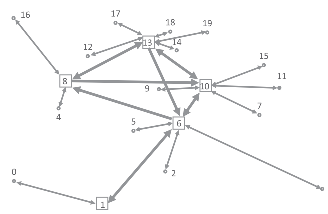
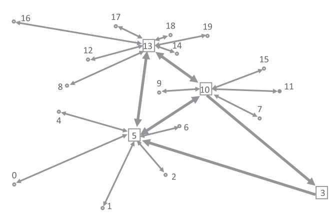
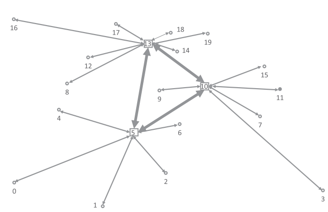
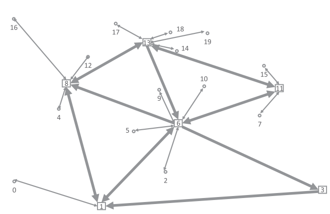
Capacity	Solution network	Hub arcs	y_{km}	% Utilization
$B = 650$		(1,6)	1	55.39
		(3,1)	1	60.45
		(6,3)	1	66.28
		(6,10)	1	86.26
		(8,13)	1	65.05
		(10,6)	1	94.26
		(10,13)	1	98.24
		(13,8)	1	79.24
		(13,10)	2	75.89
$B = 325$		(1,6)	1	79.25
		(6,1)	1	89.37
		(6,8)	1	100
		(6,10)	1	84
		(8,10)	1	44.09
		(8,13)	1	100
		(10,6)	1	100
		(10,13)	1	99.96
		(13,6)	1	100
		(13,8)	1	77.32
		(13,10)	2	81.42
$B = 206$		(3,5)	1	54.65
		(5,13)	1	98.12
		(10,3)	1	73.06
		(10,5)	1	85.48
		(5,10)	2	100
		(10,13)	2	98.88
		(13,5)	2	97.82
		(13,10)	3	100

Table 5
Effect of varying capacity of access arcs in optimal solutions with $n = 20$, $B = 250$, $b = 200$, $p = 150$.

Capacity	Solution network	Hub arcs	y_{km}	% Utilization
$R = 80$		(13,6)	4	92.14
		(6,13)	3	80.83
$R = 50$		(13,10)	3	78.13
		(13,5)	2	87
		(10,13)	1	100
		(10,5)	1	100
		(5,13)	2	71.9
		(5,10)	1	99.08
$R = 25$		(13,11)	2	99.41
		(13,8)	2	88.64
		(13,6)	1	100
		(11,13)	1	100
		(11,6)	1	100
		(8,13)	2	100
		(8,1)	1	100
		(6,11)	1	91.44
		(6,8)	1	87.94
		(6,3)	1	100
		(3,1)	1	84.83
		(1,8)	1	71.68
		(1,6)	1	100

B and R and the variable cost p are fixed. As can be seen in Table 6, as the variable cost b decreases, the MHLP tends to locate more hub facilities and activate fewer hub arcs. This behavior can be attributed to the decrease in unit flow cost on hub arcs when b decreases. For example, when $b = 600$, the MHLP locates three fully interconnected hub facilities with an average hub arc utilization of 82.72%. Decreasing the unit flow cost by decreasing b to 500 results in locating four hub facilities at nodes 5, 8, 10 and 13, and activating 10 hub arcs. The average hub arc utilization is 91.81%. As expected, further decreasing the unit flow cost, i.e. b equal to 300, the optimal solution recommends locating six hub facilities at nodes 1,3,6,8,10, and 13 and using 13 hub arcs resulting in average utilization of 92.33%.

Finally, Table 7 illustrates the effects of varying the variable costs of access arcs p on the optimal solution networks while the rest of the parameters remain fixed. As can be seen in this table,

Table 6
Effect of varying the variable cost b of hub arcs on optimal solution networks with $n = 20$, $B = 250$, $R = 25$ and $p = 150$.

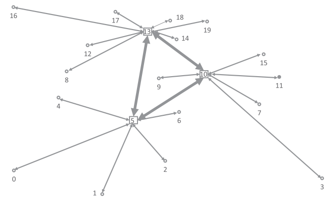
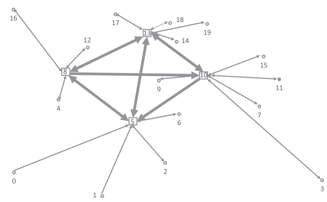
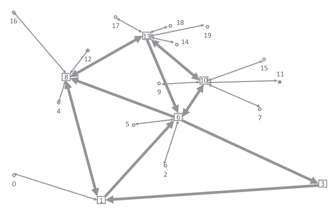
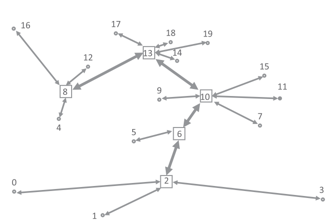
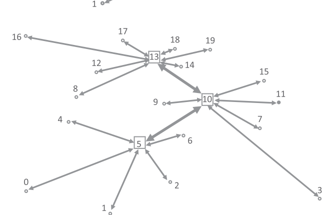
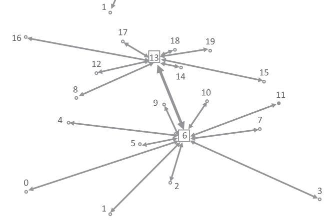
Cost	Solution network	Hub arcs	y_{km}	% Utilization
$b = 600$		(5,10)	1	100
		(5,13)	1	91.37
		(10,5)	1	65.82
		(10,13)	1	76.23
		(13,5)	2	80.31
		(13,10)	1	82.6
$b = 500$		(5,8)	1	84
		(5,10)	1	94.8
		(5,13)	1	93.59
		(8,5)	1	96
		(8,13)	1	100
		(10,5)	1	98.92
		(10,13)	1	77.74
		(13,5)	1	98.6
		(13,8)	2	74.42
		(13,10)	2	100
$b = 300$		(1,6)	1	97.5
		(1,8)	1	73
		(3,1)	1	83.6
		(6,3)	1	98.8
		(6,8)	1	87.78
		(6,10)	1	96.94
		(8,1)	1	100
		(8,13)	2	98
		(10,6)	1	78.53
		(10,13)	1	100
		(13,6)	1	100
		(13,8)	2	86.07
		(13,10)	2	100

Table 7
Effect of varying variable cost p of access arcs on optimal solution networks with $n = 20$, $B = 650$, $R = 110$ and $b = 750$.

Cost	Solution network	Hub arcs	y_{km}	% Utilization
$p = 600$		(2,6)	1	61.73
		(6,2)	1	82.59
		(6,10)	1	86.26
		(8,13)	1	65.05
		(10,6)	1	94.25
		(10,13)	1	98.24
		(13,8)	1	79.24
		(13,10)	2	75.89
$p = 317$		(5,10)	1	88
		(10,5)	1	100
		(10,13)	1	93.69
		(13,10)	2	78.54
$p = 212$		(6,13)	1	100
		(13,6)	2	74.24

as the variable costs p decreases, the MHLP tends to locate fewer hubs and connect them with fewer hub arcs. For instance, when $p = 600$, MHLP selects five locations for hub facilities, i.e., 2, 6, 8, 10, and 13, and activates eight hub arcs to route the flow between all O/D pairs. The average hub arc utilization is 80.4%. Decreasing p to 317 results in locating only three hubs at nodes 5, 10, and 13 and connecting them using four hub arcs resulting in an average utilization of 90.06%. Further decreasing p to 212 leads to selecting only two hub facilities at nodes 6 and 13 with two hub arcs with an average utilization of 87.12%.

6. Conclusions

In this paper, we studied the modular hub location problem with single assignments. The MHLP explicitly models the flow dependency of transportation cost using modular arc costs. Moreover, it does not assume a particular topological structure, instead it considers the design of the entire hub network as a part of the decision process. We presented two mixed integer programming formulations – a flow-based and a path-based formulation and compared their strengths using linear programming relaxation bounds. We proposed a Lagrangean relaxation of the path-based formulation by relaxing the linking constraints of the location/allocation and routing variables. We presented a primal heuristic to construct a feasible solution and compute an upper bounds. Further, we presented a branch-and-bound based exact algorithm that uses the Lagrangean relaxation as a bounding procedure at the nodes of an enumeration tree. Computational results on benchmark instances up to 75 nodes confirm the efficiency and the robustness of the proposed algorithms. We also analyzed the effect of changes in hub and access arcs capacities as well as changes in variable costs of hub and access arcs on optimal solutions. The obtained results showed that solution networks tend to have more open hubs when increasing the capacities of hub arcs, whereas an opposite behavior is observed for the case of access arcs. For the case of the variable costs of hub and access arcs, more hub arcs and hub nodes are activated in optimal solution networks when the variable costs on hub arcs decrease and variable costs on access arcs increase.

Acknowledgments

This research was partly funded by the Canadian Natural Science and Engineering Research Council under grants 418609-2012 and 386501-2010. This support is gratefully acknowledged. The authors thank two anonymous reviewers for their valuable comments on a previous version of the paper.

References

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., Nickel, S., Saldanha da Gama, F., 2012. Hub location under uncertainty. Transp. Res.: Part B 46, 529–543.
An, Y., Zhang, Y., Zeng, B., 2015. The reliable hub-and-spoke design problem: models and algorithms. Transp. Res. Part B: Methodol. 77, 103–122.
Beasley, J.E., 1990. Or-library: distributing test problems by electronic mail. J. Oper. Res. Soc. 41 (11), 1069–1072.
Brännlund, U., 1995. A generalized subgradient method with relaxation step. Math. Program. 71 (2), 207–219.
Bryan, D., 1998. Extensions to the hub location problem: formulations and numerical examples. Geogr. Anal. 30 (4), 315–330.
Bryan, D.L., O'Kelly, M.E., 1999. Hub-and-spoke networks in air transportation: an analytical review. J. Reg. Sci. 39 (2), 275–295.
de Camargo, R.S., de Miranda Jr, G., Luna, H.P.L., 2009. Benders decomposition for hub location problems with economies of scale. Transp. Sci. 43 (1), 86–97.
Camerini, P.M., Fratta, L., Maffioli, F., 1975. On improving relaxation methods by modified gradient techniques. In: Nondifferentiable Optimization. Springer, pp. 26–34.
Campbell, J.F., 2013. Modeling economies of scale in transportation hub networks. In: Proceedings of the 2013 46th Hawaii International Conference o nSystem Sciences (HICSS). IEEE, pp. 1154–1163.
Campbell, J.F., Ernst, A., Krishnamoorthy, M., 2005. Hub arc location problems: part i: introduction and results. Manage Sci. 51 (10), 1540–1555.
Campbell, J.F., Ernst, A., Krishnamoorthy, M., 2005. Hub arc location problems: part II: formulations and optimal algorithms. Manage Sci. 51 (10), 1556–1571.
Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transp. Sci. 46 (2), 153–169.
Contreras, I., 2015. Hub location problems. In: Laporte, G., Saldanha de Gama, F., Nickel, S. (Eds.), Location Science. Springer, New York, pp. 311–344.
Contreras, I., Cordeau, J.-F., Laporte, G., 2011. Benders decomposition for large-scale uncapacitated hub location. Oper. Res. 59 (6), 1477–1490.
Contreras, I., Cordeau, J.-F., Laporte, G., 2011. The dynamic uncapacitated hub location problem. Transp. Sci. 45 (1), 18–32.
Contreras, I., Díaz, J.A., Fernández, E., 2009. Lagrangean relaxation for the capacitated hub location problem with single assignment. OR Spectr. 31 (3), 483–505.
Contreras, I., Díaz, J.A., Fernández, E., 2011. Branch and price for large-scale capacitated hub location problems with single assignment. INFORMS J. Comput. 23 (1), 41–55.
Contreras, I., Fernández, E., 2014. Hub location as the minimization of a supermodular set function. Oper. Res. 62 (3), 557–570.
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., Tanash, M., Vidyarthi, N., 2016. Exact and heuristic approaches for the cycle hub location problem. Ann. Oper. Res. doi:10.1007/s10479-015-2091-2. 10.1007/s10479-015-2091-2
Corberán, Á., Peiró, J., Campos, V., Glover, F., Martí, R., 2016. Strategic oscillation for the capacitated hub location problem with modular links. J. Heuristics 22 (2), 221–244.
Cornuejols, G., Nemhauser, G.L., Wolsey, L.A., 1983. The Uncapacitated Facility Location Problem. Technical Report. DTIC Document.
Correia, I., Nickel, S., Saldanha de Gama, F., 2014. Multi-product capacitated single-allocation hub location problems: formulations and inequalities. Netw. Spat. Econ. 14, 1–25.
Correia, I., Nickel, S., Saldanha da Gama, F., 2010. Single-assignment hub location probems with multiple capacity levels. Transp. Res.: Part B 44, 1047–1066.
Cunha, C.B., Silva, M.R., 2007. A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. Eur. J. Oper. Res. 179 (3), 747–758.
Ernst, A.T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Locat. Sci. 4 (3), 139–154.
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.
Ernst, A.T., Krishnamoorthy, M., 1998. An exact solution approach based on shortest-paths for p-hub median problems. INFORMS J. Comput. 10 (2), 149–162.
Guignard, M., 2003. Lagrangean relaxation. Top 11 (2), 151–200.
Guta, B., 2003. Subgradient Optimization Methods in Integer Programming with an Application to a Radiation Therapy Problem. University of Kasierslautern, Germany.
Hamacher, H.W., Labbé, M., Nickel, S., Sonneborn, T., 2004. Adapting polyhedral properties from facility to hub location problems. Discrete Appl. Math. 145 (1), 104–116.
Hoff, A., Peiró, J., Corberán, Á., Martí, R., 2016. Adaptive Memory Programming for the Capacitated Modular Hub Location Problem. Technical Report. URL http://www.uv.es/rmarti/paper/routing.html.
Horner, M.W., O'Kelly, M.E., 2001. Embedding economies of scale concepts for hub network design. J Transp Geogr 9 (4), 255–265.
Kimms, A., 2006. Economies of scale in hub & spoke network design models: we have it all wrong. In: Perspectives on Operations Research. Springer, pp. 293–317.
Klincewicz, J.G., 2002. Enumeration and search procedures for a hub location problem with economies of scale. Ann. Oper. Res. 110 (1–4), 107–122.
Labbé, M., Yaman, H., 2004. Projecting the flow variables for hub location problems. Networks 44 (2), 84–93.
Labbé, M., Yaman, H., 2008. Solving the hub location problem in a star–star network. Networks 51 (1), 19–33.
Labbé, M., Yaman, H., Gourdin, E., 2005. A branch and cut algorithm for hub location problems with single assignment. Math. Program. 102 (2), 371–405.
Lawler, E.L., 1979. Fast approximation algorithms for knapsack problems. Math. Oper. Res. 4 (4), 339–356.
Magnanti, T.L., Mirchandani, P., Vachani, R., 1995. Modeling and solving the two-facility capacitated network loading problem. Oper. Res. 43 (1), 142–157.
Mirzaghafour, F., 2013. Modular hub location problems. Concordia University, Montreal, Canada.
O'Kelly, M.E., 1986. Activity levels at hub facilities in interacting networks. Geogr. Anal. 18 (4), 343–356.
O'Kelly, M.E., 1998. A geographer's analysis of hub-and-spoke networks. J. Transp. Geogr. 6 (3), 171–186.
O'Kelly, M.E., Bryan, D., 1998. Hub location with flow economies of scale. Transp. Res. Part B: Methodol. 32 (8), 605–616.
O'Kelly, M.E., Campbell, J.F., Camargo, R.S., Miranda, G., 2015. Multiple allocation hub location model with fixed arc costs. Geogr. Anal. 47 (1), 73—96.
Podnar, H., Skorin-Kapov, J., Skorin-Kapov, D., 2002. Network cost minimization using threshold-based discounting. Eur. J. Oper. Res. 137 (2), 371–386.
Racunica, I., Wynter, L., 2005. Optimal location of intermodal freight hubs. Transp. Res. Part B: Methodol. 39 (5), 453–477.

Rastani, S., Setak, M., Karimi, H., 2015. Capacity selection for hubs and hub links in hub location problems. Int. J. Manag. Sci. Eng. Manag. 1–11.

Martins de Sá, E., 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 (2), 185–202.

Martins de Sá, E., Contreras, I., Cordeau, J., 2015. Exact and heuristic algorithms for the design of hub networks with multiple lines. Eur. J. Oper. Res. 246, 186–198.

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

Sasaki, M., Fukushima, M., 2003. On the hub-and-spoke model with arc capacity constraints. J. Oper. Res. Soc. Japan-Keiei Kagaku 46 (4), 409–428.

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

Yaman, H., Carello, G., 2005. Solving the hub location problem with modular link capacities. Comput. Oper. Res. 32 (12), 3227–3245.

Yoon, M.-G., Current, J., 2006. The hub location and network design problem with fixed and variable arc costs: formulation and dual-based solution heuristic. J. Oper. Res. Soc. 59 (1), 80–89.

Zanjirani Farahani, R., 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.