CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

The Cycle Hub Location Problem

Ivan Contreras
Moayad Tanash
Navneet Vidyarthi

October 2013

CIRRELT-2013-59

The Cycle Hub Location Problem

Ivan Contreras[1,2,*], Moayad Tanash[1,2], Navneet Vidyarthi[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mechanical and Industrial Engineering, Concordia University, 1515 St-Catherine Street W., EV4 139, Montréal, Canada H3G 1M8

[3] Department of Supply Chain and Business Technology Management, Concordia University, 1455 De Maisonneuve Blvd. W., Molson Building 011-321, Montréal, Canada H3G 1M8

Abstract. This paper introduces the Cycle Hub Location Problem. It considers the location of a set of hub facilities, which are connected by means of a cycle, and the allocation of user nodes to hubs so as to minimize the total cost for routing flows through the network. This problem is useful to model applications in telecommunication and transportation systems, where large set-up costs on the links and reliability requirements make cycle topologies a prominent network architecture. We extend the commonly used path based and flow based formulations for the problem. Furthermore, we present a family of mixed-dicut inequalities that improve the LP bounds of the latter formulation. These inequalities are embedded into a branch-and-cut method to solve the problem to optimality. We also present a GRASP metaheuristic that efficiently produces high quality solutions. Numerical results on a set of benchmark instances with up to 100 nodes are reported.

Keywords. Hub location, cycle topologies, branch-and-cut, GRASP.

Acknowledgments. This research was partly founded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 418609-2012 and 386501-2010. This support is gratefully acknowledged.

---

* Corresponding author: Ivan.Contreras@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1. Introduction

Hub location problems (HLPs) arise in the design of hub-and-spoke networks. They have a wide variety of applications in airline transportation, freight transportation, rapid transit systems, trucking industries, postal operations, telecommunication networks, to name a few. These systems serve demand for transportation of passengers, commodities, and/or transmission of information (data, voice, video) between many origins and destinations. Instead of connecting every origin-destination (O/D) pair directly, hub-and-spoke networks serve customers via a small number of links, where hub facilities consolidate the flows from many origins, transfer them through the hub network, and eventually distribute them to their final destinations. The use of fewer links in the network concentrates flows at the hub facilities, allowing economies of scale to be applied on routing costs, besides helping to reduce setup costs and to centralize commodity handling and sorting operations. Broadly speaking, HLPs seek to locate a set of hubs and allocate user nodes to hubs so as to minimize the total flow cost.

Since the seminal work of O'Kelly (1986), several classes of fundamental HLPs, such as p-hub median problems, uncapacitated hub location problems, p-hub center problems, and hub covering problems, have been studied in the literature. For a detailed classification and review of these HLPs, readers are referred to Campbell et al. (2002), Alumur and Kara (2008), and Campbell and O'Kelly (2012). Even though these problems are different on a number of characteristics, primarily due to their particular applications, the vast majority of them share four assumptions in common. The first assumption is that flows have to be routed via hubs and thus, paths between O/D nodes have to include at least one hub. Secondly, it is possible to connect hubs with more effective pathways that allow a constant discount factor to be applied to the flow cost between hub nodes. The third assumption is that hub arcs have no setup cost and thus, hub facilities can be connected between them at no extra cost. The forth one is that distances between nodes satisfy the triangle inequality.

To some extent, the above mentioned assumptions and their implications simplify the network design decisions. The last two assumptions allow the backbone network to be fully interconnected (i.e. a complete graph), whereas the access network is determined by the allocation pattern of O/D nodes to hub facilities. Moreover, the combination of the first, third and fourth assumptions creates O/D paths on the solution network having at least one and at most two hub nodes. These result in HLPs to have a number of attractive theoretical features, which have given rise to mathematical models (Campbell, 1994; Ernst and Krishnamoorthy, 1998a; Labbé and Yaman, 2004; Hamacher et al., 2004) and specialized solution algorithms (Ernst and Krishnamoorthy, 1998b; Labbé et al., 2005b; Contreras et al., 2011b,a) that exploit the structure of the hub-and-spoke networks to solve real-size instances. In several applications, these assumptions are reasonable and provide a good approximation to reality. However, in other applications they can lead to unrealistic results.

It is known that fully interconnected networks may be prohibitive in applications where there is a considerable setup cost associated with the hub arcs (see, for instance, O'Kelly and Miller, 1994; Klincewicz, 1998). To overcome this deficiency, several models considering incomplete hub networks have been introduced. The so-called hub arc location problems (Campbell et al., 2005b,a), relax the assumption of full interconnection between hubs and consider the location of a set of hub arcs that may (or may not) require a particular topological structure of their induced network. Some of these models do not even require the hub arcs to define a single connected component. Alumur et al. (2009) and Calık et al. (2009) study the design of incomplete hub networks with single assignments in which no network structure other than connectivity is imposed on the backbone network. Other works have also proposed models that do not consider a complete backbone network but rather, a particular topological structure. For example, Contreras et al. (2009, 2010) and de Sá et al. (2012) study the design of tree-star hub networks in which the hubs are to be connected by means of a tree and the O/D nodes are assigned to exactly one hub. These papers considers the minimization of the total flow cost whereas Kim and Tcha (1992); Lee et al. (1996), and Lee et al. (1993) focus on minimizing the setup costs associated with the design of tree-star networks. Labbé and Yaman (2008) and Yaman (2008) consider the design of start-star networks in which hub nodes are directly connected to a central node (i.e. star backbone network) and the O/D nodes are assigned to exactly one hub node. Yaman (2009) studies the problem of designing a three-layer hub-and-spoke network, where the top layer consists of a complete network connecting the central hubs, and the second and third layers are unions of start networks connecting the remaining hubs to central hubs and the O/D nodes to hubs, respectively. Yaman and Elloumi (2012) consider the design of two-level start networks, while taking into consideration the service quality in terms of the length of paths between pair of

O/D nodes. de Sá et al. (2013) study the problem of designing a hub-line network in which hubs are connected by means of a line and the aim is to minimize the total service time between pairs of nodes. Other studies on incomplete hub networks include Campbell (2009), Contreras and Fernández (2012b), and Davari et al. (2013).

In this paper we study the *cycle hub location problem* (CHLP), which consist of locating exactly p hub facilities that are connected with a set of hub arcs with a cycle topology. Each O/D node must be allocated to exactly one hub (i.e. single assignment) and flows between pair of nodes have to be routed through the cycle-star network so as to minimize the total flow cost. The CHLP is a challenging NP-hard problem that combines location and network design decisions. The location decision focuses on the selection of the set of nodes to locate facilities, whereas the network design decisions deals with the design of the cycle-star network, by selecting the access and hub arcs as well as the routing of flows through the network. To the best of our knowledge, the CHLP was first introduced in Contreras and Fernández (2012a) in the context of general network design problems, but there is no paper in the literature dealing with approximate or exact solution methods for solving it.

Besides HLPs, the CHLP shares some similarities with other network design problems in which a cycle-star network is sought. The so-called ring-star problem (RSP), introduced by Labbe et al. (2004), aims to locate a simple cycle through a subset of nodes with the objective of minimizing the sum of setup costs of the cylce and assignment costs from the vertices (not in the cycle) to their closest vertex on the cycle. Another closely related problem is the median cycle problem (MCP), studied by Labbé et al. (2005a). This problem arises in the design of ring-shaped infrastructures and consists of finding a simple cycle that minimizes the setup costs of the cycle, such that the total assignment cost of the non-visited nodes do not exceed a given budget constraint. Current and Schilling (1994) and Gendreau et al. (1997) study covering versions of the RSP in which all nodes must be within a prespecified distance from the cycle. Baldacci et al. (2007) present the capacitated m-ring star problem, which deal with the location of m cycles that pass through a central node and the assignment of nodes to cycles. Lee et al. (1998) and Xu et al. (1999) study the Steiner ring-star problem, in which the cycle only contains Steiner nodes chosen from a given set. Current and Schilling (1994) consider the median tour problem, where a cycle with p nodes has to be located. It is a bicriterion problem, which consists of minimizing the setup cost of the cycle as well as minimizing the total assignment cost of nodes to their closest facilities. Liefooghe et al. (2010) study a bi-objective ring-star problem, in which the setup cost of the cycle and the assignment costs are considered. See Labbé et al. (1998) and Laporte and Martín (2007) for additional models related to the location of cycle structures on a network.

Potential applications of models where cycle start networks are used arise when the setup cost of the arcs of the network are very high. When minimizing such setup cost, tree-star topologies are particularly attractive as they minimize the number of links on the network as it contains exactly one path between pair of nodes. However, in the design of reliable networks, cycle topologies may be preferred to tree topologies as they offer an alternative path between any pair of nodes when a link connecting two nodes fails for some reason. That is, a cycle topology guarantees connectivity of the remaining network. Particular applications of these models arise in the design of telecommunication networks and rapid transit systems planning. In the former case, terminal nodes are usually connected to concentrators (or facilities) by point-to-point links, resulting in a star structure, and concentrators are interconnected by a ring. See Xu et al. (1999) for an example of digital data service design. In the latter case, the goal is to select a set of facilities, which are served by a single vehicle route (a cycle), and the assignment of demand nodes to their closest facility. Laporte and Martín (2007) provide additional applications considering cycle-star structures.

The contributions of this paper are threefold. Firstly, we introduce two mixed integer programming (MIP) formulations for the CHLP and to computationally compare them with a general purpose solver. One of them is based on flow variables that compute the amount of flow that passes through particular hub arcs, whereas the other is based on path variables that determine if a hub arc is used on the path between two pair of nodes. Secondly, we present an exact branch-and-cut method to solve the problem to optimality. It uses the flow-based formulation and an adaptation of the well-known mixed dicut inequalities, together with an efficient heuristic method for the separation problem, as a bounding procedure in each node of the enumeration tree. The third one is to develop a greedy randomized adaptive search procedure (GRASP) based metaheuristic to obtain high quality feasible solutions to the CHLP. A series of computational experiments are performed to compare the efficiency of both formulations and the proposed solution methods. Computational results on benchmark instances confirm the efficiency of our algorithms.

The remainder of the paper is organized as follows. Section 2 formally defines the problem and presents the two MIP formulations. In Section 3, we describe the proposed branch-and-cut method and GRASP metaheuristic. The computational results and analysis are presented in Section 4. Finally, conclusions are given in Section 5.

2. Problem Description

Let $G = (N, A)$ be a complete digraph, where $N = \{1, 2, \ldots, n\}$ represents the set of O/D nodes as well as the potential sites for locating hubs. For each ordered pair $i, j \in N \times N$, let W_{ij} denote the amount of flow between origin i and destination j. Thus, $O_i = \sum_{i \in N} W_{ij}$ is the total flow originating at node $i \in N$, and $D_i = \sum_{j \in N} W_{ij}$, is the total flow with destination node $i \in N$. The distances, or flow costs d_{ij} between nodes i and j are assumed to be symmetric, however, they need not satisfy the triangle inequality property. Given that hub nodes are no longer fully interconnected, O/D paths on the solution network may contain more than two hub nodes. Then, the per unit flow cost is given by the length of the path between an origin and destination, where the discount factor $0 < \alpha < 1$ is applied to all hub arcs contained on the path.

The CHLP seeks to determine the location of exactly p hubs that are connected by means of a cycle, and the routing of flows through the hub-and-spoke network. Each node has to be allocated to exactly one hub and if a node is selected as a hub, then it is self-assigned. The objective is to minimize the total transportation cost. In every feasible solution to the CHLP: i) there exist p hub arcs; ii) every hub node is connected with exactly two other hub nodes; iii) the graph induced by the hubs does not contain subtours, and iv) there are exactly two paths between every pair of nodes on the network. This makes the CHLP more difficult to formulate and solve, as the shortest path between O/D nodes, containing an undetermined number of hub nodes and hub arcs, needs to be determined to compute the objective function. Note that when $p \in \{1, 2, 3\}$, the CHLP reduces to a classical hub location problem in which all nodes are fully interconnected.

The routing decisions of the CHLP are affected by both the allocation and the network design decisions. Hence, formulations for this problem must keep track of O/D paths. In what follows, we present two MIP formulations for the CHLP. The first one uses path variables to determine the set of arcs on each O/D paths, whereas the second one uses flow variables to compute the amount of flow routed through a particular arc.

2.1. Path-Based Formulation

For each $i, k \in N; i \neq k$, we define binary location/allocation variables,

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

When $z_{kk} = 1$, node k is selected as a hub and assigned to itself. For each $k, m \in N$, $k < m$, we also introduce binary hub arc variables

$$y_{km} = \begin{cases} 1 & \text{if hub arc } (k, m) \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we define for each $i, j, k, m \in N$, we define binary routing variables as follows:

$$X_{ijkm} = \begin{cases} 1 & \text{if the the flow from } i \text{ to } j \text{ traverses arc } (k, m), \\ 0 & \text{otherwise,} \end{cases}$$

The CHLP can be stated as follows:

$$(PF)\ \min \quad \sum_{i\in N}\sum_{k\in N}(c_{ik}O_i + c_{ki}D_i)z_{ik} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha W_{ij}c_{km}X_{ijkm}$$

$$\text{s.t.} \quad \sum_{k\in N} z_{ik} = 1 \qquad \forall i \in N \tag{1}$$

$$\sum_{\substack{m\in N\\ m\neq k}} X_{ijkm} + z_{jk} - \sum_{\substack{m\in N\\ m\neq k}} X_{ijmk} - z_{ik} = 0, \forall i,j,k, i\neq j,\ k\neq j \tag{2}$$

$$X_{ijkm} + X_{ijmk} \leq y_{km} \qquad \forall i,j,k \in N, \forall m > k \tag{3}$$

$$\sum_{k<m} y_{km} + \sum_{k>m} y_{mk} = 2z_{kk} \qquad k \in N \tag{4}$$

$$\sum_{k\in N} z_{kk} = p \tag{5}$$

$$\sum_{k\in N}\sum_{m>k} y_{km} = p \tag{6}$$

$$X_{ijkm} \geq 0 \qquad \forall i,j,k,m \in N,\ k\neq m \tag{7}$$

$$z_{ik} \in \{0,1\} \qquad \forall i,k \in N \tag{8}$$

$$y_{km} \in \{0,1\} \qquad \forall k \in N, \forall m > k. \tag{9}$$

The first and second terms of the objective function represent the transportation cost between access arcs and hub arcs, respectively. Constraints (1) ensure that each node is allocated to one hub. Constraints (2) are the well-known flow conservation constraints used to model O/D paths. Constraints (3) ensure that paths between origins and destinations will use open hub arcs. Constraints (4) guarantees that each hub node must be connected to exactly two other hub nodes. Constraint (5) is a cardinality constraint on the number of hubs that can be opened, whereas constraint (6) state that the number of hub arcs required to define the cycle is equal to p. Finally, constraints (7)–(9) are the integrality constraints. The combination of constraints (1)–(6) will create paths between all pair of nodes and will form a cycle-star topology with a single connected component. As a consequence, classical sub-tour elimination constraints, commonly used to model cycles, are not necessary to describe the set of feasible solutions to the CHLP.

2.2. Flow-Based Formulation

In order to keep track of the path that is used to send the flow between O/D nodes, we use a flow variable that is commonly used in the hub location literature (see, for instance Ernst and Krishnamoorthy, 1998b, 1999; Boland et al., 2004; Marín, 2005; Contreras et al., 2010). For each $i \in N$ and $(k, m) \in A$, we define

x_{ikm}= amount of flow with origin in node $i \in N$ that traverses arc (k, m).

We also use the z_{ik} and y_{km} binary variables for the location/allocation and network design decisions.

The CHLP can be formulated as follows:

$$(FF) \ \min \quad \sum_{i\in N}\sum_{k\in N}(c_{ik}O_i + c_{ki}D_i)z_{ik} + \sum_{i\in N}\sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha c_{km}x_{ikm}$$

$$\text{s.t.} \quad \sum_{k\in N} z_{ik} = 1 \qquad \forall i \in N \tag{10}$$

$$O_i z_{ik} + \sum_{\substack{m\in N\\ m\neq k}} x_{imk} = \sum_{\substack{m\in N\\ m\neq k}} x_{ikm} + \sum_{m\in N} W_{im}z_{mk} \qquad \forall i, k \in N; k \neq i \tag{11}$$

$$z_{km} + y_{km} \leq z_{mm} \qquad \forall k, m \in N; m > k \tag{12}$$

$$z_{mk} + y_{km} \leq z_{kk} \qquad \forall k, m \in N; m > k \tag{13}$$

$$x_{ikm} + x_{imk} \leq O_i y_{km} \qquad \forall i, k, m \in N; m > k \tag{14}$$

$$\sum_{k<m} y_{km} + \sum_{k>m} y_{mk} = 2z_{kk} \qquad k \in N \tag{15}$$

$$\sum_{k\in N} z_{kk} = p \tag{16}$$

$$\sum_{k\in N}\sum_{m\in N} y_{km} = p \tag{17}$$

$$x_{ikm} \geq 0 \qquad \forall i, k, m \in N \tag{18}$$

$$z_{km}, y_{km} \in \{0, 1\} \qquad \forall k, m \in N \tag{19}$$

Constraints (10), (15)–(19) have the same interpretation as in the previous path based formulation. For all $m > k$, constraints (12) ensure that each demand node is assigned to an open hub. When $y_{km} = 1$, a hub is located at node k, so that $z_{kk} = 1$, and, therefore, by constraints (10), $z_{km} = 0$. Similarly, constraints (13) link x and y variables when $m < k$. Constraints (14) ensure that the flow between hubs will only be routed through the hub cycle. Constraints (11) are the flow conservation constraints. The assumption that the graph of flows contains a single connected component together with constraints (10)–(19) eliminates the need for subtour elimination constraints to describe the set of feasible solutions.

3. Solution Algorithms

In this section, we present two different solution algorithms for solving the CHLP. The first one is an exact branch-and-cut algorithm that uses the linear programming (LP) relaxation of the flow-based formulation, together with a family of strong valid inequalities, as a lower bounding procedure at nodes of the enumeration tree to obtain optimal solutions to the CHLP. The second one is a GRASP metaheuristic that is used to efficiently obtain feasible solutions to the problem.

3.1. Valid Inequalities

We present a family of valid inequalities for the CHLP, which are an extension of the so-called *mixed dicut inequalities* introduced for the fixed-charge network flow problem in Ortega and Wolsey (2003). Let Z denote the set of feasible integer solutions of FF.

Proposition 1. *For $i, m \in N$, $F \subseteq N \setminus \{m\}$, $J \subseteq N \setminus \{i, m\}$, and $Q = \sum\limits_{j \in J \cup \{m\}} W_{ij}$ the inequality*

$$\sum_{k \in N \setminus (F \cup \{m\})} x_{ikm} + Q\Big(\sum_{\substack{k \in F \\ k < m}} y_{km} + \sum_{\substack{k \in F \\ k > m}} y_{mk}\Big) \geq \sum_{j \in J \cup \{m\}} W_{ij}(z_{jm} - z_{im}) \qquad (20)$$

is valid for Z.

Proof:

Observe that the right-hand-side of inequality (20) can only be non-negative when m is a hub and i is not assigned to m since, otherwise it would be smaller than or equal to zero. When m is a hub and node i is not allocated to m, the right-hand-side $\sum_{j \in J \cup \{m\}} W_{ij}(z_{jm} - z_{im})$, denotes the total flow coming from i and destined to either the hub node m or a non-hub node j assigned to hub m. Hence, the right-hand-side of (20) is a lower bound on the total flow arriving to m from i. In the case of the left-hand-side, we note that in any feasible solution that node i is not allocated to hub m, any amount of flow routed from i to m will arrive to m via another hub $\hat{k}$ (possibly node i). Moreover, given that all flow originating at node i and entering hub k will be routed through the shortest path between these nodes, only one of the two terms in the left-hand-side will be positive, depending on whether $\hat{k} \in F$ or not. On one side, if $\hat{k} \notin F$ then the first term will be the positive one, which represents the total flow originated at i arriving to m through hub $\hat{k}$ and thus, will be an upper bound on the value of the right-hand-side of the inequality. On the other side, if $\hat{k} \in F$ the second term will be the positive one, which is the total flow from i to m and other nodes allocated to it and thus, will be an upper bound on the value of the right-and-side of the inequality and the result follows. ■

3.2. Separation Problem of Valid Inequalities

Given a fractional solution $(\bar{x}, \bar{y}, \bar{z})$ to the LP relaxation of the FF, the separation problem of inequalities (20) must be solved to find the most violated inquality at $(\bar{x}, \bar{y}, \bar{z})$ for each $i, m \in N$, if any. Contreras et al. (2010) show that the extended mixed-dicut inequalities (20) are also valid for the *tree of hubs location problem* and propose an exact algorithm to separate them. Given that this algorithm has an exponential running time due to the solution of several knapsack problems, finding the most violated inequality is rather time consuming. For this reason, we next propose a fast heuristic algorithm to find violated inequalities (20) without having to solve the separation problem to optimality.

For each pair of nodes $i, m \in N$, we want to find sets F and J such that

$$\sum_{k \in N \setminus (F \cup \{m\})} \bar{x}_{ikm} + Q(\sum_{\substack{k \in F \\ k < m}} \bar{y}_{km} + \sum_{\substack{k \in F \\ k > m}} \bar{y}_{mk}) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im}) < 0.$$

Note that the set $J \subseteq N \setminus \{i, m\}$ affects both the left-and right-hand-side of the inequality, whereas the set $F \subseteq N \setminus \{m\}$ affects only the left-hand-side. Moreover, given a set J and its associated $Q = \sum_{j \in J \cup \{m\}} W_{ij}$, we can efficiently select the set F that minimizes the value of

$$L(Q) = \min_{F \subseteq N \setminus \{m\}} \sum_{k \in N \setminus (F \cup \{m\})} \bar{x}_{ikm} + Q\Big(\sum_{k \in F : k < m} \bar{y}_{km} + \sum_{k \in F : k > m} \bar{y}_{mk}\Big), \tag{21}$$

using the following result.

Proposition 2. *Let* $i, m \in N$, $Q \geq 0$, *and* $(\bar{x}, \bar{y}, \bar{z})$ *be given. Then, a set* $\overline{F} \subseteq N \setminus \{m\}$ *that minimizes the value of* $L(Q)$ *is given by* $\overline{F} = F_{<} \cup F_{>}$, *where*

$$F_{<} = \{k \in N : k < m \text{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{km}} \geq Q\},$$

and

$$F_{>} = \{k \in N : k > m \text{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{mk}} \geq Q\}.$$

The proof of this result is given in Contreras et al. (2010) and is thus omitted.

The proposed heuristic works by iteratively evaluating different subsets $J \subseteq N \setminus \{i, m\}$ and evaluating $L(Q)$ to check whether the associated inequality is violated or not. It first constructs an initial set J_0 by considering all $j \in N$ such that $(\bar{z}_{jm} - \bar{z}_{im}) > 0$. It then modifies the set J_0 by removing elements from it one at a time and evaluating the magnitude of the (possible) violation of the inequality. The outline of the algorithm is depicted in Algorithm 3. Let δ denote the smallest difference between the left-hand-side and right-hand-side of the constraint. If the output of the algorithm gives $\delta < 0$, it means that a violated inequality has been found.

Algorithm 3: Separation of inequalities (20) for (i, m)

$J_0 \leftarrow \emptyset$
for $(j \in N)$ **do**
  if $(\bar{z}_{jm} - \bar{z}_{im} > 0)$ **then**
    $J_0 \leftarrow J_0 \cup \{j\}$
  end if
end for
$\delta \leftarrow L\left(\sum_{j \in J_0 \cup \{m\}}\right) - \sum_{j \in J_0 \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})$
$J \leftarrow J_0$
for $(l \in J_0)$ **do**
  $J \leftarrow J \setminus \{l\}$
  if $\left(\delta > L\left(\sum_{j \in J \cup \{m\}}\right) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})\right)$ **then**
    $\delta \leftarrow L\left(\sum_{j \in J \cup \{m\}}\right) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})$
  else
    $J \leftarrow J \cup \{l\}$
  end if
end for

3.3. Branch-and-Cut Algorithm

We present an exact branch-and-cut method based on the flow-based formulation for solving the CHLP to optimality. The idea is to solve the LP relaxation of FF with a cutting-plane algorithm by initially including only constraints (10)-(13), (15)-(17) at the root node and iteratively adding constraints (14) and (20) only when violated by the current LP solution. When no more violated inequalities are found, we resort to CPLEX for solving the resulting formulation by enumeration, using a call-back function for generating additional violated constraints (14) and (20) at the nodes of the enumeration tree. The separation problem of inequalities (14) is solved by inspection at every node of the tree. The separation of inequalities (20) is carried out using Algorithm 3 at the root node and at every other nodes in which its depth is multiple of 25. For the case of constraints (20), we limit the number of generated cuts at every iteration of the separation phase by using a threshold value equal to 1 for the minimum violation required for a cut to be added. We use a branching strategy in which the highest priority is given to the location variables z_{kk}, and then the hub arc variables y_{km} and finally, to the allocation variables z_{ik}.

3.4. GRASP

The Greedy Randomized Adaptive Search Procedure (GRASP) is a multi-start meta-heuristic used for solving combinatorial optimization problems (Festa and Resende, 2011). Each iteration consists of two phases: a constructive phase and a local search phase. In the constructive phase, we obtain a feasible solution using a three-stage procedure. In the first step, a set of p hubs is randomly selected among a set of candidate nodes. The remaining nodes are then assigned to their closest open hub. Finally, a set of p hub arcs, associated with the selected hub nodes, are then chosen in such a way that they form a cycle on the backbone network. A local search phase is later used to improve the initial solution by sequentially exploring different types of neighbourhoods that modify the structure of the network but always considering feasible solutions (i.e., ring-star networks).

In what follows, solutions are represented by a set of hub nodes H, a set of hub arcs E, and an assignment mapping a. Therefore, solutions are designated by the form $S = (H, E, a)$,where $H \in N$ denotes the set of selected sites to locate hubs, i.e., $H(i) = 1$ if node $i \in N$ is selected to be a hub, $E : e \to R$ represents the set of arcs between hub nodes, i.e., $E(e) = 1$ if hub arc e is exists, and $a : N \to H$ is the assigning mapping, i.e., $a(j) = m$ if node $j \in N$ is assigned to hub node $m \in H$.

3.4.1. Constructive Phase

Let $S = (H, E, a)$ be a partial solution where $H(i) = null$, $E(e) = null$ and $a(j) = null$. To generate a feasible solution, three stages are considered; the selection of a set of hubs, the assignment of nodes to hubs and the connection of hubs so as to construct a cycle structure. A restricted candidate list (RCL) is built using a greedy function, where, at each iteration t a sub-region $N_i^t(r) = \{j \in N^t : d_{ij} \le r\}$ of candidate nodes N^t around a node i with a radius of size r is considered. We define the greedy function as

$$\psi_i^t = \begin{cases} \sum\limits_{j \in N_i^t(r)} (W_{ij} + W_{ji}), & \text{if } t = 1, \\ \sum\limits_{j \in N_i^t(r)} (W_{ij} + W_{ji}) + \sum\limits_{j \in N_i^t(r)} \sum\limits_{k \in \{1,\dots,t-1\}} \sum\limits_{m \in N_{i(k)}^k(r)} (W_{jm} + W_{mj}), & \text{if } t > 1, \end{cases}$$

where $i(k)$ denotes the node selected as a hub at iteration k. The first term represents the flow originated at node i with destination $N_i^t(r)$, and the total flow going into node i coming from nodes in $N_i^t(r)$. That is, node i is considered as a potential hub to serve nodes $j \in N_i^t(r)$. The second term represents the amount of flow that needs to be routed between nodes inside the sub-region $N_i^t(r)$ of a candidate hub node i and the nodes inside the sub-regions $N_{i(k)}^k(r)$ of the open hubs $i(k)$ selected in previous iterations $k = 1, \ldots, t-1$.

In order to achieve a trade-off between quality and diversity, a partially randomized greedy procedure is considered. At each iteration, one element is randomly selected from the RCL to become a hub. The RCL is updated at each iteration of the construction phase and contains the best candidate nodes N^t with respect to the greedy function . Let $\psi_{min}^t = \min\{\psi_i^t : i \in N^t\}$ and $\psi_{max}^t = \max\{\psi_i^t : i \in N^t\}$, then

$$RCL = \{i : \psi_i^t \geq \psi_{min}^t + \alpha\left(\psi_{max}^t - \psi_{min}^t\right)\},$$

where $0 \leq \alpha \leq 1$. Once a hub is located at a candidate node i, we remove all nodes in $N_i^t(r)$ from the set of candidate nodes N^{t+1}. When p hubs are opened, all the non-hub nodes are simply assigned to their closest opened hub. In order construct a feasible solution, a *nearest neighbor algorithm* (see, Cook et al., 1998) is applied to connect the set of selected hubs by means of a cycle. This algorithm works by arbitrarily selecting hub node and connecting it to the nearest hub not yet connected. The process continues until all hubs are connected and finally connects the last hub with the first one. An outline of the constructive phase is provided in Algorithm 1.

Algorithm 1: Constructive Phase of GRASP

Let$H$ be the subset of selected hub nodes
Let N^t be the set of candidate nodes at iteration t
$H \leftarrow \phi$
$t \leftarrow 1$
$N^t \leftarrow N$
while $(\mid H \mid \neq p)$ **do**
  Evaluate ψ_i^t for all $i \in N^t$
  $RCL = \{i : \psi_i^t \geq \psi_{min}^t + \alpha\,(\psi_{max}^t - \psi_{min}^t)\}$
  Select randomly $i^* \in RCL$
  $H \leftarrow H \cup i^*$
  $N^t \leftarrow N^{t-1} \backslash \{i^* \cup N_{i^*}^t(r)\}$
  $t \leftarrow t + 1$
end while
Assign each node $j \in N^t$ to its closest hub in H.
Connect hubs using the Nearest Neighbor Algorithm.

3.4.2. Local Search Phase

The local search procedure is used to improve the initial solution obtained by the constructive phase. To do so, it explores three types of neighbourhood structures. The first type consist in the classical shift and swap neighbourhood. The latter one tries to improve the current solution by changing the assignment of one node, whereas the former one considers all solutions that differ from the current one by swapping the assignment of two nodes. Let $S = (H, E, a)$ be the current solution, then

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

and

$$N_{swap}(s) = \{s^{'} = (H, E, a^{'}) : \exists!(j_1, j_2),\ j_1^{'} = a(j_2),\ j_2^{'} = a(j_1),\ \forall j \neq j_1, j_2\}.$$

To explore N_{shift}, all pairs of the form (i, j) are considered, where $a(j) \neq i$ and for N_{swap} all pairs of the form (j_1, j_2) are considered, where $a(j_1) = a(j_2)$. In both cases we use a best improvement strategy.

The second type of neighbourhood structure affects the current set of open hubs. Let $S = (H, E, a)$ be the current solution and let $i \in N \backslash H$ be the nodes which are candidate to replace the open hubs located at site

$m \in H$, then

$$N_{open/close} = \{S^{'} = (H^{'}, E^{'}, a^{'}) : S^{'} = H^{'} \backslash \{m\} \cup \{i\}, m \in S^{'},\ i \in N \backslash H\}.$$

To explore $N_{open/close}$ all nodes $m \in N \backslash H$ are considered, and a set of solutions is obtained from the current one by interchanging an open hub by a closed one and reassigning all the non-hub nodes to their closest open hub. If we find an improved solution within this neighbourhood, we explore the N_{shift} neighbourhood to try to further improve the solution.

The third type of neighbourhood structure is the so-called *2-opt* (Cook et al., 1998), commonly used in other optimization problems in which cycle structures are sought. The procedure works by deleting two hub arcs and reconnecting the network with a new cycle. Let $S = (H, E, a)$ be the current solution, then

$$T_{2-opt} = \{S = (H, E^{'}, a) : E^{'} = E \backslash \{(i_1, j_1), (i_2, j_2) \cup \{(i_1, i_2), (j_1, j_2)\}\}\}.$$

In this neighbourhood, a best improvement strategy is also considered. An outline of the local search procedure is depicted in Algorithm 2.

Algorithm 2 Local Search phase of GRASP

$stoppingcriteria \leftarrow false$
while ($stoppingcriteria = false$) **do**
    $explore\ T_{2-opt}t$
    if (solution not improved in T_{2-opt}) **then**
        $exploreN_{shift}$
        if (solution not improved in T_{2-opt} and N_{shift}) **then**
            $exploreN_{swap}$
            if (solution not improved in T_{2-opt}, N_{shift} and N_{swap}) **then**
                $exploreN_{open/close}$
                if(solution has not been updated) **then**
                    $stoppingcriteria \leftarrow true$
                end-if
            end-if
        end-if
    end-if
end-while

4. Computational Results

We conduct computational experiments to analyse and compare the performance of the two formulations, the branch-and-cut method, and the the GRASP metaheuristic. All formulations and algorithms were coded in C and run on a Lenovo ThinkStation with an Intel Xeon CPU E31230 processor at 3.20 GHz and 16 GB of RAM under Windows 7 environment. All integer programs were solved using the callback library of CPLEX 12.4. The numerical tests were performed using the well-known Australian Post (AP) instances *(mscmga.ms.ic.ac.uk/jeb/orlib/phubinfo.html)*. These instances comprise of postal flow and Euclidean distances between 200 cites in Australia. In our experiments, we have selected instances for which $|N| =$ 10, 20, 25, 40, 50, 60, 70, 75, 90, and 100; $p =$ 4, 6 and 8; and α =0.2, 0.5 and 0.8.

The first set of computational results compares the path based formulation (PF) and flow based formulation (FF) when solved with a general purpose solver, such as CPLEX. The detailed results of this comparison on a set of instances ranging from 10 to 40 nodes are shown in Table 1. The first three columns provide the number of nodes ($|N|$), the numbers of hubs p and the discount factor α, respectively, for each instance. The next columns report the linear programming relaxation gap ($\%LP$), the percent deviation between final upper and lower bound ($\%Gap$), the CPU time in seconds (CPU), and the number of explored nodes in the branching tree ($Nodes$), for both formulations. The $\%LP$ Gap is computed as $(UB-LP)/(UB) \times 100\%$, where UB denotes the best upper bound (or optimal solution value) obtained with CPLEX and LP is the optimal value of the LP relaxation. The final gap ($\%Gap$) is computed as $(UB - LB)/(UB) \times 100\%$, where UB and LB denote the best upper and lower bounds obtained by CPLEX at termination, respectively. Throughout this experiment, the maximum time limit is set to 14,400 seconds of CPU time on CPLEX. Whenever CPLEX fails to solve an instance to optimally within the time limit, we write *time*.

Instance			Path-Based Formulation(PF)				Flow-Based Formulation(FF)			
$\|N\|$	p	$alpha$	$\%LP$	$\%Gap$	CPU	$Nodes$	$\%LP$	$\%Gap$	CPU	$Nodes$
10	4	0.2	2.25	0.0	1.5	4	3.66	0.0	2.08	25
10	4	0.5	0.00	0.0	0.48	0	5.41	0.0	0.67	43
10	4	0.8	0.00	0.0	0.5	0	6.96	0.0	0.97	16
10	6	0.2	0.00	0.0	51	0	5.66	0.0	1.14	73
10	6	0.5	0.00	0.0	0.52	0	8.70	0.0	1.26	261
10	6	0.8	0.00	0.0	0.52	0	10.92	0.0	1.54	877
20	4	0.2	1.05	0.0	890.13	9	1.73	0.0	3.78	36
20	4	0.5	0.29	0.0	470.92	2	4.33	0.0	17.69	485
20	4	0.8	0.00	0.0	346.68	0	5.17	0.0	29.16	1024
20	6	0.2	0.78	0.0	511.5	4	5.60	0.0	30.48	753
20	6	0.5	0.00	0.0	320.8	0	8.26	0.0	350.89	9693
20	6	0.8	0.60	0.0	614.46	8	9.73	0.0	1127.92	32563
20	8	0.2	1.37	0.0	753.16	15	7.35	0.0	180.73	4837
20	8	0.5	1.93	0.0	2159.55	56	12.84	0.0	2736.87	58631
20	8	0.8	1.17	0.0	883.06	20	12.78	0.0	8516.44	245771
25	4	0.2	4.89	4.0	time	6	1.79	0.0	13.65	66
25	4	0.5	0.05	0.0	2661.86	0	3.15	0.0	46.69	248
25	4	0.8	0.00	0.0	707.68	0	4.51	0.0	118.65	928
25	6	0.2	0.00	0.0	2658.85	0	3.47	0.0	32.26	312
25	6	0.5	0.00	0.0	1289.08	0	6.35	0.0	327.76	3435
25	6	0.8	0.25	0.0	7554.94	3	8.88	0.0	4065.04	37955
25	8	0.2	1.08	1.1	time	32	7.51	0.0	3169.96	22421
25	8	0.5	0.45	0.0	7285.53	12	10.12	0.0	7711.25	68475
25	8	0.8	0.98	0.0	7393.37	50	11.09	1.1	time	110994
40	4	0.2	time	time	time	time	1.65	0.0	71.232	127
40	4	0.5	time	time	time	time	3.40	0.0	3445.22	1816
40	4	0.8	time	time	time	time	2.45	1.7	time	7195
40	6	0.2	time	time	time	time	4.10	0.0	4119.21	3332
40	6	0.5	time	time	time	time	8.56	7.0	time	4056
40	6	0.8	time	time	time	time	9.21	7.5	time	4929
40	8	0.2	time	time	time	time	6.55	4.6	time	3217
40	8	0.5	time	time	time	time	12.42	11.2	time	4600
40	8	0.8	time	time	time	time	14.82	14.1	time	5093

Table 1: Comparison between path-based and flow-based formulation.

In Table 1, we can observe that path based formulation is able to solve 22 out of the 33 instances to optimality within the time limit. The % LP gap for the instances that were solved is relatively small, as is always less than 2.25% and in 11 instances it is equal to zero. However, CPLEX is not able to solve the LP relaxation of all 40-node instances in four hours of CPU time. In the case of flow based formulation, CPLEX is able to solve 26 out of the 33 instances to optimality. As expected, the % LP gap for the instances that were solved is always larger than that of the PF. Nevertheless, given that there is a considerable smaller number of variables and constraints in FF, it is able to solve three 40-node instances and one 25-node instance that the PF cannot solve.

In order to analyse the efficiency of our proposed approximate and exact algorithms we have run a second series of computational experiments using a set of instances ranging from 10 to 50 nodes. In particular, we investigate the effect of the valid inequalities (20) within a branch-and-cut framework and the quality of the obtained solutions with the GRASP algorithm. For these experiments, we set the maximum time limit to 86,400 seconds of CPU time. The results are summarized in Table 2. The first five columns have the same meaning as in Table 1. The results of the columns under the heading *branch-and-cut* give: *%LP-cuts* the LP relaxation bound at the root node after adding constraints (20), and *%Gap* the final percent deviation at termination. In order to assess the quality and robustness of GRASP, the algorithm was run 30 times for each instance and the results are reported under the heading $GRASP$. The best objective value obtained in all runs is used to compute the best percentage deviation ($\%Dev$) with respect to the optimal solution value or the best LB bound obtained ($i.e.$, $\%Dev = (best\ solution\ GRASP - LB)/(best\ solution\ GRASP) \times 100\%$). The robustness of the GRASP is measured by using the average percent deviation ($\%Avg\ Dev$) using the best solutions obtained in each or the 30 runs. An asterisk is used to indicate that GRASP was not able to obtain the optimal (or the best known) solution. The average CPU time in seconds for all runs of the GRASP is reported in the last column of the table.

Instance			Flow-based formulation (FF)				Branch-and-Cut				GRASP		
$\|N\|$	P	α	% LP	% Gap	CPU	Nodes	%LP-cuts	%Gap	CPU	Nodes	% Dev	% Avg	CPU
10	4	0.2	3.66	0.00	2.08	25	0.92	0.00	0.09	14	0.00	0.00	0.03
10	4	0.5	5.41	0.00	0.67	43	0.72	0.00	0.09	6	0.00	0.00	0.03
10	4	0.8	6.96	0.00	0.97	16	0.69	0.00	0.11	10	0.00	0.00	0.03
10	6	0.2	5.66	0.00	1.14	73	0.91	0.00	0.13	16	0.00	0.00	0.04
10	6	0.5	8.70	0.00	1.26	261	1.54	0.00	0.22	20	0.00	0.00	0.04
10	6	0.8	10.92	0.00	1.54	877	2.82	0.00	0.26	78	0.00	0.00	0.04
20	4	0.2	1.73	0.00	3.78	36	0.04	0.00	0.59	6	0.00	0.00	0.21
20	4	0.5	4.33	0.00	17.69	485	1.58	0.00	4.27	120	0.00	0.00	0.21
20	4	0.8	5.17	0.00	29.16	1024	1.53	0.00	3.45	120	0.00	0.00	0.25
20	6	0.2	5.60	0.00	30.48	753	1.23	0.00	6.65	157	0.00	0.00	0.32
20	6	0.5	8.26	0.00	350.89	9693	2.89	0.00	49.02	683	0.00	0.00	0.35
20	6	0.8	9.73	0.00	1127.92	32563	4.58	0.00	254.7	4243	0.00	0.04	0.39
20	8	0.2	7.35	0.00	180.73	4837	3.06	0.00	61.18	1365	0.00	0.00	0.45
20	8	0.5	12.84	0.00	2736.87	58631	6.51	0.00	4172.21	16191	0.00	0.00	0.52
20	8	0.8	12.78	0.00	8516.44	245771	6.32	0.00	7108.94	43676	0.00	0.03	0.51
25	4	0.2	1.79	0.00	13.65	66	0.18	0.00	2.4	14	0.00	0.00	0.42
25	4	0.5	3.15	0.00	46.69	248	0.61	0.00	4.77	30	0.00	0.00	0.49
25	4	0.8	4.51	0.00	118.65	928	1.81	0.00	9.22	131	0.00	0.00	0.5
25	6	0.2	3.47	0.00	32.26	312	0.31	0.00	5.96	32	0.00	0.00	0.66
25	6	0.5	6.35	0.00	327.76	3435	1.99	0.00	20.27	420	0.00	0.00	0.79
25	6	0.8	8.88	0.00	4065.04	37955	4.39	0.00	456.65	4195	0.00	0.00	0.88
25	8	0.2	7.51	0.00	3169.96	22421	3.59	0.00	503.71	2846	0.00	0.00	0.97
25	8	0.5	10.12	0.00	7711.25	68475	4.76	0.00	15580.55	10230	0.00	0.00	1.11
25	8	0.8	11.09	0.00	18229.57	152835	5.99	0.00	40142.86	70163	0.00	0.55	1.11
40	4	0.2	1.65	0.00	71.23	127	0.34	0.00	24.63	32	0.00	0.00	2.16
40	4	0.5	3.40	0.00	3445.22	1816	1.09	0.00	95.28	70	0.00	0.00	2.39
40	4	0.8	2.45	0.00	24354.44	16182	2.54	0.00	476.58	596	0.00	0.00	2.62
40	6	0.2	4.10	0.00	4119.21	3332	1.87	0.00	323.42	622	0.00	0.00	3.67
40	6	0.5	8.56	2.10	time	41396	4.19	0.00	12949.04	4144	0.00	0.00	3.87
40	6	0.8	9.21	1.40	time	46600	4.95	0.89	time	16503	0.89	0.89	3.68
40	8	0.2	6.55	0.70	time	52491	4.04	0.00	16318.49	9536	0.00	0.00	5.45
40	8	0.5	12.42	8.70	time	42625	6.65	4.01	time	8992	4.01	4.01	5.32
40	8	0.8	14.82	8.00	time	43053	6.16	3.59	time	11814	3.59	3.59	4.83
50	4	0.2	1.15	0.00	233.55	190	0.19	0.00	36.91	47	0.00	0.00	5.64
50	4	0.5	2.57	0.00	6729.47	4332	0.73	0.00	919.38	162	0.00	0.00	5.9
50	4	0.8	5.08	1.80	time	12659	2.65	0.00	32122.6	1795	0.00	0.00	5.41
50	6	0.2	3.56	0.00	25218.14	7791	1.42	0.00	2160	771	0.01	0.00	8.75
50	6	0.5	7.99	6.60	time	13173	5.17	2.28	time	4931	2.36*	2.36	9
50	6	0.8	9.02	7.70	time	8609	5.39	3.12	time	4901	3.12	3.14	8.44
50	8	0.2	7.98	6.60	time	11128	4.1	1.58	time	4803	1.58	1.58	12.54
50	8	0.5	10.76	9.60	time	10109	7.31	5.56	time	3628	5.56	5.56	12.37
50	8	0.8	11.26	10.10	time	14139	7.3	5.89	time	3746	5.89	5.96	10.54

Table 2: Computational results for branch-and-cut and GRASP with AP instances

The results presented in Table 2 show that the branch-and-cut algorithm was able to optimally solve 34 out of the 42 considered instances within the time limit. For the 8 unsolved instances, the final gaps are in the range of 0.8% to 6%. On the other hand, we were able to solve the flow based formulation of 31 instances to optimality and the final gaps on the remaining instances range from 0.6% to 10.10%. Column *%LP-cuts* shows that the addition of constraints (20) have a significant impact in the improvement of the lower bound at the root node of the tree. The branch-and-cut algorithm is faster than CPLEX (FF formulation) on 28 out of 31 instances that were solved to optimality using both the algorithms. Moreover, our branch-and-cut algorithm is able to solve 3 instances that CPLEX is unable to solve within the time limit. For the instances that were not solved to optimality, the branch-and-cut always provides smaller final percent gaps than CPLEX.

The results in Table 2 also show that the GRASP algorithm is very effective in finding high quality solutions for the CHLP. In particular, it finds the optimal solution (or the best known solution) for 41 out of the 42 tested instances, requiring just a fraction of time that is taken by the branch-and-cut. Just in one instance ($n = 50, p = 6, \text{and } \alpha = 0.5$), the branch-and-cut algorithm was able to improve the best solution obtained with GRASP by 0.1%. The percent average deviations over 30 runs depict the robustness of the GRASP algorithm. On 37 instances, GRASP yield the same solution in each run whereas the average deviation for the other 5 instances range from 0.02% to 0.55%.

In order to further analyse the efficiency and robustness of our algorithms, we conducted a final series of computational experiments using instances ranging from 60 to 100 nodes. The results are summarized in Table 3. The columns have the same meaning as in the previous tables.

Instance			FF	Branch-and-Cut			GRASP		
$\|N\|$	P	α	% LP	%LP-cuts	% Gap	CPU	% Dev	% Avg Dev	CPU
60	4	0.2	1.69	1.04	0.0	242.22	0.00	0.00	8.52
60	4	0.5	2.99	1.22	0.0	805.66	0.00	0.00	10.06
60	4	0.8	5.41	3.09	0.0	11398.67	0.00	0.00	11.45
60	6	0.2	3.84	2.47	0.0	16420.4	0.00	0.02	14.06
60	6	0.5	7.54	5.31	3.60	time	3.60	3.72	19.45
70	4	0.2	1.57	0.67	0.0	810.5	0.00	0.00	13.54
70	4	0.5	3.33	1.82	0.0	4982.43	0.00	0.00	15.22
70	4	0.8	5.59	3.72	0.0	66799.58	0.11*	0.22	18.37
70	6	0.2	3.88	2.21	0.0	14662.31	0.00	0.00	24.58
70	6	0.5	7.83	5.91	4.64	time	4.64	4.64	27.86
75	4	0.2	1.52	1.12	0.0	1330.54	0.00	0.00	15.83
75	4	0.5	3.40	1.99	0.0	6201.98	0.00	0.00	20.97
75	4	0.8	5.64	3.83	0.65	time	0.70*	0.70	22.83
75	6	0.2	3.94	2.66	0.22	time	0.22	0.22	28.93
75	6	0.5	7.61	5.88	4.54	time	4.54	4.54	34.91
90	4	0.2	1.45	1.37	0.0	7512.22	0.00	0.00	30.41
90	4	0.5	3.14	2.09	0.0	53820.77	0.00	0.00	34.74
90	4	0.8	5.40	4.21	3.18	time	3.18	3.30	38.60
90	6	0.2	3.91	3.35	2.61	time	2.61	2.63	47.74
90	6	0.5	7.63	6.34	5.85	time	5.85	5.85	59.61
100	4	0.2	1.50	1.41	0.00	14112.44	0.00	0.00	39.55
100	4	0.5	3.24	2.35	1.05	time	1.05	1.05	48.25
100	4	0.8	5.52	4.48	3.96	time	3.96	3.96	49.59
100	6	0.2	4.12	3.76	3.17	time	3.17	3.17	64.33

Table 3: Computational results for branch-and-cut and GRASP with large-scale instances of AP dataset

The results in Table 3 assess the efficiency of our proposed approaches for solving large-scale instances. Given the dimension and complexity of the considered instances, the exact algorithm is able to solve 13 out the 24 instances to optimality and for the remaining instances, the final % gap never exceeds 5.85%. It is worth mentioning that CPLEX is not able to solve any of the considered instances within the time limit. In the case of the GRASP algorithm, it is able to obtain the optimal solution in 12 out of the 13 instances that were solved to optimality. For the remaining one instance, the branch-and-cut was able to improve the best GRASP solution by 0.05%.

5. Conclusions

In this paper, we have introduced and studied the cycle hub location problem. Potential applications of this model appears in telecommunication and transportation systems, where large set-up costs on the links as well as the reliability requirements make cycle topologies a prominent network structure. We have presented and compared the bath based and flow based formulations for the problem. An exact algorithm based on a branch-and-cut approach was proposed to solve the problem to optimality. This algorithm uses a family of extended mixed dicut inequalities to improve the lower bound at some nodes of the enumeration tree. A GRASP metaheuristic was also presented to efficiently obtain high quality solutions. Computational results on benchmark instances involving up to 100 nodes confirm the efficiency and robustness of the proposed approaches.

6. Acknowledgements

This research was partly founded by the Canadian Natural Sciences 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, 1–21.

Alumur, S.A., Kara, B.Y., Karasan, O.E., 2009. The design of single allocation incomplete hub networks. Transportation Research Part B: Methodological 43, 936–951.

Baldacci, R., Dell'Amico, M., González, J.S., 2007. The capacitated m-ring-star problem. Operations Research 55, 1147–1162.

Boland, N., Krishnamoorthy, M., Ernst, A.T., Ebery, J., 2004. Preprocessing and cutting for multiple allocation hub location problems. European Journal of Operational Research 155, 638–653.

Calık, H., Alumur, S.A., Kara, B.Y., Karasan, O.E., 2009. A tabu-search based heuristic for the hub covering problem over incomplete hub networks. Computers & Operations Research 36, 3088–3096.

Campbell, J., Ernst, A., Krishnamoorthy, M., 2005a. Hub arc location problems: Part ii: Formulations and optimal algorithms. Management Science 51, 1556–1571.

Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72, 387–405.

Campbell, J.F., 2009. Hub location for time definite transportation. Computers & Operations Research 36, 3107–3116.

Campbell, J.F., Ernst, A., Krishnamoorthy, M., 2005b. Hub arc location problems: part i: Introduction and results. Management Science 51, 1540–1555.

Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2002. Hub location problems. Facility location: applications and theory 1, 373–407.

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

Contreras, I., Cordeau, J.F., Laporte, G., 2011a. Benders decomposition for large-scale uncapacitated hub location. Operations research 59, 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, 41–55.

Contreras, I., Fernández, E., 2012a. General network design: A unified view of combined location and network design problems. European Journal of Operational Research 219, 680–697.

Contreras, I., Fernández, E., 2012b. hub location as the minimization of a supermodular set function. submitted to Operations Research .

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, 3117–3127.

Contreras, I., Fernández, E., Marín, A., 2010. The tree of hubs location problem. European Journal of Operational Research 202, 390–400.

Cook, W., Cunningham, W., Pulleybank, W., Schrijver, A., 1998. Combinatorial Optimization. Wiley-Interscience.

Current, J.R., Schilling, D.A., 1994. The median tour and maximal covering tour problems: Formulations and heuristics. European Journal of Operational Research 73, 114–126.

Davari, S., Fazel Zarandi, M., Turksen, I., 2013. The incomplete hub-covering location problem considering imprecise location of demands. Scientia Iranica .

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, 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, 149–162.

Ernst, A.T., Krishnamoorthy, M., 1999. Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86, 141–159.

Festa, P., Resende, M., 2011. Grasp: basic components and enhancements. Telecommunication Systems 46, 253–271.

Gendreau, M., Laporte, G., Semet, F., 1997. The covering tour problem. Operations Research 45, 568–576.

Hamacher, H.W., Labbé, M., Nickel, S., Sonneborn, T., 2004. Adapting polyhedral properties from facility to hub location problems. Discrete Applied Mathematics 145, 104–116.

Kim, J.G., Tcha, D.w., 1992. Optimal design of a two-level hierarchical network with tree-star configuration. Computers & industrial engineering 22, 273–281.

Klincewicz, J., 1998. Hub location in backbone/tributary network design: a review. Location Science 6, 307–335.

Labbe, M., Laporte, G., Martín, I.R., Gonzalez, J.J.S., 2004. The ring star problem: Polyhedral analysis and exact algorithm. Networks 43, 177–189.

Labbé, M., Laporte, G., Rodríguez-Martín, I., 1998. Path, tree and cycle location. Springer.

Labbé, M., Laporte, G., Rodrıguez Martın, I., González, J.J.S., 2005a. Locating median cycles in networks. European Journal of Operational Research 160, 457–470.

Labbé, M., Yaman, H., 2004. Projecting the flow variables for hub location problems. Networks 44, 84–93.

Labbé, M., Yaman, H., 2008. Solving the hub location problem in a star–star network. Networks 51, 19–33.

Labbé, M., Yaman, H., Gourdin, E., 2005b. A branch and cut algorithm for hub location problems with single assignment. Mathematical programming 102, 371–405.

Laporte, G., Martín, I.R., 2007. Locating a cycle in a transportation or a telecommunications network. Networks 50, 92–108.

Lee, Y., Chiu, S., Sanchez, J., 1998. A branch and cut algorithm for the steiner ring star problem. International Journal of Management Science 4, 21–34.

Lee, Y., Lim, B.H., Park, J.S., 1996. A hub location problem in designing digital data service networks: Lagrangian relaxation approach. Location Science 4, 185–194.

Lee, Y., Lu, L., Qiu, Y., Glover, F., 1993. Strong formulations and cutting planes for designing digital data service networks. Telecommunication Systems 2, 261–274.

Liefooghe, A., Jourdan, L., Talbi, E.G., 2010. Metaheuristics and cooperative approaches for the bi-objective ring star problem. Computers & Operations Research 37, 1033–1044.

Marín, A., 2005. Uncapacitated euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. Journal of Global Optimization 33, 393–422.

O'Kelly, M.E., 1986. The location of interacting hub facilities. Transportation Science 20, 92–106.

O'Kelly, M.E., Miller, H.J., 1994. The hub network design problem: a review and synthesis. Journal of Transport Geography 2, 31–40.

Ortega, F., Wolsey, L.A., 2003. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks 41, 143–158.

de Sá, E.M., de Camargo, R.S., et al., 2012. An improved benders decomposition algorithm for the tree of hubs location problem. European Journal of Operational Research .

de Sá, E.M., Contreras, I., Cordeau, J.F., de Camargo, R.S., de Miranda, G., 2013. The hub line location problem. Submitted to Transportation Science .

Xu, J., Chiu, S.Y., Glover, F., 1999. Optimizing a ring-based private line telecommunication network using tabu search. Management Science 45, 330–345.

Yaman, H., 2008. Star p-hub median problem with modular arc capacities. Computers & Operations Research 35, 3009–3019.

Yaman, H., 2009. The hierarchical hub median problem with single assignment. Transportation Research Part B: Methodological 43, 643–658.

Yaman, H., Elloumi, S., 2012. Star p-hub center problem and star p-hub median problem with bounded path lengths. Computers & Operations Research 39, 2725–2732.