Formulations and Decomposition Methods for the Incomplete Hub Location Problem With and Without Hop-Constraints

Ricardo S. de Camargo
Department of Industrial Engineering, Universidade Federal de Minas Gerais, MG, Brazil

Gilberto de Miranda Jr.
Department of Applied Mathematics, Universidade Federal do Espírito Santo, ES, Brazil

Morton E. O’Kelly
Department of Geography, Ohio State University, OH, USA

James F. Campbell
College of Business Administration, University of Missouri – St. Louis, MO, USA

Abstract
The incomplete hub location problem with and without hop-constraints is modeled using a Leontief substitution system approach. The Leontief formalism provides a set of important theoretical properties and delivers formulations with tight linear bounds that can explicitly incorporate hop constraints for each origin-destination pair of demands. Furthermore, the proposed formulations are amenable to a Benders decomposition technique which can solve large scale test instances. The performance of the devised algorithm is primarily due to a new general scheme for separating Benders feasibility cuts. This scheme relies on a stabilization step that is directly responsible for the solution of instances up to 80 nodes.

Keywords: Hub location problems, Hub and spoke networks, Leontief flow substitution systems, Benders decomposition method, Improved Benders feasibility cuts.

1. Introduction
Hub-and-spoke networks have been a rich source of research for over a quarter century (Campbell and O’Kelly, 2012), since the seminal work of O’Kelly (1986). Hubs are facilities which are located to connect and route flow demands between many pairs of origin and destination (OD) nodes in a network. They can function as switching/sorting/distribution and/or as consolidation/break-bulk centers. When operating as consolidation facilities, hubs allow flows to exploit economies of scale on inter-hub links when flows are transported by high volume carriers. The way that the OD nodes are connected to the hubs, forming the spokes of the network, and how the hubs are inter-connected lead to a great variety of different topologies with applicability to numerous areas in both telecommunications and transportation. The exploitation of scale economies provides one strong incentive for the creation and use of hub-and-spoke networks.

To provide efficient communication and transportation between many origins and destinations, the design of hub-and-spoke networks combines elements of classical facility location and multi-commodity network design problems.
The facility location features correspond to the location of the hubs and the assignment of the non-hub nodes to the installed hubs; while the multi-commodity network design aspects relate to the linkage of the hubs by arcs with proper scale economies, and to the routing of the many to many origin-destination demands through the network. The exact form of hub and non-hub nodes inter-connections is application dependent.

Despite technological differences, passengers connecting between two flights at an airport and freight consolidation in transportation networks can be considered to be analogous to switching and concentration operations in telecom networks, respectively. Hence it is possible to adopt a generic terminology to facilitate hub-and-spoke modeling to both telecommunication and transportation applications. Without loss of context, hereafter, a transfer arc will denote the connection between two hubs; an access arc will represent the link between a non-hub node and a hub; finally, a direct arc will express the linkage between an OD pair of demand nodes. The backbone of a network will refer to all the installed transfer arcs and the incident hubs; while the fixed access arcs will be regarded as the access or tributary subnetwork. Usually access arcs have a simple unitary variable cost, whereas transfer arcs have their costs discounted by a constant scalar \(0 \leq \alpha \leq 1\) to represent the effects of scale economies. Further the number of hubs to be installed can be endogenously decided when hub setup costs are prescribed; or exogenously, when the number of hubs to be locate are known a priori.

Early hub location problems (O’Kelly, 1986, 1987; O’Kelly et al., 1996) assume an access subnetwork in which non-hub nodes can directly interact with only one of the hubs on the backbone, resulting in a single allocation structure; or with several hubs, yielding in a multiple allocation structure. These problems also assume that hubs are fully interconnected by transfer arcs generating complete hub networks. Complete hub networks allow for the lowest transport-\(\)transmission cost possible (due to the aforementioned discount), on the other hand, they may actually have quite small optimal flows on some of the transfer arcs (see Campbell (2013a)), at a level insufficient to justify the assumed economies of scale. Thus, while these assumptions simplify the modeling and computational analysis, they create unrealistic networks in many cases.

In contrast to complete hub networks, incomplete hub networks allow for more interesting, flexible linking structures with wider applicability. In these networks, the installed hubs are interconnected by only some of the available transfer arcs resulting in an underlying structure which is dependent on topological aspects of the application being addressed or emerge from the assumed cost components. The former requires the incorporation of specific technological constraints during modeling, while the latter requires far more elaborate formulations which can yield many types of backbone network. In a recent work, O’Kelly et al. (2015) describe a model for such cases with direct arcs and a generalized cost function that includes fixed and variable cost components for arcs and hubs. The proposed formulation can produce many different system designs for the access and backbone networks depending on the relative magnitudes of the cost coefficients. From a mathematical point of view, the formulation blends aspects of generalized network design and quadratic location models, making the solution of large instances a computational challenge. The increased difficulty is due to the large number of integer variables and to the absence of topological restrictions typically used in hub location models to reduce the range of admissible solutions.

Though other authors (e.g., Alumur et al. (2009); Contreras et al. (2010); Campbell (2013b); Campbell et al. (2005a,b)) have addressed hub location models with some type of incomplete backbone network; the compact model in O’Kelly et al. (2015) is one starting point for the present research. Their model uses a well known divergence theorem to deploy an aggregated representation for the flows transported through the network. Rather than tracing every explicit path, the formulation relies on flow conservation constraints to track the flows on the arcs only by their origin. The resultant mathematical program is a natural extension of the 3-index model of Ernst and Krishnamoorthy (1996) with the addition of direct arcs, but with better modeling for the economies of scale. By changing the relative magnitudes of the cost components for the different types of arcs and for hubs, it is possible to obtain a wide range of topologies and protocols (see O’Kelly and Miller (1994) for the available types) for the backbone. For thorough surveys on different types of hub-and-spoke networks and their variants please refer to Farahani et al. (2013), Alumur and Kara (2008), Campbell et al. (2002), and Klincewicz (1998).

The formulation of O’Kelly et al. (2015) is quite compact, but it does not allow for problems with more than 25 nodes to be solved in a reasonable time on a regular computer, nor it is amenable to extension for other issues, such as controlled service levels for each OD demand. For instance, when designing incomplete hub networks, it is customary to specify the number \(p\) of hubs to be installed in order to assure that every OD pair demand has at most \(p + 1\) arcs on its path. (That is fact is the worst case; more fully interconnected backbone networks will have fewer hops.) Fewer hubs may improve the service level of a network by restricting the number of sorting/connection points an OD flow
can face on its path, i.e. the path is implicitly hop constrained. While the prescription of installing a small number of hubs is an easy way to limit sorting points, it may yield poor system designs. By limiting the number of transfer arcs that are installed, to at most \( p(p-1)/2 \) arcs, this idea may prevent the OD demands from exploiting scale economies on the transfer arcs, resulting in networks with much higher costs. The trade off between hubs and hops is subtle as shown in the following discussion. One way to pose the trade-off is to see a small number of fully connected hubs or a large number of partially interconnected hubs. To gauge the merits of these options, it is necessary to formulate (explicitly) the concept of hop constrained path for each OD pair. This requires a much more elaborate mathematical program that leads to the design of more interesting, flexible, and lower cost hub networks; this formulation, extension and solution is the subject of the present paper.

Figures 1 and 2 illustrate the aforementioned issue by showing the resultant incomplete hub networks for the same data test problem when the number of hubs to be installed is set first to \( p = 3 \), and second when the paths of the OD demands are actually hop constrained to \( p + 1 \) (4) arcs, respectively. Direct and transfer arcs are represented by red, and black lines, whereas dotted blue lines serve as access arcs. Hubs and non-hub nodes are depicted by black dots and triangles, respectively. Only two transfer arcs are installed in the network of Figure 1 compared to nine in Figure 2. This explains the lower costs of the latter, since more transfer arcs are available allowing for the OD demands to better exploit the scale economies. Notice that imposing the limit of 4 arcs on any OD path is respected in Figure 2. One might be tempted to claim that by making \( p = 6 \), i.e. to install the same number of hubs of Figure 2, the same network would be obtained. Figure 3 shows that this is not the case. In brief, if a given hub solution produces an unsatisfactory service level, re-solving the problem with hop constraints can produce different (more) hubs (Fig. 2), and for a given number of hubs, (say 6) optimization without regard to hop constraints can produce different locations (Fig. 3). For this reason, we refer to the hub and hop versions of the problem as interdependent.

Figure 1: p-hub constrained problem with \( p = 3 \).
One way to properly model hop constrained hub networks is to use a Leontief Substitution System (see Veinott (1968) and Provan and Billera (1982)). The Leontief formalism provides a nice modeling technique that allows one to translate the required flow balance constraints into recourse constraints which, combined with variable redefinitions, lead to very large formulations but with very attractive theoretical properties, and stronger bounds that can solve rather large test problems. Inspired by the works of Eppen and Martin (1987); Martin (1987); Martin et al. (1990); Jeroslow et al. (1992), and more specifically by the work of Gouveia (1998) on solving the hop constrained minimum spanning and Steiner tree problems, the Leontief substitution technique is applied here to the problem addressed by O’Kelly et al. (2015), but explicitly incorporating hop constrains and cleaning up a minor formulation subtlety (see below).

The model of O’Kelly et al. permits that direct arcs are established parallel to other types of arcs, e.g. a pair of hubs can be connected by both a transfer arc and a direct arc. This is similar to a result for bridge arcs in the general hub arc models in Campbell et al. (2005a,b). While there are some practical instances with parallel arcs of different types, in a strategic hub location model one can view the costs for movement between a pair of nodes as reflecting use of a bundle of vehicles (e.g., aircraft of different types) to optimally serve the demand being carried, as in O’Kelly (2012). Then it is quite reasonable to model only a single arc between any pair of nodes. Further, the parallel arcs in O’Kelly et al. (2015) creates inefficiencies in the formulation, which the new proposed formulation avoids. Further, the proposed formulation has one interesting characteristic. When the hop constraints are disregarded, the model can be simplified resulting in a formulation that is similar to the one devised by Gelareh and Nickel (2011), but with fewer constraints and stronger linear relaxation bounds.

Exploiting this feature then, in the present paper, two formulations for the incomplete hub location problem with direct arcs and having relative costs (fixed and variable cost for arcs, and fixed costs for hubs) to determine the optimal
network topology, which can include single and multiple allocation access networks, and complete, incomplete or tree backbone networks. One of the formulations has hop constraints that limit the number of arcs as well as the number of hubs in any OD path. As both formulations have stair-case constraint matrices with subsystems, one for each OD pair of demand, being coupled by complicating variables, they are amenable to be solved by decomposition methods, such as the Benders decomposition technique (Benders, 1962).

Generally speaking, the technique consists of leaving the arc and hub decisions in a master problem (MP), and the routing decisions distributed in several linear programming subproblems (SPs), one for each OD demand pair. The SPs are assembled by using the arcs proposed by the MP. The cost of the routing decisions is then approximated by Benders cuts on the MP, which are generated after solving the SPs. The method requires a coordination procedure that iterates between the solution of the MP and the SPs until optimality is reached. Though the Benders decomposition method has been already successfully used on other hub location problems (Camargo et al., 2009; Contreras et al., 2011a), its success has been largely explained by the use of formulations with strong linear relaxations, and a stair-case constraint matrix, and by having feasible SPs on all the iterations. When infeasible SPs are found on some of the iterations, the procedure tends to have a slow convergence rate (Gelareh and Nickel, 2011), being normally not the most efficient method. However, recently, de Sá et al. (2014) have shown that it is possible to greatly speed up the Benders technique even when facing infeasible SPs during the Benders’ cycles. Their scheme for generating Benders cuts is further improved here allowing for greater speedups for solving larger, more difficult problems.

The main contributions of this paper are threefold: (i) a flexible hop constrained formulation for the incomplete hub location problem; (ii) an improved formulation for the incomplete p-hub location problem; (iii) a well deployed Benders decomposition algorithm for the aforementioned formulation. The proposed method relies on a new scheme for selecting Benders feasibility cuts and a stabilization step that are directly responsible for the solution of instances up to 80 nodes. The remainder of this paper is organized as follows: §2 and §3 present the two proposed formulations and the Benders Decomposition algorithm to solve them, respectively, while §3.4 introduces the new cut selection procedure. The attained computations results are reported in §4. Finally, §5 concludes this paper with final remarks and future works.

2. Notation and Definitions

The critical insight here results from the computation experiments done in O’Kelly et al. (2015) which allows an enhanced understanding of the structure of hub networks and their flows. The key development is the specification of three types of interaction on an incomplete hub network: direct connections, access and transfer arcs. In the proposed models, direct connections can only be used to connect and to transport the demands of non-hub nodes. All other traffic travels via one or more hubs, so that each OD path starts with an access arc that conveys the flow to a hub, where it can depart on a transfer arc to another hub or on another access arc to the destination.

While the distinction between nodes and hubs is well known in this area, an important feature of hubs is that they permit traffic to make connections. Imagine a given node which is not a hub. This node will have a relatively simple installed infra-structure, compared with that at a hub, which must accommodate connecting, sorting, consolidation, etc. So in the formulations, flows to hubs are designed to take advantage of the potential to make a connection there. This is slightly different than the model in O’Kelly et al. (2015), which allows direct connections to start or end at hubs, and thus can be considered to be slightly less general. On the other hand, the improved structure of the resulting models is better suited to decomposition techniques, such as the one developed in this paper.

As all formulations share the same sets, parameters, and some common decision variables, these are shown in Tables 1 and 2, respectively. Further, before introducing the proposed models, the three-index formulation of O’Kelly et al. (2015) is here restated for sake of completeness. Its additional variables are described in Table 3.

$$
\begin{align*}
\min & \sum_{k \in N} f^H_k z_k + \sum_{i \in N} \sum_{j \in N, j \neq i} z_{ij} + \sum_{i \in N} \sum_{k \in N} (\hat{c}_{ik} + \hat{c}_{ik}^{2} Z_{ik}) \\
& + \sum_{i \in N} \sum_{m \in N} (\hat{c}_{im}^{2} Y_{im}) + \sum_{i \in N} \sum_{j \in N \setminus \{i\}, m \in m} (\hat{c}_{mj}^{3} X_{mj}) \\
& \quad + \sum_{i \in N} \sum_{j \in N \setminus \{i\}, m \in m} (\hat{c}_{jm}^{3} X_{jm})
\end{align*}
$$

(1)
**Table 1: Sets and parameters.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>Set of points/nodes.</td>
</tr>
<tr>
<td>([0,1,2,3])</td>
<td>Set of connection types: (0) direct arc, (1) starting access arc, (2) transfer arc, (3) finishing access arc.</td>
</tr>
<tr>
<td>( w_{ij} )</td>
<td>Flow (e.g., of goods or passengers) required to be routed for each ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( c_{ij} )</td>
<td>Distance matrix for each ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( f_i, f^1, f^2, f^3 )</td>
<td>Fixed cost per distance unit for each type of available connection.</td>
</tr>
<tr>
<td>( b_i, b^1, b^2, b^3 )</td>
<td>Variable cost per distance unit per unit flow for each type of available connection.</td>
</tr>
<tr>
<td>( A_{ij} )</td>
<td>Arc specific fixed cost per unit distance related to suitable vehicle issues to connect nodes ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( M )</td>
<td>Large constant (e.g. ( \sum_{i\in N} w_{ij} )).</td>
</tr>
<tr>
<td>( O_i )</td>
<td>Aggregated demand at origin ( i \in N ), i.e. ( O_i = \sum_{j \in N} w_{ij} ).</td>
</tr>
<tr>
<td>( D_j )</td>
<td>Aggregated demand at destination ( j \in N ), i.e. ( D_j = \sum_{i \in N} w_{ij} ).</td>
</tr>
</tbody>
</table>

**Table 2: Common notation for all formulations.**

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_k \in {0,1} )</td>
<td>Decides the installation of a hub at node ( k \in N ).</td>
</tr>
<tr>
<td>( y_{ij}^4 \in {0,1} )</td>
<td>Decides the installation of a direct arc between nodes ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( y_{ij}^1 \in {0,1} )</td>
<td>Decides the installation of a starting access arc between nodes ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( y_{ij}^2 \in {0,1} )</td>
<td>Decides the installation of a transfer arc between nodes ( i, j \in N: i \neq j ).</td>
</tr>
<tr>
<td>( y_{ij}^3 \in {0,1} )</td>
<td>Decides the installation of a finishing access arc between nodes ( i, j \in N: i \neq j ).</td>
</tr>
</tbody>
</table>

**Table 3: Notation for O’Kelly and Campbell compact model.**

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Z_k \geq 0 )</td>
<td>Measures the flow from origin ( i \in N ) transported through hub ( k \in N ).</td>
</tr>
<tr>
<td>( Y_{km} \geq 0 )</td>
<td>Measures the flow from origin ( i \in N ) transported through hubs ( k, m \in N ).</td>
</tr>
<tr>
<td>( X_{klj} \geq 0 )</td>
<td>Measures the flow from origin ( i \in N ) coming from hub ( k \in N ) arriving at destination ( j \in N ).</td>
</tr>
</tbody>
</table>

s.l.: \[
\sum_{k \in N} Z_k = O_i - \sum_{j \in N} w_{ij} y_{ij}^4 \quad \forall i \in N \tag{2}
\]
\[
\sum_{m \in N} X_{imj} = w_{ij}(1 - y_{ij}^4) \quad \forall i \in N, j \in N \tag{3}
\]
\[
\sum_{m \in N} Y_{ilm} - \sum_{k \in N} Y_{ikl} = Z_{il} - \sum_{j \in N} X_{ijl} \quad \forall i \in N, l \in N \tag{4}
\]
\[
Z_k \leq O_i z_k \quad \forall i \in N, k \in N \tag{5}
\]
\[
\sum_{r \in N} X_{rm} \leq D_j y_{jm} \quad \forall j \in N, m \in N \tag{6}
\]
\[
Z_k \leq O_i y_{ik}^4 \quad \forall i \in N, k \in N \tag{7}
\]
\[
\sum_{r \in N} Y_{kim} \leq M y_{km}^1 \quad \forall k \in N, m \in N \tag{8}
\]
\[
\sum_{r \in N} X_{rmj} \leq D_j y_{km}^2 \quad \forall m \in N, j \in N \tag{9}
\]
\[
y_{ik}^1 \leq z_k \quad \forall i, k \in N: i \neq k \tag{10}
\]
\[
y_{jm}^2 \leq z_m \quad \forall j, m \in N: j \neq m \tag{11}
\]
\[
y_{km} \leq z_k \quad \forall k, m \in N: k \neq m \tag{12}
\]
\begin{align*}
y_{km}^3 & \leq z_m & \forall k, m \in N : k \neq m \quad (13) \\
y_{ik}^1 & \leq (1 - z_i) & \forall i, k \in N : k \neq i \quad (14) \\
y_{mj}^2 & \leq (1 - z_j) & \forall j, m \in N : m \neq j \quad (15)
\end{align*}

where \( c_{ij}^0 = c_{ij}(f^0 + A_{ij} + b^0w_{ij}) \), \( c_{ik}^1 = c_{ik}(f^1 + A_{ik}) \), \( c_{km}^2 = c_{km}(f^2 + A_{km}) \), \( c_{mj}^3 = c_{mj}(f^3 + A_{mj}) \), \( c_{ik}^1 = c_{ik}b^1 \), \( c_{km}^2 = c_{km}b^2 \), \( c_{mj}^3 = c_{mj}b^3 \), and The objective function (1) accounts for all the fixed and variable costs occurring in the system. Constraints (2)-(4) are flow balancing equations based on the well known Divergence Theorem. Constraints (5)-(9) are activation constraints that install arcs and hubs in order to enable the proper flows. Constraints (10)-(15) tie hubs and nodes to the suitable connections, enforcing the design of feasible systems. Formulation (1)-(15) is able to provide any kind of network topology for the backbone, including all the cases described in O’Kelly and Miller (1994) taxonomy. This formulation is efficient and quickly reaches small optimality gaps, because it has a very good response to standard cutting planes usually available on the majority of the commercial optimization platforms (e.g. IBM CPLEX, GUROBI and XPRESS-MP). On the other hand, closing these gaps requires a great computational effort. Anyway, the model is very useful as a standard to test the correctness and sharpness of alternative modeling efforts, given its compactness on the number of variables.

To avoid direct arcs established in parallel to other types of arcs, i.e. having direct arcs starting or ending at hubs, formulation (1)-(15) can easily be augmented by including the following constraints:

\begin{align*}
y_{ij}^0 & \leq (1 - z_i) & \forall i, j \in N : i \neq j \quad (16) \\
y_{ij}^0 & \leq (1 - z_j) & \forall i, j \in N : i \neq j \quad (17)
\end{align*}

Nevertheless, formulation (1)-(15) does not have the desirable properties that allow for the solution of larger test problems.

A more explicit arc-path flow based formulation, which accounts for all the possible feasible paths for a given OD pair, can be devised along the lines of the model of Hamacher et al. (2004) for the uncapacitated multiple allocation hub location problem. This avenue leads to an extremely large formulation which requires the generation of the paths and their associated costs on the fly, since it is well known that the number of existing paths in a complete graph is not a polynomial function of its number of nodes, suggesting thus the use of decomposition/column generation schemes. A simple way to obtain an improved formulation, without the need for a non-polynomial number of decision variables, is to disaggregate the flows and track each origin-destination pair. This type of resort is inspired by the original model of Koopmans and Beckmann (1957) for the Quadratic Assignment Problem, and can be found in other formulations for other hub location problems (e.g. Gelareh and Nickel (2008, 2011) and de Sá et al. (2014)). In fact, with some rather simple modifications, the model of Gelareh and Nickel (2011) can be adapted to handle the problem addressed by O’Kelly et al. (2015). For sake of comparisons, the formulation of Gelareh and Nickel (2011) is here restated with the decision variables explained in Table 4:

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Decision Variables} & \textbf{Description} \\
\hline
\( h_{ik} \geq 0 \) & Measures the flow of the OD pair \( i, j \in N, i \neq j \), on the starting access connection from node \( i \) to hub \( k \in N : i \neq k \). \\
\( x_{km} \geq 0 \) & Measures the flow of the OD pair \( i, j \in N, i \neq j \), on the transfer connection \( k - m, k, m \in N : k \neq m, i \neq m, j \neq k \). \\
\( t_{km} \geq 0 \) & Measures the flow of the OD pair \( i, j \in N, i \neq j \), on the finishing access connection from hub \( m \in N \) to node \( j, j \neq m \). \\
\( c_{ij} \geq 0 \) & Measures the flow of the OD pair \( i, j \in N, i \neq j \), on the direct connection from non-hub node \( i \in N \) to non-hub node \( j \). \\
\hline
\end{tabular}
\end{table}

\[
\min \sum_{kmN} f_k^{ik} z_k + \sum_{kmN} f_{ik}^{0} y_{ik}^1 + \sum_{kmN} c_{km}^{2} z_{km}^{2} + \sum_{mN} \sum_{mN} \sum_{mN} c_{mj}^{3} y_{mj}^{3} + \sum_{iN} \sum_{iN} \sum_{j \neq j} c_{ij}^{1} y_{ij}^{0}
\]
OD demands to flow through the network, implying that to install hubs is not the most suitable assumption to make if a hub \( k \in N \) is installed if any \( i \rightarrow j \) OD demand \((i, j \in N : i \neq j)\) enters to or departs from it whenever \( k \) is not \( i \) or \( j \). Constraints (24)-(27) are arc activation constraints. Constraints (28) and (29) ensure that flows on starting and finishing access arcs can only happen if the hubs on the access arcs are installed; likewise constraints (30) and (31) guarantee that flows on starting and finishing arcs for a given \( i \rightarrow j \) OD demand \((i, j \in N : i \neq j)\) can only exist if \( i \) and \( j \) are not hubs themselves, respectively.

Whereas formulation (1)-(15) is more compact; model (18)-(31) renders a good balance between size and improved linear relaxation. Both formulation though can be used to solve the incomplete p-hub location problem by just incorporating the following constraint

\[
\sum_{k \in N} z_k = p
\]

where \( p \) is the number of hubs to be installed. Implicitly this equation assures that at most \((p + 1)\) hops (arcs) are present in any OD demand path. But, as already explained, resorting to such constraint prevents the OD demands from exploiting the benefits of scale economies—the key aspect of hub networks—since the number of transfer arcs that can be installed is indirectly restricted by the number of installed hubs. This greatly limits the available paths for the OD demands to flow through the network, implying that to install \( p \) hubs is not the most suitable assumption to make during the design of hub networks.
An alternative is to explicitly hop constrain each OD path and to unrestrict the number of hubs to be installed. While formulation (1)-(15) cannot directly accomplish that, formulation (18)-(27) requires that each flow variable \( h_{ijk}x_{ijk} \) and \( t_{ijm} \) is turned into a binary variable and that the following constraints are added to limit the number of hops on each OD path:

\[
\sum_{k \in N}^{N} h_{ijk} + \sum_{k \in N}^{N} \sum_{m \in N}^{N} x_{ijkm} + \sum_{m \in N}^{N} t_{ijm} \leq S \quad \forall i, j \in N : i \neq j
\]

where \( S \) is the number of allowed hops on a path. These additions greatly increase the solution complexity and deteriorate the linear relaxation of the formulation. Though both formulations can be used to address the hop-constrained incomplete hub location problem, nevertheless requiring great modifications, they are not the most suitable models to the task.

A model with the best possible balance between size and strength has an appealing motivation. The ideal model should deliver linear programming bounds stronger than formulations (1)-(15) and (18)-(27), but without the need to enumerate a large number of paths; it should also be flexible enough to handle several variants of the problem: fixed hub installation costs, \( p \)-hub type constraints and some measures of service, such as hop constraints. It is possible to design a formulation that it is tailor-made for the problem by employing a modeling technique known as Leontief substitution system (Veinott, 1968; Koehler et al., 1975). This technique allows for the development of a formulation which balances good linear relaxation bound and size, besides being amenable to decomposition methods. Furthermore, when simplified, i.e. when the hop constrained assumption is disregarded, the resulting formulation is similar, but smaller than formulation (18)-(27), and with a better linear relaxation bound.

### 2.1. A Leontief substitution system based formulation for the hop-constrained incomplete hub location problem

A (pre-)Leontief substitution system is a linear system \( Ax = b, x \geq 0 \), where \( b \) is a non-negative vector, \( A \) is a matrix containing at most one positive element in each column (Veinott, 1968; Koehler et al., 1975). When \( A \) represents a vertex-hyper-arc incidence matrix of a Leontief directed hyper-graph, i.e. a generalization of a directed graph in which arcs have multiple or no tails and at most one head (Coullard and Ng, 1995), then the matrix is unimodular and the vertices of the feasible region are all integer. By modeling the hyper-graph of an OD demand \( i - j \) in a hop constrained incomplete hub location problem as a discrete dynamic programming (Martin et al., 1990) or as Finite-State Machine, it is then possible to produce a subsystem with such property.

To clarify the concept of a Leontief hyper-graph, observe Figure 4. This example illustrates one possible hyper-graph with the available paths for an OD demand being hop constrained to \( S = 5 \). Black arrows are transfer arcs, while dashed blue links are starting and finishing access arcs. The red connection is the direct arc from \( i \) to \( j \). Bold triangles are installed hubs. Empty ones represent just transitional hubs between hops, i.e. the OD flow artificially moves from a hop to another, but remains at the same hub. All paths, but the one represented by the direct arc, are artificially constructed such that they have exactly \( S \) connections. Paths with fewer arcs are stretched by inserting artificial transitional points (empty triangles), i.e. the OD flow actually remains at the same hub, but moves from one lower level hop to the next level one. Arcs are constructed such that the number of hops in any path is exactly \( S \). For instance an actual path starting from node \( i \), passing through hubs \( k_1 \) and \( k_2 \), and finishing at node \( j \) or \( i - k_1 - k_2 - j \) can be represented as \( i - k_1 - k_2 - k_2' - k_1' - j \) on the hyper-graph, where \( k_2' \) is a transitional hub. Another example is the lowermost path in Figure 4. This paths visits all four hubs \( k_1, k_2, k_3, \) and \( k_4 \); while the path just below the direct arc (red arc) visits only one hub \( (k_1) \) and includes three transitional hubs and three null hops, which involves staying at the same node \( (k_1) \) in this case, having zero cost because \( c_{ik} = 0 \).

The characterization proposed by Martin et al. (1990) can be used to properly formulate the flow interactions in the hyper-graph of Figure 4 with the help of variables of Table 5. By modeling the flow recurrence and balances on each node to represent the changing stages or hops, a Leontief substitution system can be obtained with the variables \( \xi \). Then, following a similar reasoning to the one presented in the work of Gouveia (1998) for the minimum spanning and Steiner trees with hop constraints, these various systems, one for each OD demand, can be combined with variables of Table 2, i.e. with the hub and arcs infra-structure variables, to form a Leontief substitution system based formulation for the hop-constrained incomplete hub location problem. This mathematical program can be written as:
Figure 4: One of the possible hyper-graphs for the paths from origin $i$ to destination $j$ for $S = 5$.

| Decision Variables $\xi_{i,j,s} \geq 0$ | Description | |
|-----------------------------------------|-------------|
| Decides if a transition from $u$ to $v$ ($u, v \in N$) happens in position $s \in \{1..S\}$ for the $i-j$ pair ($i, j \in N : i \neq j$) on the Leontief hyper-graph. |

\[
\begin{align*}
\min & \quad \sum_{i \in N} \sum_{k \in N} f_{ik} + \sum_{i \in N} \sum_{j \in N} c_{ij}^0 \xi_{ij} + \sum_{i \in N} \sum_{k \in N} c_{ik}^1 \xi_{ik} + \sum_{k \in N} \sum_{m \in N} c_{km}^2 \xi_{km} + \sum_{m \in N} \sum_{j \in N} c_{mj}^3 \xi_{mj} \\
& \quad + \sum_{i \in N} \sum_{j \in N} \sum_{u \in N} w_{ij} \left( \sum_{k \in N} \sum_{v \in N} c_{k}^{ij} \xi_{iv} + \sum_{k \in N} \sum_{m \in N} \sum_{s \in \{2..(S-2)\}} c_{km}^{ij} \xi_{km} + \sum_{m \in N} \sum_{j \in N} c_{mj}^{ij} \xi_{mj} \right) \\
\text{s.t.:} & \quad \xi_{ij}^{1} = \sum_{v \in N} \xi_{iv}^{2} \quad \forall i, j \in N : i \neq j \\
& \quad \xi_{ij}^{2} = \sum_{u \in N} \xi_{ij}^{2} \quad \forall i, j \in N : i \neq j, u \neq i, u \neq j \\
& \quad \xi_{ij}^{3} = \xi_{ij}^{2} \quad \forall i, j \in N : i \neq j \\
& \quad \xi_{ij}^{4} = \sum_{v \in N} \xi_{ij}^{2} \quad \forall i, j \in N, s \in \{2..(S-2)\} : i \neq j \\
& \quad \sum_{v \in N} \xi_{ij}^{2} = \xi_{ij}^{2} + \xi_{ij}^{3} \quad \forall i, j \in N, s \in \{2..(S-2)\} : i \neq j, u \neq i, u \neq j \\
& \quad \sum_{v \in N} \xi_{ij}^{2} = \xi_{ij}^{2} \quad \forall i, j \in N, s \in \{2..(S-2)\} : i \neq j \\
\end{align*}
\]
The objective function (32) minimizes the total cost which consists of the fixed and variable costs. The first line of the objective function (32) has the installation costs for hubs, and the fixed and variable costs for direct arcs; while the next three terms in the second line are the setup costs for the starting access, the transfer, and the finishing access arcs, respectively. The last three terms are the variable costs of the path on the hyper-graph for each OD demand. Constraints (33)-(35) state the three possible types of transitions from origin \( i \) in position \( s = 1 \): from \( i \) to \( i \), if and only if \( i \) is a hub, then connecting this transition to any node \( v \) in position \( s = 2 \); from \( i \) to \( u (u \neq i \text{ and } u \neq j) \), \( u \) being a hub, then linking this transition to a given node \( v, v \neq i \text{ in position } s = 2 \); from \( i \) to destination \( j \), if and only if \( j \) is a hub, after which is only possible to reach \( j \).

Equations (36)-(38) are responsible for implementing a recursion from positions \( s = 2 \) to \( s = (S - 1) \) in the following manner: at a given node at any position within the specified range, it is possible to stay in that node or go to another location. The noticeable exceptions are: one can only leave the origin node \( i \) in a given position, but never enter it, and one can only enter the destination \( j \), but never leave it. Constraints (39)-(41) describe the end of feasible paths: from the origin node \( i \) in position \( s = (S - 1) \) one must reach the destination \( j \) in a single hop; at any other node except \( i \) and \( j \) in position \( s = (S - 1) \) then one must reach \( j \) in the next hop; from \( i \) in position \( s = (S - 1) \) then one must stay there and \( j \) is necessarily a hub. Finally, constraints (43)-(48) just provide the activation of the necessary infrastructure to render feasible paths.

As it is common in the hub location literature, formulation (32)-(48) prevents multiple hops on the access network so only a single non-null connection is allowed in a path before entering and after leaving the backbone network. However it is straightforward to adapt the aforementioned ideas to address cases in which more hops are permitted on the access network. This can be accomplished by just expanding the corresponding directed acyclic hyper-graph in Figure 4 to include the extra access hops. Moreover, as claimed by Gouveia (1998), the resulting model satisfies all the assumptions given by Martin (1987), i.e. given a feasible infra-structure of hubs and arcs, the resulting routing subsystem for a single OD demand is a pre-Leontief substitution system, and therefore all the extreme points of its associated polyhedron are integer-valued (Martin, 1987; Eppen and Martin, 1987; Gouveia, 1998).

Finally, a few remarks are in order: (i) for a single \( i - j \) pair, formulation (32)-(48) is actually a dynamic program as can be seen by the recurrence relations explicitly written as equations (33)-(42) (for further reference on how to write mathematical programs for dynamic programs refer to Martin et al. (1990)). (ii) As noted by Gouveia (1998), by tying all the Leontief substitution systems for the OD demands, the whole formulation is not capable producing only integer valued solutions for the infrastructure variables (refer to Table 2); nevertheless, it does provide very tight linear relaxation bounds for the problem. (iii) Parameter \( S \) controls the maximum number of hops on the network for the OD demands. By making \( S = |N| - 1 \) the resulting problem reduces to the incomplete hub location problem.
Further, it is possible to have $S$ individually indexed for each OD flow, i.e $S_{ij}$. (iv) Moreover, formulation (32)-(48) has a staircase matrix structure, being thus suited to be solved by decomposition techniques.

**Proposition 1.** Formulations (32)-(48), (1)-(15), and (18)-(31) are equivalent for the incomplete hub location hub location problem when $S = |N| - 1$.

**Proof.** To prove the equivalence one has just to find a linear transformation map from one system to the other. The following linear transformation equations allow construction of a one-to-one mapping between the flow variables of formulation (1)-(15) (Table 3) and the arc transition variables of formulation (32)-(48) (Table 5):

\[
Z_{ik} = \sum_{j \in N \atop j \neq i} w_{ij} \xi_{ik}^{(j)} \quad \forall i, k \in N : i \neq k
\]

\[
Y_{km} = \sum_{j \in N \atop j \neq k} \sum_{s=2}^{(S-1)} w_{js} \xi_{km}^{(j,s)} \quad \forall i, k, m \in N : k \neq m
\]

\[
X_{mi} = w_{im} \xi_{mi}^{(j)} \quad \forall i, m, j \in N : i \neq j
\]

Likewise, the following linear transformation equations map the flow variables of formulation (18)-(31) (Table 4) to the arc transition variables of formulation (32)-(48) (Table 5):

\[
h_{iu} = \xi_{iu}^{(j)} \quad \forall i, j, u \in N : i \neq j, u \neq i
\]

\[
x_{jmv} = \sum_{s=2}^{(S-1)} \xi_{jmv}^{(s)} \quad \forall i, j, u, v \in N : i \neq j, v \neq i, u \neq j, u \neq v
\]

\[
t_{iuj} = \xi_{iuj}^{(j)} \quad \forall i, j, u \in N : i \neq j, u \neq j
\]

These linear mappings allow for one integer valued feasible solution from one formulation to be translated to the variables of the other models showing then that the formulations are equivalent.

2.2. A tighter reformulation for the incomplete hub location problem

By carefully examining the linear transformations of Proposition 1 and using its mappings, it is possible to simplify formulation (32)-(48), for the case in which the hop constraints are disregarded, i.e. for the incomplete hub location problem, in order to get another model which can be seen as an improved version of formulation (18)-(31). The new formulation has not only fewer constraints, but it has stronger linear relaxation bounds than the model (18)-(31). The enhanced formulation can be written as:

\[
\min \sum_{k \in N} f_k z_k + \sum_{r \in N \atop r \neq i} \sum_{j \in N \atop j \neq i} \tilde{c}_{ij}^0 \tilde{y}_{ij} + \sum_{i \in N} \sum_{k \in N \atop k \neq i} \tilde{c}_{ik}^1 \tilde{y}_{ik} + \sum_{k \in N \atop k \neq m} \sum_{m \in N} \tilde{c}_{km}^2 \tilde{y}_{km} + \sum_{m \in N \atop m \neq j} \sum_{j \in N \atop j \neq i} \tilde{c}_{mj}^3 \tilde{y}_{mj}
\]

\[
+ \sum_{r \in N \atop r \neq i} \sum_{j \in N \atop j \neq i} w_{ij} \left( \sum_{k \in N \atop k \neq i} \tilde{c}_{ik} h_{ijk} + \sum_{j \in N \atop j \neq k} \sum_{k \in N \atop k \neq i} \tilde{c}_{km} x_{jkm} + \sum_{m \in N \atop m \neq j} \tilde{c}_{mj} t_{jum} \right) \quad (49)
\]

\[
s.t. : \sum_{m \in N \atop m \neq j} t_{jum} + \sum_{k \in N \atop k \neq j} \tilde{z}_{jk} + \tilde{h}_{ij} + \tilde{y}_{ij} = 1 \quad \forall i, j \in N : i \neq j \quad (50)
\]
\[ h_{jm} + \sum_{k \in N, k \neq j} x_{jkm} = \sum_{k \in N, k \neq m} x_{jmk} + t_{jm} \quad \forall i, j, m \in N : i \neq j, i \neq m, j \neq m \]  
(51)

\[ t_{ji} + \sum_{m \in N, m \neq i} x_{ijm} = z_i \quad \forall i, j \in N : i \neq j \]  
(52)

\[ h_{jk} + \sum_{m \in N, m \neq k} x_{ijm} \leq z_k \quad \forall i, j, k \in N : i \neq j, k \neq i, k \neq j \]  
(53)

\[ h_{ij} + \sum_{k \in N, k \neq j} x_{jkm} = z_j \quad \forall i \in N : i \neq j \]  
(54)

\[ h_{jk} \leq y^j_k \quad \forall i, j, k \in N : i \neq j, k \neq i \]  
(55)

\[ x_{jkm} \leq \frac{1}{2} \quad \forall i, j, k, m \in N : i \neq j, j \neq m, i \neq k, m \neq j \]  
(56)

\[ t_{jm} \leq y^j_m \quad \forall i, j, k \in N : i \neq j, m \neq j \]  
(57)

The objective function (49) minimizes the total cost consisted of the installation and variable costs. Constraints (50) indicate that the flow for an OD pair \( i - j \) either travels on a direct arc, or a starting access arc, or a finishing access arc or goes through a transfer arc from some other node \( m \). Constraints (51) are the divergence equation for node \( m \), while constraints (52) - (54) establish the hub nodes appropriately. Constraints (55) - (57) link the flow variables to the arc variables. Formulation (49)-(57) could only be achieved by the simplification of the Leontief substitution system based formulation (32)-(48).

**Proposition 2.** Formulation (49)-(57) is stronger than formulation (18)-(31).

**Proof.** Let \( P_1 = \{(z, y^0, y^1, y^2, y^3) : (49)-(57)\} \) and \( P_2 = \{(z, y^0, y^1, y^2, y^3, h, t, x) : (18) - (31)\} \) be the continuous feasible region for each formulation, respectively. To prove that formulation (49)-(57) is stronger than formulation (18)-(31), i.e. to prove that \( P_1 \subseteq P_2 \), one can show that a point \((z, y^0, y^1, y^2, y^3, h, t, x) \in P_2 \setminus P_1 \).

Suppose that each integer infra-structure variable \( z = y^0 = y^1 = y^2 = y^3 = 0.5 \) and that there is at least an integer number \( \Pi \geq 3 \) of paths that allows for an OD demand \( i - j \) to be flown through. Such configuration allows for variables \((h, t, x) \) of formulation (49)-(57) leaving node \( i \), for instance, to assume the value \( 1/\Pi \) by observing constraints (19) and (24)-(27), i.e. \( (1/\Pi) < 0.5 \). However, by observing the equality constraints (50), (52) and (54) of \( P_1 \) and the presence of the integer variables \( y^0 \) and \( z \) on them, this solution would not be possible. After replacing equation of (54) into (50) for the OD pair \( i - j \), one gets

\[ \sum_{m \in N, m \neq j} t_{jm} + z_j + y^0_{ij} = 1, \]

which sets all variables \( t \) to 0, since \( z_j = y^0_{ij} = 0.5 \). In other words, the aforementioned solution does not belong to \( P_1 \), thus proving that formulation (49)-(57) is stronger than formulation (18)-(31).

The same reasoning used for Proposition 2 can be employed to show that formulation (32)-(48) is stronger than formulation (18)-(31) augmented by the hop constraints and integer requirements for variables \((h, t, x)\). For sake of presentation, Table 6 shows the number of integer and continuous variables, and constraints as a function of \( n = |N| \) for each formulation for \( p \)-hub and hop constrained variants. Besides being a tighter formulation, model (49)-(57) has fewer continuous variables \((n^2)\) and constraints \((n^3 + n)\) than formulation (18)-(31). Though model (32)-(48) is more suitable to the hop constrained problem, it can still be used to the \( p \)-hub variant. It is important to highlight that though traditional incomplete hub location problems have the same base formulations, the problems here addressed have three additional different integer sets of variables which greatly increase the combinatorial nature and the computational burden to achieve the optimal solution. To overcome this extra overhead, decomposition strategies that exploit the formulation features and mathematical structure are a key finding for successfully solving
large instances. Therefore, given the properties of the new proposed formulations (32)-(48) and (49)-(57), two specialized Benders decomposition algorithms, one for each model, is used to solve the incomplete hub location with and without hop constraints, respectively.

The Benders decomposition method (Benders, 1962) is a partitioning technique designed to handle large-scale systems having the so called complicating variables and a constraint matrix with a staircase structure. Normally, for problems with these features, it is possible to fix the values for the complicating variables to obtain a linear subsystem that can be decomposable into smaller ones, which are usually much easier to be solved. Whenever these conditions are met, a Benders decomposition algorithm can be designed to project these smaller subsystems out, replacing them by a large number of constraints, possibly exponential in size, involving only the complicating variables. These constraints are known as Benders cuts (BC). The resulting mathematical program, called Benders master problem (MP), is equivalent to the original formulation, yielding the same optimal solution.

As most of the BCs will not be active for an optimal solution, a relaxation policy, in which all but a few of BCs are disregarded, can be implemented through an iterating procedure that generates them on a needed basis. At each iteration, the BCs can be separated by solving the dual subproblem (SP) of the projected subsystems with the complicating variables temporally fixed by the relaxed Benders master problem. The produced BCs are later added to the MP, guiding it to propose new solutions for the complicating variables at the next iteration. Whenever the resolution of dual SP results into an unbounded solution, then a valid ray is found to generate a feasibility BC; otherwise, an optimality BC is separated. At any iteration, the method readily renders a lower bound (LB) to the original problem, since the relaxed MP is a relaxation of the original problem. Likewise, an upper bound (UB) can be easily constructed by combining part of objective function of the MP with the resulting value of bounded SPs. The method stops when the LB and the UB converge.

Since its introduction by Benders (1962), the Benders decomposition method has found many successful applications related to location, transportation and logistics systems design: Geoffrion and Graves (1974), França and Luna (1982), Magnanti and Wong (1981, 1984), Magnanti et al. (1986), Cordeau et al. (2001), Costa (2005), Mercier et al. (2005), Papadakos (2008, 2009), and Contreras et al. (2011b,c,a). Further, during the last four decades, several research teams are responsible for providing improvements for the original algorithm, including the works of Geoffrion (1972) to extend the technique to mixed-integer nonlinear programs; McDaniel and Devine (1977) to propose an efficient warm-start phase, Magnanti and Wong (1981) to introduce the concepts of Pareto-optimal cuts, Papadakos (2008) to facilitate the generation of Pareto-Optimal cuts, Mercier (2008), Fischetti et al. (2010), de Sá et al. (2013) to propose efficient ways to better exploit feasibility BCs.

In the present work, besides presenting two efficient Benders decomposition algorithms to solve the hop-constrained incomplete hub location problem with and without hop constraints, the method of de Sá et al. (2013) to separate feasibility BCs is further improved. This enhancement, which consists of embedding a counter intuitive normalization constraint on the dual SPs when generating feasibility BCs, is general and not problem dependent. This simple add-on greatly speeds up the convergence of the method. Further, for sake of simplicity in the presentation, as the complicating variables \( (z, y^1, y^2, y^3) \) are the same for both variants of the problem being addressed, only one description for the MPs is here presented.

### Table 6: Formulations size comparison.

<table>
<thead>
<tr>
<th>prob. type</th>
<th>formulation</th>
<th># integer vars.</th>
<th># continuous vars.</th>
<th># constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>p-hub</td>
<td>(1)-(15)</td>
<td>4n^2 + n</td>
<td>2n^2 + n^2</td>
<td>13n^2 + n + 1</td>
</tr>
<tr>
<td></td>
<td>(18)-(31)</td>
<td>4n^2 + n</td>
<td>n^2 + 2n + n^2</td>
<td>n^4 + 10n^2 + 3n^2 + n + 1</td>
</tr>
<tr>
<td></td>
<td>(32)-(48)</td>
<td>4n^2 + n</td>
<td>n^2 + n^2</td>
<td>3n^2 + 3n^2 + 7n^2 + n^2 + 2n^2p + 1</td>
</tr>
<tr>
<td></td>
<td>(49)-(57)</td>
<td>4n^2 + n</td>
<td>n^2 + 2n^2</td>
<td>n^4 + 9n^2 + 5n^2 + n + 1</td>
</tr>
<tr>
<td>hop constrained</td>
<td>(1)-(15)</td>
<td>n^4 + 2n^2 + 5n^2 + n</td>
<td>-</td>
<td>n^4 + 10n^2 + 4n^2 + n</td>
</tr>
<tr>
<td></td>
<td>(18)-(31)</td>
<td>n^4 + 2n^2 + 5n^2 + n</td>
<td>-</td>
<td>n^4 + 10n^2 + 4n^2 + n</td>
</tr>
<tr>
<td></td>
<td>(32)-(48)</td>
<td>4n^2 + n</td>
<td>n^2S</td>
<td>3n^2 + 3n^2 + 7n^2 + n^2S + 2n^2S</td>
</tr>
<tr>
<td></td>
<td>(49)-(57)</td>
<td>n^4 + 2n^2 + 4n^2 + n</td>
<td>-</td>
<td>n^4 + 9n^2 + 4n^2</td>
</tr>
</tbody>
</table>

### 3. Two specialized Benders decomposition algorithms

The Benders decomposition method is a partitioning technique designed to handle large-scale systems having the so called complicating variables and a constraint matrix with a staircase structure. Normally, for problems with these features, it is possible to fix the values for the complicating variables to obtain a linear subsystem that can be decomposable into smaller ones, which are usually much easier to be solved. Whenever these conditions are met, a Benders decomposition algorithm can be designed to project these smaller subsystems out, replacing them by a large number of constraints, possibly exponential in size, involving only the complicating variables. These constraints are known as Benders cuts (BC). The resulting mathematical program, called Benders master problem (MP), is equivalent to the original formulation, yielding the same optimal solution.

As most of the BCs will not be active for an optimal solution, a relaxation policy, in which all but a few of BCs are disregarded, can be implemented through an iterating procedure that generates them on a needed basis. At each iteration, the BCs can be separated by solving the dual subproblem (SP) of the projected subsystems with the complicating variables temporally fixed by the relaxed Benders master problem. The produced BCs are later added to the MP, guiding it to propose new solutions for the complicating variables at the next iteration. Whenever the resolution of dual SP results into an unbounded solution, then a valid ray is found to generate a feasibility BC; otherwise, an optimality BC is separated. At any iteration, the method readily renders a lower bound (LB) to the original problem, since the relaxed MP is a relaxation of the original problem. Likewise, an upper bound (UB) can be easily constructed by combining part of objective function of the MP with the resulting value of bounded SPs. The method stops when the LB and the UB converge.

Since its introduction by Benders (1962), the Benders decomposition method has found many successful applications related to location, transportation and logistics systems design: Geoffrion and Graves (1974), França and Luna (1982), Magnanti and Wong (1981, 1984), Magnanti et al. (1986), Cordeau et al. (2001), Costa (2005), Mercier et al. (2005), Papadakos (2008, 2009), and Contreras et al. (2011b,c,a). Further, during the last four decades, several research teams are responsible for providing improvements for the original algorithm, including the works of Geoffrion (1972) to extend the technique to mixed-integer nonlinear programs; McDaniel and Devine (1977) to propose an efficient warm-start phase, Magnanti and Wong (1981) to introduce the concepts of Pareto-optimal cuts, Papadakos (2008) to facilitate the generation of Pareto-Optimal cuts, Mercier (2008), Fischetti et al. (2010), de Sá et al. (2013) to propose efficient ways to better exploit feasibility BCs.

In the present work, besides presenting two efficient Benders decomposition algorithms to solve the hop-constrained incomplete hub location problem with and without hop constraints, the method of de Sá et al. (2013) to separate feasibility BCs is further improved. This enhancement, which consists of embedding a counter intuitive normalization constraint on the dual SPs when generating feasibility BCs, is general and not problem dependent. This simple add-on greatly speeds up the convergence of the method. Further, for sake of simplicity in the presentation, as the complicating variables \( (z, y^1, y^2, y^3) \) are the same for both variants of the problem being addressed, only one description for the MPs is here presented.
3.1. A Benders decomposition for the hop-constrained incomplete hub location problem

For fixed values for the complicating variables, e.g. \((z, y^0, y^1, y^2, y^3) = (\bar{z}, \bar{y}^0, \bar{y}^1, \bar{y}^2, \bar{y}^3)\), an implied dual SP can be obtained after the dual variables of Table 7 are associated to their respective constraints:

Table 7: Notation for the dual Benders SP for the hop-constrained incomplete hub location problem.

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu_{ij}^0) (\in\mathbb{R})</td>
<td>Dual variables associated with constraints (33)-(42), (\forall u, i, j \in N, s \in {1...S} : i, j \in N : i \neq j).</td>
</tr>
<tr>
<td>(\rho_{uv}^{1j} \geq 0)</td>
<td>Dual variables associated with constraints (43), (\forall u, v, i, j \in N : u \neq v, i \neq j).</td>
</tr>
<tr>
<td>(\rho_{uv}^{2j} \geq 0)</td>
<td>Dual variables associated with constraints (44), (\forall u, v, i, j \in N : u \neq v, i \neq j).</td>
</tr>
<tr>
<td>(\rho_{uv}^{3j} \geq 0)</td>
<td>Dual variables associated with constraints (45), (\forall u, v, i, j \in N : u \neq v, i \neq j).</td>
</tr>
<tr>
<td>(\bar{v}_i \geq 0)</td>
<td>Dual variables associated with constraints (47), (\forall u, i, j \in N : i \neq j, u \neq i, u \neq j).</td>
</tr>
<tr>
<td>(\bar{v}_j \in \mathbb{R})</td>
<td>Dual variables associated with constraints (48), (\forall j \in N : i \neq j).</td>
</tr>
</tbody>
</table>

\[
\max \sum_{u \in N} \sum_{j \in S} \left(1 - y^0_{ij} \mu_{ij}^0 - \bar{z}_i \rho_{ij}^0 - \sum_{u \in N} \sum_{v \neq u} \left(\bar{v}_u \rho_{uv}^{1j} - \bar{v}_u \rho_{uv}^{2j} - \bar{v}_u \rho_{uv}^{3j}\right)\right)
\]

s.t. : \(\mu_{ij}^0 - \rho_{ij}^0 \leq 0\) \(\forall i, j \in N : i \neq j\)

\[
\mu_{uv}^{1j} - \rho_{uv}^{1j} - \rho_{uv}^{1ij} \leq w_{ij}^0 c_{uv}^j
\]

\(\forall i, j \in N : i \neq j, u \neq j, u \neq i\)

\[
\mu_{uv}^{2j} - \rho_{uv}^{2j} \leq w_{ij} \rho_{ij}^j
\]

\(\forall i, j \in N : i \neq j\)

\[
\mu_{uv}^{2ij} - \rho_{uv}^{2ij} \leq w_{ij} \rho_{ij}^j
\]

\(\forall i, j \in N : i \neq j, u \neq j, v \neq i, v \neq j\)

\[
\mu_{uv}^{3j} - \rho_{uv}^{3j} - \rho_{uv}^{3ij} \leq w_{ij}^0 \rho_{ij}^j
\]

\(\forall i, j \in N : i \neq j, u \neq j, u \neq i\)

\[
\mu_{uv}^{3ij} - \rho_{uv}^{3ij} \leq 0
\]

\(\forall i, j \in N : i \neq j\)

\[
\mu_{uv}^{3ij} - \rho_{uv}^{3ij} \leq 0
\]

\(\forall i, j \in N : i \neq j, s \in \{2,...,(S-1)\} : i \neq j, u \neq j, v \neq i, v \neq j\)

At any iteration \(h = 1, \ldots, H\) of the algorithm, in which a bounded dual SP is found, an optimality BCs can be obtained from the objective function of the dual SP:

\[
\psi_{ij} \geq (1 - y^0_{ij}) \mu_{ij}^{0j} + \sum_{u \in N} \sum_{v \neq u} \left(\bar{v}_u \rho_{uv}^{1j} - \bar{v}_u \rho_{uv}^{2j} - \bar{v}_u \rho_{uv}^{3j}\right) \forall i, j \in N : i \neq j
\]

(58)

where \(h = 1, \ldots, H\), and the vector \((\bar{p}^0, \bar{p}^1, \bar{p}^2, \bar{p}^3)\) is the optimal dual values associated with iteration \(h\); and \(\psi_{ij}\) is a non-negative variable for estimating the variable cost for transporting the flow for the \(i-j\) OD pair. For the iterations \(r = 1, \ldots, R\), in which unbounded dual SPs are found, a feasibility BC can be separated from the ray direction or:

\[
0 \geq (1 - y^0_{ij}) \mu_{ij}^{0jr} - \sum_{u \in N} \sum_{v \neq u} \left(\bar{v}_u \rho_{uv}^{1jr} - \bar{v}_u \rho_{uv}^{2jr} - \bar{v}_u \rho_{uv}^{3jr}\right) \forall i, j \in N : i \neq j
\]

(59)

where \(r = 1, \ldots, R\), and the vector \((\bar{p}^r, \bar{p}^1, \bar{p}^2, \bar{p}^3)\) is the ray direction values associated with iteration \(r\).
3.2. A Benders decomposition for the incomplete hub location problem

Likewise, for fixed values for the complicating variables, e.g. \((z, y^0, y^1, y^2, y^3) = (\bar{z}, \bar{y}^0, \bar{y}^1, \bar{y}^2, \bar{y}^3)\), the implied dual SP can be written as the following mathematical program after the dual variables of Table 8 are associated to their respective constraints:

Table 8: Notation for the dual form of Benders subproblem – improved heads and tails model.

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta_i^j \in \mathbb{R})</td>
<td>Dual variables associated with constraints (50), (\forall i, j \in N : i \neq j).</td>
</tr>
<tr>
<td>(\gamma_i^j \in \mathbb{R})</td>
<td>Dual variables associated with constraints (51), (\forall i, j \in N : i \neq j, k \neq j, k \neq i).</td>
</tr>
<tr>
<td>(\delta_i^j \in \mathbb{R})</td>
<td>Dual variables associated with constraints (52), (\forall i, j \in N : i \neq j).</td>
</tr>
<tr>
<td>(\varphi_i^j \geq 0)</td>
<td>Dual variables associated with constraints (53), (\forall i, j \in N : i \neq j, k \neq j, k \neq i).</td>
</tr>
<tr>
<td>(\beta_i^j \in \mathbb{R})</td>
<td>Dual variables associated with constraints (54), (\forall i \in N : i \neq j).</td>
</tr>
<tr>
<td>(\psi_i^j \geq 0)</td>
<td>Dual variables associated with constraints (55), (\forall i, j \in N : i \neq j, k \neq i).</td>
</tr>
<tr>
<td>(\tau_i^j \geq 0)</td>
<td>Dual variables associated with constraints (56), (\forall i, j, k, m \in N : i \neq j, m \neq i, i \neq j, k, k \neq m).</td>
</tr>
<tr>
<td>(\varsigma_i^j \geq 0)</td>
<td>Dual variables associated with constraints (57), (\forall i \neq j, k \neq j).</td>
</tr>
</tbody>
</table>

max \[\sum_{i \in N} \sum_{j \in N} \left(1 - y^0_i j \theta_i^j - \bar{z}_i \delta_i^j - \bar{y}_i \beta_i^j - \sum_{k \in N} \left( \bar{z}_k \varphi_k^j - \bar{y}_k \beta_k^j \right) - \sum_{k \in N} \sum_{m \in N} \left( \bar{z}_m \psi_m^j - \bar{y}_m \beta_m^j \right) - \sum_{m \in N} \sum_{k \in N} \left( \bar{z}_m \phi_m^j - \bar{y}_m \beta_m^j \right) \right) \]

s.t.: \[\theta_i^j - \beta_i^j - \xi_i^j \leq w_i \bar{z}_i^j \] \(\forall i, j \in N : i \neq j\)

\[\gamma_k^j - \varphi_k^j - \xi_k^j \leq w_i \bar{z}_i^k \] \(\forall i, j, k \in N : i \neq j, k \neq i, k \neq j\)

\[\gamma_j^k - \delta_j^k - \psi_j^k \leq w_i \bar{z}_i^j \] \(\forall i, j, k \in N : i \neq j, k \neq i, k \neq j\)

\[\gamma_m^j - \varphi_m^j \leq \bar{y}_i \bar{z}_i^m \] \(\forall i, j, m \in N : i \neq j, k \neq i, k \neq j\)

\[\theta_i^j - \gamma_i^j - \beta_i^j - \bar{y}_i \bar{z}_i^j \leq w_i \bar{z}_i^j \] \(\forall i, j \in N : i \neq j\)

\[\theta_i^j - \delta_i^j - \xi_i^j \leq w_i \bar{z}_i^j \] \(\forall i, j \in N : i \neq j\)

\[\theta_i^j - \varphi_i^j \leq w_i \bar{z}_i^j \] \(\forall i, j \in N : i \neq j, m \neq i, m \neq j\)

Once again, at any iteration \(h = 1, \ldots, H\) of the algorithm, in which a bounded dual SP is found, an optimality BCs can be obtained from the objective function of the dual SP:

\[\psi_{ij} \geq (1 - y^0_{ij}) \theta_{ij} - z_i \delta_{ij} - z_j \beta_{ij} - \sum_{k \in N} \left( \bar{z}_k \varphi_k^j - \bar{y}_k \beta_k^j \right) - \sum_{m \in N} \sum_{k \in N} \left( \bar{z}_m \psi_m^j - \bar{y}_m \beta_m^j \right) - \sum_{m \in N} \sum_{k \in N} \left( \bar{z}_m \phi_m^j - \bar{y}_m \beta_m^j \right) \] \(\forall i, j \in N : i \neq j\)

where \(h = 1, \ldots, H\) and the vector \((\bar{\theta}, \bar{\delta}, \bar{\psi}, \bar{\varphi}, \bar{\beta}, \bar{\phi})\) is the optimal dual values associated with iteration \(h\); and \(\psi_{ij}\) is a non-negative variable for estimating the variable cost for transporting the flow for the \(i - j\) OD pair. For the iterations \(r = 1, \ldots, R\), in which unbounded dual SPs are found, a feasibility BC can be separated from the ray direction or:

\[0 \geq (1 - y^0_{ij}) \theta_{ij} - z_i \delta_{ij} - z_j \beta_{ij} - \sum_{k \in N} \left( \bar{z}_k \varphi_k^j - \bar{y}_k \beta_k^j \right) - \sum_{m \in N} \sum_{k \in N} \left( \bar{z}_m \psi_m^j - \bar{y}_m \beta_m^j \right) - \sum_{m \in N} \sum_{k \in N} \left( \bar{z}_m \phi_m^j - \bar{y}_m \beta_m^j \right) \] \(\forall i, j \in N : i \neq j\)

where \(r = 1, \ldots, R\), and the vector \((\bar{\theta}, \bar{\delta}, \bar{\beta}, \bar{\varphi}, \bar{\psi}, \bar{\phi}, \bar{\psi'})\) is the ray direction values associated with iteration \(r\).
3.3. The Benders master problem

The Benders MP is responsible for fixing the values for the complicating variables, i.e. the MP installs the infrastructure of the network. It has all the complicating variables with some additional constraints involving only these variables. The role of such redundant constraints is to reduce as much as possible the number of iterations having unbounded SPs. They only allow for arc installation if the proper hubs are installed or not. These additional constraints usually become inactive as more BCs are iteratively added to the MP. However, they are useful in the first cycles to ensure a better practical behavior. Further, variables \( \psi_{ij} \geq 0 \) for all \( \forall i, j \in N \) are used to sub-estimate the variable cost for \( i - j \) OD demand on the MP, which can then be written as:

\[
\begin{align*}
\min & \sum_{i \in N} \sum_{j \in N} \psi_{ij} + \sum_{k \in N} f_k \cdot z_k + \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot y_{ij} + \sum_{i \in N} \sum_{k \in N} c_{ik} \cdot y_{ik} + \sum_{k \in N} \sum_{m \in N} c_{km} \cdot y_{km} + \sum_{j \in N} \sum_{m \in N} c_{mj} \cdot y_{mj} \\
\text{s.t.:} & (58) - (59) \text{ or } (60) - (61)
\end{align*}
\]

Both formulations (32)-(48) and (49)-(57) share the same MP with the difference of having different Benders cuts, constraints (58)-(59) or (60)-(61), depending on which formulation is currently being solved by the Benders decomposition technique, which is illustrated at Algorithm 1. Recall that \( UB \) and \( LB \) stand for current upper and lower bounds, the iteration counter, and the solution for the dual SP of iteration \( h \), respectively. The algorithm iterates between the solution of the MP and the SP. Optimality and feasibility BCs are added to the MP depending if the dual SPs are bounded or not, respectively. The procedure stops when the \( UB \) and \( LB \) converge. Note that the \( UB \) can only be updated when the dual SPs are bounded.

Algorithm 1 Basic Benders decomposition

\[
\begin{align*}
UB & \leftarrow +\infty, \quad LB \leftarrow 0, \quad h \leftarrow 1 \\
\text{while} \ (UB \neq LB) \ \text{do} & \\
\quad LB & \leftarrow \text{SOLVE MP} \\
\quad S^P & \leftarrow \text{SOLVE SP} \\
\quad \text{if} \ (S^P < \infty) \ \text{then} & \\
\quad \quad \text{ADD} \ \text{optimality BCs to MP} \\
\quad \quad UB & \leftarrow \min(UB, S^P + (LB - \sum \psi)) \\
\quad \text{else} & \\
\quad \quad \text{ADD} \ \text{feasibility BCs to MP} \\
\quad \text{end if} & \\
h & \leftarrow h + 1 \\
\text{end while} &
\end{align*}
\]

3.4. Improving the performance of the Benders decomposition algorithms

In addition to the two new formulations and their dominance relations to the other known formulations (1)-(15) and (18)-(31), this paper also provides a new contribution to the methodological aspects of the Benders decomposition method. More specifically, an improved BC selection scheme is proposed for the iterations in which the MP
renders unbounded SPs, i.e. for generating feasibility BCs. Note that, for the problems addressed in this work, it is not straightforward to ensure that the MP will always generate feasible topologies (bounded SPs) at each iteration. Actually as there are no specific topology been sought, e.g. a tree or a ring for the backbone network, an excessive number of feasibility cuts is expected to be added to the MP prior to the convergence to the optimal solution by the Benders decomposition method.

When some specific type of topology is being addressed, like in the work of de Sá et al. (2013), it is sometimes possible to take advantage of the problem structure and to devise a small set of additional variables and/or constraints to enforce such network design in the MP. However in the present case, as the formulation can render all kinds of topologies for the system being designed (the network topology emerges from the cost matrices), such variables and/or constraints are not readily available. One may be naively tempted to use formulation (1)-(15) embedded into the MP to avoid having feasibility BCs, not knowing of the undesired results of having a larger, burdensome MP. An alternative way of handling these iterations with unbounded SPs is to select stronger BCs at these cycles to ensure a better performance of the Benders method when tackling large scale instances. Two good starting points are the Pareto-Optimal optimality generation subproblem of Papadakos (2008) and the cut selection procedure introduced by de Sá et al. (2013). Prior to presenting them and the new generation scheme for feasibility BCs, some general notation is introduced.

For sake of completeness and as these schemes are here adapted to two different formulations, they are presented in a general matrix form since they are not problem specific. Let a mathematical programming problem have a subset of its decision variables considered to be complicating, e.g. y, and another subset considered to be easy, e.g. x. Further, let its whole matrix system have a stair-case structure. Given these assumptions, let this mathematical program be written as \[ \min \{ f^T y + c^T x : F y + A x \geq b, y \in Y, x \geq 0 \}, \] without loss of generality, where f, c, and b, and A and F are column vectors and matrices, respectively, of appropriate sizes. Y is a set of feasible points involving constraints having only the y variable. To project the easy variables x out, one first obtains the following primal SP, for fixed values of the complicating variables \( y = \bar{y} \), \[ \min \{ c^T x : A x \geq b - F \bar{y}, x \geq 0 \}. \] Dualizing this problem by associating the dual variables \( u \geq 0 \) to the constraints yields the dual SP \[ \max \{ (b - F \bar{y})^T u : A^T u \leq c, u \geq 0 \}. \]

3.4.1. Pareto-optimal cuts

At some iteration, the dual SP can be bounded or unbounded, depending on the values of \( \bar{y} \). In the former case, the dual SP can be degenerate having multiple optimal solutions. This is very common when addressing formulations with embedded flow network problems, like in the present work. Whenever degeneracy occurs, several alternative optimality BCs can be generated having different strengths. As shown by Magnanti and Wong (1981), a relationship of dominance among these cuts can be established, if a reference point is used to assess these strengths. This reference point is known as a Magnanti-Wong point (\( y^{MW} \)), and it is defined as:

**Definition 1.** (i) If a point \( y^{MW} \) is not one of the points that spans the set of all possible projections of the original problem into the variables of the Benders MP; (ii) if the primal Benders subproblem is feasible for \( y^{MW} \); (iii) and if the dual feasible solution set is nonempty, then \( y^{MW} \) is a Magnanti-Wong point.

When a given cut is not dominated by other cuts in reference to a Magnanti-Wong point, then this cut is said to be a Pareto-optimal cut. To render such Pareto-optimal cuts, Papadakos (2008) states that one can judiciously modify the search direction of the objective function of the dual SP at each iteration, i.e. one can alter the direction of the cut, by replacing the usually integer-valued \( \bar{y} \) by a Magnanti-Wong point \( y^{MW} \) or \[ \max \{ (b - F y^{MW})^T u : A^T u \leq c, u \geq 0 \}. \] Though not trivial to find a valid \( y^{MW} \), but provided that one has an initial Magnanti-Wong point obeying Definition 1, a sequence of Magnanti-Wong points can be produced by linearly combining the \( y^{MW} \) an iteration \( h \) with the current Benders MP \( y^h \) solution in the form \( y^{MW} = (1 - \lambda) y^{MW} + \lambda y^h \), where \( 0 < \lambda < 1 \), to obtain a new valid Magnanti-Wong point. Papadakos (2008) empirically demonstrates that the most effective value for \( \lambda \) is 1/2, if all properties stated in the definition are respected for the initial \( y^{MW} \). An initial \( y^{MW} \) can be found by linearly combining \( |N| + 1 \) feasible solutions for the problem.

3.4.2. Getting optimality cuts on iterations with unbounded SPs

Unfortunately, the Papadakos (2008)’s procedure cannot be applied when a given MP solution \( y^h \) renders unbounded dual SPs. In such cases, is not possible to obtain a new Magnanti-Wong point using \( \lambda = 1/2 \) and no new
Pareto-optimal cut can be separated. To overcome this difficulty, de Sá et al. (2013) propose an auxiliary SP that searches for the largest $\lambda$ possible, but respecting all of the properties of Definition 1:

$$\max \lambda$$

s.t.:

$$Ax \geq b - F((1 - \lambda)y^{MW} + \lambda y^h)$$

$$0 \leq \lambda \leq 1/2$$

This additional subproblem computes a valid value for $\lambda$ which allows for the Magnanti-Wong point $y^{MW}$ to be updated and a Pareto-Optimal cut to be generated. However, a feasibility BCs still needs to be separated.

### 3.4.3. A new normalization constraint to improve feasibility BCs

As there are no practical alternatives to assess the performance of feasibility BCs, since there is a lack of dominance information regarding alternative cuts because of the nonexistence of natural costs to function as a metric, it is not possible to establish which cuts are dominant. The only assessment possible about alternative feasibility cuts is to determine if they are associated with extreme directions of the Benders subproblem dual polyhedron or if they are just combinations of such extremal entities.

Furthermore, it is not easy to collect extreme rays from standard optimization packages, like IBM CPLEX, since they are not designed to provide those directions. The majority of commercial solvers usually render only unbounded directions, not necessarily extremal ones. To circumvent these limitations, Mercier (2008) proves that a given direction is extremal by establishing a one-to-one correspondence between the extreme rays of the Benders dual SP polyhedron and the extreme points of an alternative polyhedron obtained by bounding the Benders dual SP. To obtain such alternative polyhedron, an error variable to measure the system’s infeasibility is inserted into the Benders primal SP and then minimized:

$$\min e$$

s.t.:

$$Ax + e \geq b - F y^h$$

$$e \geq 0$$

This extra error variable renders a normalization constraint, responsible for determining bounded solutions in the dual Benders SP or:

$$\max (b - F y^h)^T u$$

s.t.:

$$A^T u \leq 0$$

$$\|u\| \leq 1$$

$$u \geq 0$$

Several authors (Fortz and Poss, 2009; Fischetti et al., 2010; Mercier, 2008) experiment with variations for the normalization constraint. The achieved results show that some versions are more adequate than others depending on the problem being addressed. However no definite conclusion has been achieved in what appears to be an ongoing research effort. In this work, a new normalization constraint is proposed by following the guidelines of Mercier (2008) combined with Proposition 3 and the subproblem of §3.4.2. This new approach is capable of generating better feasibility BCs.

**Proposition 3.** When the Benders primal SP is infeasible (i.e. the dual SP is unbounded), determining the maximum $\lambda$ of the SP of §3.4.2 is equivalent to calculating the minimal error, like done in the primal SP $\min\{e : Ax + e \geq b - F y^h, e \geq 0\}$. Further, finding this maximum $\lambda$ provides dual information about unbounded extremal rays on the dual system.

**Proof.** By dropping constraint $0 \leq \lambda \leq 1/2$, one obtain the following SP $\max\{\lambda : Ax \geq b - F((1 - \lambda)y^{MW} + \lambda y^h), \lambda \geq 0\}$. As $\lambda$, by definition, is restricted to $0 \leq \lambda \leq 1$ (recall that $\lambda$ is the weight used in a linear combination of two points), it is possible to do a variable transformation $\lambda = 1 - e$, where $e$ is defined as a non-negative error. $\lambda$ can
be seen as the minimum value in which the system will render feasible; while $e$ is the maximum value in which the system will yield infeasible. After some simple manipulations, the SP of §3.4.2 becomes:

$$\begin{align*}
\min e \\
\text{s.t.}: & Ax + F(y^{MW} - y^h)e \geq b - Fy^h \\
& e \geq 0
\end{align*}$$

which in dual form can be written as:

$$\begin{align*}
\max (b - Fy^h)^Tu \\
\text{s.t.}: & A^Tu \leq 0 \\
& (F(y^{MW} - y^h))^Tu \leq 1 \\
& u \geq 0
\end{align*}$$

Note that the above problem is in the same form as the dual SP of Mercier (2008) or max\{$(b - Fy^h)^Tu : A^Tu \leq 0, ||u|| \leq 1, u \geq 0$\}, but having a different normalization constrain. Instead of $||u|| \leq 1$, there is now a constraint $(F(y^{MW} - y^h))^Tu \leq 1$ that is weighted by a vector difference between $y^{MW}$ and $y^h$. As a direct consequence of the application of Lemmas 6 and 7 from Mercier (2008), an one-to-one correspondence for each extreme point of this bounded dual polyhedron and each extreme ray of the original unbounded Benders dual SP can be established. Therefore, dual information about unbounded extremal rays can be obtained to generate feasibility BCs. □

Notice that the proposed variable transformation can only be carried out when the maximum error in a given system is bounded by 1, or if a proper normalized error procedure is at hand. Extreme rays of the dual polyhedron and the maximum possible global weight $\lambda$ can now be found, as well as new Maganti-Wong points and Pareto-optimal BCs at each iteration by solving a single auxiliary subproblem in the form max\{$(b - Fy^h)^Tu : A^Tu \leq 0, ||u|| \leq 1, u \geq 0$\}. The scale coefficients of the new normalization constraint allow to obtain better, smaller values for the dual variables $u$, but still assessing the errors in the system. Smaller values for $u$, imply smaller coefficients for the feasibility BCs, which renders stronger cuts. When $\lambda$ is sufficiently close to 1.0, no feasibility cut needs to be assembled. In this case instead, one can use the values for the dual variables to separate an optimality BC. The handy additional SP is thus capable of yielding improved Benders feasibility cuts, and it is responsible for the noted enhanced performance during the computational experiments.

### 3.4.4. Benders cuts inside the branch-and-bound tree

Finally, valid BCs can be produced at any node of a branch-and-bound tree of a Benders MP, because they can be separated from any solution of the MP, fractional or not (McDaniel and Devine, 1977). This can be achieved through the use of advanced features of modern mixed integer solvers, such as callback instructions which allow for the addition of BCs to the MP. An outline of such strategy is displayed in Algorithm 2. The branch-and-bound tree $T$ of the MP is initialized with the root node having no branching constraints. While the tree $T$ is not empty or the gap is greater then zero, a node of tree is selected and removed from $T$. Then this node is now augmented by the branching constraints and the generated BCs so far obtaining values for the LB and $(z^0, y^1, y^2, y^3)$. If the maximum number of BCs $\Pi_{\text{max}}$ is not reached, then BCs are separated observing if the SP is bounded or unbounded. In the former case, the Maganti-Wong point is updated before new Pareto-optimal BCs are generated and added to the pool $\Omega_{\text{cut}}$ of BCs; while in the latter, the new auxiliary Benders SP with the new normalization constraint is solved. BCs are separated and the Maganti-Wong point update so that new Pareto-Optimal BCs can be produced. Note that at each generation of BCs, they are added to the pool of BCs $\Omega_{\text{cut}}$. Whenever a node of the branch-and-bound tree yields integer values for $(z, y^0, y^1, y^2, y^3)$, a potential incumbent solution is available and the upper bound may be updated if possible. Otherwise the branching is carried out generating new nodes to be added to the branch-and-bound $T$. If the maximum number of cuts $\Pi_{\text{max}}$ is reached, then no BCs is generated and the branch-and-bound algorithm runs its usual course. When it stops, or a new Benders iteration is performed following the same logical steps taken inside of the branch-and-bound tree or an optimal solution is achieved.

20
4. Computational experiments

The computational experiments devised for this work aim at three distinct aspects: first, providing a fair comparison of the proposed formulations for some of the more challenging variants of the problem; then, establishing some of the interesting features of the new system optimal designs; and finally, providing assessment of which large instances are within reach of our improved technique. For all those experiments, the package *IBM CPLEX 12.5* is used as a solver for Benders master program and the associated subproblems. The computational platform adopted is a *Dell PowerEdge T620* workstation, equipped with two *Intel Xeon E5-2600v2* processors and 96 Gbytes of RAM memory.

The arc specific fixed costs ($A_{ij}$) were the same ones adopted in O’Kelly et al. (2015).

4.1. Comparing the available formulations

The first testbed is designed to compare the available and the proposed formulations for the problem on the p-hub variant. The instance *CAB25* is selected, together with the parameters already used in O’Kelly et al. (2015). The results are shown in Table 9, where the column for *Fixed Costs* shows $f_0, f_1, f_2, f_3$ and the column for *Variable Costs* shows $b_0, b_1, b_2, b_3$. This table displays the number of cuts employed by *IBM CPLEX*, the linear programming optimality gap, the number of nodes on the *branch-and-bound* tree, and the cpu time in seconds for the p-hub variant, using $p = 4$ in all cases. All the tests in Table 9 were carried out using a stand-alone, straightforward implementation of the
compared models, without decomposition or any other special technique. Recall that the formulation of O’Kelly et al. (2015) is modified by the addition of constraints (16) and (17). As expected, the relative dominance of the proposed models is experimentally verified, and the augmented computational effort for solving the larger formulations as well, justifying the decomposition approach. The results in Table 9 show that formulation (49)-(57) is fastest on 14 of the 21 problems, and in aggregate the cpu time required for the formulation in O’Kelly et al. (2015) is 3.8 times that for formulation (49)-(57). The model (49)-(57) also dominates (32)-(48) for these problems, as the latter requires over 19 times more cpu time than former.

The second testbed provides a comparison of the proposed models, (49)-(57) and (32)-(48) respectively, for the hop constrained case. To generate a fair comparison, formulation (49)-(57) needed to be modified by the redefinition of the variables \( x \) as binary and by the addition of an extra constraint limiting the number of hops in the form:

\[
\sum_{k \in S} h_{ijk} + \sum_{k \in S} \sum_{m \in N} x_{ijkm} + \sum_{m \in N} t_{ijm} \leq S \quad \forall i, j : i \neq j
\]

Again, all the computations here use straightforward implementations of the devised formulations. The obtained results are displayed in Table 10, where the second column defines the data set as either CAB15, CAB20 or CAB25. The following columns are the same as in Table 9. Columns Diff. [%] and Rel. Time show the difference between integrality gaps for the solved models and the ratio between the solution time of formulations (32)-(48) and (49)-(57), respectively. As expected, the dominance of the Leontief based model is empirically demonstrated. In fact, even when the difference between the bounds is small, a high number of nodes on the branch-and-bound tree is observed, resulting in very large solution times for the formulation (49)-(57). Naturally, the improvements of the larger model tend to have a higher impact as the difference of linear programming bounds is increased. Further, it can be noted that the hop constrained variant is even harder than the p-hub case.

4.2. Comparing the optimal topologies

This third testbed aims to provide insight on the differences of the typical optimal topologies that may be found for different problem variants. We consider three types of hub location problems with our generalized cost structure: (1) \( p \)-hub constrained problems have the number of hubs to locate specified as \( p \), (2) hop constrained problems have a specified maximum number of hops allowed, and (3) cost driven problems allow the fixed and variable costs.
costs for the hubs and arcs to determine the optimal solution, without restrictions on the path lengths or the number of hubs. The first three tests use the same parameters for fixed and variable costs, [2500, 3000, 3500, 3000] and [0.08, 0.04, 0.03, 0.04] respectively. Moreover, in order to provide a fair comparison with the fixed costs case, a fixed cost of $1.0 \times 10^7$ for all hubs is adopted for all the three tests.

The obtained topologies are shown in Figures 5, 6 and 7. In all the diagrams, the black links are inter-hub connections, the blue links are heads or tails and the red links are direct paths connecting two nodes. In spite of the fact that only a small subset of results are plotted, for the sake of simplicity, the task of establishing the structure of each of the addressed variants is well-illustrated.

A range of effects can be seen in the following sequence of three figures. Figure 5 shows the optimal network for the cost driven problem, where there is a relatively large number of hubs (6) connected via six transfer arcs. There is a path with six hops to travel between the eastern-most node 3 (Boston) and the western-most node 22 (San Francisco). Figure 6 adds the hop constraint with $S = 5$, and the resulting optimal network replaces the two transfer arcs between nodes 12 and 8 and nodes 8 and 21 in Figure 5 with a single transfer arc between nodes 12 and 21, thereby providing a 5 hop path between node 3 and node 22. Node 8 is then allocated to both hubs at 12 and 21, but the rest of the network is the same as in Figure 5. With the p-hub type constraints in Figure 7 (in this case $p = 4$), and with no hop constraints, the design is quite sparse with a tree of hubs as the backbone. The limitation to four hubs in Figure 7 does cause the network to respond with more head and tail arcs in the access network, compared to Figure 5 and Figure 6. Note that the restriction to four hubs in Figure 7 does limit the longest possible path to 5 hops; however, this is an inefficient way to achieve that goal as shown by the cost of the hop constrained model being 1.6% less expensive.

Table 10: A comparison of the proposed models for the hop constrained problem variant for $S = 5$ using CAB instances.

<table>
<thead>
<tr>
<th>Test</th>
<th>Instance Data</th>
<th>Formulation (49)-(57)</th>
<th>Formulation (32)-(48)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[10000, 30000, 60000, 30000]</td>
<td>[0.06, 0.04, 0.03, 0.03]</td>
<td>7.47350 927 10251</td>
</tr>
<tr>
<td>2</td>
<td>[10000, 30000, 60000, 30000]</td>
<td>[0.06, 0.04, 0.02, 0.03]</td>
<td>9.35500 1927 24304</td>
</tr>
<tr>
<td>3</td>
<td>[10000, 3000, 6000, 3000]</td>
<td>[0.06, 0.04, 0.03, 0.03]</td>
<td>0.29030 11 369</td>
</tr>
<tr>
<td>4</td>
<td>[10000, 3000, 12000, 3000]</td>
<td>[0.06, 0.04, 0.02, 0.03]</td>
<td>1.72840 34 691</td>
</tr>
<tr>
<td>5</td>
<td>[5000, 3000, 12000, 3000]</td>
<td>[0.06, 0.04, 0.02, 0.03]</td>
<td>1.61500 25 674</td>
</tr>
<tr>
<td>6</td>
<td>[2500, 2500, 2500, 2500]</td>
<td>[0.04, 0.04, 0.04, 0.04]</td>
<td>0.03050 3 502</td>
</tr>
<tr>
<td>7</td>
<td>[2500, 2500, 5000, 2500]</td>
<td>[0.05, 0.04, 0.03, 0.04]</td>
<td>0.29930 18 2110</td>
</tr>
<tr>
<td>8</td>
<td>[2500, 2500, 7500, 2500]</td>
<td>[0.05, 0.04, 0.03, 0.04]</td>
<td>0.00000 0 1035</td>
</tr>
<tr>
<td>9</td>
<td>[2500, 2500, 10000, 2500]</td>
<td>[0.05, 0.04, 0.02, 0.04]</td>
<td>0.01390 3 1628</td>
</tr>
<tr>
<td>10</td>
<td>[2500, 3500, 10000, 3500]</td>
<td>[0.06, 0.04, 0.01, 0.04]</td>
<td>2.76130 267 42703</td>
</tr>
<tr>
<td>11</td>
<td>[10000, 15000, 15000, 15000]</td>
<td>[0.06, 0.04, 0.03, 0.04]</td>
<td>0.08504 11 6876</td>
</tr>
<tr>
<td>12</td>
<td>[2500, 2500, 2500, 2500]</td>
<td>[0.04, 0.04, 0.04, 0.04]</td>
<td>0.08504 11 6876</td>
</tr>
<tr>
<td>13</td>
<td>[2500, 3500, 5000, 3500]</td>
<td>[0.08, 0.04, 0.02, 0.04]</td>
<td>1.48316 153 99403</td>
</tr>
<tr>
<td>14</td>
<td>[2500, 3000, 3500, 3000]</td>
<td>[0.08, 0.04, 0.03, 0.04]</td>
<td>0.00000 0 8281</td>
</tr>
<tr>
<td>15</td>
<td>[2500, 3000, 3500, 3000]</td>
<td>[0.08, 0.04, 0.02, 0.04]</td>
<td>0.00000 0 3245</td>
</tr>
</tbody>
</table>
Observing Figures 5–7, one might be tempted to infer that a hop constrained network with $S = 5$ (which has at most 4 hubs in a given path) would be the same as a $p$-hub problem network with $p = 5$, instead of $p = 4$. In order to clarify that, the aforementioned experiment was extended now using $[1500, 1500, 1500, 1500]$ and $[0.08, 0.04, 0.03, 0.04]$ as
fixed and variable costs, changing also the flat fixed costs to \(0.3 \times 10^7\) for all hubs, and taking \(S = 4\). The perturbation on the parameters was selected to amplify the possible differences among the optimal designs obtained by the distinct variants under investigation. The resulting optimal networks are displayed in Figures 1-3 on the Introduction §1.

Figures 1 and 3 have no hop constraints, but require the use of 3 and 6 hubs, respectively. In contrast, Figure 2 imposes a hop constraint (with \(S = 4\)), but the optimal number of hubs is determined by the cost trade-offs, and found to be 6. While both Figures 1 and 2 produce networks where all paths have at most 4 hops, the hop constrained model provides a lower cost by using the flexibility of more hubs, and consequently fewer direct arcs and fewer multiple allocations of non-hub nodes. Comparing Figures 1 and 3 shows how allowing more hubs (6 vs. 3) allows more transfer arcs and more hops in a path, and provides a lower cost. Note though that the longest path in Figure 3 is only 5 hops, not the maximum possible of seven hops if all six hubs were on some OD path. Finally, a comparison of Figures 2 and 3 shows the effects from relaxing the hop constraint. Both of these networks use six hubs, but the added flexibility of no hop constraints in Figure 3 allows fewer transfer arcs and a lower cost.

In order to investigate further the possible shape of the hop constrained optimal solutions, the former experiment is expanded to provide a better view of how the proposed models should respond to changes on the ratio of fixed and variable costs on the connections. The optimal networks are displayed in Figure 8, for the parameters displayed in Table 11, recalling that for all cases we are using \(S = 5\).

### Table 11: Parameters used for the generation of the optimal solutions in Figure 8.

<table>
<thead>
<tr>
<th>Test #</th>
<th>Flat Hub Installation Costs</th>
<th>Fixed Costs</th>
<th>Variable Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.3 \times 10^7)</td>
<td>([1500, 1500, 1500])</td>
<td>([0.04, 0.04, 0.04, 0.04])</td>
</tr>
<tr>
<td>2</td>
<td>(0.3 \times 10^7)</td>
<td>([2500, 2500, 2500])</td>
<td>([0.04, 0.04, 0.04, 0.04])</td>
</tr>
<tr>
<td>3</td>
<td>(0.3 \times 10^7)</td>
<td>([2500, 3500, 5000, 3500])</td>
<td>([0.08, 0.04, 0.02, 0.04])</td>
</tr>
<tr>
<td>4</td>
<td>(1.0 \times 10^7)</td>
<td>([2500, 3000, 3500, 3500])</td>
<td>([0.08, 0.04, 0.03, 0.04])</td>
</tr>
<tr>
<td>5</td>
<td>(1.5 \times 10^7)</td>
<td>([2500, 3000, 3500, 3500])</td>
<td>([0.08, 0.04, 0.02, 0.04])</td>
</tr>
</tbody>
</table>

![Total cost = 418569314.9498](image1)

(a) Test 1  

![Total cost = 451792408.4048](image2)

(b) Test 2
Comparing Test 1 and Test 2 in Figure 8 shows that increasing the arc fixed costs uniformly causes a decrease in the total number of arcs (from 41 in Test 1 to 37 in Test 2) as 5 access arcs are removed and two transfer arcs and one hub are added. This addition of a hub and transfer arcs helps reduce cost as this effectively can replace several head and tail arcs by handling flows for multiple non-hub nodes. Test 3 shows the influence from two changes: an increase in the variable costs for direct arcs and in the fixed costs for all other arcs. The net result (Test 3 vs. Test 2) is a sharp reduction in the total number of arcs (27 for Test 3 vs. 37 for Test 2), including no direct arcs due to their high variable cost, but the addition of two hubs (10 hubs in Test 3 vs. 8 hubs in Test 2) as the fixed hub cost did not increase. Both Test 4 and Test 5 have an increase in the hub fixed costs, and consequently they both use only 5 hubs. The effect of the lower variable cost for transfer arcs in Test 5 is quite apparent in the rearrangement of transfer arcs to convey more of the large flows between Florida (nodes 14 and 24) and the eastern US cites. Overall, these figures display how the subtle interplay of the fixed and variable cost components determine the optimal network structure.

4.3. Evaluating the impact of the improved Benders feasibility cuts

In order to provide experimental evidence that adding Benders feasibility cuts obtained from the auxiliary subproblem of §3.4.2 renders an improved algorithm when compared to the selection scheme described in de Sá et al. (2013) a small batch of tests was designed. For all these tests, the CAB data set is employed, where for the cases with 30 and more nodes, a selection of the first \(|N| = n\) nodes from data set \(CAB100\) is adopted.

For this entire testbed the number of cut rounds and the computing time needed for reaching the linear programming solution of formulation (49)-(57) for the p-hub variant is displayed in Table 12. (This is the version with the number of hubs set to \(p\) and without hop constraints.) The last two columns of Table 12 display the improvement...
ratios on the number of cuts and the computing times, respectively. At first glance, it becomes clear that the new cuts yield better improvements only when the target instance has an aggressive profile of fixed costs. Otherwise, its performance is not remarkably different.

Table 12: A comparison of the standard Benders feasibility cuts and the new improved version.

<table>
<thead>
<tr>
<th>Test Instance Data</th>
<th>Standard</th>
<th>Improved</th>
<th>Imp. Cuts</th>
<th>Imp. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Cuts</td>
<td># Cuts</td>
<td># Cuts</td>
<td># Cuts</td>
</tr>
<tr>
<td></td>
<td>Rounds</td>
<td>Rounds</td>
<td>Time [s]</td>
<td>Time [s]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>8</td>
<td>7.42</td>
<td>56.36</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>6</td>
<td>9.24</td>
<td>50.00</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>9</td>
<td>7.42</td>
<td>56.36</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>7</td>
<td>6.77</td>
<td>69.95</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>2.00</td>
<td>5.00</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>15</td>
<td>3.00</td>
<td>10.19</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>22</td>
<td>5.00</td>
<td>10.19</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>27</td>
<td>7.00</td>
<td>14.41</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>20</td>
<td>9.00</td>
<td>11.41</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>13</td>
<td>4.00</td>
<td>10.35</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>15</td>
<td>5.00</td>
<td>12.38</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>12</td>
<td>0.00</td>
<td>10.47</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>10</td>
<td>2.00</td>
<td>5.99</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>17</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>15</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>15</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>15</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>15</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>19</td>
<td>11</td>
<td>11</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>14</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>21</td>
<td>12</td>
<td>12</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td>10</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>23</td>
<td>12</td>
<td>12</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>24</td>
<td>15</td>
<td>15</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>25</td>
<td>9</td>
<td>9</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>26</td>
<td>14</td>
<td>14</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>27</td>
<td>16</td>
<td>16</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>28</td>
<td>6</td>
<td>6</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>29</td>
<td>6</td>
<td>6</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>6</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>6</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>33</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>7</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>35</td>
<td>7</td>
<td>7</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>36</td>
<td>10</td>
<td>10</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>37</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>39</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>40</td>
<td>9</td>
<td>9</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>41</td>
<td>12</td>
<td>12</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>42</td>
<td>14</td>
<td>14</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>43</td>
<td>6</td>
<td>6</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>44</td>
<td>8</td>
<td>8</td>
<td>0.00</td>
<td>7.00</td>
</tr>
<tr>
<td>45</td>
<td>9</td>
<td>9</td>
<td>0.00</td>
<td>7.00</td>
</tr>
</tbody>
</table>

Examination of the convergence process of all the instances reveals that the new cuts are responsible for an increased computational burden, introducing some issues regarding numerical instabilities. This could even lead to poor performance, eventually. On the other hand, they are efficient in ensuring the computation of larger λ values. The sense in which the new computational strategy is better suited to instances displaying a certain cost profile is explained below.

Typically, in instances with aggressive fixed costs, the Benders master program will produce solutions with a poor allocation of infrastructure, in an effort to avoid fixed costs. This may imply a large number of master program solutions which have infeasible subproblems, reducing λ and the impact of Pareto optimal cuts as a result. Those observations suggest that the new cuts may be better employed in an adaptive way: one should add these cuts only when small λ’s are observed. For the aforementioned reasons, a final refinement of Algorithm 2 where the improved cuts are deployed adaptively was implemented. For this new algorithm, feasibility cuts from new auxiliary subproblem
from §3.4.2 are added to the master problem only when small values of $\lambda$ are computed, typically only when $\lambda < 0.5$ is found.

The results for instances with 50 and 60 nodes are shown in Table 13. For all these additional tests a time limit of 24 hours of computing was established, and from now on, all the tabulated entries display the computing times necessary to reach integer proven optimality.

Table 13: A comparison of the standard Benders feasibility cuts and the new improved version on large scale instances.

<table>
<thead>
<tr>
<th>Test</th>
<th>Instance Data</th>
<th>Standard</th>
<th>Improved</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>N</td>
<td>p</td>
<td>Fixed Costs</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>[1000, 20000, 30000, 20000]</td>
<td>[0.60, 0.08, 0.06, 0.08]</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>4</td>
<td>&gt;</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>4</td>
<td>&gt;</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>4</td>
<td>&gt;</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

In Table 13, the six additional test results show the decisive role played by the feasibility cuts recovered from computations of the auxiliary subproblem of §3.4.2. These cuts are responsible for considerable speed-ups, sometimes halving the computational burden needed to reach proven optimality. Moreover, the impact on the number of cut rounds needed to optimize the instance is also remarkable. Once again, the impact of the new cuts tend to be more important when small values of $\lambda$ are found during the convergence of the algorithm.

In order to refine the illustration of the aforementioned effect, a tracking of the resulting $\lambda$ values and the associated lower-bound quality for the 60 nodes and 5 hubs instance of Table 13 is displayed in Figure 9. The figure shows the stabilizing effect provided by the new cuts on the $\lambda$ values, typically avoiding the calculation of small or null lambdas, and the associated improvements in the respective lower-bounds during the convergence of the algorithm.

It is important to focus on cut round #4: On the original procedure (blue lines), after this first computation of a $\lambda$ value below 0.5, the algorithm becomes quite unstable, computing $\lambda = 0$ several times, and as shown on the lower-bounds comparison, delaying convergence. On the improved algorithm however (black lines), after cut round #4, a single round of the improved feasibility cuts suffices to avoid the instabilities, always rendering $\lambda \geq 0.5$ and therefore providing better lower-bounds and a faster convergence.

4.4. Tackling large-scale instances

On the last set of experiments, the devised Benders algorithms for the formulations (49)-(57) and (32)-(48) were tested against larger instances for the p-hub and hop constrained variants, respectively. For all instances, the fixed and variables costs were set to [1000, 1000, 1000, 1000] and [0.10, 0.04, 0.02, 0.04], respectively; however there are no installation costs for the hubs. Tables 14 and 15 show the attained results.
As the number of nodes increases, the computational effort required to attain optimality increases considerably for both algorithms. The hop constrained variant requires more time and more branch-and-bound nodes for the master problems when solving instances of the same size compared to the p-hub constrained variant. The extra cut rounds and branch-and-bound nodes allow for the generation of more cuts, which in turn improve the bounds at the price of slowing down the search process in the future rounds and adding branch-and-bound nodes. Just for illustration purposes, tests 10-12 of Table 14 are depicted in Figures 10-12 for the p-hub constrained version. This shows the 3, 4 and 5-hub optimal solutions for the generalized cost formulation (49)-(57) with 80 nodes. Figure 10 with 3 hubs (in New York, Indianapolis, and Las Vegas) uses 11 direct arcs, mainly to reduce cost in the southeast (e.g., Texas and Florida). Allowing a fourth hub in Figure 12 effectively splits the nodes served through Indianapolis between new hubs in Atlanta and Chicago, with a corresponding elimination of some direct arcs due to the better (lower cost) service via Atlanta. Figure 12 shows the further reduction in use of direct arcs with one more hub (in Texas). These figures and Table 14 demonstrate the strength of our formulation and our tailored Benders decomposition approach to solve rather large and complex hub location problems with a generalized cost function.

Table 14: Results for the Benders algorithm based on the formulation (49)-(57) on larger instances for the p-hub constrained variant.

<table>
<thead>
<tr>
<th>Test</th>
<th># of Hubs</th>
<th># Cut Rounds</th>
<th># b&amp;b Rounds</th>
<th>Time [s] (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>9,417.74 (2.61)</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>4</td>
<td>11</td>
<td>13,311.79 (3.70)</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>11,548.01 (3.21)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>25,302.97 (7.00)</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>4</td>
<td>10</td>
<td>40,704.83 (11.31)</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>43,419.86 (12.06)</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>56,258.37 (15.63)</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>4</td>
<td>8</td>
<td>63,503.07 (17.64)</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>44,668.49 (12.41)</td>
</tr>
</tbody>
</table>

Table 15: Results for the Benders algorithm based on the formulation (32)-(48) on larger instances for the hop constrained variant, here $S = 4$. 

<table>
<thead>
<tr>
<th>Test</th>
<th># of Hubs</th>
<th># Cut Rounds</th>
<th># b&amp;b Rounds</th>
<th>Time [s] (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>4</td>
<td>10</td>
<td>3,675.70 (1.02)</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>6</td>
<td>14</td>
<td>6,869.27 (1.91)</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>4</td>
<td>11</td>
<td>12,400.93 (3.44)</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>6</td>
<td>12</td>
<td>17,138.58 (4.76)</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>5</td>
<td>12</td>
<td>40,452.37 (11.23)</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>7</td>
<td>31</td>
<td>345,602.19 (96.00)</td>
</tr>
<tr>
<td>7</td>
<td>70</td>
<td>4</td>
<td>17</td>
<td>135,896.98 (37.75)</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>5</td>
<td>12</td>
<td>366,159,912.38 (44.42)</td>
</tr>
</tbody>
</table>
Figure 10: The optimal network for an 80 nodes instance, using $p = 3$ and p-hub constraints.

Figure 11: The optimal network for an 80 nodes instance, using $p = 4$ and p-hub constraints.
5. Final remarks

The paper has provided a wide range of results for the incomplete hub location problem with and without hop-constraints. The optimal network designs and traffic flows are determined by the interplay of the fixed costs for the network components and the variable costs for using the network. Models with hop constraints allow aspects of service to be incorporated along with costs, though they are generally more challenging to solve. The results display all the expected nuances for a realistic hub and spoke system: direct connection, incomplete hub backbone, and various type of distribution and collection arcs. That this is accomplished for a system with 80 nodes represents a strong demonstration that the many refined techniques and insights gained from careful study and exploration of a standard formal model have paid off in terms of relatively large scale computability. As described in Campbell and O’Kelly (2012) there are major hurdles to a model of greater size with greater realism. The current paper achieves a good balance between the twin goals. Moreover, a new general scheme for generation better Benders feasibility cuts has been devised which greatly speeds up the Benders decomposition method.

Acknowledgments

O’Kelly is grateful for support from CNPq 402651/2012-0, and for Award BCS-1125840 from NSF. Miranda and Camargo thank CNPq for the financial support, projects: 302986/2012-0, 202765/2013-0, 402651/2012-0 and 304603/2013-0. 


