Expert Systems with Applications 39 (2012) 3385–3391

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Single allocation hub location problem under congestion: Network owner and user perspectives

R.S. de Camargo *, G. Miranda

Department of Industrial Engineering, Federal University of Minas Gerais Av., Antonio Carlos, 6627, 31270-901 Belo Horizonte, MG, Brazil

ARTICLE INFO

Keywords:
Congestioned hub-and-spoke networks
Generalized Benders decomposition
Nonlinear mixed-integer programming

ABSTRACT

The single allocation hub location problem under congestion is addressed in this article. This mixed integer non-linear programming problem is referential in discrete location research having many real applications. Two different network design perspectives are proposed: the network owner and the network user. These perspectives can be translated into mathematical programming problems that are very hard to solve due to their inherently high combinatorial nature combined to the nonlinearities associated to congestion. A very efficient and effective generalized Benders decomposition algorithm is then deployed, enabling the solution of large scale instances in reasonable time.

1. Introduction

In many-to-many transportation systems, hub-and-spoke (HS) networks have an important role. They can efficiently route demand between many origins and destinations nodes using fewer links. Instead of directly connecting each pair of origin–destination, which is usually extremely expensive, transshipment facilities, named hubs, are employed to aggregate, route and distribute the flow, sometimes via other hubs, towards their final destination. Hubs are then responsible for the centralization of the handling and sorting processes and for flow consolidation. The aggregation of traffic at these facilities allows the exploitation of scale economies due to the use of more efficient and higher volume carriers on inter-hub connections. For a general review of different applications and problems of HS networks on the telecommunication systems refer to Klincewicz (1998), while for freight and passenger transportation sector see Campbell, Ernst, and Krishnamoorthy (2002), Alumur and Kara (2008).

In the literature on HS networks, there are several versions of hub location problems, mainly differing from one another by the nature of the connection of the nonhub nodes to the hubs, the number and type of hubs to be located, the presence or absence of hub capacities or if the graph induced by the hub nodes is complete or incomplete. Independently of which problem is addressed, one of the main overall advantages of such networks is the exploitation of scale economies. However this exploitation may induce the formation of networks that tend to overload a small number of hubs, resulting in some inter-hub connections more heavily-utilized than others. Hence it is unavoidable to take congestion effects into consideration.

A common way of addressing congestion is to limit the amount of traffic a installed hub can handle (Aykin, 1994, 1995; Contreras, Díaz, & Fernández, 2011; Costa, Captivo, & Clímaco, 2008; Ernst & Krishnamoorthy, 1999; Labbé, Yaman, & Gourdin, 2005; Marianov & Serra, 2003; Rodriguez, Alvarez, & Barcos, 2007). Unfortunately, capacity constraints do not mimic the explosive nature of congestion: the more flow a hub attracts, the harder the handling process becomes resulting then in greater costs. Usually these costs increase extremely rapid due to queueing and delay effects. Thus elaborate cost functions are needed such as the one employed by Elhedhli and Hu (2005). They are the first authors to consider explicitly the congestion effects of each located hub as a cost on the objective function for the HS problems. Using a power-law function widely utilized to estimate delay costs in airport applications (Gillen & Levinson, 1999), they propose a non-linear formulation where this convex cost function, that increases rapidly as more traffic flows through the installed hubs, is present on the objective function. They linearize the model utilizing a set of infinite piece-wise linear and tangent hyperplanes, and then solve it by means of a Lagrangean relaxation algorithm. They solve only small instances (up to 25 nodes) with an average optimality gap of 0.92%. The obtained solutions have a more balanced overall distribution of flows through the network than the ones attained by disregarding the congestion effects. More recently Elhedhli and Wu (2010) proposed the same approach but considering each hub as a $M/M/1$ queue and use the Kleinrock's average delay function Kleinrock (1964) as a representation of the congestion effects. Once again, only small size instances (up to 25 nodes) with an average optimality gap of 1.00% are solved in reasonable time.

In this paper, the single allocation hub location problem under congestion (SAHLPC) is addressed, where the number of hubs to be located is undetermined beforehand, but there are fixed costs for locating hubs on the nodes, each nonhub node is allocated to

* Corresponding author. Tel.: +55 31 3409 4881; fax: +55 31 3409 4883.
E-mail addresses: rcamargo@dep.ufmg.br (R.S. de Camargo), miranda@dep.ufmg.br (G. Miranda).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.026

a single hub, and congestion effects are addressed. Further, for each pair of origin–destination demand, a path containing at least one and at most two hubs is established. The objective is to minimize the installation, congestion, and routing costs over the network. Moreover, two different perspectives are analyzed: The network owner (NO) who aims at a network design with the least cost and the network user (NU) who is willing to accept the minimum of congestion effect at reasonable cost. Two similar nonlinear mixed integer linear formulations are used and solved by means of the generalized Benders decomposition (GBD) method (Geoffrion, 1972). The proposed algorithm optimally solves large instances (up to 100 nodes) in reasonable time. The outline of this paper is as follows: general notation, definitions and the formulations are provided in Section 2. The GBD algorithm is presented in Section 3. Finally, an illustrative scenario and the computational results are shown in Sections 4 and 5, respectively, while the final remarks and future research plans are done in Section 6.

2. Definitions and formulation

The SAHLPC is formulated as a mixed integer nonlinear program (MINLP) and requires the following definitions: Let N be the set of demand node locations which exchange flows and let K be the set of node candidates to become hubs. Usually $K \subseteq N$, but it is assumed henceforth that all demand nodes are candidates to have a installed hub, implying $K \equiv N$. For all node pairs i and $j(i,j \in N : i \neq j)$, w_{ij} represents the flow demand from origin node i to destination node j which is routed through either one or two installed hubs. Let also $O_i = \sum_{j \in N} w_{ij}$ and $D_i = \sum_{j \in N} w_{ji}$ be the total of demand that is originated from and destined to node $i \in N$, respectively. Further, let f_k be the fixed installation cost of a hub at node $k \in N$ and let c_{ik} and c_{mj} be the standard transportation costs per unit of flow from node i to hub k and from hub m to node j, and αc_{km} be the discounted standard transportation cost between hubs k and m. The discount factor $0 \leqslant \alpha \leqslant 1$ represents the scale economies on the inter-hub connections. In the remainder of this paper, for the sake of simplicity in presentation, one must consider $i, j, k, m \in N$ and $i < j$.

The MINLP uses flow variables $x_{ijkm} \geqslant 0$ to represent the fraction of demand w_{ij} that is routed through hubs k and m, in this order; the variables g_k to account for the total flow passing through hub k; and the integer variables $z_{ik} \in \{0,1\}$ to indicate if node i is allocated to hub k ($z_{ik} = 1$) or not ($z_{ik} = 0$). When a hub is located at node k, then $z_{kk} = 1$; otherwise $z_{kk} = 0$. Variables x_{ijkm} are equal to the product of variables $z_{ik}z_{jm}$. The adopted congestion cost function considers congestion effects only after a given flow threshold $\gamma\Gamma_k$ is trespassed (usually γ is set to 70% of the hub nominal capacity Γ) and it is written as $\tau_k(g_k) = max\{0, a\,(g_k - \gamma\Gamma_k)^b\}$, where the parameters $a > 0$ and $b \geqslant 1$ are scalars related to the hub features.

From the perspective of the NO, the implied formulation is given as:

$$\min \sum_k [f_k z_{kk} + \tau_k(g_k)] + \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik} + \sum_{i<j} \sum_{k \neq m} c_{ijkm} x_{ijkm} \quad (1)$$

$$\text{s.t.}: \sum_k z_{ik} = 1 \quad \forall\ i \quad (2)$$

$$\sum_m x_{ijkm} = z_{ik} \quad \forall i < j, k \quad (3)$$

$$\sum_k x_{ijkm} = z_{jm} \quad \forall\ i < j, m \quad (4)$$

$$z_{ik} \leqslant z_{kk} \quad \forall\ i \neq k \quad (5)$$

$$x_{ijkm} \geqslant 0 \quad \forall\ i < j, k, m \quad (6)$$

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

$$\sum_i (O_i + D_i) z_{ik} - \sum_{i<j} (w_{ij} + w_{ji}) x_{ijkk} = g_k \quad \forall\ k \quad (8)$$

$$g_k \geqslant 0 \quad \forall\ k \quad (9)$$

where $c_{ijkm} = (w_{ij}c_{km} + w_{ji}c_{mk})\alpha$. The objective function (1) minimizes the total cost associated with the demand transportation, the congestion effects and the hub installation costs. Constraints (2) assure that all nodes are allocated to a hub. Constraints (3) guarantee that routes beginning at origin node i, then passing firstly at hub k, and finishing at destination node j will only exist if node i is allocated to hub k. Likewise, constraints (4) guarantee that routes beginning at origin i and passing at hub m just before finishing at destination j will only exist if node j is allocated to hub m. Constraints (5) allow a node i to be allocated to hub k only if hub k is installed. Constraints (8) are responsible for accounting the total hub traffic, avoiding the double computation of the local traffic component. Constraints (6), (9) are the non-negativity of variables x_{ijkm} and g_k, respectively, while (7) are the integrality constraints of variables z_{ik}.

From the perspective of the NU, the maximum congestion effect needs to be minimized, hence resulting in the following formulation:

$$\min \tau(\theta) + \sum_k f_k z_{kk} + \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik} + \sum_{i<j} \sum_{k \neq m} c_{ijkm} x_{ijkm} \quad (10)$$

$$\text{s.t.}: \ (3)-(8)$$

$$\theta \geqslant \sum_i (O_i + D_i) z_{ik} - \sum_{i<j} (w_{ij} + w_{ji}) x_{ijkk} - \gamma\Gamma_k \quad \forall\ k \quad (11)$$

$$\theta \geqslant 0 \quad (12)$$

where the non-negative variable θ accounts for the largest congestion effect of the network using constraints (11).

3. Generalized Benders decomposition

The Benders decomposition algorithm (Benders, 1962) is a partition method for solving mixed-integer linear and non-linear programming problems. In general terms, the algorithm relies on a projection problem manipulation, followed by solution strategies of dualization, outer linearization and relaxation. The complicating variables (integer variables) of the original problem are projected out, resulting into an equivalent model with fewer variables, but many more constraints. When attaining optimality, a large number of these constraints will not be binding, suggesting then a strategy of relaxation that ignores all but a few of these constraints. So, these constraints are added on demand by using two levels of coordination. At a higher level, known as master problem (MP), a relaxed version of the original problem having the set of the integer variables and its associated constraints is responsible for fixing the values of these integer variables and for providing a lower bound (LB) for the problem. At a lower level, known as subproblem (SP), the dual linear programming of the original problem with the values of the integer variables temporarily fixed by the MP is responsible for rendering a new cut or a Benders cut to be added to the MP and for generating a upper bound (UB) for the problem. The algorithm iterates, solving the MP and the SP one at a time, until the UB and the LB converge towards an optimal solution, if one exists. The presentations of the GBD algorithms for the NO and the NU perspectives are going to be presented simultaneously.

3.1. Benders master program

Projecting the problem (1)–(9) and (10)–(12) onto the space of the z variables results into the equivalent problem:

$$\min_{z \in Z} \sum_k f_k z_{kk} + \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik} + \phi(z)$$

where $\mathbf{Z} = \{z \in \{0,1\}|$ constraints (2) and (5) hold$\}$ and $\phi(z)$ is the following SP:

$$\phi(z) = \min_{(x,\Phi)\in G}\{\Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm}\}$$

being $\Phi = g, \Theta = \sum_k \tau_k(g_k)$ and $G = \{(x,g) \geqslant 0|$ constraints (3), (4) and (8) hold} for the case of NO and $\Phi = \theta$, $\Theta = \tau(\theta)$ and $G = \{(x,\theta) \geqslant 0 \mid$ constraints (3), (4) and (11) hold} for the perspective of the NU.

Since the constraints defining z are enough to ensure feasibility, the SP is bounded. Further, as $\phi(z)$ has a convex and differentiable objective function and linear constraints, its Karush–Kuhn–Tucker conditions are necessary and sufficient for optimality, hence amenable to dualization techniques. So associating vectors of dual variables u, v and β to constraints (3)–(5) ((11)), respectively, and because there is no duality gap, $\phi(z)$ can be re-written as:

$$\begin{aligned}\phi(z) = \max_{u,v,\beta}\Bigg\{\min_{(x,\Phi)\in G}\Bigg\{&\Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} \\ &+ \sum_{i<j}\sum_{k,m} u_{ijk}x_{ijkm}\sum_{i<j}\sum_{k,m} v_{ijm}x_{ijkm} - \sum_{i<j}\sum_{k}\beta_k(w_{ij}+w_{ji})x_{ijkk} - \Upsilon\Bigg\} \\ &- \sum_{i<j}\sum_{k} u_{ijk}z_{ik} - \sum_{i<j}\sum_{m} v_{ijm}z_{jm} + \sum_{i,k}\beta_k(O_i+D_i)z_{ik}\Bigg\}\end{aligned}$$

where $\Upsilon = \sum_k \beta_k g_k$ for the case of NO and $\Upsilon = \sum_k \beta_k g_k(\gamma\Gamma_k + \theta)$ for the perspective of the NU. Since the supremum is the least upper bound and with the help of variable $\eta \geqslant 0$, the whole problems (1)–(9) and (10)–(12) are then equivalent to following MP:

$$\min_{z\in Z}\sum_k f_k z_{kk} + \sum_{k\neq i}(O_i+D_i)c_{ik}z_{ik} + \eta \tag{13}$$

$$\begin{aligned}\text{s.t.: } \eta \geqslant \min_{(x,\Phi)\in G}\Bigg\{&\Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} + \sum_{i<j}\sum_{k,m} u_{ijk}x_{ijkm} \\ &\sum_{i<j}\sum_{k,m} v_{ijm}x_{ijkm} - \sum_{i<j}\sum_{k}\beta_k(w_{ij}+w_{ji})x_{ijkk} - \Upsilon\} \\ &- \sum_{i<j}\sum_{k} u_{ijk}z_{ik} - \sum_{i<j}\sum_{m} v_{ijm}z_{jm} + \sum_{i,k}\beta_k(O_i+D_i)z_{ik} \qquad \forall\, u, v, \beta\end{aligned} \tag{14}$$

$$\eta \geqslant 0 \tag{15}$$

Because a large number of the constraints of the MP (13)–(15) will not be binding when optimality is attained, the GBD algorithm solves the MP through a strategy of relaxation that ignores all but a few of the constraints (14). These constraints are then added, via a iterated procedure, to the MP as needed. Thus for a given iteration t, where $z = z^t$ and after the solution of the associated SP and the recovery of the optimal values of u^t, v^t and β^t, the optimal value of $\phi(z^t)$ is given by:

$$\begin{aligned}\phi(z^t) = \min_{(x,\Phi)\in G}\Bigg\{&\Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} + \sum_{i<j}\sum_{k,m} u^t_{ijk}x_{ijkm}\sum_{i<j}\sum_{k,m} v^t_{ijm}x_{ijkm} \\ &- \sum_{i<j}\sum_{k}\beta^t_k(w_{ij}+w_{ji})x_{ijkk} - \Upsilon\Bigg\} - \sum_{i<j}\sum_{k} u^t_{ijk}z^t_{ik} \\ &- \sum_{i<j}\sum_{m} v^t_{ijm}z^t_{jm} + \sum_{i,k}\beta^t_k(O_i+D_i)z^t_{ik}\end{aligned} \tag{16}$$

Further, constraints (14) can be rewritten for iteration t in the form:

$$\begin{aligned}\eta \geqslant \min_{(x,\Phi)\in G}\Bigg\{&\Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} + \sum_{i<j}\sum_{k,m} u^t_{ijk}x_{ijkm}\sum_{i<j}\sum_{k,m} v^t_{ijm}x_{ijkm} \\ &- \sum_{i<j}\sum_{k}\beta^t_k(w_{ij}+w_{ji})x_{ijkk} - \Upsilon\Bigg\} - \sum_{i<j}\sum_{k} u^t_{ijk}z_{ik} - \sum_{i<j}\sum_{m} v^t_{ijm}z_{jm} \\ &+ \sum_{i,k}\beta^t_k(O_i+D_i)z_{ik}\end{aligned} \tag{17}$$

Therefore, by eliminating the minimum in (17) by using (16), the relaxed Benders master program (RMP) is stated as:

$$\min_{z\in Z}\sum_k f_k z_{kk} + \sum_{k\neq i}(O_i+D_i)c_{ik}z_{ik} + \eta \tag{18}$$

$$\begin{aligned}\text{s.t.: } \eta \geqslant \phi(z^t) &- \sum_{i<j}\sum_{k} u^t_{ijk}\left(z_{ik} - z^t_{ik}\right) - \sum_{i<j}\sum_{m} v^t_{ijm}\left(z_{jm} - z^t_{jm}\right) \\ &+ \sum_{i,k}\beta^t_k(O_i+D_i)\left(z_{ik} - z^t_{ik}\right) \qquad \forall\ t = 1\ldots\ T\end{aligned} \tag{19}$$

$$\eta \geqslant 0 \tag{20}$$

where T is the maximum number of iterations in order to attaining the optimal solution.

3.2. Benders subproblem

For a hub structure z^t fixed by the MP (18)–(20) at iteration t, the SP $\phi(z^t)$ is given by:

$$\min \Theta + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} \tag{21}$$

$$\text{s.t.: } \Omega \tag{22}$$

$$\sum_m x_{ijkm} = z^t_{ik} \qquad \forall i<j, \forall\ k \tag{23}$$

$$\sum_k x_{ijkm} = z^t_{jm} \qquad \forall i<j, \forall\ m \tag{24}$$

$$x_{ijkm} \geqslant 0 \qquad \forall i<j, \forall k, m \tag{25}$$

$$\Phi \tag{26}$$

where Ω is $g_k + \sum_{i<j}(w_{ij}+w_{ji})x_{ijkk} = \sum_i(O_i+D_i)z^t_{ik}, \forall\ k$, for the case of NO and Ω is $\theta \geqslant \sum_i(O_i+D_i)z_{ik} - \sum_{i<j}(w_{ij}+w_{ji})x_{ijkk} - \gamma\Gamma_k, \forall\ k$, for the perspective of the NU. In the SP (21)–(26), there are two sets of variables, x and Φ, coupled by the constraints (22). After dualizing these constraints by the associated dual multipliers β, a decomposable problem is implied:

$$\begin{aligned}d(\beta) = \min \Lambda &- \sum_{i<j}\sum_{k}\beta_k(w_{ij}+w_{ji})x_{ijkk} + \sum_{i<j}\sum_{k\neq m} c_{ijkm}x_{ijkm} \\ &+ \sum_{i,k}\beta_k(O_i+D_i)z^t_{ik}\end{aligned}$$

subject to constraints (23)–(26), where $\Lambda = \sum_k(\tau_k(g_k) - \beta_k g_k)$ for the case of the NO and $\Lambda = \tau(\theta) - \sum_k\beta_k\theta - \sum_k\beta_k\gamma\Gamma_k$ for the perspective of the NU. The problem $d(\beta)$ is decomposable in two smaller SPs and a constant: A linear problem, $d_L(\beta)$, having only the x_{ijkm} variables; a non-linear problem $d_{NL}(\beta)$, having just the Φ variables; and a fixed term. Further, the SP $d_{NL}(\beta)$ is convex and differentiable, being therefore its Karush–Kuhn–Tucker conditions necessary and sufficient for optimality. Hence, the optimal solution of β^t_k that minimizes $d_{NL}(\beta)$ for a given structure z^t fixed by the RMP at iteration t is given by $\tau'_k(g^t_k) = \beta^t_k$, for each k, in the case of the NO. Recall that the variables x_{ijkm} are equal to the product $z_{ik}z_{jm}$ allowing then the value of g^t_k (constraints (8)) to be computed as $\sum_i(O_i+D_i)z^t_{ik} - \sum_{i<j}(w_{ij}+w_{ji})z^t_{ik}z^t_{jk}$. In the perspective of the NU, the optimal solution of β^t_k that minimizes $d_{NL}(\beta)$ is given by $\tau'_k(\theta) = \sum_k\beta_k$, where $\theta = \max_k\{0, \sum_i(O_i+D_i)z^t_{ik} - \sum_{i<j}(w_{ij}+w_{ji})z^t_{ik}z^t_{jk} - \gamma\Gamma_k\}$, and, for each k, if $\sum_i(O_i+D_i)z^t_{ik} - \sum_{i<j}(w_{ij}+w_{ji})z^t_{ik}z^t_{jk} - \gamma\Gamma_k$ equals θ then $\beta^t_k = a(\theta)^b$; else $\beta^t_k = 0$. This small and apparently innocuous detail of having $x_{ijkm} = z_{ik}z_{jm}$ makes it possible to easily calculate the values of g^t and θ, avoiding therefore the use of non-linear programming methods for evaluating $\phi(z^t)$. Moreover, once the values of β_k are attained, it is possible to decompose the SP $d_L(\beta)$ in smaller problems, one for each $i - j$ pair. Henceforth the optimal values of u^t and v^t can now be computed by solving the dual linear programming of these smaller problems:

$$\max \sum_k u_{ijk}\, z^t_{ik} + \sum_m v_{ijm}\, z^t_{jm} \quad (27)$$

$$\text{s.t.}: u_{ijk} + v_{ijm} \leqslant c_{ijkm} \qquad \forall\, k \neq m \quad (28)$$

$$u_{ijk} + v_{ijk} \leqslant -\beta_k (w_{ij} + w_{ji}) \qquad \forall\, k \quad (29)$$

$$u_{ijk} \in IR \qquad \forall\, k \quad (30)$$

$$v_{ijm} \in IR \qquad \forall\, m \quad (31)$$

The efficiency of GBD algorithm depends mainly on the number of iterations required to attain global convergence. This number is intimately related to the quality of the Benders cuts assembled. Strong cuts usually mean fewer iterations. In order to have strong cuts, the solution of the SP has to be judiciously done, since the Benders algorithm is very sensitive to the selection of the dual variables. If care is not taken, a poor behavior of the algorithm may be expected. Papadakos (2008) shows that it is possible to improve the algorithm performance by means of an initial core point z^0, i.e. a point that belongs to the relative interior of the convex hull **Z**, and that is updated by a scheme akin to $z^0_{ik} = \lambda z^0_{ik} + (1-\lambda)\, z^t_{ik}, \forall\, i,k,t$, where $0 < \lambda < 1$ is a scalar parameter (usually $\lambda = 1/2$). When this core point z^0 is used in the objective function (27) instead of the variables z, a large speedup in the algorithm is

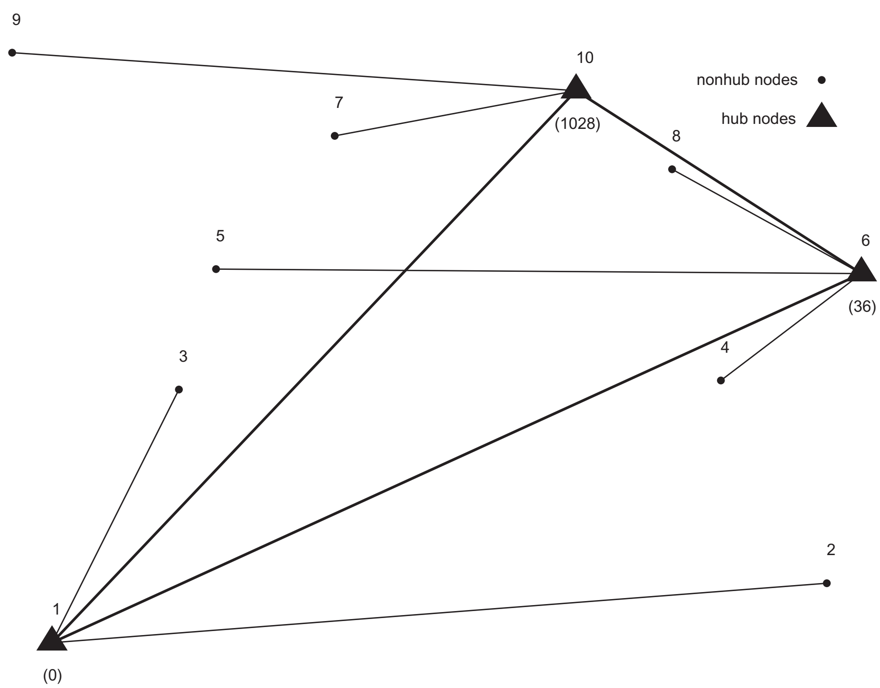

Fig. 1. Network structure designed under the perspective of the NO.

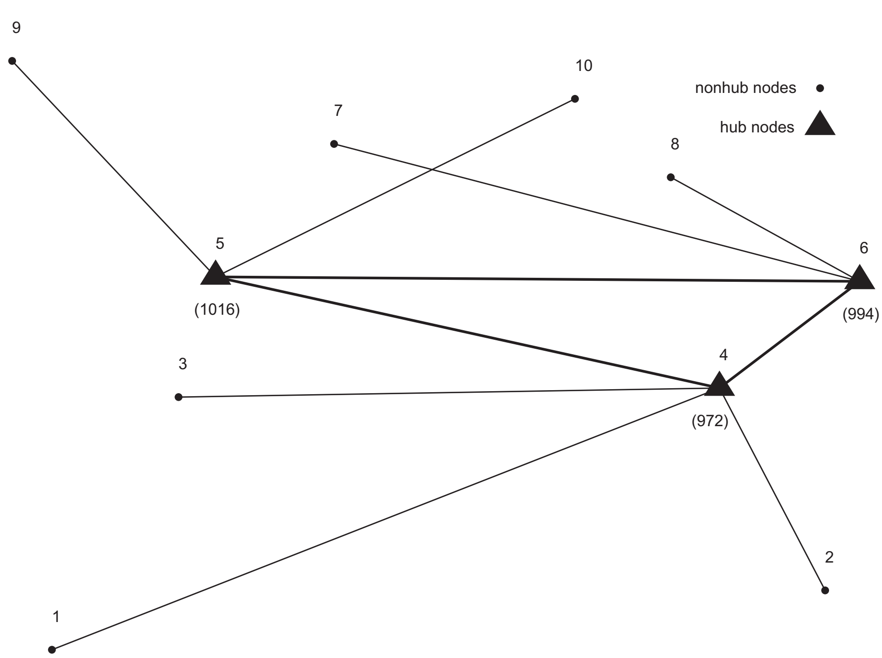

Fig. 2. Network structure designed under the perspective of the NU.

perceived. The starting core point that demonstrated the best overall performance during the computer experiments is taken as: $z^0_{kk} = 0.5, \forall k$, and $z^0_{ik} = 0.5/(n-1)$, $\forall i \neq k$, where $n = |N|$. A sketch of the implemented algorithm is depicted below, where $UB, LB, \varepsilon, \theta^*_{RMP}$ and ϑ^* are the upper bound, lower bound, stopping precision, the objective function optimal value of the RMP, and the objective function value of the current solution, respectively.

Algorithm: Generalized Benders decomposition

1 Set $UB = +\infty$, $LB = -\infty$, $t = 1$, z^0
2 If $(UB - LB) < \varepsilon$, then stop. Terminate, a near optimal solution has been obtained
3 Calculate the values of β^t
4 Solve the subproblem $d_L(\beta)$ for z^0
5 Add a new Benders cut to the RMP using (19).
6 Solve the RMP (18)–(20), obtaining θ^*_{RMP} and the optimal values for the integer variables z^t.
7 Set $LB = \theta^*_{RMP}$ and update z^t in subproblems $d_L(\beta)$ and $d_{NL}(\beta)$
8 Calculate the values of β^t
9 Solve the subproblem $d_L(\beta)$
10 Compute the optimal value of $\phi(z^t)$ and set:

$$\vartheta^* = \phi(z^t) + \sum_k f_k\ z^t_{kk} + \sum_{k \neq i} (O_i + D_i)\ c_{ik}\ z^t_{ik}$$

11 Add a new Benders cut to the RMP using (19)
12 Update core point z^0
13 If $\vartheta^* < UB$, then set $UB = \vartheta^*$
14 Increment t and go to **2**

4. An illustrative scenario

In order to illustrate the main differences between the two perspectives and how the algorithm behaves, a small example is constructed by using one instance of the the well-known Australian Post (AP) standard data set (Ernst & Krishnamoorthy, 1996, 1999). This instance has 10 nodes, $\alpha = 0.2$, $a = 0.005$ and $b = 2$. The algorithm was coded in $C++$ using the IBM CPLEX 12 Concert Technology under a Linux operating system running on a regular PC desktop with a Intel Core 2 Quad 3.2 GHz processor and 8 Gb of RAM.

Figs. 1 and 2 display the designed network structure with total cost of 14268.34 and 13669.84 for the perspectives of the NO and the NU, respectively. Although, on one hand, the percentage difference between the two solutions may be considered not so large (4.38%), on the other hand, the installed infrastructure is completely different. Furthermore, on the first case (Fig. 1), there are flows perceiving small or no congestion cost (values between parenthesis under each hub), see nodes 3 and 2, and demands that come upon high congestioned hubs, refer to nodes 7 and 9. While, on the second situation (Fig. 2), all the traffic between the different source–destination pairs experience hubs with similar congestion costs.

The algorithm performs differently for both perspectives in the proposed example. The algorithm version of NO takes 22 iterations to find the optimal solution against 8 iterations of the approach of NU. Consequently, it takes longer, 3 times more, for the upper bound (UB) and the lower bound (LB) of the former to converge to the optimal solution (Fig. 3) than the latter (Fig. 4). Notice also that although the UB improves much faster in the first situation than in the last one, the evolution of both LBs has a similar behavior.

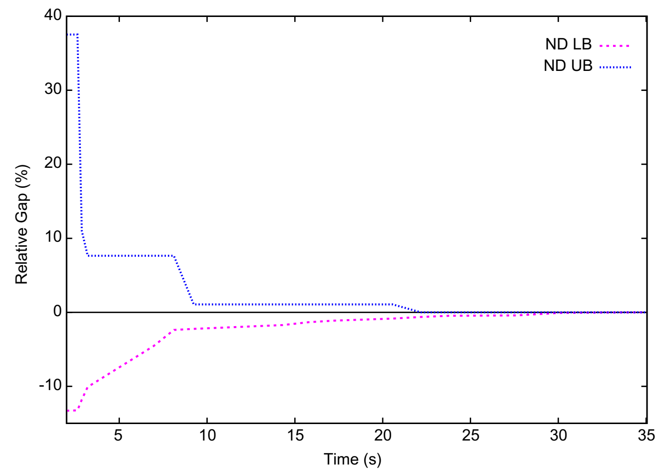

Fig. 3. Convergence of the algorithm for the perspective of the NO (relative gap x time[s]).

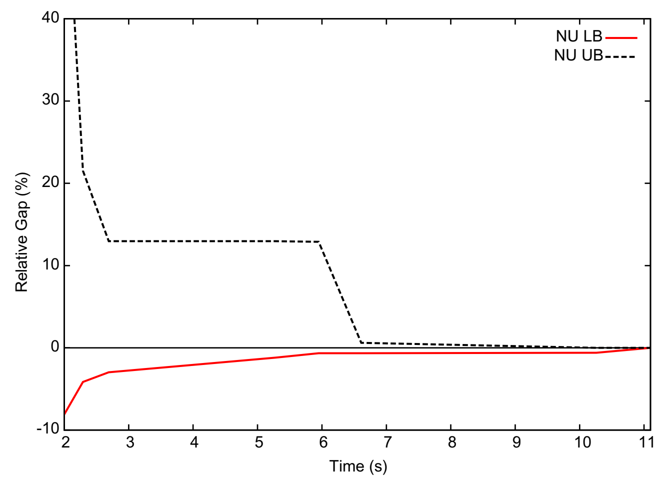

Fig. 4. Convergence of the algorithm for the perspective of the NU (relative gap x time[s]).

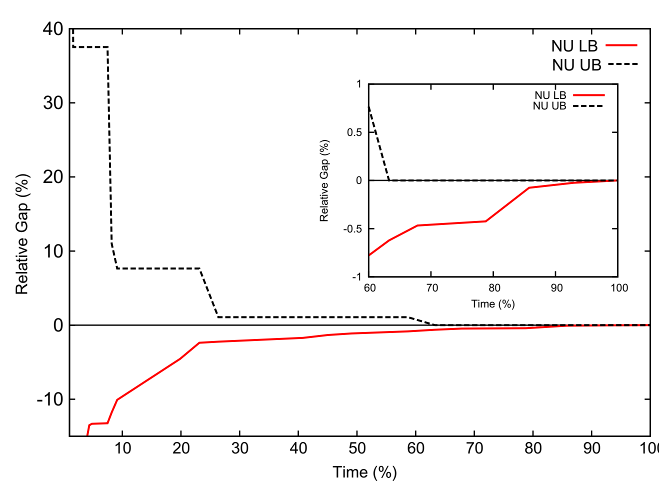

Fig. 5. Convergence of the algorithm for the perspective of the NO (relative gap x time[%]).

When the evolution of the UBs and LBs are considered observing the time as percentage (see Figs. 5 and 6), the algorithm takes

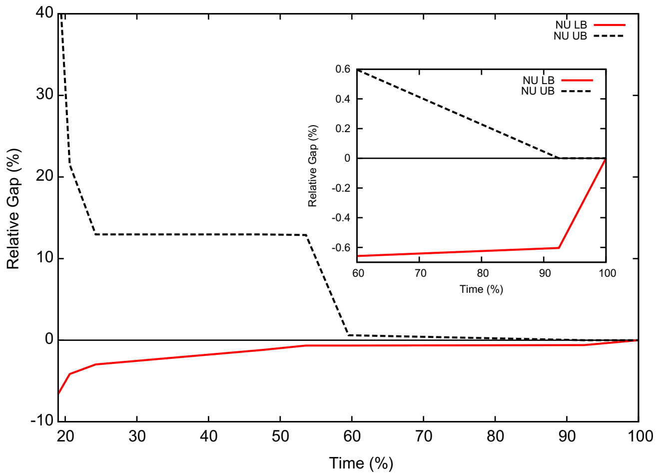

Fig. 6. Convergence of the algorithm for the perspective of the NU (relative gap *x* time[%]).

40% of the total time to close an optimality gap of less than 1% for both perspectives. In other others, after 60% of the algorithm running time, a very close optimal solution is available. Further, it takes 50% of the total time to the UB of the perspective of NU to improve close to the optimal solution, while it requires 30% of the total time to have the a similar improvement for the case of the NO.

5. Computational study

To assess the performance of the proposed algorithm on solving different test problems, the well-known Australian Post (AP) standard data set introduced by Ernst and Krishnamoorthy (1996, 1999) is used in the experiments. The names of the instances are coded as *APn.α*, where n is the number of nodes and $\alpha = \{2, 4, 6\}$ represents the selected discount factors 0.2, 0.4 and 0.6, respectively. Instances have sizes ranging from 20 to 100 nodes were selected. As $K \equiv N$ is considered in the experiments and as the AP data set has setup cost for the first 50 nodes only, hub fixed costs were generated for all instances using a Gaussian distribution with average equal to f_o and variance set to 40% to mimic how different installation costs vary in realistic problems. Here f_o represents the scaled difference in objective value between a scenario in which a virtual hub is located in the center of mass and a scenario in which all nodes are hubs, as introduced by Ebery, Krishnamoorthy, Ernst, and Boland (2000). Further, as done by Ebery et al. (2000), the nodes with higher flows are selected to have the higher fixed costs, hardening in general the selection of potential hubs. The value Γ_k of each hub k was generated by taking the total demand of the node $O_k + D_k$ added to a random fraction (U[15%, 50%]) of the total demand. The values of parameters a of the powerlaw congestion function and ϵ were set to 0.001 and 0.0001 respectively.

Table 1 presents the computational effort required to solve the chosen test bed. The instances were solved for both perspectives and the values for the number of iterations, the computer running time in seconds and the number of installed hubs recorded (see columns 2 through 7). The number of times the perspective of the NO takes longer to converge to the optimal solution than the case of the NU is also displayed in column *"Time NO/Time NU"*. Entries in column "Δ *# hubs*" represent the difference in the number of installed hubs.

In 44% of the instances, not only the allocation of the nonhub nodes to the hubs was different, but as well as the installed hub infrastructure. In average, the proposed algorithm takes 4.7 more time and 7.6 more iterations to converge to the optimal solution when addressing the perspective of the NO. However there are entries for the case of the NU that requires more iterations. For example, for instance AP100.2, the algorithm uses 7 fewer iterations to finish for the case of the NO than for the NU, but it consumes 76% more time. In average, each iteration of the algorithm requires 2.1 more computer effort for the perspective of the NO than for the NU. Averaging the time for each scale economy for both perspectives (see Table 2), an interesting feature can be seen. On average, instances having $\alpha = 0.6$ are harder to solve

Table 2
Average computing time for each economy of scale

α	Avg. Time (s) NO	Avg. Time (s) NU
0.2	3205.82	1752.34
0.4	2807.51	2150.95
0.6	6699.39	4687.20

Table 1
Computational effort.

Instance	NO			NU			Time NO/ Time NU	Δ # hubs
	# item	Time (s)	# hubs	# item	Time (s)	# hubs		
AP20.2	17	285.71	5	6	31.83	4	8.98	1
AP20.4	21	957.98	4	2	37.56	3	25.51	1
AP20.6	41	412.56	6	15	43.52	5	9.48	1
AP30.2	10	349.76	5	11	126.12	4	2.77	1
AP30.4	26	558.09	6	14	161.38	5	3.46	1
AP30.6	6	298.97	5	5	142.09	5	2.1	0
AP40.2	11	406.67	6	12	454.33	6	0.9	0
AP40.4	56	5811.28	6	18	628.4	6	9.25	0
AP40.6	62	10536.42	6	15	1084.21	5	9.72	1
AP50.2	12	874.28	7	14	450.19	6	1.94	1
AP50.4	13	444.41	7	17	518.38	7	0.86	0
AP50.6	13	412.92	7	12	186.66	7	2.21	0
AP70.2	12	2908.06	9	12	1286.36	9	2.26	0
AP70.4	18	6884.66	11	29	9758.49	11	0.71	0
AP70.6	26	10797.19	17	23	9208.9	17	1.17	0
AP100.2	6	14410.44	16	13	8165.22	14	1.76	2
AP100.4	6	2188.62	15	6	1801.51	15	1.21	0
AP100.6	30	17738.27	20	25	17457.8	20	1.02	0

for both perspectives, while instances with α equal to 0.4 and 0.2 are less demanding for the cases of the NO and the NU, respectively.

6. Conclusions

In this paper, two different perspectives to model the SAHLPC, a MINLP, are proposed. A very efficient generalized Benders decomposition algorithm is presented to solve both approaches. Instances having up to 10,000 integer and 49,500,000 continuous variables are solved to optimality in reasonable time. Future work could address a bi-criteria solution strategy.

Acknowledgments

The authors' research has been funded by CNPq, project 480757/2008–9.

References

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research, 190*, 1–21.

Aykin, T. (1994). Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research, 79*, 501–523.

Aykin, T. (1995). Network policies for hub-and-spoke systems with applications to the air transportation system. *Transportation Science, 29*, 201–221.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik, 4*, 238–252.

Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2002). Hub location problems. In Z. Drezner & H. W. Hamacher (Eds.), *Facility Location: Applications and Theory* (1^a ed., pp. 373–407). Springer. chapter 12.

Contreras, I., Díaz, J. A., & Fernández, E. (2011). Branch and price for large-scale capacitated hub location problems with single assignment. *INFORMS Journal on Computing, 23*, 41–55.

Costa, M. d. G., Captivo, M. E., & Clímaco, J. (2008). Capacitated single allocation hub location problem a bi-criteria approach. *Computers and Operations Research, 35*, 3671–3695.

Ebery, J., Krishnamoorthy, M., Ernst, A., & Boland, N. (2000). The capacitated multiple allocation hub location problema: Formulations and algorithms. *European Journal of Operational Research, 120*, 614–631.

Elhedhli, S., & Hu, F. (2005). Hub-and-spoke network design with congestion. *Computers and Operations Research, 32*, 1615–1632.

Elhedhli, S., & Wu, H. (2010). A lagrangean heuristic for hub-and-spoke system design. *INFORMS Journal on Computing, 22*, 282–296.

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science, 4*, 139–154.

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

Geoffrion, A. (1972). Generalized Benders decomposition. *Journal of optimization Theory and Applications, 10*, 237–260.

Gillen, D., & Levinson, D. (1999). The full cost of air travel in the california corridor. *Transportation Research Record: Journal of the Transportation Research Board, 1662*, 1–9.

Kleinrock, L. (1964). *Communication nets: Stochastic message flow and delay*. New York: McGraw-Hill.

Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: A review. *Location Science, 6*, 307–335.

Labbé, M., Yaman, H., & Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. *Mathematical Programming: Series A, 102*, 371–405.

Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as $m/d/c$ queues. *Computer and Operations Research, 30*, 983–1003.

Papadakos, N. (2008). Practical enhancements to the Magnanti–Wong method. *Operations Research Letters, 36*, 444–449.

Rodriguez, V., Alvarez, M., & Barcos, L. (2007). Hub location under capacity constraints. *Transportation Research Part E, 43*, 495–505.