European Journal of Operational Research 202 (2010) 390–400

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

# The Tree of Hubs Location Problem

Ivan Contreras [a,*], Elena Fernández [a], Alfredo Marín [b]

[a] Dpt. d' Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Barcelona, Spain
[b] Departamento de Estadística e Investigación Operativa, Universidad de Murcia, Murcia, Spain

ARTICLE INFO

*Article history:*
Received 17 October 2007
Accepted 30 May 2009
Available online 17 June 2009

*Keywords:*
Hub location
Spanning trees
Valid inequalities

ABSTRACT

This paper presents the Tree of Hubs Location Problem. It is a network hub location problem with single assignment where a fixed number of hubs have to be located, with the particularity that it is required that the hubs are connected by means of a tree. The problem combines several aspects of location, network design and routing problems. Potential applications appear in telecommunications and transportation systems, when set-up costs for links between hubs are so high that full interconnection between hub nodes is prohibitive. We propose an integer programming formulation for the problem. Furthermore, we present some families of valid inequalities that reinforce the formulation and we give an exact separation procedure for them. Finally, we present computational results using the well-known AP and CAB data sets.



## 1. Introduction

Hub networks have their origin in transportation and telecommunication systems where several origin/destination points send and receive some product. The key feature of these networks is to route products via a specific subset of links, rather than routing each product with a direct link from its origin to its destination point. In particular, hub networks use a set of hub nodes to consolidate and reroute the flows, and a reduced number of links, where economies of scale are applied, to connect the (usually large) set of origins/destination points. Broadly speaking, Hub Location Problems (HLP) consider the location of a set of hub nodes and the design of the hub network.

Two basic HLP models appeared in the literature more than a decade ago: the multiple allocation and the single allocation uncapacitated hub location problems. The first formulation for the multiple allocation case was given by Campbell (1994). Later, Klincewicz (1996), Skorin-Kapov et al. (1996), Ernst and Krishnamoorthy (1998a,b) and Mayer and Wagner (2002) studied several improvements, while Hamacher et al. (2004) and Cánovas et al. (2006) began the polyhedral study of the problem. Recent advances have been presented in Marín (2005b) and Cánovas et al. (2007). The single allocation case has been studied by O'Kelly (1987), Klincewicz (1991), Skorin-Kapov et al. (1996), Aykin (1995) and Ernst and Krishnamoorthy (1998a). The capacitated multiple allocation case was studied by Aykin (1994), Ebery et al. (2000), Campbell (1994), Boland et al. (2004), and by Marín (2005a). The capacitated single allocation problem has also been studied by Ernst and Krishnamoorthy (1999), Labbé et al. (2005), Contreras et al. (2008, 2009), and by Contreras (2009). The interested reader is referred to the comprehensive surveys on the matter Campbell et al. (2002) and Alumur and Kara (2008).

Traditionally, most hub location models assume that hubs are fully interconnected, that is to say, that there exists a link connecting any pair of hubs. However, there exist many applications in which the backbone network (i.e. the network connecting the facilities) is not fully interconnected (see, for instance O'Kelly and Miller, 1994; Klincewicz, 1998). The so-called hub-arc location models Campbell et al. (2005a,b) impose no condition on the structure of the arcs that connect hubs, and they do not even require these arcs to define one single connected component. Moreover, some hub location models have been proposed recently where two hubs are not necessarily connected by some link (see, for instance Nickel et al., 2001; Gelareh, 2008).

In this paper we propose the Tree of Hubs Location Problem (THLP). The THLP is a single allocation hub location problem where $p$ hubs have to be located on a network and connected by means of a (non-directed) tree. Then each non-hub node must be connected (allocated) to a hub and all the flow between nodes must use these connections to circulate, i.e., excepting the arcs that connect each non-hub node with its allocated hub, the arcs that route the flows must be links connecting hubs. There is a per unit transportation cost associated with each arc. The objective is to minimize the operation costs of the system.

* Corresponding author.
*E-mail addresses:* ivan.contreras@upc.edu (I. Contreras), e.fernandez@upc.edu (E. Fernández), amarin@um.es (A. Marín).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.05.044

As already stated in Klincewicz (1998) and Campbell et al. (2002) there are some problems in the literature, under slightly different names, dealing with the location of a tree on a network in the context of hub location. In the case of hierarchical networks, Kim and Tcha (1992) presented a problem with a tree-star topological configuration. In the case of digital data service networks, Lee et al. (1994, 1996) consider the problem of locating a set of hubs and connecting them by means of a tree. All these problems have the fixed installation costs as their dominant cost components, while the THLP has the routing or flow cost as its dominant cost component.

In addition to the above problems and to other hub location problems, our problem is also related to other types of problems studied in the literature where location, design or routing decisions are considered. To mention just a few: (a) the location of tree-shaped facilities (also called extensive facilities) on a network, which has been studied within location theory (Hakimi et al., 1993; Kim et al., 1996; Puerto and Tamir, 2005); (b) network design problems that study the arcs to be used in the optimal routing of flows within pairs of nodes on a network, like the Optimum Communication Spanning Tree Problem (Hu, 1974) or the Minimum Sum Violation Tree Problem (Chen et al., 2008); (c) rapid transit network design problems that locate lines in rail transit systems (Marín, 2007); and (d) some location-routing problems, like the ring-star problem (Labbé et al., 2004; Laporte and Rodríguez-Martín, 2007).

Potential applications of models where facilities are connected by means of a tree arise when the cost of the links between facilities is very high, and as a consequence full interconnection is prohibitive. For sending all the flows in the network a path must exist between each pair of facilities, i.e., a connected graph must be built. But due to the high cost of the connections, connectivity must be achieved using the minimum number of links. Specific applications of such problems arise mostly in telecommunications (see Hu, 1974; Nguyen and Knippel, 2007) and in transportation (see, for instance, the recent work of Chen et al. (2008) for an excellent description of the practical relevance of tree-backbone problems in small package delivery). One concrete example of an application of the THLP is the design of the high-speed train network in Spain, which is currently under construction and which is planned to be completed by 2020. This train network has been designed with a tree structure and it is intended that, when finished, each city with more than 10,000 inhabitants will be within 50 km of some high-speed train station. Another application is the design of rapid transit systems for urban areas where there are flows of users (citizens) who must travel between origin/destination pairs. Depending on the criterion used for establishing the allocation of users, these applications would fit in the THLP model.

In this paper, we propose a mixed integer programming formulation for the THLP. This formulation uses two sets of two indices binary variables to model the location/allocation and the design decisions, respectively, and one set of three indices continuous variables to model the flows which are routed through the tree of hubs. Moreover, we present several families of valid inequalities to strengthen the formulation of the proposed model (and, thus, its LP bound) together with their exact separation procedures. We have performed a series of computational experiments in order to assess the quality of the formulation and the strength of the proposed valid inequalities. The obtained numerical results confirm the efficiency of the proposed inequalities since, in all tested instances ranging from 10 to 25 nodes, they are able to considerably reduce both the duality gap and the required computational time to optimally solve them.

The paper is organized as follows. First, we formally define the problem. In Section 3, we present a mathematical programming formulation for the THLP. Later, in Section 4, we derive some families of valid inequalities and separation procedures to strengthen the formulation. The computational results obtained are given in Section 5. The paper finishes with a section of conclusions and some possible future lines of research.

## 2. Description of the problem

Consider a complete digraph without loops $G = (N,A)$ whose set of nodes, $N = \{1,\dots,n\}$, represents the set of origins and destinations of a certain product that is routed through $G$ via some transshipment nodes that are called *hubs*. For each pair of nodes $i, j \in N$, let $W_{ij}$ denote the demand of product from $i$ to $j$. When some of the end-nodes of the arc $(r,s) \in A$ is not a hub, each unit of product that traverses $(r,s)$ incurs a cost $c_{rs} \geqslant 0$, whereas when both $r$ and $s$ are hubs a discount factor $0 \leqslant \alpha \leqslant 1$ is applied, and the per unit cost associated with arc $(r,s)$ is $\alpha c_{rs}$.

Any node of $N$ can be chosen to become a hub, and there is a fixed number $p$ of nodes of $N$, $3 \leqslant p \leqslant n-1$, that must be chosen to be hubs. For each pair $i, j \in N$, if neither $i$ nor $j$ are hubs the flow $W_{ij}$ must go from $i$ to $j$ through one or more intermediate hubs. When $i$ or $j$ are hubs $W_{ij}$ can go directly from $i$ to $j$, although it is also possible that $W_{ij}$ uses one or more intermediate hubs. Non-hub nodes cannot be used to transship the product. We also require *single allocation* and, thus, every non-hub node $i$ must be assigned (allocated) to one single hub, so that all the flow from/to $i$ to/from any other node must pass through this hub.

We require the hubs to be connected by means of a (non-directed) tree. Observe that since the links that connect hubs are required to define a tree (that we denote *small tree*), the structure that results after adding to the small tree the links connecting non-hub nodes to their allocated hub nodes defines a tree in $G$ (that we denote *large tree*). Thus, in the large tree the path between each pair $i$ and $j$ will be unique. In this path all the transshipment nodes will be hubs so all the arcs, excepting (possibly) the first and last ones, link hubs.

The cost per unit of flow of the path between a pair $i$ and $j$ is the sum of the costs of the arcs in the path, where the discount factor is applied to all the inter-hubs links. Therefore, if $i$ and $j$ are both hubs, $W_{ij}$ could be sent from $i$ to $j$ (directly, if there is such an arc in the small tree, or through additional intermediate hubs, otherwise) and the discount factor will be applied to all the arcs in the route. If either $i$ or $j$ are hubs, the product could be sent directly (if such an arc is part of the large tree) without discount, or through some other hubs, otherwise; if $i$ (resp. $j$) was the hub, the discount will be applied to all the arcs except the last (resp. first) one. If neither $i$ nor $j$ are hubs, $W_{ij}$ will again be sent from $i$ to $j$ through at least one intermediate hub. Now, $W_{ij}$ must go from $i$ to some hub, then possibly to some other hub(s), and finally to $j$, and the discount factor will be applied to all the intermediate arcs, but will not be applied neither to the first nor to the last arcs. Note that there can be a flow of product $W_{ii}$ with origin and destination at the same node $i$. We assume free self service, i.e., $c_{ii} = 0$ $\forall i \in N$. Thus, if $i$ is a hub, $W_{ii}$ can stay in $i$ at null cost, whereas if $i$ is not a hub, $W_{ii}$ must go to some hub before returning to $i$. Note also that triangle inequality of costs $c_{ij}$ is not assumed. Some formulations in the literature (see Cánovas et al., 2006) are only valid if this property holds, but our formulation is also valid without this assumption. We do not assume symmetry either, i.e., costs $c_{ij}$ and $c_{ji}$ can be different.

We define the *graph of flows*, $G_F$, as the undirected graph with vertex set $N$ and an edge associated with each pair $(i,j) \in N \times N$ such that $W_{ij} + W_{ji} > 0$. We assume that $G_F$ has one single connected component since, otherwise, it would be more natural to define as many hub trees as connected components in $G_F$. However, if for some strategic decision $G_F$ would not be connected but we were required to define just one tree of hubs, then we would replace demands $W_{ij}$ equal to zero by $W_{ij} = \epsilon$.

We will assume throughout that $p \geqslant 3$. Note that in the case of $p \in \{1,2\}$ the problem reduces to a classical hub location problem. Moreover, for the sake of simplicity, hereafter the total flow with origin and destination in node $i$ is defined as

$$O_i = \sum_{j \in N} W_{ij} \quad \forall i \in N$$

and

$$D_i = \sum_{j \in N} W_{ji} \quad \forall i \in N,$$

respectively.

We want: (i) to locate $p$ hubs (at null or equal costs); (ii) to link them (at null or equal costs) so as to define a tree – the *small tree*; and (iii) to allocate every non-hub node to one single hub – obtaining the so-called *large tree* – in such a way that the overall cost is minimized. The cost is the sum of the routing costs associated with the flows between all the elements in $N \times N$. Given $(i,j) \in N \times N$, the routing cost of the flow $W_{ij}$ is given by that of the unique path from $i$ to $j$ in the large tree which traverses at least one hub times the amount of flow $W_{ij}$.

**Proposition 1.** *The THLP is $\mathcal{NP}$-hard.*

**Proof.** When both the set of hubs to be located and the allocation pattern of the rest of the nodes is given, the subproblem of connecting them so as to define a tree, and of routing the flows between the nodes at minimum total cost is the so-called Optimum Communication Spanning Tree Problem (Hu, 1974), which is known to be $\mathcal{NP}$-hard (Johnson et al., 1987), even if there is a unit flow between each pair of nodes ($W_{ij} = 1, \forall (i,j) \in N \times N$). Hence, the result follows. □

The above result is important not only because it establishes the complexity of the problem but also because it highlights one of the main differences between classical hub location problems and the THLP. In the case of classic models, once we know the location and allocation pattern of nodes to hub facilities the resulting problem of routing the flow through the network is trivial. (Since they assume that the backbone network is complete and can be easily determined.) In the case of the THLP, routing the flow through the network is still $\mathcal{NP}$-hard. This means that the THLP is much more difficult to solve than the classical models that assume a complete backbone network.

## 3. Integer programming formulation

The aim of the problem we are dealing with is the minimization of the total cost of sending the flow between each pair of nodes. The product sent from an origin to a destination flows through a route with an undetermined number of arcs, whose costs depend on whether or not the end-nodes of the arcs are both hubs. Therefore, in order to evaluate the objective function, the formulation must (i) represent (or be able to reproduce) the complete routes – note here that the routes have an orientation – and (ii) know if the discount must be applied to each arc of a route or not.

In order to represent the routes between origins and destinations, we will consider variables with three indices, proposed by Ernst and Krishnamoorthy (1998b) for the uncapacitated multiple allocation hub location problem. Nowadays, this type of variables are commonly used in the hub location literature (see, for instance Ernst and Krishnamoorthy, 1999; Boland et al., 2004; Marín, 2005a). These variables are defined as continuous ones, representing the amount of flow with the origin given which traverses a given arc. That is, to model the flow coming from every node $i \in N$ that traverses arc $(k,m) \in A$ where $m \neq k$, we define the following set of continuous variables:

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

Integer Programming formulations for several kinds of spanning tree problems have been studied, for example, in Gouveia (1995), Pop et al. (2006) and references therein. In our work we have followed the strategy of building a tree and imposing all the routes to use it. From our connectivity assumption on the graph of flows, we know *a priori* that the graph of routes for the flows between nodes will also be connected. Hence, to represent the trees that must be determined, we will simply need to constrain the number of edges of the graph of links between nodes. Thus, for every $k, m \in N$; $m > k$ we define the following binary variable:

$$y_{km} = \begin{cases} 1 & \text{if arc } (k,m) \text{ links two hubs,} \\ 0 & \text{otherwise.} \end{cases}$$

To model the location of hubs as well as the allocations of non-hubs, for every $i, k \in N$; $i \neq k$ we define the following $z$-variables:

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

When $i = k$, variable $z_{kk}$ represents whether or not a hub is located at node $k$. The $z$ variables will be referred to as location/allocation variables.

Using these sets of variables we can formulate the THLP as follows,

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

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

$$\sum_{k \in N} z_{kk} = p, \tag{2}$$

$$z_{km} + y_{km} \leqslant z_{mm} \quad \forall k,\ m \in N;\ m > k, \tag{3}$$

$$z_{mk} + y_{km} \leqslant z_{kk} \quad \forall k,\ m \in N;\ m > k, \tag{4}$$

$$x_{ikm} + x_{imk} \leqslant O_i y_{km} \quad \forall i,\ k,\ m \in N;\ m > k, \tag{5}$$

$$O_i z_{ik} + \sum_{\substack{m \in N \\ m \neq k}} x_{imk} = \sum_{\substack{m \in N \\ m \neq k}} x_{ikm} + \sum_{m \in N} W_{im} z_{mk} \quad \forall i,\ k \in N;\ k \neq i, \tag{6}$$

$$\sum_{k \in N} \sum_{m \in N} y_{km} = p - 1, \tag{7}$$

$$x_{ikm} \geqslant 0 \quad \forall i,\ k,\ m \in N,$$

$$z_{km},\ y_{km} \in \{0, 1\} \quad \forall k,\ m \in N.$$

Constraints (1)–(4) imply the usual ones in the formulation of the $p$-median problem: they guarantee that there are exactly $p$ hubs and that non-hubs are allocated to open hubs. In the context of THLP, the usual constraints $z_{km} \leqslant z_{mm}$, can be reinforced to (3) when $m > k$. (When $y_{km} = 1$, then a hub is located at node $k$, so that $z_{kk} = 1$. Thus – by constraint (1) – $z_{km} = 0$.) Similarly, when $m < k$ the usual constraints can be reinforced to (4). Constraints (5) determine that the flow between hubs will only move through the small tree ($y$-variables). Each constraint in family (6) guarantees the conservation of the flow with origin $i$ in node $k$. On the left-hand-side we add up the flow incoming node $k$ directly from node $i$ (if $i$ is not a hub and it is allocated to $k$), plus the flow with origin in $i$ incoming node $k$ from another hub $m$. On the right-hand-side, we add up the flow with origin in $i$ going from hub $k$ to another hub (if $k$ is a hub), plus the flow with origin in $i$ and destination in any non-hub node allocated to hub $i$ (again if $k$ is a hub). Finally, through constraints (7) the variables $y$ define a tree, since $p - 1$ edges are chosen which are connected due to constraints (5) and to the flow conservation constraints (6).

## 4. Coefficient reduction and valid inequalities

In this section we improve the constraints used in formulation THP and present new inequalities which can be added to THP in order to get better bounds and less computational times when solving (TFLP).

### 4.1. Coefficient reduction

It is well-known that, in general, mixed integer programming formulations based on aggregated flows tend to suffer from weak LP bounds. One of the main reasons is that the right-hand-side coefficient of constraints (5) usually overestimates the amount of flow that actually passes through a given arc $(k,m)$. Thus, the first step for improving the LP bound of the proposed formulation THP is to tighten as much as possible this coefficient by taking into account the structure of the problem. To do so, we define the following values,

$$m_{ikm} = \min\{W_{ik}, W_{im}\} \quad \forall i,\ k,\ m \in N.$$

Using these values we can tighten constraints (5) in the following way,

$$x_{ikm} + x_{imk} \leqslant (O_i - W_{ii} - m_{ikm}) y_{km} \quad \forall i,\ k,\ m \in N;\ m > k,\ i \neq k,\ m, \tag{8}$$

$$x_{iim} \leqslant (O_i - W_{ii}) y_{im} \quad \forall i,\ m \in N;\ m > i, \tag{9}$$

$$x_{iim} \leqslant (O_i - W_{ii}) y_{mi} \quad \forall i,\ m \in N;\ m < i. \tag{10}$$

When $i \neq k, m$, the maximum amount of flow with origin in $i$ traversing edge $\{k,m\}$ will be $O_i$ minus the flow $W_{ii}$ – which visits only one hub – minus the flow with destination in the first hub of the edge. Since we do not know whether such a hub is $k$ or $m$, we use the value $m_{ikm}$. The cases where $i = k$ of $i = m$ are written separately, since the bound is different. Moreover, in some very special cases, the $x$-variables can be fixed to zero:

$$x_{iki} = 0 \quad \forall i,\ k \in N;\ k \neq i. \tag{11}$$

### 4.2. Cut and mixed-cut inequalities

These inequalities we will present in the next subsection are based on the *cut* and *mixed-cut* inequalities for the uncapacitated fixed-charge network flow problem (Ortega and Wolsey, 2003). For that particular problem, they are known to be facet-defining inequalities. We first adapt the cut and mixed-cut inequalities for the THLP. Then, in the next subsection, we present a generalization of the latter, which explicitly takes into account the structure of the THLP. Let $X$ be the set of feasible solutions of THP.

**Proposition 2.** *For $m \in N$, the cut inequality*

$$\sum_{k<m} y_{km} + \sum_{k>m} y_{mk} \geqslant z_{mm} \quad \forall m \in N \tag{12}$$

*is valid for X.*

**Proof.** If $z_{mm} = 1$, inequality (12) holds trivially. If $z_{mm} = 1$, then at least one arc incident to node $m$ must be part of the small tree and the inequality is satisfied. So, the inequality (12) is valid. □

**Proposition 3.** *For $i, m \in N$ and $F \subseteq N \backslash \{m\}$, the mixed-cut inequality*

$$\sum_{k \in N \backslash F} x_{ikm} + W_{im} \left( \sum_{\substack{k \in F \\ k<m}} y_{km} + \sum_{\substack{k \in F \\ k>m}} y_{mk} \right) \geqslant W_{im}(z_{mm} - z_{im}) \tag{13}$$

*is valid for X.*

**Proof.** Note that the right-hand-side of inequality (13) can only be non-negative when $m$ is a hub and $i$ is not assigned to $m$. Otherwise, if $m$ were not a hub, $z_{mm} = z_{im} = 0$, so that the right-hand-side would be zero. Also, if $m$ were a hub and $i$ were assigned to $m$, $z_{mm} = z_{im} = 1$ so that the term $z_{mm} - z_{im} = 0$.

When $m$ is a hub and $i$ is not assigned to $m$ the right-hand-side of inequality (13), $W_{im}(z_{mm} - z_{im})$, represents the flow with origin in $i$ and destination in $m$. Thus, it is clear that this right-hand-side is a lower bound on the total flow that arrives to $m$ from $i$. Note that, in addition to the flow $W_{im}$, this total flow may have its destination in (i) a non-hub node which is assigned to $m$; (ii) a hub different from $m$; or, (iii) a non-hub node assigned to a hub different from $m$.

To analyze the left-hand-side, we take into account that in any feasible solution, any amount of flow arriving to $m$ from $i$ will reach hub $m$ through one intermediate hub $\hat{k}$ (possibly node $i$). Therefore, in the left-hand-side of the inequality, only one of the two terms will be positive, depending on whether or not $\hat{k} \in F$. If $\hat{k} \notin F$, the positive term is $\sum_{k \in N \backslash (F \cup \{m\})} x_{ikm}$, that represents the total flow leaving $i$ arriving to hub $m$ through hub $\hat{k}$, and which is an upper bound on the value of the left-hand-side of the inequality. On the other hand, if $\hat{k} \in F$ the positive terms is $W_{im}(\sum_{k \in F: k<m} y_{km} + \sum_{k \in F: k<m} y_{mk})$, that is the total flow from $i$ to hub $m$. Now the value of this positive term coincides with the value of the right-hand-side of the inequality when $m$ is a hub and $i$ is not assigned to $m$. We have, thus, proved the validity of the inequality. □

As we next see, the above inequalities can be extended, to take into account non-hub nodes assigned to hub node $m$.

### 4.3. New valid inequalities

Next we derive a family of valid inequalities which can be used to tighten formulation THP, based in the inequalites stated in the previous subsection.

**Proposition 4.** *For $i, m \in N$, $F \subseteq N \backslash \{m\}$, $J \subseteq N \backslash \{i, m\}$, the inequality*

$$\sum_{k \in N \backslash (F \cup \{m\})} x_{ikm} + \left( \sum_{j \in J \cup \{m\}} W_{ij} \right) \left( \sum_{\substack{k \in F \\ k<m}} y_{km} + \sum_{\substack{k \in F \\ k>m}} y_{mk} \right) \geqslant \sum_{j \in J \cup \{m\}} W_{ij}(z_{jm} - z_{im}) \tag{14}$$

*is valid for X.*

**Proof.** The proof follows the same idea as that of Proposition 3. Again, the right-hand-side of inequality (14) can only be non-negative when $m$ is a hub and $i$ is not assigned to $m$ since, otherwise the right-hand-side would be smaller than or equal to zero.

Now, when $m$ is a hub and $i$ is not assigned to $m$ the right-hand-side of inequality (14), $\sum_{j \in J \cup \{m\}} W_{ij}(z_{jm} - z_{im})$, represents the total flow with origin in $i$ and destination in $m$ or in some node indexed in $J$ and assigned to hub $m$ that, again, is a lower bound of the total flow that arrives to $m$ from $i$.

By the same argument as in Proposition 3, only one of the two terms of the left-hand-side will be positive, depending on whether or not the intermediate node $\hat{k} \in F$. As before, when $\hat{k} \notin F$, the positive term is an upper bound on the value of the left-hand-side of the inequality, since it represents the total flow leaving $i$ arriving to hub $m$ through hub $\hat{k}$. Now, when $\hat{k} \in F$ the positive term is $\left(\sum_{j \in J \cup \{m\}} W_{ij}\right)(\sum_{k \in F: k<m} y_{km} + \sum_{k \in F: k<m} y_{mk})$ that is the total flow from $i$ to either hub $m$ or a non-hub node indexed in $J$ and assigned to $m$. In this generalization again the value of this positive term coincides with the value of the right-hand-side of the inequality when $m$ is a hub and $i$ is not assigned to $m$. □

### 4.4. Separation problem of inequalities (14)

In this section we solve the so-called separation problem: given a fractional solution $(\bar{x}, \bar{y}, \bar{z})$ to the linear relaxation of THP, find one or more inequalities in family (14) which are not satisfied by $(\bar{x}, \bar{y}, \bar{z})$.

To this end, two sets have to be identified. On the one hand the set $J \subseteq N \backslash \{i, m\}$ that affects both the left- and right-hand-sides of the inequality and determines the coefficient $Q = \sum_{j \in J \cup \{m\}} W_{ij}$. On the other hand, the set $F \subseteq N \backslash \{m\}$, that only affects the left-hand-side of the inequality which, in turn, depends on the value of the coefficient $Q$. The following proposition indicates how to determine a set $F$ that minimizes the left-hand-side of the inequality, for a given value of a coefficient $Q$. Then, we will see how to identify the set $J$ that gives the best trade-off between the values of the left- and right-hand-sides of the inequality, in terms of finding a violated inequality. For a given coefficient $Q$, let

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

and

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

Observe that if $k \in N$ is such that $\bar{y}_{mk} = 0$, constraints (5) ensure that also $\bar{x}_{ikm} = 0$. Therefore, any such index can be arbitrarily assigned to either $F_<$ or $F_>$ without affecting the value of the left-hand-side of the corresponding inequality (14).

**Proposition 5.** *Let $i, m \in N$, $Q \geqslant 0$, and $(\bar{x}, \bar{y}, \bar{z})$ be given. Then, a set $\overline{F} \subseteq N \setminus \{m\}$ that minimizes the value of*

$$L(Q) = \min_{F \subseteq N \setminus \{m\}} \sum_{k \in N \setminus (F \cup \{m\})} \bar{x}_{ikm} + Q\left(\sum_{k \in F: k<m} \bar{y}_{km} + \sum_{k \in F: k>m} \bar{y}_{mk}\right), \tag{15}$$

*is given by $\overline{F} = F_< \cup F_>$.*

**Proof.** A set $\overline{F} \subseteq N \setminus \{m\}$ that minimizes the value of $L(Q)$ can be obtained by solving the optimization problem:

$$\min \sum_{k \in N \setminus \{m\}} \bar{x}_{ikm}(1-\delta_k) + Q\left(\sum_{k<m} \bar{y}_{km}\delta_k + \sum_{k>m} \bar{y}_{mk}\delta_k\right), \quad \delta_k \in \{0,1\},\ \forall k \in N \setminus \{m\}. \tag{16}$$

Therefore, the result follows, since the objective function (16) can be rewritten as

$$\sum_{k \in N \setminus \{m\}} \bar{x}_{ikm} + \min\left\{\sum_{k<m}(-\bar{x}_{ikm} + Q\bar{y}_{km})\delta_k + \sum_{k>m}(-\bar{x}_{ikm} + Q\bar{y}_{mk})\delta_k\right\}. \qquad \square$$

As a consequence of the above result, for a given value of $Q$, we have that

$$L(Q) = \sum_{k \notin F_< \cup F_> \cup \{m\}} \bar{x}_{ikm} + Q\left(\sum_{k \in F_<} \bar{y}_{km} + \sum_{k \in F_>} \bar{y}_{mk}\right).$$

Therefore, the function $L : \mathrm{IR}_+ \to \mathrm{IR}$ defined as in (15), is piecewise linear. Hence, by evaluating for each $k \in N$ the "break" value $\bar{x}_{ikm}/\bar{y}_{mk}$, we can easily identify a series of intervals $I_t = [A_t, B_t]$, $t = 1, \ldots, r-1$, and $I_r = [A_r, \infty)$ with $A_1 = 0$, and $A_t = B_{t-1}$, $t = 2, \ldots, r$, and a series of sets $F^t = F^t_< \cup F^t_>$ such that for all $Q \in I_t$, $L(Q) = a^t + Qb^t$, with

$$a^t = \sum_{k \notin F^t_< \cup F^t_>} \bar{x}_{ikm}, \tag{17}$$

$$b^t = \sum_{k \in F^t_<} \bar{y}_{km} + \sum_{k \in F^t_>} \bar{y}_{mk}. \tag{18}$$

Now, for solving the separation problem for inequality (14) with given $i, m \in N$, we observe that the candidate values for the coefficient $Q$ are of the form $Q = \sum_{j \in N} \beta_j W_{ij}$, with $\beta_j \in \{0,1\}$, $j \in N$ and $\beta_m = 1$. Therefore, we have the following result.

**Proposition 6.** *Let $i, m \in N$, and $(\bar{x}, \bar{y}, \bar{z})$ be given. There exists an inequality (14) violated by $(\bar{x}, \bar{y}, \bar{z})$ if and only if $v < 0$, where*

$$v = \min L\left(\sum_{j \in N} W_{ij}\beta_j\right) - \sum_{j \in N} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})\beta_j, \tag{19}$$

$$\beta_m = 1, \tag{20}$$

$$\beta_j \in \{0,1\}, \quad j \in N. \tag{21}$$

Since the function $L(\cdot)$ is piecewise linear, the linear expression that gives the value of the first term of the objective function (19) depends on the interval $I_t$ where the value $\sum_{j \in N} W_{ij}\beta_j$ lies. Therefore, only the intervals that contain a possible value for $\sum_{j \in N} W_{ij}\beta_j$ need to be considered, so that we first identify $r_{max} = \arg\min\left\{t : \sum_{j \in N} W_{ij} \in I_t\right\}$ that is the index of the last interval to take into account. When $r_{max} = r$, $I_r$ is not a closed interval, which can cause difficulties for solving the associated optimization problem. In this case, for the purpose of solving the separation problem, we substitute the upper limit $B_r = \infty$ with $B_r = \sum_{j \in N} W_{ij}$, given that this is the maximum possible value for $\sum_{j \in N} W_{ij}\beta_j$. Then, for solving the separation problem (19)–(21), we proceed as follows:

1. Consider the intervals $I_t$, $t = 1, \ldots, r$ that determine the expression of the piecewise linear function $L(\cdot)$. For $t = 1, \ldots, r$ let $F^t = F^t_< \cup F^t_>$ denote an optimal solution to $L(W)$ for $W \in I_t$, and let $L(W) = a^t + Wb^t$, with $a^t$ and $b^t$ be defined as in (17) and (18), respectively.
2. For $t = 1, \ldots, r$,
   2.1 Solve the optimization problem:

$$v^t = \min\left[a^t + \left(\sum_{j \in N} W_{ij}\beta_j\right)b^t\right] - \sum_{j \in N} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})\beta_j,$$

$$A_t \leqslant \sum_{j \in N} W_{ij}\beta_j \leqslant B_t,$$

$$\beta_m = 1,$$

$$\beta_j \in \{0,1\}, \quad j \in N.$$

   Let $J^t$ denote the set of indices of variables at value one in the optimal solution to the above problem.
   2.2 If $v^t < 0$, the inequality (14) associated with $J \cup \{m\} = J^t$ and $F = F^t$ is violated by $(\bar{x}, \bar{y}, \bar{z})$.
3. If $\min\{v^t : t = 1, \ldots, r\} \geqslant 0$, no inequality (14) violated by $(\bar{x}, \bar{y}, \bar{z})$ exists.

### 4.5. Overall solution algorithm

The initial formulation is THP with inequalities (5) replaced by their tighter version (8)–(10) and with constraints (11) also included. We use a standard cut-and-branch algorithm to solve the problem. This means that separation of inequalities (14) is carried out only at the root node of the branching tree. The separated inequalities are added to the formulation and the problem is re-optimized. The separation phase is repeated until a stop condition is satisfied (see next section for details about the implementation of the algorithm).

Preliminary computational experiments indicated that even though the separation of inequalities (14) at some other nodes of the branching tree were effective in the reduction of the total number of nodes in the tree, they were not so effective on reducing the requiered cpu burden for solving to optimality the porblem. Moreover, these results indicated that the addition of inequalities (12) was no effective in reducing neither the duality gap nor the required cpu time to optimally solve the instances. Therefore, we did not consider these inequalities in the formulation. Neither are inequalities (13) considered in the solution, as they are dominated by inequalities (14).

## 5. Computational study

A thorough computational study has been carried out in order to test the performance of the formulation and the proposed valid inequalities. For the computational experiments we have used two data sets of benchmark instances commonly used in the literature, called AP (Australian Post) and CAB (Civil Aeronautics Board). Both sets of instances have been downloaded from the well-known OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html). The AP data set contains information on 200 postcode districts in Australia, together with a code to reduce the size of the set by grouping districts and the values of $W_{ij}$ (postal flow between districts). The CAB data set contains data of the 25 US cities with the highest traffic of airline passengers in 1970. The solver used in the study was Xpress-Mosel v2.0.0, run under Windows, and the processor was a Intel(R) Core(TM)2 T7600 with 2.33 GHz and 3 GB of RAM memory.

In the first part of the computational study we investigated the effectiveness of valid inequalities (14). To this end, we used the AP data set with $n = 10, 20$. For each instance size, we combined three different values of $p$, namely 3, 5 and 8, and three discount factors: 0.2, 0.5 and 0.8. The results are given in Table 1. For these instances we did the following experiments:

- We solved formulation THP without any type of inequalities. The results of this experiment are referred to as *LP*.
- We solved formulation THP with the cuts generated automatically by the Xpress solver. The results of this experiment are referred to as *XC*.
- We solved formulation THP with the valid inequalities (14) separated as explained in the previous section. The results of this experiment are referred to as *OC*.
- We solved formulation THP with both inequalities (14) and cuts generated by Xpress. The results of this experiment are referred to as *XC + OC*.

After some tuning, we set the following parameters when generating our cuts. We limited the number of generated valid inequalities at every iteration of the separation phase to 50, the total number of iterations for generating cuts to 30, the maximum number of generated cuts to 1500, and the cut tolerance (minimum violation for generating a cut) to 5. Also, for all the experiments a time limit of 40 hours was set. The optimal values for these instances are given in column under *Optimum*. The next four columns under *%Gap* allow us to compare the values of the bounds obtained at the root node with each of the experiments. In particular, we give the percent deviation with respect to the optimal solution ($100(Optimum - z^{\mathrm{COL}})/Optimum$), where COL = $LP, XC, OC, XC + OC$ depending on the considered column. The next four columns under *Time (seconds)* give the cpu times, in seconds, required to obtain the optimal solution of the problem with each experiment.

**Table 1**
Comparison of our valid inequalities and Xpress cuts for the AP data set.

| *n* | *p* | α | *Optimum* | *%Gap* | | | | *Time (seconds)* | | | | *Nodes* | | | | *# Cuts* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | *LP* | *XC* | *OC* | *XC + OC* | *LP* | *XC* | *OC* | *XC + OC* | *LP* | *XC* | *OC* | *XG + OC* | |
| 10 | 3 | 0.2 | 52,541.0 | 2.6 | 0.7 | 0.3 | 0.0 | 0.5 | 0.7 | 0.7 | 0.5 | 25 | 10 | 17 | 1 | 97 |
| | | 0.5 | 63,166.8 | 6.7 | 2.8 | 1.6 | 0.4 | 0.6 | 1.4 | 1.0 | 2.7 | 77 | 17 | 9 | 3 | 81 |
| | | 0.8 | 72,640.8 | 8.5 | 3.4 | 2.5 | 1.1 | 1.1 | 1.9 | 1.4 | 3.0 | 121 | 53 | 15 | 17 | 80 |
| | 5 | 0.2 | 34,340.0 | 5.9 | 4.1 | 1.8 | 0.6 | 0.9 | 1.4 | 1.5 | 3.1 | 213 | 107 | 33 | 17 | 200 |
| | | 0.5 | 49,418.7 | 11.0 | 7.9 | 4.2 | 1.9 | 3.7 | 4.6 | 3.2 | 5.9 | 789 | 329 | 61 | 49 | 187 |
| | | 0.8 | 64,013.2 | 14.1 | 10.4 | 5.7 | 3.3 | 10.0 | 12.5 | 5.6 | 8.1 | 2055 | 1177 | 155 | 107 | 153 |
| | 8 | 0.2 | 20,513.4 | 13.5 | 12.2 | 7.0 | 4.7 | 27.9 | 33.3 | 18.9 | 11.5 | 11,686 | 5225 | 606 | 209 | 312 |
| | | 0.5 | 39,288.1 | 18.2 | 17.4 | 8.9 | 7.2 | 66.3 | 78.1 | 27.1 | 23.6 | 24,000 | 15,612 | 761 | 507 | 304 |
| | | 0.8 | 57,953.4 | 20.1 | 19.4 | 10.3 | 8.5 | 180.3 | 166.3 | 53.6 | 36.6 | 61,413 | 37,838 | 1361 | 801 | 231 |
| 20 | 3 | 0.2 | 58,761.2 | 1.9 | 1.6 | 0.1 | 0.0 | 15.6 | 5.9 | 15.5 | 8.5 | 51 | 7 | 9 | 1 | 120 |
| | | 0.5 | 69,515.9 | 3.6 | 2.9 | 1.7 | 0.8 | 103.5 | 38.0 | 46.7 | 49.3 | 237 | 59 | 49 | 37 | 96 |
| | | 0.8 | 78,177.6 | 5.3 | 4.0 | 2.2 | 1.2 | 137.4 | 78.8 | 78.8 | 60.1 | 409 | 139 | 99 | 69 | 118 |
| | 5 | 0.2 | 46,480.4 | 3.6 | 3.3 | 1.4 | 0.6 | 198.1 | 143.8 | 84.9 | 71.6 | 603 | 351 | 91 | 49 | 313 |
| | | 0.5 | 61,061.4 | 6.9 | 6.2 | 3.6 | 2.4 | 2190.9 | 1234.9 | 735.0 | 436.8 | 5136 | 2662 | 734 | 394 | 237 |
| | | 0.8 | 73,592.9 | 9.1 | 8.0 | 5.4 | 4.1 | 3 hours 29 minutes | 1 hours 45 minutes | 3472.5 | 2780.5 | 29,451 | 13,902 | 4396 | 2496 | 165 |
| | 8 | 0.2 | 35,421.0 | 6.5 | 6.3 | 3.2 | 2.6 | 4 hours 22 minutes | 2 hours 57 minutes | 2106.5 | 1637.7 | 85,142 | 38,540 | 2215 | 1460 | 661 |
| | | 0.5 | 52,294.3 | 10.3 | 9.6 | 5.8 | 4.6 | 40 hours | 40 hours | 4 hours 32 minutes | 2 hours 13 minutes | n.a. | n.a. | 29,189 | 10,696 | 762 |
| | | 0.8 | 68,272.7 | 12.6 | 11.8 | 7.6 | 5.9 | 40 hours | 40 hours | 40 hours | 22 hours 31 minutes | n.a. | n.a. | n.a. | 105,744 | 333 |

Times greater than 3600 seconds are indicated in hours and minutes. If an entry of this set of columns is "40 hours", it means that the problem could not be solved within the given time limit. The next four columns under *Nodes* give the number of nodes of the branch and bound tree explored to obtain the optimal solution. Entries with *n.a.* correspond to instances that could not be solved to optimality. Finally, the last column gives the number of valid inequalities (14) added at the root node of the enumeration tree in the experiments that used them.

The obtained results allow us to make several observations, which apply to each individual experiment. First, we can appreciate that the quality of the results highly depends on the size and characteristics of the instances. In particular, the percent deviation tends to deteriorate as the value of $p$ increases. This behavior is analogous to that of $p$-hub median problems and just reflects the fact that the number of potential combinations for the location of the hubs increases with $p$ (up to a certain limit). We can also observe that the percent deviation also gets worse as the discount factor increases, indicating that the instances where more economies of scale are applied tend to be easier than those with a small reduction on the transportation costs between hub nodes.

From a complementary point of view, we can also compare the results of the different experiments among them. For each of the considered instances, it is easy to appreciate how the different experiments progressively give better results. In column *LP*, the percent deviation of the LP bound from the optimum value is, in general, not so good, ranging from 2% to 20%. In particular, in this column, the average percent deviation for the instances with $n = 10$ is 11.2% and for the instances with $n = 20$ is 6.6%. Using only the basic THP formulation, we were able to optimally solve all the 10 nodes instances, but only seven of the 20 nodes instances within the given time limit. The results in column *XC* indicate that using the cuts generated by Xpress reduces the percent deviation at the root node in all tested instances, although the improvement is not substantial. In particular, the average percent deviation decreases to 8.7% for the 10 nodes instances and to 5.9% for the 20 nodes instances. However, with the Xpress cuts we were able to solve exactly the same instances as before within the time limit, meaning that the improvement in the LP bound at the root node is not enough. On the other hand, the results of the experiments that use our proposed valid inequalities illustrate the considerable improvement with respect to the previous ones. In particular, the percent deviation for all instances is reduced in at least one half with respect to column *LP*, and generally much more. The average deviation of column *OC* decreases to 4.7% for the 10 nodes instances and to 3.5% for the 20 nodes instances. This improvement is also reflected in the cpu times required to optimally solve the instances that are greately reduced in most of the tested instances, particularly for the larger ones, allowing us to solve one more instance within the maximum cpu time. Finally, column *XC* + *OC* shows that the best results are obtained when combining our valid inequalities with the ones generated by Xpress. Now, the average percent deviation reduces to 3.1% for the 10 nodes instances and to 2.5% for the 20 nodes instances. What is more important, now we can solve all tested instances in the given time limit. In addition, it is interesting to observe that the number of generated inequalities is reasonably small and it depends on the characteristics of the instances.

The results of the previous experiments also give us valuable information on how, for the same instance, the structure of an optimal solution changes depending on the values of the parameters $p$ and $\alpha$. Next we describe such structures for an arbitrarily selected 20 nodes instance of the AP data set although, in general terms, the analysis is quite similar for the rest of the instances. For this analysis Fig. 1 depicts the change in the topological structure of the optimal solution of the selected instance when applying different discount factors $\alpha$ for a fixed value of $p = 3$. In all cases, nodes 6 and 14 are chosen to become hub facilities. However, as can be seen, the location of the hub facilities tend to become closer one to another as the $\alpha$ value increases. Fig. 2 depicts the difference in the topological structure of the optimal solution of the same 20 nodes instance when the discount factor is set to $\alpha = 0.2$ and the number of required hubs $p$ varies. It can be seen that again, in all cases, nodes 6 and 14 are chosen to become hub facilities. Now, when $p = 5$, 8 node 4 is selected to become a hub given that it is one of the most isolated nodes with a high amount of incoming/outgoing flow.

For the second part of our computational study we considered the set of the larger AP instances with $n = 25$, combining the three different values of $p$ and $\alpha$ as before. Now, we give the results of the root node of experiments *LP*, *OC* and *OC* + *XC*, but we only give the results of the branch and bound tree for the most successful experiments, namely *OC* and *OC* + *XC*. As before the cpu time limit was set to 40 hours. The obtained results are given in Table 2. The columns in this table have the same meaning as in the previous one. We were not able to solve the instances with $p = 8$ and $\alpha = 0.5$, 0.8 within the given time limit. Thus, for these two instances we report the best solutions found by Xpress within the time limit.

As can be seen, the results obtained with this set of larger instances follow the same tendency than the previous ones. The deviations of the LP bounds of formulation THP range from 1% to 13%, with an average value over the nine considered instances of 6.2%. In fact, these bounds only allowed to optimally solve the first 4 instances of this set witin the given time limit. As before, the results obtained with our valid inequalities were much better, improving once more the results of column *LP*. In particular, our inequalities were able to considerably reduce the percent deviation of all tested instances, yielding an average percent deviation of 3.6%. With these improved bounds we could solve to optimality 7 out of the 9 test instances of the set within the time limit. Columns under *XC* + *OC* again indicate that the best results are obtained when combining our valid inequalities with the ones of Xpress. The average percent deviation is reduced to 2.7% and the com-

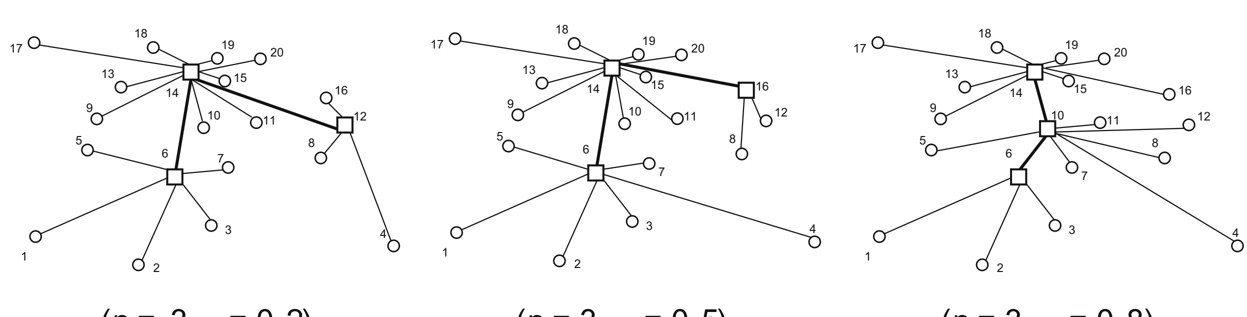


**Fig. 1.** Optimal solutions for different values of $\alpha$ for a 20 nodes instance.

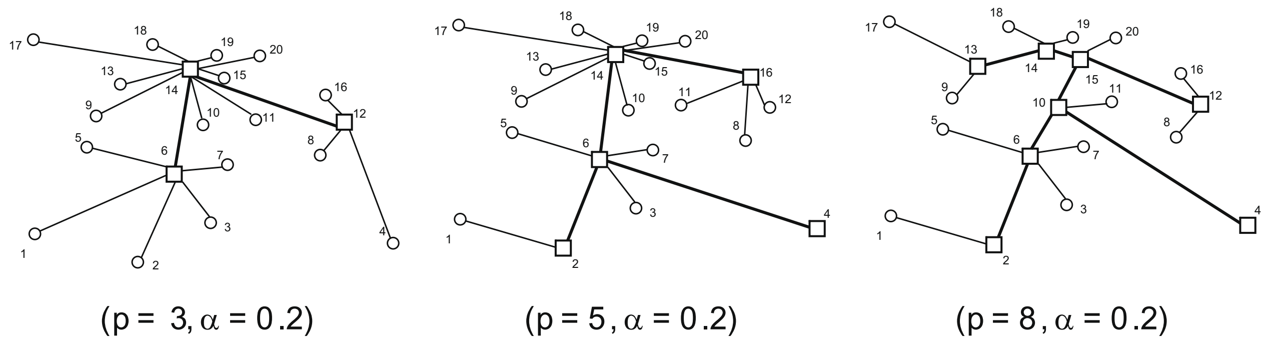


**Fig. 2.** Optimal solutions for different values of $p$ for a 20 nodes instance.

**Table 2**
Comparison of our valid inequalities and Xpress cuts for the AP data set.

| *n* | *p* | *α* | *Optimum* | *%Gap* | | | *Time (seconds)* | | *Nodes* | | *# Cuts* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | *LP* | *OC* | *XC + OC* | *OC* | *XC + OC* | *OC* | *XC + OC* | |
| 25 | 3 | 0.2 | 60,602.3 | 1.0 | 0.1 | 0.0 | 27.5 | 26.3 | 3 | 1 | 159 |
| | | 0.5 | 70,130.9 | 2.7 | 0.8 | 0.2 | 283.0 | 122.9 | 91 | 11 | 126 |
| | | 0.8 | 79,442.4 | 5.2 | 2.7 | 1.8 | 599.8 | 488.1 | 185 | 119 | 137 |
| | 5 | 0.2 | 47,432.7 | 3.2 | 1.2 | 0.4 | 276.8 | 278.4 | 99 | 49 | 235 |
| | | 0.5 | 61,046.7 | 6.2 | 3.2 | 2.0 | 1955.4 | 1175.7 | 796 | 317 | 194 |
| | | 0.8 | 73,569.9 | 8.4 | 5.2 | 3.7 | 4 hours 25 minutes | 2 hours 31 minutes | 5402 | 2206 | 128 |
| | 8 | 0.2 | 37,295.6 | 5.6 | 2.8 | 2.3 | 2 hours 44 minutes | 2 hours 30 minutes | 3136 | 2137 | 632 |
| | | 0.5 | *54,318.7 | 10.4 | 6.7 | 5.5 | 40 hours | 40 hours | n.a. | n.a. | 717 |
| | | 0.8 | *70,072.5 | 13.2 | 9.8 | 8.3 | 40 hours | 40 hours | n.a. | n.a. | 511 |

putational burden to obtain the optimal solution is also reduced. However, as mentioned above, this combination of inequalities is not enough for solving the two hardest instances of this set.

We finally performed a last series of computational tests, using now the CAB data set. We used instances with $n = 10, 15, 20, 25$ combining the three different values of $p$ and $\alpha$ as before. Again the time limit was set to 40 hours. The results are given in Table 3 where we display information of the root node for *LP*, *OC* and *OC* + *XC*, and results of the branch and bound tree for *OC* + *XC*, which like in the previous tests was the winning strategy. The columns in this table have the same meaning as before. The instances marked with an asterisk could not be solved to optimality within the time limit, so that for these instances we report the best solution found by Xpress in the given time limit.

As can be seen, the obtained results with this set of instances are quite good. Once more, the results are highly dependent of the characteristics of the instances. The percent deviation of the bounds in column *LP* ranges from 0.3% to 18%, with an average percent deviation over the 32 considered instances of 6.4%. Using these bounds we were able to optimally solve 19 of the 32 instances to optimality within the given time limit. Like in the previous tests, the results obtained with our valid inequalities were much better. In particular, using only our inequalities, the percent deviation from the optimum value was considerably reduced for of all tested instances, yielding an average percent deviation over the 32 instances of 2.8%. Now, we were able to solve to optimality 29 of the 32 test instances within the time limit. It is quite remarkable that with these inequalities we were able to close the optimality gap at the root node in 6 instances. Finally, like in the previous tests, the best results were obtained when combining our valid inequalities and the ones generated by Xpress. The average percent deviation is now reduced to 2.0% and the computational burden to obtain the optimal solution is also reduced. Furthermore, this combination is able to close the optimality gap at the root node in 11 out of the 32 instances. Nevertheless, the three more difficult instances of the set could also not be optimally solved using this combination within the maximum cpu time.

## 6. Conclusions and further research

In this paper we have presented the Tree of Hubs Location Problem. Similar problems requiring a tree-star network configuration in which the installation costs have to be minimized have been studied previously in the literature but, to the best of our knowledge, this is the first attempt to minimize the transportation costs over the network. We have presented an integer programming formulation for the problem together with several families of valid inequalities and an exact separation procedure for them. The numerical results of a series of computational experiments confirm the effectiveness of the proposed valid inequalities since using them it is possible to considerably improve both the LP bound at the root node and the computational time to optimally solve the instances. Using these inequalities and the ones generated by Xpress we have been able to optimally solve instances up to 25 nodes in reasonable computational time. These initial results should not be underestimated given the difficulty of the considered problem in which for a given selection of the hubs and the allocation pattern, the design of the backbone network is already an $\mathcal{NP}$-hard problem. This is a very important difference with classical hub location problems, where the backbone network is always a complete graph so that no design decision is required at this level.

Future research contemplates the design of an specialized exact algorithm for the problem. This will require a deep polyhedral study of the THLP for improving the lower bounds associated with the LP relaxations of the reinforced formulations together with the design of efficient heuristics able to provide tight upper bounds for larger and more realistic instances. Other avenues of research include the study

**Table 3**
Comparison of our valid inequalities and Xpress cuts for the CAB data set.

| *n* | *p* | *α* | *Optimum* | *%Gap* | | | Time (seconds) | *Nodes* | *# Cuts* |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *LP* | *OC* | *XC + OC* | *XC + OC* | *XC + OC* | |
| 10 | 3 | 0.2 | 494.5 | 0.7 | 0.0 | 0.0 | 0.2 | 1 | 77 |
| | | 0.5 | 613.0 | 2.5 | 0.0 | 0.0 | 0.2 | 1 | 70 |
| | | 0.8 | 719.0 | 5.1 | 0.2 | 0.0 | 0.7 | 1 | 423 |
| | 5 | 0.2 | 322.9 | 3.9 | 0.2 | 0.0 | 0.8 | 1 | 250 |
| | | 0.5 | 499.4 | 8.2 | 1.1 | 0.3 | 2.9 | 1 | 429 |
| | | 0.8 | 667.4 | 10.4 | 2.4 | 1.2 | 10.4 | 9 | 439 |
| | 8 | 0.2 | 190.5 | 13.7 | 3.0 | 2.8 | 10.5 | 35 | 442 |
| | | 0.5 | 411.8 | 16.6 | 5.2 | 4.2 | 18.3 | 107 | 180 |
| | | 0.8 | 631.6 | 17.7 | 7.1 | 5.6 | 34.8 | 371 | 409 |
| 15 | 3 | 0.2 | 1915.2 | 2.1 | 0.2 | 0.0 | 3.4 | 1 | 232 |
| | | 0.5 | 2324.4 | 3.9 | 0.6 | 0.0 | 57.9 | 3 | 474 |
| | | 0.8 | 2666.1 | 4.4 | 0.8 | 0.2 | 93.9 | 1 | 1346 |
| | 5 | 0.2 | 1299.6 | 4.3 | 0.6 | 0.0 | 22.3 | 1 | 369 |
| | | 0.5 | 1935.1 | 8.2 | 2.4 | 1.7 | 415.9 | 36 | 1497 |
| | | 0.8 | 2454.2 | 9.1 | 3.2 | 1.9 | 2953.7 | 119 | 1057 |
| | 8 | 0.2 | 876.4 | 8.9 | 3.4 | 1.7 | 1 hours 12 minutes | 73 | 1051 |
| | | 0.5 | 1590.3 | 11.7 | 4.9 | 3.2 | 2317.8 | 417 | 755 |
| | | 0.8 | 2250.3 | 11.9 | 5.2 | 4.0 | 3 hours 3 minutes | 1191 | 906 |
| 20 | 3 | 0.2 | 4170.1 | 0.3 | 0.0 | 0.0 | 7.6 | 1 | 90 |
| | | 0.5 | 5234.9 | 1.6 | 0.0 | 0.0 | 18.6 | 1 | 156 |
| | | 0.8 | 6279.4 | 4.6 | 1.6 | 0.9 | 351.5 | 69 | 520 |
| | 5 | 0.2 | 2808.7 | 4.4 | 0.8 | 0.1 | 67.3 | 45 | 518 |
| | | 0.5 | 4384.3 | 8.5 | 3.0 | 1.8 | 604.0 | 171 | 855 |
| | | 0.8 | 5663.5 | 8.2 | 3.8 | 2.3 | 1 hours 51 minutes | 945 | 844 |
| | 8 | 0.2 | 2057.0 | 8.0 | 3.1 | 2.6 | 1798.2 | 368 | 1122 |
| | | 0.5 | 3700.2 | 11.1 | 5.2 | 4.3 | 6 hours 44 minutes | 5634 | 811 |
| | | 0.8 | *5283.1 | 12.3 | 6.9 | 5.5 | 40 hours | n.a. | 1381 |
| 25 | 3 | 0.2 | 6554.6 | 0.8 | 0.0 | 0.0 | 15.3 | 1 | 127 |
| | | 0.5 | 8274.0 | 2.2 | 0.0 | 0.0 | 36.6 | 1 | 245 |
| | | 0.8 | 9923.9 | 3.6 | 0.8 | 0.2 | 1131.2 | 7 | 321 |
| | 5 | 0.2 | 4791.1 | 4.3 | 0.9 | 0.3 | 430.0 | 10 | 862 |
| | | 0.5 | 7190.7 | 7.9 | 3.1 | 2.3 | 4 hours 34 minutes | 683 | 1085 |
| | | 0.8 | 9173.4 | 7.7 | 3.7 | 2.9 | 7 hours 50 minutes | 1717 | 1472 |
| | 8 | 0.2 | 3752.9 | 6.4 | 2.7 | 1.9 | 4 hours 23 minutes | 710 | 1756 |
| | | 0.5 | *6272.9 | 10.5 | 5.7 | 4.3 | 40 hours | n.a. | 1036 |
| | | 0.8 | *8756.7 | 13.1 | 8.4 | 7.3 | 40 hours | n.a. | 1542 |

of more realistic models that incorporate capacities onto the hubs and/or the arcs of the small tree, as well as additional constraints on the performance of the system.

## Acknowledgements

The authors thank Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I + D + I) – projects TIC2006-14961-C05-(01,04) and HA2005-56 – and ERDF funds for the support given to the research which led to this article. The research of the first author has been partially supported through grant 197243/217997 from CONACYT, México. The research of the third author has been partially supported by Fundación Séneca – projects 02911/PI/05 and 08716/PI/08. Thanks are also due to the anonymous referee for his valuable comments.

## 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.
Aykin, T., 1994. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. European Journal of Operational Research 79, 505–523.
Aykin, T., 1995. Networking policies for hub-and-spoke systems with application to the air transportation system. Transportation Science 29, 201–221.
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.
Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72, 387–405.
Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2002. Hub location problems. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location Applications and Theory. Springer, Heidelberg, pp. 373–408.
Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005a. Hub arc location problems: Part I. Introduction and results. Management Science 51 (10), 1540–1555.
Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2005b. Hub arc location problems: Part II. Formulations and optimal algorithms. Management Science 51 (10), 1556–1571.
Cánovas, L., García, S., Marín, A., 2007. Solving the uncapacitated multiple allocation hub location problem by means of a dual ascent technique. European Journal of Operational Research 179, 990–1007.
Cánovas, L., Landete, M., Marín, A., 2006. New formulations for the uncapacitated multiple allocation hub location problem. European Journal of Operational Research 172, 274–292.
Chen, H., Campbell, A.M., Thomas, B.W., 2008. Network design for time-constrained delivery. Naval Research Logistics 55, 493–515.
Contreras, I., 2009. Network Hub Location: Models, algorithms and related problems. PhD. Thesis, Department of Statistics and Operations Research, Technical University of Catalonia, Spain.

Contreras, I., Díaz, J.A., Fernández, E., 2009. Lagrangean relaxation for the capacitated hub location problem with single assignment. Operations Research Spectrum 31, 483–505.
Contreras, I., Díaz, J.A., Fernández, E., 2008. Branch and price for large-scale capacitated hub location problems with single assignment. Optimization (Online).
Ebery, J., Krishnamoorthy, M., Ernst, A., Boland, N., 2000. The capacitated multiple allocation hub location problem: Formulations and algorithms. European Journal of Operational Research 120, 614–631.
Ernst, A., Krishnamoorthy, M., 1998a. An exact solution approach based on shortest-paths for *p*-hub median problems. INFORMS Journal on Computing 10, 149–162.
Ernst, A., Krishnamoorthy, M., 1998b. 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., 1999. Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86, 141–159.
Gelareh, S., 2008. Hub Location Models in Public Transport Planning. PhD. Thesis, University of Saarlandes, Germany.
Gouveia, L., 1995. Using the Miller–Tucker–Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. Computers and Operations Research 22, 959–970.
Hakimi, S.L., Schmeichel, E.F., Labbé, M., 1993. On locating path or tree shaped facilities on networks. Networks 23, 543–555.
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.
Hu, T.C., 1974. Optimum communication spanning trees. SIAM Journal on Computing 3 (3), 188–195.
Johnson, D.S., Lenstra, J.K., Rinooy Kan, J.K., 1987. The complexity of the network design problem. Networks 8, 279–285.
Kim, T.U., Lowe, T.J., Tamir, A., Ward, J.E., 1996. On the location of a tree-shaped facility. Networks 28, 167–175.
Kim, H.J., Tcha, D.W., 1992. Optimal design of a two-level hierarchical network with tree-star configuration. Computers and Industrial Enginneering 22 (3), 273–281.
Klincewicz, J.G., 1991. Heuristics for the *p*-hub location problem. European Journal of Operational Research 53, 25–37.
Klincewicz, J.G., 1996. A dual algorithm for the uncapacitated hub location problem. Location Science 4, 173–184.
Klincewicz, J.G., 1998. Hub location in backbone/tributary network design: A review. Location Science 6, 307–335.
Labbé, M., Laporte, G., Rodríguez-Martín, I., Salazar-González, J.J., 2004. The ring star problem: Polyhedral analysis and exact algorithm. Networks 43 (3), 177–189.
Labbé, M., Yaman, H., Gourdin, E., 2005. A branch and cut algorithm for hub location problems with single assignment. Mathematical Programming 102, 371–405.
Laporte, G., Rodríguez-Martín, I., 2007. Locating a cycle in a transportation or a telecommunications network. Networks 50 (1), 92–108.
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 (3), 185–194.
Lee, Y., Lu, L., Qiu, Y., 1994. Strong formulations and cutting planes for designing digital data service networks. Telecommunications Systems 2, 261–274.
Marín, A., 2005a. Formulating and solving the splittable capacitated multiple allocation hub location problem. Computers and Operations Research 32, 3093–3109.
Marín, A., 2005b. Uncapacitated Euclidean hub location: Strengthened formulation new facets and a relax-and-cut algorithm. Journal of Global Optimization 33, 393–422.
Marín, Á., 2007. An extension to rapid transit network design problems. TOP 15, 231–241.
Mayer, G., Wagner, B., 2002. HubLocator: An exact solution method for the multiple allocation hub location problem. Computers and Operations Research 29, 715–739.
Nickel, S., Schöbel, A., Sonneborn, T., 2001. Hub location problems in urban traffic networks. In: Niittymaki, J., Pursula, M. (Eds.), Mathematics Methods and Optimization in Transportation Systems. Kluwer Academic Publishers.
Nguyen, V.H., Knippel, A., 2007. On tree star network design. In: INOC 2007: International Network Optimization Conference, Spa, Belgium.
O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research 32, 393–404.
O'Kelly, M.E., Miller, H.J., 1994. The hub network design problem: A review and synthesis. Journal of Transport Geography 2 (1), 31–40.
Ortega, F., Wolsey, L., 2003. A branch-and-cut algorithm for the single-commodity uncapacitated fixed-charge network flow problem. Networks 41 (3), 143–158.
Pop, P.C., Kern, W., Still, G., 2006. A new relaxation method for the generalized minimum spanning tree problem. European Journal of Operational Research 170, 900–908.
Puerto, J., Tamir, A., 2005. Locating tree-shaped facilities using the ordered median objective. Mathematical Programming 102, 313–338.
Skorin-Kapov, D., Skorin-Kapov, J., O'Kelly, M., 1996. Tight linear programming relaxations of uncapacitated *p*-hub median problems. European Journal of Operational Research 94, 582–593.