

An Exact Algorithm for the Modular Hub Location Problem with Single Assignments

Moayad Tanash, Ivan Contreras, Navneet Vidyarthi

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

Abstract

A key feature of hub-and-spoke networks is the consolidation of flows at hub facilities. The bundling of flows allows to reduce transportation costs, which are frequently modelled 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 dependence of transportation costs on all links of a hub network using modular arc costs. It neither assumes a fully interconnection between hub nodes nor a particular topological structure, instead it considers link activation decisions as part of the decision process. 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.

Keywords: Hub location; flow-dependent costs; Lagrangean relaxation

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

Email addresses: m_tanash@encs.concordia.ca (Moayad Tanash),
icontrer@encs.concordia.ca (Ivan Contreras), navneetv@jmsb.concordia.ca
(Navneet Vidyarthi)

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 fourth 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 deci-

sions as they are determined mainly by the allocation 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 (Ernst and Krishnamoorthy, 1998a; Labbé and Yaman, 2004; Hamacher et al., 2004; Correia et al., 2010; Alumur et al., 2012; Correia et al., 2014; Contreras and Fernández, 2014) and to sophisticated solution algorithms that are able to solve real-size instances (Ernst and Krishnamoorthy, 1998b; Labbé et al., 2005; Contreras et al., 2011a,b; 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 the *modular hub location problem with single assignments* (MHLP). The MHLP considers explicitly the flow dependence of transportation costs in all the arcs in the network based on modular arc costs. Thus, the total transportation cost is measured not in terms of the per unit flow cost but in terms of the number of facility links used on each arc, avoiding 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. The MHLP and other variants 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 authors 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), Klinecicz (2002) and de Camargo et al. (2009). The model uses 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. Kimms (2006) and O’Kelly et al. (2015) adopted a different approach, 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. 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,b) 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 (Yaman, 2008; Labbé and Yaman, 2008), tree-star networks (Contreras et al., 2009; Martins de Sá et al., 2013), cycle-star networks (Contreras et al., 2016), and hub line networks (Martins de Sá et al., 2015a,b).

In this paper, we present two new 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 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 the 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 4 describes the primal heuristic algorithm. While in Section 5, we present a branch-and-bound algorithm. The computational results and analysis are presented in Section 6, followed by conclusions in Section 7.

2. Problem Definition and Formulation

Let $G = (N, A)$ be a complete digraph, 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, the amount of flow that is routed on each arc is used to explicitly determine the number of facility links with a given capacity that will be needed to route the flow on that arc. That is, we model the transportation costs on arcs using a step-wise function. In particular, for each pair of hub nodes (k, m) , let $c_{km} = l_c + bd_{km}$ denote the transportation cost for using one facility link with capacity B on hub arc (k, m) , where l_c and b represent the fixed and variable costs, respectively. Similarly, for each pair of non-hub node and hub node (i, k) , let $q_{ik} = l_q + pd_{ik}$ denote the transportation cost for using one facility link with capacity R on access arc (i, k) , where l_q and p represent the fixed and variable costs, respectively. For each $i \in N$, let $v_i^1 = \lceil \sum_{j \in N} W_{ij}/R \rceil$ denote the number of facility links required to route the flow originating from i directly to a hub, and let $v_i^2 = \lceil \sum_{j \in N} W_{ji}/R \rceil$ denote the number of facility links required to route the flow from a hub directly to destination i . In order to accurately account for the economies of scale when consolidating flows at hub facilities and when using more efficient paths between hubs, we assume that the unit transportation cost on hub arcs is smaller than the unit transportation cost on access arcs. That is, $\frac{c_{km}}{B} < \frac{q_{km}}{R}$, where $B > R$, $b > p$ and $l_c > l_q$. The MHLP consists of locating a set of hub facilities, activating a set of facility links (hub and access arcs), and determining the route of flows through the network such that the total setup

and transportation cost is minimized.

The MHLP 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 (Ernst and Krishnamoorthy, 1996; Contreras et al., 2011b), 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 $i, k \in N; i \neq k$, 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 pair of nodes $k, m \in N$, 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:

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

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

$$z_{ik} \leq z_{kk} \quad i, k \in N \quad (2)$$

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

$$\sum_{i \in N} \sum_{j \in N} W_{ij} x_{ijkm} \leq B y_{km} \quad k, m \in N \quad (4)$$

$$y_{mk} \leq Q z_{mm} \quad k, m \in N \quad (5)$$

$$y_{mk} \leq Q z_{kk} \quad k, m \in N \quad (6)$$

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

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

$$0 \leq x_{ijkm} \leq 1 \quad i, j, k, m \in N. \quad (9)$$

The objective function minimizes the sum of setup costs for locating hub facilities and the transportation cost between access and hub arcs. 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. Constraints (7)–(8) 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 $i \in N$ 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:

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

subject to (1) – (2), (5) – (9)

$$\begin{aligned} & \sum_{j \in N} W_{ij} z_{jk} + \sum_{m \in N} X_{ikm} \\ & - \sum_{m \in N} X_{imk} - O_i z_{ik} = 0 \quad i, k \in N \quad (10) \end{aligned}$$

$$\sum_{i \in N} X_{ikm} \leq B y_{km} \quad k, m \in N. \quad (11)$$

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 lower bounds on the optimal solution value. In the case of MHLP, if we relax constraints (3), (5), and (6), in a Lagrangean fashion, weighting their violations with multiplier vectors $\lambda^1, \lambda^2, \lambda^3$ of appropriate dimension, we obtain the following Lagrangean function:

$$\begin{aligned}
L(\lambda^1, \lambda^2, \lambda^3) = \min & \sum_{k \in N} f_k z_{kk} + \sum_{i \in N} \sum_{k \in N} q_{ik} (v_i^1 + v_i^2) z_{ik} + \sum_{m \in N} \sum_{k \in N} c_{km} y_{km} \\
& + \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \lambda_{ijk}^1 (z_{ik} + \sum_{m \in N} x_{ijmk} - z_{jk} - \sum_{m \in N} x_{ijkm}) \\
& + \sum_{k \in N} \sum_{m \in N} \lambda_{km}^2 (y_{km} - Q z_{mm}) + \sum_{k \in N} \sum_{m \in N} \lambda_{km}^3 (y_{km} - Q z_{kk}) \\
\text{s.t.} & \text{ (1) - (2), (4), and (7) - (8).}
\end{aligned}$$

For a given values 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:

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

where the coefficients of the objective function are:

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

Observe that the $L_z(\lambda^1, \lambda^2, \lambda^3)$ can be evaluated by solving a classical *uncapacitated facility location problem* (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:

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

where the coefficients of the objective function are:

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

Given that each of the y_{km} variables appear only 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:

$$L_{x,y}^{k,m}(\lambda^1, \lambda^2, \lambda^3) = \min \quad \bar{R}_{km}y_{km} + \sum_{i \in N} \sum_{j \in N} \bar{M}_{ijkm}x_{ijkm}$$

s.t. (4), (8), (9).

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

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

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. Note that only a fraction of the last considered item (denoted as r) may have been added, i.e., $x_{(r)km} = (By_{km} - \sum_{s=1}^{r-1} W_{(s)}) / W_{(r)} < 1$.

To determine the optimal value of y_{km} , the algorithm 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., $\bar{M}_{(s)km} < 0$). A value of $y_{km} = 0$ is selected as optimal whenever setting $y_{km} \geq 1$ 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:

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

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):

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

where $0 \leq \pi \leq 2$. 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

$$\begin{aligned} \gamma(\lambda_1, \lambda_2, \lambda_3) = & \left(\left(z_{ik}(\lambda) + \sum_{m \in N} x_{ijmk}(\lambda) - z_{jk}(\lambda) - \sum_{m \in N} x_{ijkm}(\lambda) \right)_{(i,j,k)}, \right. \\ & \left(\sum_{i \in N} \sum_{j \in N} W_{ij} x_{ijkm}(\lambda) - B y_{km}(\lambda) \right)_{(k,m)}, \\ & \left(y_{km}(\lambda) - Q z_{kk}(\lambda) \right)_{(k,m)}, \\ & \left. \left(y_{km}(\lambda) - Q z_{mm}(\lambda) \right)_{(k,m)} \right). \end{aligned}$$

At each iteration t of the subgradient algorithm, the dual multipliers are update 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, $D((i, j)) : A \rightarrow Z^+$ represents the number of facility links installed on hub arcs (i, j) and $M : N \rightarrow H$ 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) in an auxiliary network.

Let $\hat{G} = (\hat{H}, \hat{A})$ be a directed graph where $\hat{H} = \{k \in N : \hat{z}_{kk} = 1\}$ is the set of open hubs at iteration t and $\hat{A} = \{(k, m) : k, m \in \hat{H}\}$ is the set of candidate hub arcs. For each pair $(k, m) \in \hat{H} \times \hat{H}$, let $w_{km} = \sum_{i \in N(k)} \sum_{j \in N(m)}$ denote the amount of flow that needs to be routed from k to m , where $N(k) = \{i \in N : \hat{z}_{ik} = 1\}$. 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:

$$\begin{aligned} & \text{minimize} && \sum_{(k,m) \in \hat{A}} c_{km} y_{km} \\ & \text{subject to} && \sum_{i \in \hat{H}} \sum_{j \in \hat{H}} W_{ij} x_{ijkm} \leq B y_{km} && (k, m) \in \hat{A} \end{aligned}$$

$$\begin{aligned}
\sum_{m \in \hat{H}} x_{ijkm} - \sum_{m \in \hat{H}} x_{jikm} &= \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j. \\ 0 & \text{if } k \neq i, j. \end{cases} & i, j, k \in \hat{H} \\
0 \leq x_{ijkm} \leq 1 & & i, j, k, m \in \hat{H} \\
y_{ij} \in Z^+ & & (i, j) \in \hat{A}.
\end{aligned}$$

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.

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. The shift neighborhood improves the current solution by changing the assignment of one node, whereas the swap neighborhood considers all solution that differ from the current one by swapping the assignment of two nodes. Let $s = (H, A, M)$ be the current solution, then

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

and

$$N_{swap}(s) = \{s' = (H, A, M') : \exists!(j_1, j_2), j'_1 = M(j_2), j'_2 = M(j_1), \forall j \neq j_1, j_2\}.$$

All pairs of the form (i, j) in N_{shift} are considered, where $M(j) \neq i$. Similarly, all pairs of the form (j_1, j_2) in N_{swap} are considered, where $M(j_1) = M(j_2)$. The local search procedure explores N_{shift} first until a local optimal solution is found. The algorithm then tries to improve the solution by exploring N_{swap} . Each time the search improves the best known solution, the procedure starts with 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 neighbour, we solve one NLP to optimality. The overall Lagrangean relaxation algorithm is depicted in Algorithm 1.

Algorithm 1: Lagrangean relaxation heuristic

Initialize $z_D \leftarrow -\infty$; $(\lambda^1, \lambda^2, \lambda^3)^0 \leftarrow 0$; $\delta^0 \leftarrow 2$; $\bar{\phi} \leftarrow \infty$; $t \leftarrow 0$

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**

$\theta^t = \rho \|\gamma(\lambda^1, \lambda^2, \lambda^3)^t\| / \|d^{t-1}\|$

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

Apply local search on best found solution

4. Branch-and-Bound Algorithm

We now describe a branch-and-bound algorithm for solving the MHLPP to optimality. It uses the Lagrangean relaxation algorithm 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 $(\bar{z}, \bar{y}, \bar{x})$ 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 1-branch. We store the associated 0-branch on a list of unexplored nodes for later. When there are no more unfixed location variables 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 1-branch. The first phase is completed once all locational decisions have been fixed.

When some of the nodes of the first phase have not been eliminated, 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 using a general purpose solver. 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, Lagrangean relaxation and branch-and-bound algorithm. All formulations and algorithms were 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 were solved using Concert technology of CPLEX 12.5.1. We generate a set of benchmark instances for the MHLP by using the well known Australian post

(AP) instances which can be downloaded from the OR library (<http://mscmga.ms.ic.ac.uk/jeb/orlib/phubinfo.html>). 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. For each N , we generated 9 instances in such a way that an equivalent discount factor of $\alpha = \{0.2, 0.4, 0.63\}$ is obtained. In particular, each instance comprises a hub facility link capacity B with a variable cost b and a facility link capacity on access arcs H with a variable cost p . For $\alpha = 0.2$, we generated three instances: $L1 : (B = 750, R = 100, b = 600, p = 400)$, $L2 : (B = 750, R = 100, b = 450, p = 300)$ and $L3 : (B = 600, R = 100, b = 600, p = 500)$. For $\alpha = 0.4$, we generated three instances $L4 : (B = 400, R = 100, b = 800, p = 500)$, $L5 : (B = 650, R = 150, b = 600, p = 345)$ and $L6 : (B = 500, R = 100, b = 600, p = 300)$. Finally, For $\alpha = 0.63$, we generated three instances: $L7 : (B = 200, R = 100, b = 500, p = 400)$, $L8 : (B = 300, R = 150, b = 500, p = 400)$ and $L9 : (B = 400, R = 200, b = 500, p = 400)$.

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

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

The first set of computational experiment is performed to compare the path-based formulation with the flow-based formulation when solved using CPLEX. For this experiment, 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 is 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* is 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 fail to solve to optimality within this time limit are marked with the label "time".

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

Instance	Path-based formulation (<i>PF</i>)					Flow-based Formulation (<i>FF</i>)				
	% LP	% <i>LP_{cut}</i>	%GAP	CPU	Nodes	% LP	% <i>LP_{cut}</i>	%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	2,521	8.30	4.31	0.00	35	6,576
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	time	3,529	6.14	4.01	0.00	1,027	20,977
20-L2	2.96	1.44	0.00	6,543	603	3.36	2.10	0.00	67	1,451
20-L3	8.22	2.85	2.74	time	3,098	7.44	4.64	0.00	8,708	98,281
20-L4	5.93	2.17	0.00	78,393	6,547	6.68	4.04	0.00	1,707	34,623
20-L5	3.48	1.69	0.00	1,286	472	3.92	3.70	0.00	72	1,485
20-L6	5.05	3.74	0.00	24,329	3,276	5.24	5.18	0.00	133	5,224
20-L7	3.35	2.80	0.00	24,049	3,960	4.15	3.53	0.00	166	8,135
20-L8	2.98	2.32	0.00	6,055	575	3.80	3.05	0.00	101	2,633
20-L9	2.97	2.26	0.00	3,116	357	3.67	2.78	0.00	56	1,091
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	3,300	14,906
25-L3	9.15	3.97	3.97	time	0	8.50	5.76	2.15	time	262,447
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	1,330	4.68	4.34	0.00	1,503	14,439
25-L6	4.11	2.89	2.89	time	539	4.96	4.72	0.00	2,180	24,962
25-L7	2.21	1.74	1.16	time	1,329	3.97	3.69	0.00	1,928	17,138
25-L8	2.45	1.77	0.00	time	873	3.39	3.05	0.00	560	4,023
25-L9	3.36	2.15	2.00	time	1,038	4.61	3.94	0.00	646	4,987
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	7.38	5.14	time	162,711
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

As can be seen in the Table 1, the *PF* is able to optimally solve 16

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 were solved using PF are slightly better than that for obtained in the FF . 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 six columns have the same meaning as in Table 1. The next two columns under heading LR provide duality gap to the best lower bound obtained with Lagrangean relaxation ($\%LR$) 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 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 considered instances, the $\%$ 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 CPLEX is unable to solve within the time limit. For the instances that were not solved to optimality, our algorithm always provides much smaller $\%$ gaps as compared to CPLEX. 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% for the larger instances with n equal to 40 and 50 nodes.

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	4.31	0.00	35	6,576	2.48	50	0.00	88	624
10-L4	6.96	0.00	129	43,548	4.52	22	0.00	337	4,537
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	1,027	20,977	1.79	35	0.00	102	1,268
20-L2	2.10	0.00	67	1,451	1.47	23	0.00	48	383
20-L3	4.64	0.00	8,708	98,281	1.83	83	0.00	308	4,624
20-L4	4.04	0.00	1,707	34,623	2.05	47	0.00	241	4,173
20-L5	3.70	0.00	72	1,485	1.50	42	0.00	58	349
20-L6	5.18	0.00	133	5,224	3.75	37	0.00	149	3,102
20-L7	3.53	0.00	166	8,135	3.22	79	0.00	356	5,116
20-L8	3.05	0.00	101	2,633	2.38	39	0.00	78	818
20-L9	2.78	0.00	56	1,091	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	3,300	14,906	0.95	87	0.00	167	700
25-L3	5.76	2.15	time	262,447	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	1,957
25-L6	4.72	0.00	2,180	24,962	2.79	82	0.00	439	5,147
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	3,020
25-L9	3.94	0.00	646	4,987	1.77	101	0.00	249	2,185
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	1,177
40-L3	7.38	5.14	time	162,711	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	1,502	1.84	time	239,520
50-L2	6.52	5.86	time	21,077	2.35	1,613	2.00	time	250,642
50-L3	9.74	8.87	time	56,291	2.41	1,903	2.16	time	166,954
50-L4	8.62	7.49	time	154,566	2.56	3,675	2.01	time	190,193
50-L5	9.21	8.56	time	14,841	4.45	1,224	2.61	time	210,039
50-L6	5.83	5.23	time	30,198	2.23	1,305	1.67	time	226,644
50-L7	5.03	3.56	time	105,850	3.34	2,169	1.79	time	173,901
50-L8	6.53	5.75	time	76,346	3.41	1,963	2.72	time	176,399
50-L9	5.40	5.30	time	9,144	1.87	1,319	0.12	time	283,396

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} reports the CPU time in seconds to solve the LP relaxation and whereas the other columns have the same meaning as in the previous tables.

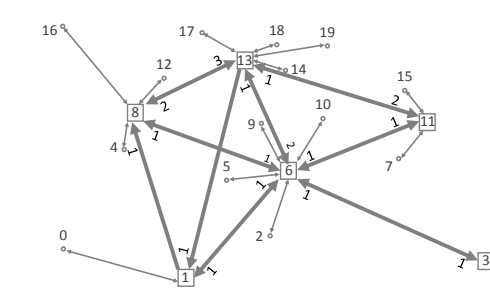
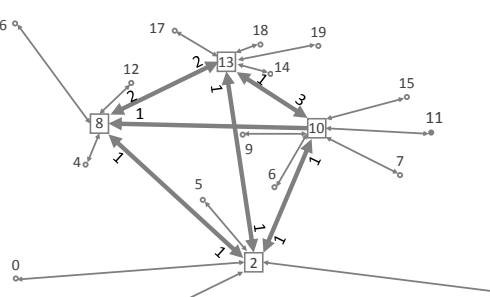
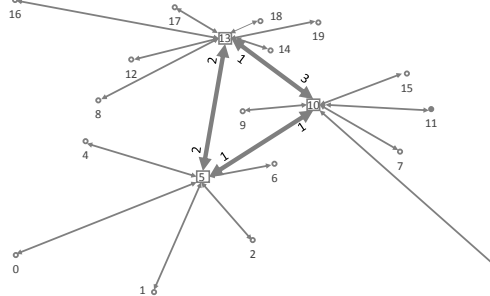
Table 3: Results of branch and bound algorithm for 60 and 75-node instances

Instance	Flow-based Formulation		LR		Branch and Bound		
	$\%LP_{cuts}$	CPU_{LP}	$\%LR$	CPU	$\%GAP$	CPU	Nodes
60-L1	8.76	4,419	2.00	3,129	1.73	time	118,013
60-L2	7.82	3,338	1.75	2,523	1.53	time	131,604
60-L3	9.78	2,408	2.29	3,347	2.09	time	21,369
60-L4	8.57	1,877	2.95	4,191	2.13	time	15,657
60-L5	8.70	2,682	1.90	2,880	1.61	time	130,091
60-L6	8.03	3,392	2.69	3,128	2.13	time	98,955
60-L7	6.02	1,553	4.33	4,037	2.45	time	14,426
60-L8	7.60	1,920	3.69	3,346	2.56	time	15,460
60-L9	7.04	1,543	2.60	4,772	1.58	time	93,931
75-L1	11.05	11,339	2.30	9,678	2.21	time	5,721
75-L2	8.79	10,851	1.83	8,219	1.52	time	7,680
75-L3	10.09	8,390	2.00	15,620	0.86	time	4,940
75-L4	11.01	11,640	3.18	23,013	2.11	time	2,249
75-L5	12.64	11,727	2.80	9,069	0.28	time	4,391
75-L6	10.85	13,972	3.55	8,070	2.67	time	4,198
75-L7	8.25	11,235	4.96	13,183	3.05	time	2,488
75-L8	9.12	8,639	3.55	13,160	1.70	time	3,883
75-L9	8.61	9,888	2.39	10,451	0.44	time	4,959

As can be seen in Table 3, the lower bounds obtained from Lagrangean relaxation are significantly tighter than those obtained with CPLEX. In particular, the LP gaps of FF range from 6% to 13%, whereas the LP gaps of the Lagrangean relaxation algorithm never exceeds 5%. It is worth mentioning that both CPLEX and our algorithm fail to solve any of these instances within the time limit due to the size and complexity of the problem. However, the final gap of our algorithm is always below 3.2%.

Tables 4-6 show how optimal network configurations change depending on the capacity B and the variable cost b of hub arcs as well as the capacity R and the variable cost p of access arcs. The tables present the optimal network configuration - hub nodes, hub arcs, facility links, the actual economies of scale achieved on each hub arc, and the % hub arc utilization. That is, hub arc utilization measures how much of the available capacity is being used in each hub arc and is computed as $flow_{km}/y_{km}R \times 100\%$, where $flow_{km}$ denotes the amount of flow routed on hub arc (k, m) at the optimal solution. In Table 4, we report the changes in solution network when varying the capacities of facility links on hub arcs B and access arcs R , while the variable costs p and b are fixed, in such a way to achieve a target discount factor of $\alpha = 0.2$.

Table 4: Effect of varying capacities of hub and access arcs in optimal solutions with $b = 200$, $p = 150$ and target $\alpha = 0.2$.

Instance	Solution network	Hub arcs	% Utilization	Discount
$B = 167$ $R = 25$		1-6	80.24	0.25
		1-8	100.00	0.20
		3-6	67.41	0.30
		6-1	100.00	0.20
		6-3	90.13	0.22
		6-8	100.00	0.20
		6-11	100.00	0.20
		6-13	100.00	0.20
		8-6	100.00	0.20
		8-13	81.59	0.28
		11-6	73.26	0.27
		11-13	91.60	0.22
		13-1	100.00	0.20
		13-6	100.00	0.20
		13-8	100.00	0.20
		13-11	100.00	0.20
		Avg	<u>92.76</u>	<u>0.22</u>
$B = 200$ $R = 30$		2-8	66.5	0.30
		2-10	89.0	0.22
		2-13	99.7	0.20
		8-2	93.0	0.21
		8-13	87.6	0.23
		10-2	93.0	0.21
		10-8	53.1	0.38
		10-13	100.0	0.20
		13-2	100.0	0.20
		13-8	97.5	0.21
		13-10	100.0	0.20
		Avg	<u>89</u>	<u>0.23</u>
		$B = 250$ $R = 38$		5-10
5-13	71.9			0.33
10-5	97.5			0.21
10-13	100.0			0.2
13-5	85.1			0.24
13-10	79.4			0.31
Avg	<u>87.8</u>			<u>0.25</u>

From Table 4, we observe that as the capacities of facility links on hub and access arcs increase while variable costs are fixed, the difference of the unit flow cost between hub and access arcs decreases, therefore, MHLP locates fewer hub facilities as well as fewer hub arcs to route the flow between all $O - D$ nodes. For instance, when $B = 167$ and $R = 200$ (i.e. unit flow cost difference is 4.8), the model selects 6 hub locations (1, 3, 6, 8, 11, 13) and activates only 16 hub arcs to route the flow between all $O - D$ nodes. Moreover, the average hub arc utilization in the network is 92.76% which implies that higher flows through the hub arc facilities links and hence higher flow discounts. As can also be seen in column *discount*, the average *discount* is 0.22. Increasing B to 200 and R to 30 (i.e. unit flow cost difference is 4.0) leads to locating 4 hub facilities at nodes (2, 8, 10, 13) and activating 13 hub arcs. In this case, the average utilization is 89% with an average *discount* of 0.23. Similarly, when B increases to 250 and R to 38 (i.e. unit flow cost difference is 3.1) 3 fully interconnected hub facilities are located at nodes (5, 10, 13). The average utilization is 87.8% with an average *discount* equal to 0.25. It can also be seen that, as the difference in unit flow cost decreases, the location of hub facilities tend to be closer to each other.

Table 5 illustrates the effects of varying the variable costs b and p on the optimal solution networks while the capacities B and R are fixed with a target discount factor α of 0.20. As can be seen in the tables, as the variable costs b and p increase, the MHLP tends to locate more hubs and connect them with fewer hub arcs. For instance, when $b = \$600$ and $p = \$500$, MHLP selects four locations for hub facilities (i.e., 2, 8, 10, 13) and activates six hub arcs to route the flow between all $O - D$ pairs. The average hub arc utilization is 80.83% and the average *discount* is 0.26. Decreasing b to \$450 and p to \$370 results in locating only three hubs at nodes (5, 10, 13) and using four hub arcs. Although the average utilization increases to 86.28%, the average *discount* increases to $\alpha = 0.28$. Further decreasing b to \$300 and p to \$250 leads to location of three hub facilities but at locations 2, 10 and 13 with four hub arcs. Note that the average utilization increases to 87.1% with an average *discount* of 0.26. In all cases, nodes 10 and 13 are chosen as hubs.

Table 5: Effect of varying variables costs of hub and access arcs in optimal solutions with $B = 600$, $R = 100$ and target $\alpha = 0.20$.

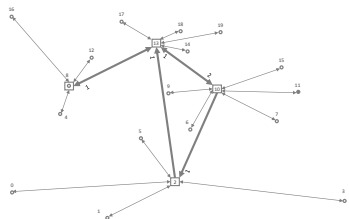
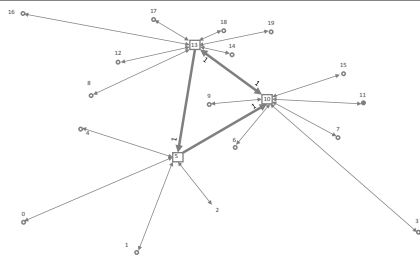
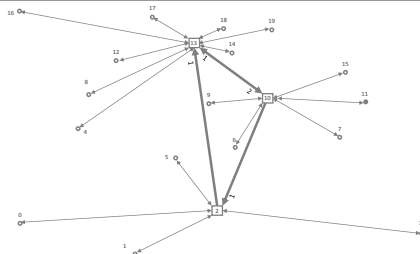
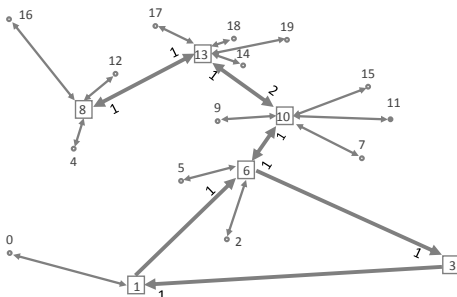
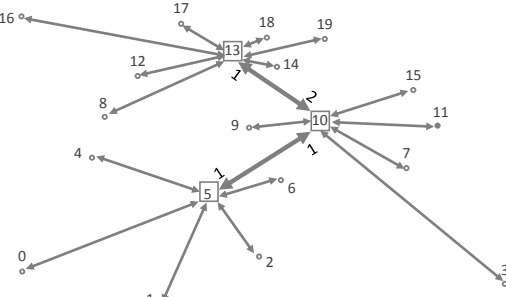
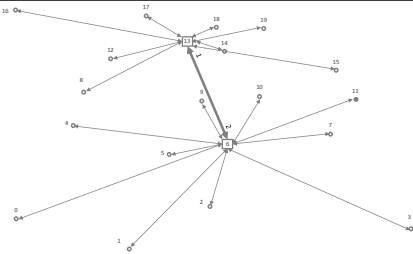
Instance	Solution network	Hub arcs	% Utilization	Discount
$b = \$600$ $p = \$500$		2-13	85.1	0.24
		8-13	70.5	0.28
		10-2	95.5	0.21
		10-13	51.0	0.39
		13-8	85.8	0.23
		13-10	97.1	0.21
		Avg	<u>80.83</u>	<u>0.26</u>
$b = \$450$ $p = \$370$		5-10	83.2	0.24
		10-13	64.5	0.45
		13-5	97.65	0.21
		13-10	99.78	0.20
		Avg	<u>86.28</u>	<u>0.28</u>
$b = \$300$ $p = \$250$		2-13	85.09	0.24
		10-2	95.44	0.21
		10-13	51.02	0.39
		13-10	97.05	0.21
		Avg	<u>87.1</u>	<u>0.26</u>

Table 6 illustrates the effect of changing the target discount factor α on optimal solution networks. Through this experiment, the target α values are achieved by varying the variable cost p while keeping the other parameters fixed. As expected, we observe that at lower α , more hubs are opened. However, the MHLP tries to utilize hub arc facilities by activating fewer hub arcs and thus, resulting in higher discount factors. For instance, with $p = \$634$ and $\alpha = 0.2$, MHLP opens six hubs at nodes (1, 3, 6, 8, 10, 13) with only nine hub arcs. The average arc utilization is 75.7% and the overall *discount* is 0.23. Furthermore, a lower discount factor leads to the selection of the most isolated node 3 as a hub given that it is one with a highest amount

of the incoming/outgoing flows. In the second solution, we decrease p to \$317 resulting in an increase of α to 0.4. In this setting, the MHLP locates three hub facilities and activates four hub arcs with an average utilization of 87.7% and an average *discount* of 0.47. Finally, with $p = \$212$ and $\alpha = 0.6$, only two hubs are opened with an average hub arc utilization of 87.1% and an average *discount* of 0.76.

Table 6: Effect of varying discount factor on optimal solution networks with $B = 650$, $R = 110$ and $b = 750$.

Instance	Solution network	Hub arcs	% Utilization	Discount
$p = 634$ $\alpha = 0.20$		1-6 3-1 6-3 6-10 8-13 10-6 10-13 13-8 13-10 Avg	55.39 60.45 66.28 86.26 65.05 94.26 98.24 79.24 75.89 <u>75.7</u>	0.36 0.33 0.30 0.23 0.31 0.21 0.20 0.25 0.29 <u>0.23</u>
$p = 317$ $\alpha = 0.40$		5-10 10-5 10-13 13-10 Avg	83.3 95.2 93.8 78.54 <u>87.7</u>	0.48 0.42 0.43 0.55 <u>0.47</u>
$p = 212$ $\alpha = 0.60$		6-13 13-6 Avg	100 74.2 <u>87.1</u>	0.60 0.92 <u>0.76</u>

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 on all arcs of the hub-and-spoke network. 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 present 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 bound. 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 analyzed the effect of changing capacities of hub and access arcs, variable costs of hub and access arcs, and discount factor on the optimal network configuration. Results show that the proposed model captures the interdependency of the hub arc utilization and the actual economies of the scale on transportation costs.

7. 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.

References

- Alumur, S., Kara, B. Y., 2008. Network hub location problems: The state of the art. *European Journal of Operational Research* 190 (1), 1–21.
- Alumur, S., Nickel, S., Saldanha da Gama, F., 2012. Hub location under uncertainty. *Transportation Research: Part B* 46, 529–543.
- Brännlund, U., 1995. A generalized subgradient method with relaxation step. *Mathematical Programming* 71 (2), 207–219.

- Bryan, D., 1998. Extensions to the hub location problem: Formulations and numerical examples. *Geographical Analysis* 30 (4), 315–330.
- Bryan, D. L., O’Kelly, M. E., 1999. Hub-and-spoke networks in air transportation: An analytical review. *Journal of regional science* 39 (2), 275–295.
- 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: *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, pp. 1154–1163.
- Campbell, J. F., Ernst, A., Krishnamoorthy, M., 2005a. Hub arc location problems: part I: Introduction and results. *Management Science* 51 (10), 1540–1555.
- Campbell, J. F., Ernst, A., Krishnamoorthy, M., 2005b. Hub arc location problems: Part II: Formulations and optimal algorithms. *Management Science* 51 (10), 1556–1571.
- Campbell, J. F., O’Kelly, M. E., 2012. Twenty-five years of hub location research. *Transportation Science* 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., 2011a. Benders decomposition for large-scale uncapacitated hub location. *Operations Research* 59 (6), 1477–1490.
- Contreras, I., Díaz, J. A., Fernández, E., 2011b. Branch and price for large-scale capacitated hub location problems with single assignment. *INFORMS Journal on Computing* 23 (1), 41–55.
- Contreras, I., Fernández, E., 2014. Hub location as the minimization of a supermodular set function. *Operations Research* 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. *Computers & Operations Research* 36 (12), 3117–3127.
- Contreras, I., Tanash, M., Vidyarthi, N., 2016. Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research* 10.1007/s10479-015-2091-2.
- Cornuejols, G., Nemhauser, G. L., Wolsey, L. A., 1983. The uncapacitated facility location problem. Tech. rep., DTIC Document.
- Correia, I., Nickel, S., Saldanha da Gama, F., 2010. Single-assignment hub location problems with multiple capacity levels. *Transportation Research: Part B* 44, 1047–1066.
- Correia, I., Nickel, S., Saldanha da Gama, F., 2014. Multi-product capacitated single-allocation hub location problems: Formulations and inequalities. *Networks and Spatial Economics* 14, 1–25.
- 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. *European Journal of Operational Research* 179 (3), 747–758.
- de Camargo, R. S., de Miranda Jr, G., Luna, H. P. L., 2009. Benders decomposition for hub location problems with economies of scale. *Transportation Science* 43 (1), 86–97.
- Ernst, A. T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science* 4 (3), 139–154.
- Ernst, A. T., Krishnamoorthy, M., 1998a. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research* 104 (1), 100–112.
- Ernst, A. T., Krishnamoorthy, M., 1998b. An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS Journal on Computing* 10 (2), 149–162.
- Guta, B., 2003. Subgradient optimization methods in integer programming with an application to a radiation therapy problem. Ph.D. thesis, University of Kaiserslautern, Germany.

- Hamacher, H. W., Labbé, M., Nickel, S., Sonneborn, T., 2004. Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics* 145 (1), 104–116.
- Horner, M. W., O’Kelly, M. E., 2001. Embedding economies of scale concepts for hub network design. *Journal of Transport Geography* 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. *Annals of Operations Research* 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. *Mathematical programming* 102 (2), 371–405.
- Lawler, E. L., 1979. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research* 4 (4), 339–356.
- Magnanti, T. L., Mirchandani, P., Vachani, R., 1995. Modeling and solving the two-facility capacitated network loading problem. *Operations Research* 43 (1), 142–157.
- Martins de Sá, E., Contreras, I., Cordeau, J., 2015a. Exact and heuristic algorithms for the design of hub networks with multiple lines. *European Journal of Operational Research* 246, 186–198.
- Martins de Sá, E., Contreras, I., Cordeau, J.-F., Saraiva de Camargo, R., de Miranda, G., 2015b. The hub line location problem. *Transportation Science* 49 (3), 500–518.

- Martins de Sá, E., de Camargo, R. S., de Miranda, G., 2013. An improved benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research* 226 (2), 185–202.
- Mirzaghafour, F., 2013. Modular hub location problems. Master’s thesis, Concordia University, Montreal, Canada.
- O’Kelly, M. E., 1986. Activity levels at hub facilities in interacting networks. *Geographical Analysis* 18 (4), 343–356.
- O’Kelly, M. E., 1998. A geographer’s analysis of hub-and-spoke networks. *Journal of transport Geography* 6 (3), 171–186.
- O’Kelly, M. E., Bryan, D., 1998. Hub location with flow economies of scale. *Transportation Research Part B: Methodological* 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. *Geographical Analysis* 47 (1), 73–96.
- Podnar, H., Skorin-Kapov, J., Skorin-Kapov, D., 2002. Network cost minimization using threshold-based discounting. *European Journal of Operational Research* 137 (2), 371–386.
- Racunica, I., Wynter, L., 2005. Optimal location of intermodal freight hubs. *Transportation Research Part B: Methodological* 39 (5), 453–477.
- Yaman, H., 2008. Star p-hub median problem with modular arc capacities. *Computers & Operations Research* 35 (9), 3009–3019.
- Zanjirani Farahani, R., Hekmatfar, M., Arabani, A. B., Nikbakhsh, E., 2013. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering* 64 (4), 1096–1109.