Hop-Level Flow Formulation for the Survivable Network Design with Hop Constraints Problem

A. Ridha Mahjoub
Université Paris-Dauphine, LAMSADE, Place du Maréchal De Lattre de Tassigny, 75775, Paris Cedex 16, France

Luidi Simonetti and Eduardo Uchoa
Departamentos de Computação e Engenharia de Produção, Universidade Federal Fluminense, Rua Passo da Pátria, 156 CEP 24210-240, Niterói, Brazil

The hop-constrained survivable network design problem consists of finding a minimum cost subgraph containing *K* edge-disjoint paths with length at most *H* joining each pair of vertices in a given demand set. When all demands have a common vertex, the instance is said to be rooted. We propose a new extended formulation for the rooted case, called hop-level multicommodity flow (MCF), that can be significantly stronger than the previously known formulations, at the expense of having a larger number of variables and constraints, growing linearly with the number of edges and demands and quadratically with *H*. However, for the particular case where $H = 2$, it can be specialized into a very compact and efficient formulation. Even when $H = 3$, hop-level-MCF can still be quite efficient and it has solved several instances from the literature for the first time. **NETWORKS, Vol. 61(2), 171–179 2013**

Keywords: network design; survivability; extended formulations

1. INTRODUCTION

Let $G = (V, E)$ be an undirected graph with $n+1$ vertices, numbered from 0 to n, and m edges with positive costs c_e, $e \in E$; let $D \subseteq V \times V$ be a set of demands; and let $K \geq 1$ and $H \geq 2$ be natural numbers. The hop-constrained survivable network design problem (HSNDP) consists of finding a subgraph of G with minimum cost containing, for each demand $(u, v) \in D$, K edge-disjoint (u, v)-paths with at most H edges. If all demands have a common vertex, w.l.o.g. the vertex 0, we say that the instance is rooted, otherwise it is unrooted. In this article, we only consider rooted instances. Therefore, to simplify the notation, a demand $(0, d)$ will be identified by its destination vertex d, so the set D is assumed to be a set of destinations. A vertex that does not belong to any demand is a Steiner vertex. Instances without Steiner vertices are said to be spanning. Figure 1 depicts an optimal solution of a rooted spanning instance with $K = 3$ and $H = 3$. For example, in that solution, we can find edge-disjoint paths 0-18-14-15, 0-14-16-15, and 0-13-15 joining vertices 0 and 15.

The HSNDP is a quite general network design problem; the parameter H models quality of service requirements, whereas parameter K models the desired level of network survivability (see [14]). An even more general version of the HSNDP can consider potentially distinct values $K(d)$ and $H(d)$ for each $d \in D$ to model demand importance. Anyway, some particular cases of the HSNDP are well-known NP-hard problems:

- When $|D| = 1$ (single-demand HSNDP), the problem is polynomial for $H \in \{2, 3\}$ and NP-hard for $H \geq 4$ [13].
- When the instance is rooted and $K = 1$, the HSNDP is equivalent to the Steiner Tree with hop constraints problem. When the instance is also spanning, the HSNDP is equivalent to the spanning tree with hop constraints problem. This latter problem is NP-hard even when $H = 2$ (see [9]).

This work proposes a new extended formulation for the rooted HSNDP, called hop-level-multicommdity flow (HL-MCF), inspired by a previous extended formulation known as Hop-MCF [3]. Hop-MCF is already large, having $O(|D| \cdot H \cdot m)$ variables and $O(|D| \cdot H \cdot n)$ constraints. The new formulation increases both dimensions by a factor of H. The lower bounds given by the linear relaxation (LR) of HL-MCF can be significantly stronger than those given by Hop-MCF. However, when solving an instance to optimality by branch-and-bound, the time to solve each node can be much larger. Overall, HL-MCF is still usually better when $H \leq 3$, solving several open instances from the literature with up to 40 vertices and 30 demands. The computational results of HL-MCF

Received September 2011; accepted July 2012
Correspondence to: E. Uchoa; e-mail: uchoa@producao.uff.br
Contract grant sponsors: CNPq and FAPERJ
DOI 10.1002/net.21483
Published online 7 September 2012 in Wiley Online Library (wileyonlinelibrary.com).

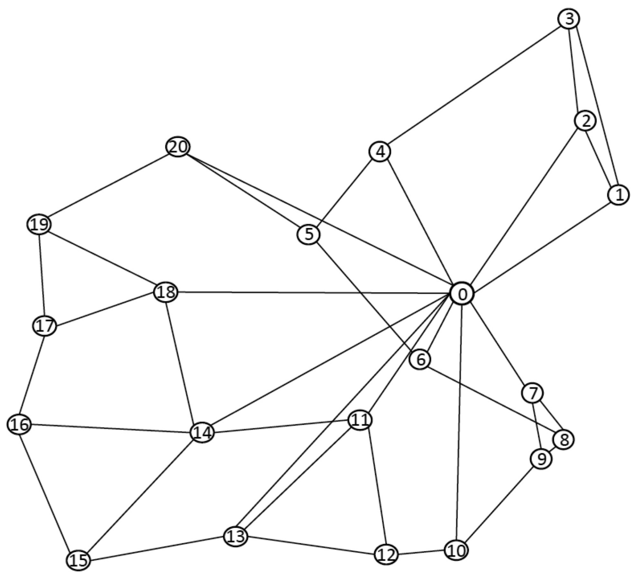

FIG. 1. Optimal solution of a rooted spanning instance with $K = 3$ and $H = 3$; $n = 20$, complete graph, Euclidean costs.

for that case are also better than those obtained by branch-and-cut algorithms that separate inequalities over the natural edge variables. Moreover, when $H = 2$, HL-MCF can be specialized into a formulation having less than $3m$ variables and $4m + 2|D|$ constraints. This represents a breakthrough in the practical solution of that case, as quite large instances can be quickly solved to optimality.

2. BACKGROUND

First, we define the polyhedra $P(H, K, d)$ associated with the single-demand HSNDP. Let $\mathcal{F}(H, K, d)$ be the collection of all edge subsets $F \subseteq E$ such that (V, F) contains K edge-disjoint $(0, d)$-paths with at most H edges. For every $F \subseteq E$, the 0-1 vector $\chi^F \in R^m$ such that $\chi^F(e) = 1$ iff $e \in F$ is the incidence vector of F. Then,

$$P(H, K, d) = \text{Conv}\{\chi^F : F \in \mathcal{F}(H, K, d)\}.$$

Dahl initiated the investigation of these polyhedra by characterizing them in the case where $K = 1$ and $H \in \{2, 3\}$ [4]. For the case where $K = 2$ and $H \in \{2, 3\}$, a complete description of $P(H, K, d)$ in the natural space of edge variables was provided by Huygens et al. [12]. Dahl et al. [5] found the description for the case where $H = 2$, for any K. All those results were generalized by Bendali et al. [1] who proved that *st*-cut and *L*-path-cut inequalities suffice for describing $P(H, K, d)$ when $H \in \{2, 3\}$, for any K.

There is little hope of finding a full description of $P(H, K, d)$ when $H \geq 4$, as the associated single-demand HSNDP is NP-hard. Huygens and Mahjoub [11] give an integer programming formulation (i.e, a polyhedron that relaxes $P(H, K, d)$ by including additional fractional points) for the case $H = 4$ and $K = 2$. No formulation in the natural space of the edge variables is known for any H and any K. Botton et al. [2] provided such a general extended formulation over a hop-indexed network (called HopE) for the single-demand HSNDP. In fact, hop-indexed networks have been used to formulate network design problems with hop-constraints since [8]. When $H \in \{2, 3\}$, it was proved that HopE is integral. The reported experiments suggest that the LR of that formulation is very strong even when $4 \leq H \leq 6$, obtaining integral optimal solutions in 27,994 out of the 28,000 tested instances. This means that the projection of that extended formulation is, for most algorithmic purposes, almost equivalent to a description of $P(H, K, d)$, at least when H is in that range.

The polyhedron $P(H, K, D)$ associated with the multi-demand HSNDP is defined next. Let $\mathcal{F}(H, K, D)$ be the collection of all edge subsets $F \subseteq E$ such that (V, F) contains K edge-disjoint $(0, d)$-paths with at most H edges for every $d \in D$. Then,

$$P(H, K, D) = \text{Conv}\{\chi^F : F \in \mathcal{F}(H, K, D)\}.$$

It can be seen that $\bar{P}(H, K, D) = \cup_{d \in D} P(H, K, d)$ is a formulation for $P(H, K, D)$ (in fact, the intersection of any formulation for each individual demand would still be a valid formulation for the multidemand HSNDP). However, practical experiments showed that $\bar{P}(H, K, D)$ is usually not a good approximation of $P(H, K, D)$, even when $H \in \{2, 3\}$.

To strengthen $\bar{P}(H, K, D)$, it is necessary to look for additional joint inequalities, that is, inequalities that cannot be derived by only considering each isolated demand. If possible, one would like to find joint inequalities that may define facets of $P(H, K, D)$. Huygens et al. [10] investigated the case where $K = 2$ and $H \in \{2, 3\}$, identifying facet-defining inequalities and constructing a branch-and-cut algorithm. Similar work was done by Diarrassouba et al. [6, 7] for the case where $K = 3$ and $H \in \{2, 3\}$. In both works, experimentation showed that the separation of joint inequalities can improve significantly the lower bounds; however, the integrality gaps are still too large for solving many instances with more than 20 demands to optimality. There are some potential drawbacks in continuing that approach:

- The currently known families of joint inequalities are complex enough to require complicated heuristic separation algorithms to be used. New families of strong joint inequalities are likely to be even more complex (finding them may be difficult research work) and require even more complicated algorithms.
- The HSNDP is a very generic problem. Yet the known families of strong joint inequalities are effective (or even valid) only for quite particular cases. The simple fact that there is no known HSNDP formulation in the space of the edge variables for $H \geq 5$ attests to the difficulties of finding generic inequalities.

In this work, we try to find lower bounds significantly better than those provided by $\bar{P}(H, K, D)$ in an alternative way, by exploring extended formulations. See [15] for a recent discussion on the pros and cons of the "extended formulation approach" when compared to the "facet-finding approach".

3. HOP-MCF FORMULATION

An extended formulation was recently proposed for the general HSNDP [3]. It basically consists of combining the extended formulations given in [2] for each single demand into a single formulation. That article proposes handling the large size of the resulting extended formulation by means of a Benders decomposition. Let $V' = V - \{0\}$ and $E' = E \backslash \delta(0)$, where $\delta(i)$ represents the set of edges adjacent to a vertex i. For each demand $d \in D$, define the hop layered directed graph $G^d_H = (V^d_H, A^d_H)$, where $V^d_H = \{(0,0)\} \cup \{(i,h) : i \in V';\ 1 \leq h \leq H-1\} \cup \{(d,H)\}$. Assuming that G is a complete graph, $A^d_H =$

$$\{[0,j,1] = [(0,0),(j,1)] : j \in V'\}$$
$$\cup\ \{[i,j,h] = [(i,h-1),(j,h)] : i,j \in V',$$
$$i \neq d, i \neq j;\ 2 \leq h \leq H-1\}$$
$$\cup\ \{[i,d,h] = [(i,h-1),(d,h)] : i \in V' - \{d\};\ 2 \leq h \leq H\}.$$

Each arc in A^d_H is identified by a triple $[i,j,h]$, giving its origin, destination, and hop. When G is not complete, if $(i,j) \notin E$, arcs of form $[i,j,h]$ and $[j,i,h]$ are omitted from A^d_H. Figure 2 depicts an example of such an auxiliary network. For each $d \in D$, and for each arc $[i,j,h]$ in A^d_H, define binary flow variables f^{dh}_{ij}. For each edge (i,j) in E, define design binary variables x_{ij}. Let $\delta^-(i,h,d)$ and $\delta^+(i,h,d)$ denote, respectively, the set of arcs in A^d_H entering and leaving vertex (i,h). The Hop-MCF formulation follows:

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{1}$$

$s.t.$

$$\sum_{a\in\delta^-(i,h,d)} f_a - \sum_{a\in\delta^+(i,h,d)} f_a = 0, \qquad \forall d \in D; (i,h) \in V^d_H, i \notin \{0,d\} \tag{2}$$

$$\sum_{h=1}^{H} \sum_{a\in\delta^-(d,h,d)} f_a = K, \qquad \forall d \in D \tag{3}$$

$$f^{d1}_{0j} \leq x_{0j}, \qquad \forall d \in D; (0,j) \in \delta(0) \tag{4}$$

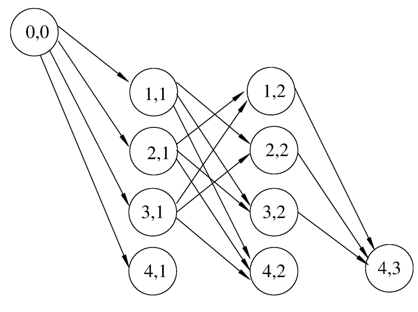

FIG. 2. Example of auxiliary graph G^d_H: G complete, $n = 4$, $d = 4$, and $H = 3$.

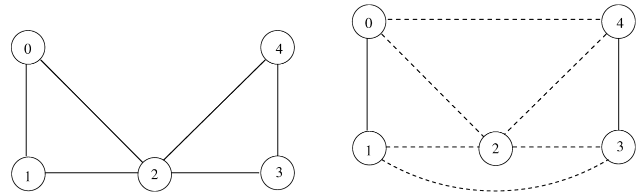

FIG. 3. Instance *inst4*: optimal solution and fractional solution of Hop-MCF.

$$\sum_{h=2}^{H-1} \left(f^{dh}_{ji} + f^{dh}_{ij}\right) \leq x_{ij}, \qquad \forall d \in D; (i,j) \in E' \backslash \delta(d) \tag{5}$$

$$\sum_{h=2}^{H} f^{dh}_{jd} \leq x_{jd}, \qquad \forall d \in D; (j,d) \in \delta(d) \backslash \delta(0) \tag{6}$$

$$0 \leq x, f \leq 1. \tag{7}$$

In the single-demand case, when $H \in \{2,3\}$, the projection of Hop-MCF onto the x space is exactly $P(H,K,d)$. Therefore, Hop-MCF projects onto $\bar{P}(H,K,D)$ in the multi-demand case, when $H \in \{2,3\}$. The typical integrality gaps of Hop-MCF over Euclidean instances, as reported in [3], are between 5% and 25%.

To illustrate the potential weakness of the Hop-MCF formulation when several demands are present, we define a rooted spanning instance with $n = 4$, $K = 2$, and $H = 3$, called *inst4*. The graph is complete and the costs correspond to Euclidean distances. Vertex coordinates are $((0,1),(0,0),(0,1),(0,2),(2,1))$. Figure 3 depicts on its left the optimal integral solution x^* of *inst4* with cost 6.83. On the right, the LR of Hop-MCF $\bar{x}$ with cost 6.41 is shown; dashed lines represent variables with value $1/2$. As Hop-MCF is integral for a single demand when $H = 3$, for each demand $d \in D$, there must be fractional $0-d$ paths with length up to 3 over $\bar{x}$ summing 2. For example, for demand 1 there is path 0-1 with value 1, path 0-2-1 with value $1/2$, and path 0-4-3-1 with value $1/2$. When $H \in \{2,3\}$, it is not possible to cut a fractional solution of Hop-MCF with inequalities that only consider individual demands.

4. HOP-LEVEL MCF FORMULATION

It is well known that directed formulations of network design problems, when available, are much stronger that their undirected counterparts. The relative weakness of known HSNDP formulations, including Hop-MCF, is related to the impossibility of directing the solutions, as both orientations of an edge can be used in the paths for different demands. The proposed formulation tries to remedy this difficulty by introducing the concept of solution level. Given a solution T, we can partition V into $L+2$ levels, according to their distances to 0 in T. In the basic HL-MCF presented in this article, L is set as equal to H (in its generalizations mentioned in Section 7, L can be greater than H). The set of Steiner vertices is defined as $S = V' \setminus D$. Level 0 only contains vertex 0; level l, $1 \le l \le L$, contains vertices at distance l from 0; and level $L+1$ contains the vertices in S that are not connected to 0 in T. Besides the design variables x, HL-MCF also has:

- Binary variables w_i^l, $i \in D, 1 \le l \le L$ and $i \in S, 1 \le l \le L+1$, indicating that vertex i is in level l. As optimal solutions never have Steiner vertices in level L, variables w_i^L, $i \in S$, can be fixed to 0. Variable w_0^0 is fixed to 1.
- Binary variables $y_{ij}^{l_1 l_2}$ indicating that edge (i,j) belongs to T, i is in level l_1 and j in level l_2. For each $(0,j) \in \delta(0)$, there is a single variable y_{0j}^{01}. Each $e = (i,j) \in E'$ is associated with a set of $3(L-1)$ variables $\{y_{ij}^{ll} : 1 \le l \le L-1\} \cup \{y_{ij}^{l(l+1)}, y_{ji}^{l(l+1)} : 1 \le l \le L-1\}$. Variables y_{ij}^{LL} are not necessary because an edge between vertices in level L can not belong to a path from 0 with length at most H.
- Binary flow variables $g_{ij}^{dhl_1 l_2}$ associated with $|D|$ auxiliary hop-level networks to be defined.

The x and (w,y) variables are linked by the following constraints:

$$w_0^0 = 1, \tag{8}$$

$$\sum_{l=1}^{L} w_i^l = 1, \quad \forall i \in D \tag{9}$$

$$\sum_{l=1}^{L+1} w_i^l = 1, \quad \forall i \in S \tag{10}$$

$$w_i^L = 0, \quad \forall i \in S \tag{11}$$

$$w_j^1 = y_{0j}^{01} = x_{0j}, \quad \forall (0,j) \in \delta(0) \tag{12}$$

$$\sum_{l=1}^{L-1} y_{ij}^{ll} + \sum_{l=1}^{L-1} \left(y_{ij}^{l(l+1)} + y_{ji}^{l(l+1)} \right) = x_{ij}, \quad \forall (i,j) \in E' \tag{13}$$

$$\left.\begin{aligned} y_{ij}^{11} + y_{ij}^{12} &\le w_i^1, \\ y_{ij}^{11} + y_{ji}^{12} &\le w_j^1, \end{aligned}\right. \quad \forall (i,j) \in E' \tag{14}$$

$$\begin{aligned} y_{ij}^{ll} + y_{ij}^{l(l+1)} + y_{ji}^{(l-1)l} &\le w_i^l, \\ y_{ij}^{ll} + y_{ji}^{l(l+1)} + y_{ij}^{(l-1)l} &\le w_j^l, \\ \forall (i,j) \in E'; l = 2, \ldots, L-1 & \end{aligned} \tag{15}$$

$$\left.\begin{aligned} y_{ji}^{(L-1)L} &\le w_i^L, \\ y_{ij}^{(L-1)L} &\le w_j^L, \end{aligned}\right. \quad \forall (i,j) \in E' \tag{16}$$

$$w_i^l \le \sum_{(j,i) \in E'} y_{ji}^{(l-1)l}, \quad \forall i \in V'; l = 2, \ldots, L-1 \tag{17}$$

$$K \cdot w_i^L = \sum_{(j,i) \in E'} y_{ji}^{(L-1)L}, \quad \forall i \in D \tag{18}$$

$$0 \le x, w, y \le 1. \tag{19}$$

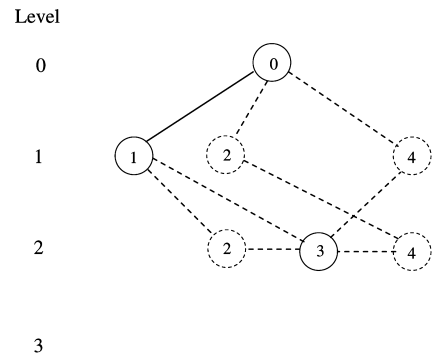

FIG. 4. Instance *inst4*: level-expanded representation of the fractional solution of Hop-MCF that is not feasible in HL-MCF.

Constraints (8)–(11) state that each vertex should be in exactly one of the possible levels. Constraints (12) and (13) make the connection between the new (w,y) variables and the original edge variables x. Constraints (14)–(16) assure that a variable $y_{ij}^{l_1 l_2}$ can only be 1 if both $w_i^{l_1}$ and $w_j^{l_2}$ are one. Constraints (17) state that a vertex can only be in level l if it is reached by at least one edge from level $l-1$. Constraints (18) are based on the fact that a demand vertex in level L must be reached by exactly K vertices from level $L-1$ in any optimal solution. It can be checked that for any fixed binary solution x, there is a single binary solution (w,y) that satisfies (8)–(19). That unique solution indeed sorts the vertices according to their distances from 0 in the solution x. For example, the integral solution of *inst4* shown in Figure 3 would force vertices 1 and 2 to be in level 1 and 3 and 4 to be in level 2.

However, fixing a fractional $\bar{x}$ into the right-hand side of (12) and (13) usually forces (w,y) solutions of the linear system (8)–(19) that split vertices and edges into different levels. Note that those levels are the same for every demand. For example, consider the fractional solution $\bar{x}$ of *inst4* shown in Figure 3. Constraints (12) imply that $w_1^1 = 1$, $w_2^1 = 1/2$, $w_3^1 = 0$, and $w_4^1 = 1/2$. By taking the other constraints into account, it can be shown (computationally or by a long *ad hoc* argument) that there is a single solution $(\bar{w}, \bar{y})$ that is compatible with $\bar{x}$ (this is not a general property, there are other cases where there are several solutions compatible with a given $\bar{x}$). The solution is depicted in the level-expanded graph in Figure 4; the horizontal positioning of the vertices and edges indicates the values of $(\bar{w}, \bar{y})$. For instance, $y_{43}^{12} = y_{34}^{22} = 1/2$. Consider the demand (0,1). In the level-expanded graph given by $(\bar{w}, \bar{y})$, there is path 0-1 with value 1 and path 0-4-3-1 with value 1/2. But the remaining path 0-2-4-3-2-1 is not valid because its length is greater than 3. What happened is that the splitting of vertex 2 removed the path 0-2-1 that existed in the original fractional solution. Now, it is possible to cut the fractional $(\bar{w}, \bar{y})$ solution by only considering demand 1, which indirectly also cuts the fractional $\bar{x}$ solution.

We need to define auxiliary networks to enforce the existence of the required paths in the level-expanded graph. For each $d \in D$, we define hop-level directed graphs $G_{HL}^d =$

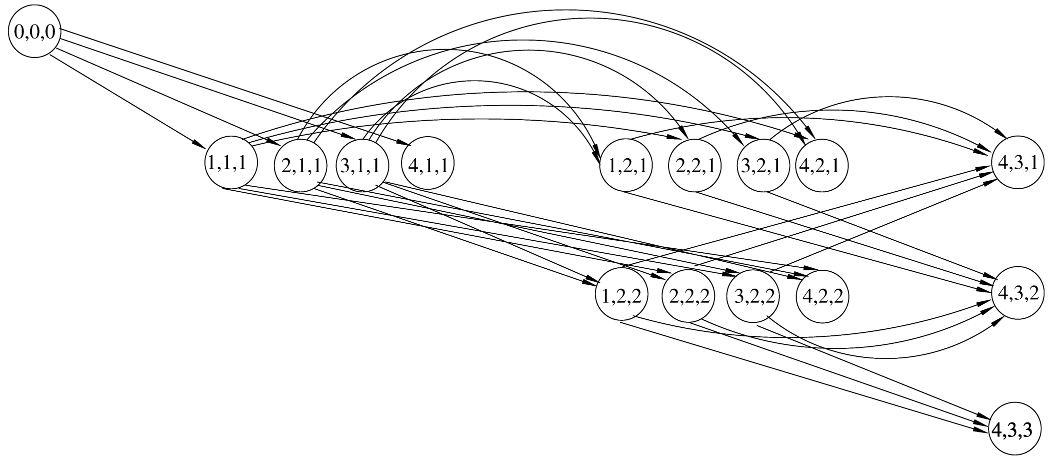

FIG. 5. Example of auxiliary graph G^d_{HL}: G complete, $n = 4$, $d = 4$, and $H = L = 3$.

(V^d_{HL}, A^d_{HL}), where $V^d_{HL} = \{(0,0,0)\} \cup \{(i,h,l) : i \in V';\ 1 \leq h \leq H-1;\ 1 \leq l \leq h\} \cup \{(d,H,l) : 1 \leq l \leq L\}$, and $A^d_{HL} =$

$$\begin{aligned}
&\{[0,j,1,0,1] = [(0,0,0),(j,1,1)] : j \in V'\}\\
&\cup \{[i,j,h+1,l,l'] = [(i,h,l),(j,h+1,l')] :\\
&\quad i,j \in V', i \neq j, i \neq d;\ 1 \leq h \leq H-2;\\
&\quad 1 \leq l \leq h;\ \max(l-1,1) \leq l' \leq l+1\}\\
&\cup \{[i,d,H,l,l'] = [(i,H-1,l),(d,H,l')] : i \in V', i \neq d;\\
&\quad 1 \leq l \leq L-1;\ \max(l-1,1) \leq l' \leq l+1\}
\end{aligned}$$

Again, if G is not complete, the arcs corresponding to missing edges are removed. Each arc in A^d_{HL} is identified by a tuple $[i,j,h,l_1,l_2]$, giving its origin, destination, hop, origin level, and destination level. For each such arc, we define a binary flow variable $g^{dhl_1l_2}_{ij}$. Let $\delta^-(i,h,l,d)$ and $\delta^+(i,h,l,d)$ denote, respectively, the set of arcs in A^d_H entering and leaving vertex (i,h,l). Figure 5 depicts an example of such an auxiliary network. The new formulation HL-MCF is defined by the objective function (1), subjected to (8)–(19) and to the following constraints:

$$\sum_{a\in\delta^-(i,h,l,d)} g_a - \sum_{a\in\delta^+(i,h,l,d)} g_a = 0, \qquad d \in D; (i,h,l) \in V^d_{HL}, i \notin \{0,d\} \quad (20)$$

$$\sum_{h=1}^{H} \sum_{a\in\delta^-(d,h,l,d)} g_a = K \cdot w^l_d, \quad d \in D;\ 1 \leq l \leq L \quad (21)$$

$$g^{d101}_{0j} \leq y^{01}_{0j}, \quad d \in D; (0,j) \in \delta(0), j \neq d \quad (22)$$

$$g^{d101}_{0d} = y^{01}_{0d}, \quad d \in D; (0,d) \in \delta(0) \quad (23)$$

$$\sum_{h=l+1}^{H-1} \left(g^{dhll}_{ji} + g^{dhll}_{ij}\right) \leq y^{ll}_{ij}, \qquad d \in D; (i,j) \in E'\backslash\delta(d); 1 \leq l \leq L-2 \quad (24)$$

$$\sum_{h=l+2}^{H-1} g^{dh(l+1)l}_{ji} + \sum_{h=l+1}^{H-1} g^{dhl(l+1)}_{ij} \leq y^{l(l+1)}_{ij}, \qquad d \in D; (i,j) \in E'\backslash\delta(d); 1 \leq l \leq L-2 \quad (25)$$

$$\sum_{h=l+2}^{H} g^{dh(l+1)l}_{jd} \leq y^{l(l+1)}_{dj}, \qquad d \in D; (j,d) \in \delta(d)\backslash\delta(0); 1 \leq l \leq L-2 \quad (26)$$

$$\sum_{h=l+1}^{H} g^{dhll}_{jd} \leq y^{ll}_{jd}, \qquad d \in D; (j,d) \in \delta(d)\backslash\delta(0); 1 \leq l \leq L-1 \quad (27)$$

$$\sum_{h=l+1}^{H} g^{dhl(l+1)}_{jd} \leq y^{l(l+1)}_{jd}, \qquad d \in D; (j,d) \in \delta(d)\backslash\delta(0); 1 \leq l \leq L-1. \quad (28)$$

Constraints (20) and (21) define a flow of K units from $(0,0,0)$ to vertices (d,h,l), vertices in level l receive $K \cdot w^l_d$ units. Constraints (22)–(28) link the flow to the y variables. The complete HL-MCF formulation has $O(|D| \cdot H \cdot L \cdot m)$ variables and $O(|D| \cdot H \cdot L \cdot n)$ constraints, an increase by a factor of $L = H$ (in both dimensions) with respect to Hop-MCF.

Theorem 1. *HL-MCF is at least as strong as Hop-MCF in terms of the bounds provided by their LRs.*

Proof 1. *A fractional solution $(\bar{x}, \bar{w}, \bar{y}, \bar{g})$ of HL-MCF can be converted into a fractional solution $(\bar{x}, \bar{f})$ of Hop-MCF by defining $\bar{f}^{dh}_{ij} = \sum_{l_1,l_2} \bar{g}^{dhl_1l_2}_{ij}$, where the summation is defined over the appropriate indices. It can be checked that if $\bar{g}$ satisfies (20) and (21) then $\bar{f}$ satisfies (2) and (3). Moreover, if $(\bar{x}, \bar{w}, \bar{y}, \bar{g})$ satisfies (8)–(17) and (22)–(28), then $(\bar{x}, \bar{f})$ satisfies the coupling constraints (4)–(6).*

However, HL-MCF can be significantly stronger than Hop-MCF. For example, in *inst4*, the fractional solution of

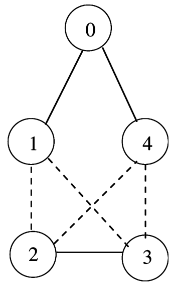

FIG. 6. Instance *inst4*: Level-expanded representation of the fractional solution of HL-MCF.

HL-MCF has value 6.71 and is depicted as a level-expanded graph in Figure 6. We remark that this new fractional solution avoids splitting vertices.

5. HL2: SPECIALIZING THE HL-MCF WHEN $H = 2$

In the particular case when $H = 2$, the HL-MCF can be rewritten in a much more compact way, without the need of having an auxiliary network flow for each demand. The crucial observation is that each variable g is associated with a single y variable in constraints (22)–(28) in that case. This allows replacing all those networks by constraints over the y variables. The following constraints are valid when $H = 2$:

$$\sum_{(i,j)\in\delta(d)\setminus\delta(0)} y_{ij}^{11} \geq (K-1)\cdot y_{0d}^{01} \quad d \in D. \tag{29}$$

Theorem 2. *When $H = 2$, the formulation (1), subjected to (8)–(19) and (29), that will be called* HL2*, is equivalent to HL-MCF in terms of the bounds provided by their LRs.*

Proof 2. *Given a solution $(\bar{x}, \bar{w}, \bar{y})$ of (8)–(19) and (29), Algorithm 1 gives a solution $(\bar{x}, \bar{w}, \bar{y}, \bar{g})$ also satisfying (20)–(28). To better understand how the algorithm constructs a $\bar{g}$ flow for a given demand $d \in D$, let us consider what happens in the extreme cases where $\bar{y}_{0d}^{01}$ is binary. If that variable has value zero, this means that vertex d is in level 2. The loop in lines 3-6 will then construct the appropriate $\bar{g}$ flow and the remaining lines will have no effect. Conversely, if $\bar{y}_{0d}^{01} = 1$, this means that d is in level 1. Then lines 3-6 will have no effect and the code in lines 7-17 will then construct the $\bar{g}$ flow. Of course, a fractional $\bar{y}_{0d}^{01}$ will yield a $\bar{g}$ flow that is a combination of the extreme cases. Now, we check that $(\bar{x}, \bar{w}, \bar{y}, \bar{g})$ indeed has the desired properties:*

1. *By construction, flow conservation constraints (20) are satisfied by $\bar{g}$.*
2. *Constraints (18) imply that constraints (21) for $l = 2$ are satisfied.*
3. *Constraints (29) imply the satisfaction of (21) for $l = 1$. As (29) is an inequality, Algorithm 1 has a break condition to make sure that the $\bar{g}$ variables in those constraints are only increased until (21) for $l = 1$ is satisfied as an equality.*
4. *By construction, $\bar{g}_{0j}^{d101} \leq \bar{y}_{jd}^{11} + \bar{y}_{jd}^{12}$. Then, constraints (14) imply that the corresponding inequalities (22) are satisfied.*
5. *Constraints (23) are satisfied by construction.*
6. *Constraints (24)–(26) do not apply when $H = 2$.*
7. *Constraints (27)–(28) are satisfied by construction.*

Conversely, constraints (21) for $l = 1$ and constraints (23), (27) already imply constraints (29).

Algorithm 1. Converting a solution of HL2 to a solution of Hop-MCF

Input: $(\bar{x}, \bar{w}, \bar{y})$ satisfying (8)–(19) and (29)
Output: $(\bar{x}, \bar{w}, \bar{y}, \bar{g})$ satisfying (20)–(28)

1 $\bar{g} = 0$;
2 **foreach** $d \in D$ **do**
3   **foreach** $(j, d) \in \delta(d)\setminus\delta(0)$ **do**
4     $\bar{g}_{jd}^{d212} = \bar{y}_{jd}^{12}$;
5     $\bar{g}_{0j}^{d101} = \bar{y}_{jd}^{12}$;
6   **end**
7   $\bar{g}_{0d}^{d101} = \bar{y}_{0d}^{01}$;
8   $L1F = (K-1)\cdot\bar{y}_{0d}^{01}$;
9   **foreach** $(j, d) \in \delta(d)\setminus\delta(0)$ **do**
10     $z = \min\{\bar{y}_{jd}^{11}, L1F\}$;
11     $\bar{g}_{jd}^{d211} = z$;
12     $\bar{g}_{0j}^{d101} += z$;
13     $L1F -= z$;
14   **if** $L1F = 0$ **then**
15   break;
16   **end**
17   **end**
18 **end**

Formulation HL2 can be further simplified. Noting that $w_d^2 = 1 - y_{0d}^{01}$ when $d \in D$, variables w can be eliminated without increasing the number of non-zeros. By defining the objective function in terms of the y variables, the x variables can be eliminated too. Eliminating the constraints in (8)–(19) that do not apply or are now redundant, HL2 can be written as:

$$\min \sum_{(0,j)\in\delta(0)} c_{0j} y_{0j}^{01} + \sum_{(i,j)\in E'} c_{ij}\left(y_{ij}^{11} + y_{ij}^{12} + y_{ji}^{12}\right) \tag{30}$$

$$s.t.$$

$$\begin{aligned} y_{ij}^{11} + y_{ij}^{12} &\leq y_{0i}^{01}, \\ y_{ij}^{11} + y_{ji}^{12} &\leq y_{0j}^{01}, \end{aligned} \qquad \forall (i,j) \in E' \tag{31}$$

$$\begin{aligned} y_{ij}^{12} + y_{0j}^{01} &\leq 1, \\ y_{ji}^{12} + y_{0i}^{01} &\leq 1, \end{aligned} \qquad \forall (i,j) \in E' \tag{32}$$

$$\sum_{(j,d)\in E'} y_{jd}^{12} + K\cdot y_{0d}^{01} = K, \qquad \forall d \in D \tag{33}$$

$$\sum_{(i,j)\in\delta(d)\setminus\delta(0)} y_{ij}^{11} \geq (K-1)\cdot y_{0d}^{01}, \qquad \forall d \in D \quad (34)$$

$$y \geq 0. \quad (35)$$

6. COMPUTATIONAL EXPERIMENTS

The experiments were performed on an Intel Core Duo P7350@2.00 GHz machine, with CPLEX 12.1 Mixed-Integer Program (MIP) solver. The first set of tests were conducted on the medium-sized rooted instances used in [3], complete graphs with 21 vertices associated with random points in a square, Euclidean distances. The root vertex is in the center on instances TC20-5 and TC20-10, and on a corner on instances TE20-5 and TE20-10. The numbers 5 and 10 refer to the number of demands. To cover an important case, we also added spanning instances TC20-20 and TE20-20. We used the LRs of Hop-MCF and HL-MCF on all those instances, for each $H \in \{2,3,4,5\}$ and each $K \in \{1,2,3\}$. Each row in Table 1 presents averages over a group of six instances, except the last row that presents averages over all the 72 instances. Columns (Gap) are the average percentage gaps of the LRs with respect to the optimal solutions (or the best known solutions, in some of the instances with 20 demands) and columns (TLR) give the time in seconds to solve the LR. Note that, when $H = 2$, the times correspond to solving HL-MCF, not HL2. Column (Gap red(%)) is the percentage of the gap of the Hop-MCF that was "cut" by HL-MCF; a figure of 100 indicates that HL-MCF reduced the gap to zero, a figure of 0 would indicate that both formulations gave the same lower bound. Column (T factor) gives the factor of increase in time.

The results in Table 1 indicate that the gap reduction obtained by HL-MCF decreases with increasing H and K. Formulation HL-MCF is stronger than Hop-MCF because the concept of solution level forces vertices to be split in typical fractional solutions. This splitting can break many fractional $0-d$ paths of small length, turning them into paths with larger length (as in the example of Fig. 4). However, as H increases, the resulting paths are more likely to be still feasible. A possible explanation for the effect of K on the gap reduction is the following. Instances with larger K have denser fractional solutions, therefore most of the vertices are likely to have at least one short path linking it to the root. The concentration of vertices in levels 1 and 2 leads to fewer splittings. We provide additional statistics about the effect of the number of demands on the integrality gaps. The respective average gaps achieved by Hop-MCF and HL-MCF were 7.96% and 1.46%, when $|D| = 5$; 14.79% and 2.73%, when $|D| = 10$; and 19.36% and 3.45%, when $|D| = 20$. It is remarkable that the average gap reduction is almost constant (about 82%) on all those three cases.

The practical relevance of the new formulation to improving the state-of-the-art on the algorithmic solution of rooted HSNDP instances can be summarized as follows:

- Formulation HL-MCF is much stronger than Hop-MCF when $K = 1$. However, this fact does not lead to an improvement of the state-of-the-art. The rooted HSNDP with $K = 1$ is equivalent to the Steiner/spanning tree with hop constraints problem (HSTP). A stronger directed variant of Hop-MCF [8] can be applied in those problems. More recently, it was found that it is even stronger to model the HSTP as a directed Steiner tree problem, without the need of having separated auxiliary networks for each demand [9]. This enables the solution of instances with a few hundred demands to optimality without branching. Anyway, the strength of HL-MCF when $K = 1$ can be viewed as an indication of the potential power of the new modeling idea behind it, which makes an undirected formulation competitive with a directed formulation.
- Formulation HL-MCF is much stronger than Hop-MCF when $H = 2$, a fact that has direct practical importance. Although previous works agreed that the HSNDP with $H = 2$ is easier than with $H = 3$, solving instances with more 30 demands by branch-and-cut can still be very time consuming [10]. The existence of the equivalent compact formulation HL2 for that case makes the new approach even more practical. As will be shown next, quite large instances can be easily solved to optimality.
- When $H = 3$ and $K = 2$, HL-MCF is significantly stronger than Hop-MCF, more than compensating for the need of solving larger Linear Programs (LPs). As the next experiments will show, solving HL-MCF with CPLEX outperforms the branch-and-cut in [10] and the Benders decomposition over Hop-MCF [3] on larger instances.
- When $K = 2$ and $H = 4$ or when $K = 3$ and $H = 3$, HL-MCF is still much stronger than Hop-MCF. However, the decreased gaps and smaller enumeration trees are roughly compensated by the burden of the larger LPs; the average results are comparable to those by the methods in [3].
- In the remaining three cases, HL-MCF is only slightly stronger than Hop-MCF, which performs much better on solving those instances to optimality [3].

TABLE 1. Average percentage gaps and times for LRs of Hop-MCF and HL-MCF on instances TC20-5, TC20-10, TC20-20, TE20-5, TE20-10, and TE20-20.

		Hop-MCF		HL-MCF			
H	K	Gap	LRT	Gap	LRT	Gap red(%)	T factor
2	1	14.99	0.35	0.00	0.33	100	0.95
2	2	12.92	0.05	0.40	0.47	97	9.3
2	3	7.38	0.05	0.08	0.20	99	4.0
3	1	23.91	0.97	0.00	6.03	100	6.2
3	2	13.13	0.57	2.83	9.98	78	17.6
3	3	7.64	0.62	3.27	7.45	57	12.1
4	1	25.82	1.55	0.83	33.33	97	21.5
4	2	13.05	1.65	5.62	49.97	57	30.3
4	3	7.00	1.43	5.20	25.75	26	18.0
5	1	26.94	0.45	1.94	26.73	93	59.4
5	2	9.60	0.52	5.08	53.57	47	103.7
5	3	6.01	0.58	5.27	30.18	12	51.7
Avg.		14.03	0.73	2.54	20.33	82	27.8

Now we present the additional experiments made to support those claims. Table 2 presents results on rooted instances from [11], for cases $K = 2$ and $H = 2, 3$. We compare the performance of the CPLEX MIP solver over formulations

TABLE 2. Comparison of methods on the rooted instances from Huygens et al. and Mahjoub ($K = 2$).

		$H = 2$							$H = 3$						
		HL2			Hop-MCF			[11]	HL-MCF			Hop-MCF			[11]
n	$\|D\|$	Gap	Nd	TT	Gap	Nd	TT	Gap	Gap	Nd	TT	Gap	Nd	TT	Gap
20	5	0.00	1	0.01	6.30	2	0.03	0.9	3.23	6	2.75	8.15	12	0.66	5.2
	10	0.00	1	0.01	12.73	22	0.44	6.8	4.09	16	10.6	12.52	181	26.0	6.8
	15	0.00	1	0.03	12.90	55	0.40	9.2	3.82	141	76.6	16.33	6,788	1,635	–
30	8	0.22	3	0.06	9.28	5	0.99	3.3	5.28	14	25.2	10.75	53	8.45	5.3
	15	0.00	1	0.05	14.27	36	0.44	7.4	4.23	86	217	18.36	7,026	2,110	–
	22	0.00	1	0.17	18.31	685	8.29	–	3.37	1,608	6,754	22.01	–	>24 h	–
40	10	0.00	1	0.05	12.64	22	0.46	7.4	2.24	5	100	8.84	117	43.7	7.4
	20	0.00	1	0.09	14.96	263	5.47	–	6.23	6,819	21,866	18.06	–	>24 h	–
	30	0.00	1	0.23	16.60	8,456	198	–	4.11	23,014	83,150	20.98	–	>24 h	–
Avg.		0.02	1	0.08	13.11	1,061	23.8	5.8	4.07	3,523	12,466	15.11	–	–	6.18

Hop-MCF and HL-MCF, in terms of root gaps (columns Gap), number of nodes in the branch tree (columns Nd), and total time to solve the instance (columns TT). The root gaps of the branch-and-cut in [11] are also presented (only for the instances that the branch-and-cut could solve to optimality within the time limit). The results in Table 2 make clear that the instances with $H = 2$ are easily solved by the HL2 formulation. The results for $H = 3$ show that HL-MCF is better than Hop-MCF, at least when those formulations are given to the MIP solver. It should be noted that the root gaps of the branch-and-cut in [11] are not much worse than those obtained with HL-MCF. The superiority of HL-MCF in solving the instances to optimality can be partially attributed to slow cut separation but also to the lack of some advanced algorithmic features that are part of the commercial MIP solvers, like sophisticated primal heuristics or branching by pseudocosts.

Table 3 presents a detailed comparison of HL-MCF and Hop-MCF over the larger instances in Botton et al. [3]. The results of solving HL-MCF with CPLEX are compared with the solution of Hop-MCF with CPLEX, but also with the method bc-n-heur, that implements a branch-and-cut algorithm based on the Benders decomposition of Hop-MCF. The root of that branch-and-cut tree obtains the same lower bounds as the LR of Hop-MCF. However, as cut separation is only performed up to a certain depth of the tree (except on integral solutions to check feasibility), the number of nodes can be much larger. The experiments in [3] were also performed with CPLEX 12, using a Intel Core Duo T7200@2.00 GHz machine. As our P7350 is essentially a more power efficient version of the T7200 processor, the running times can be directly compared. Columns give the value of the LR and the time to obtain it (TLR), the best integral solution found Upper Bound (UB), the number of nodes explored (Nd), and the total time (TT). This time is limited to 10,800 s. The columns (Gap) have value 0.0 if the instance is solved to optimality; otherwise they are the final gaps, when the run was stopped. Values in bold indicate the method that

TABLE 3. Comparison of methods on the larger instances from Botton et al. [3] ($H \in \{3, 4\}$, $K \in \{2, 3\}$).

			Hop-MCF - CPLEX						Hop-MCF - Benders					HL-MCF - CPLEX					
Instance	H	K	LR	TLR	UB	Nd	TT	FGap	TLR	UB	Nd	TT	FGap	LR	TLR	UB	Nd	TT	FGap
TC40-10	3	2	354.0	5	395	683	64	0.0	3	395	21,374	100	0.0	381.9	71	395	37	104	0.0
TC40-10	3	3	539.6	5	574	284	23	0.0	4	574	14,894	96	0.0	553.1	78	574	103	310	0.0
TC40-10	4	2	334.7	26	364	1,249	2,671	0.0	38	364	57,347	566	0.0	353.0	303	364	68	7,951	0.0
TC40-10	4	3	507.9	20	525	112	200	0.0	40	525	1,383	64	0.0	514.9	352	525	35	4,400	0.0
TE40-10	3	2	411.2	4	488	226	74	0.0	5	488	222,212	2003	0.0	485.5	65	488	3	68	0.0
TE40-10	3	3	628.1	6	703	409	142	0.0	6	703	564,487	6,768	0.0	683.2	73	703	65	364	0.0
TE40-10	4	2	371.5	27	455	1,475	10,800	11.0	56	453	276,695	10,800	9.5	414.8	1,226	**452**	63	10,800	**7.1**
TE40-10	4	3	577.0	28	616	445	1,754	**0.0**	60	616	450,810	10,703	**0.0**	591.2	297	616	52	10,800	2.5
TC40-20	3	2	516.9	19	632	11,373	10,800	7.7	6	632	194,901	10,800	5.7	595.8	209	632	1,534	6,757	**0.0**
TC40-20	3	3	795.0	16	889	11,656	10,800	4.2	5	889	174,101	10,800	3.7	840.2	193	889	757	10,800	**3.2**
TC40-20	4	2	456.4	146	548	125	10,800	14.4	54	556	58,801	10,800	12.1	515.4	772	**540**	6	10,800	**4.4**
TC40-20	4	3	710.4	134	803	206	10,800	10.2	47	**785**	151,278	10,800	**5.2**	738.6	759	876	1	10,800	15.7
TE40-20	3	2	616.9	34	810	1,300	10,800	20.4	13	930	39,901	10,800	29.8	758.3	177	**790**	1,142	6,811	**0.0**
TE40-20	3	3	932.5	32	1,125	2,570	10,800	13.0	13	1,197	34,800	10,800	18.5	1,025.2	198	**1,103**	523	10,800	**5.6**
TE40-20	4	2	528.4	196	**707**	99	10,800	**24.9**	137	896	13,301	10,800	39.5	607.5	5,174	837	1	10,800	27.4
TE40-20	4	3	808.6	143	**967**	92	10,800	**15.2**	117	1,073	18,201	10,800	22.6	856.7	858	–	1	10,800	–

TABLE 4. Formulation HL2 on new larger instances ($H = 2$, $K \in \{2, 3\}$).

		HL2 - CPLEX					
Instance	K	LR	TLR	UB	Nd	TT	FGap
TC160-40	2	14,85.0	1.1	1,485	1	1.1	0.0
TC160-40	3	2,138.0	1.5	2,149	1	6.0	0.0
TC160-80	2	2,439.5	8.4	2,246	46	20.9	0.0
TC160-80	3	3,524.5	10.8	3,544	67	37.9	0.0
TC160-160	2	3,921.7	14.0	3,930	224	93.3	0.0
TC160-160	3	5,679.7	24.0	5,718	8,534	2,605.4	0.0
TE160-40	2	1,856.0	1.2	1,856	1	2.6	0.0
TE160-40	3	2,681.1	2.2	2,693	3	9.5	0.0
TE160-80	2	2,889.0	6.8	2,891	1	14.7	0.0
TE160-80	3	4,203.1	14.7	4,229	1,854	247.4	0.0
TE160-160	2	4,618.6	11.0	4,626	88	42.5	0.0
TE160-160	3	6,755.4	20.8	6,805	15,246	10,800.0	0.2

provided the smaller gap, for the instances that are not solved to optimality by all the methods. Superior upper bounds are also indicated by values in bold. We did not run the cases $H = 2$ or $K = 1$, as HL-MCF would clearly perform much better; neither did we run the case $H = 5$, as the Hop-MCF based methods would perform much better. Therefore, we test the cases where $H \in \{3, 4\}$ and $K \in \{2, 3\}$.

The last experiments test the capacity of solving larger instances in the case $H = 2$. We created instances TC160-40, TC160-80, TC160-160, TE160-40, TE160-80, and TE160-160, with $n = 160$ and 40, 80, and 160 demands. The instances were run with $K \in \{2, 3\}$. The results in Table 4 make clear that formulation HL2 is indeed very effective. The instances with 40 demands are quickly solved to optimality. Only the instances with 160 demands and $K = 3$ are still difficult.

7. CONCLUSIONS

This work presented an extended formulation for the rooted HSNDP based on the concept of sorting the vertices and edges in a solution into levels, given by their distance to the root. We believe that this original modeling idea can be useful for other network design problems. Anyway, the computational results already obtained on the rooted HSNDP are encouraging, improving the practical capability of solving some important classes of instances. Further progress may be obtained by working on two points:

1. Reducing the time to solve the linear programs associated with HL-MCF. A promising approach would be applying a Benders decomposition by demands, as done with success in [3].
2. Further improving the quality of the lower bounds provided by HL-MCF. A possible way of doing that is by generalizing the concept of solution level, associating the edges with arbitrary integer distances. In that generalization, the parameter L, the largest level of a vertex connected to the root, can be greater than H. Preliminary experiments have shown that significantly improved lower bounds can indeed be obtained. However, the very large size of the resulting linear programs hinders the practical use of this generalized HL-MCF, at least until the first point is addressed.

REFERENCES

[1] F. Bendali, I. Diarrassouba, A. Mahjoub, and J. Mailfert, The k edge-disjoint 3-hop-constrained paths polytope, Discrete Optim 7 (2010), 222–233.

[2] Q. Botton, B. Fortz, and L. Gouveia, "Le polyèdre du problème de conception de réseaux robustes k-arête connexe avec 3 sauts," Journées Polyèdres et Optimisation Comb, Université Blaise-Pascal, Clermont-Ferrand, France, 2008, pp. 13–18.

[3] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, Benders decomposition for the hop-constrained survivable network design problem, INFORMS J Comput (2011), in press.

[4] G. Dahl, Notes on polyhedra associated with hop-constrained paths, Oper Res Lett 25 (1999), 97–100.

[5] G. Dahl, D. Huygens, A. Mahjoub, and P. Pesneau, On the k edge-disjoint 2-hop-constrained paths polytope, Oper Res Lett 34 (2006), 577–582.

[6] I. Diarrassouba, Survivable network design problems with high survivability requirements, Ph.D. Thesis, Université Blaise Pascal Clermond-Ferrand II, France, 2009.

[7] I. Diarrassouba, V. Gabrel, and A. Mahjoub, "Le problème de conception de réseau fiable avec contrainte de borne," Proc ROADEF, Toulouse, 2010, pp. 299–301.

[8] L. Gouveia, Multicommodity flow models for spanning trees with hop constraints, Eur J Oper Res 95 (1996), 178–190.

[9] L. Gouveia, L. Simonetti, and E. Uchoa, Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs, Math Program 128 (2011), 123–148.

[10] D. Huygens, M. Labbé, A. Mahjoub, and P. Pesneau, The two-edge connected hop-constrained network design problem: Valid inequalities and branch-and-cut, Networks 49 (2007), 116–133.

[11] D. Huygens and A. Mahjoub, Integer programming formulations for the two 4-hop-constrained paths problem, Networks 49 (2007), 135–144.

[12] D. Huygens, A. Mahjoub, and P. Pesneau, Two edge-disjoint hop-constrained paths and polyhedra, SIAM J Discr Math 18 (2004), 287–312.

[13] A. Itai, Y. Perl, and Y. Shiloach, The complexity of finding maximum disjoint paths with length constraints, Networks 12 (1982), 277–286.

[14] H. Kerivin and A. Mahjoub, Design of survivable networks: A survey, Networks 46 (2005), 1–21.

[15] E. Uchoa, "Cuts over extended formulations by flow discretization," Progress in combinatorial optimization, A. Mahjoub (Editor), ISTE-Wiley, London, 2011, pp. 255–282.