Yugoslav Journal of Operations Research
27 (2017), Number 4, 427–438
DOI: 10.2298/YJOR160512004F

A FORMULATION FOR A HOP CONSTRAINED SURVIVABLE NETWORK DESIGN PROBLEM

Graciela FERREIRA
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
gferreira@fing.edu.uy

Sergio NESMACHNOW
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
sergion@fing.edu.uy

Franco ROBLEDO
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
frobledo@fing.edu.uy

Received: May 2016 / Accepted: March 2017

Abstract: This article presents an integer linear model for the hop constrained node survivable network design problem. The formulation is focused on networks represented by undirected graphs with not rooted demands, considering costs in arcs and in optional (Steiner) nodes, too. The proposed model allows setting different values of parameters for constraints between each pair of terminal nodes, including hop length and number of node disjoint paths constraints. This work includes calculating lower and upper bounds to the optimal solution. Since this kind of problems are NP-hard, it is useful to combine the presented formulation with heuristic methods in order to solve effectively large problem instances. The model was tested over the graphs with up to 85 nodes and 148 arcs, in order to validate it in cases with known solution.

Keywords: Network Design, Hop Constrained, Survivability.

MSC: 90B06, 90C05, 90C08.

1. INTRODUCTION

In network design, the survivability property enables the network to maintain a certain level of network connectivity and quality of service under failure conditions. Survivability has been considered as one of the critical requirements in network planning and design [5]. It often involves considering design requirements on the network topology, or in the case of communication networks, constraints could be associated with protocol, bandwidth allocation, etc. For instance, a topology requirement would achieve a design that keeps a minimum two-connected network against any failure of a single link or node. This concept can be applied to multiple types of networks, such as communication, power, transportation network, etc. (see for instance [3] [8]).

Formally, survivability is defined as the capacity of a network to remain operational after disturbances or failures in some components [1]. The survivable network design problem has been extensively studied [5, 3, 8, 10] and it is known to be NP-hard[6]. Survivability properties are usually modeled by requiring a minimal number of node -or edge- disjoint paths between certain pairs of nodes.

In this article, we combine survivability and quality of service concepts for the problem that imposes additionally hop-constraints when designing survivable networks. This approach ensures that for every distinct pair of nodes, there exists a predefined number of edge/node disjoint paths, so that each such path does not exceed a given hop limit.

Recent literature calls this kind of problems as *Hop Constrained Survivable Network Design Problem* (HCSNDP) [5, 2, 7]. The HCSNDP proposes finding the optimal network design with survivability requirements and effectiveness in quality of service (e.g., the maximum length of paths is bounded).

We focus on solving a variant of the HCSNDP that is applied to model networks represented by undirected graphs with not rooted demands, considering costs in arcs and in optional (Steiner) nodes, too. Different values of parameters for constraints between each pair of terminal nodes are allowed in the problem formulation, including hop length and number of node disjoint paths constraints. This is a generalization of HCSNDP that we named as "Generalized Steiner Problem with weighted Steiner nodes and diameter constrained" (GSPWDC).

We introduce an integer linear model for the GSPWDC. The exact model is tested over some graphs in order to perform a validation in cases with known solutions. Since this kind of problems is NP-hard, we propose and test some procedures to calculate lower and upper bounds to the optimal solution, which allow to approach effectively an optimal solution. These bounds are very useful when facing large problem instances.

The paper is organized as follows. Next section introduces the main concepts and definitions releted to the problem model and formulations. The proposed problem formulation is presented in Section 3. Lower and upper bounds to the optimal solution are analyzed in Section 4. The experimental evaluation is reported and discussed in Section 5. Finally, Section 6 presents the conclusions and the main lines for future work.

2. BACKGROUND

A network is represented as a graph, $G = (V, E)$. We consider only two possible types of nodes: *terminal* nodes, for which connectivity requirements are defined (set T), and optional or *Steiner* nodes (set S).

There are two models to specify the survivability conditions [6]. In this work we follow the *Generalizad Steiner Problem* (GSP) approach by Winter [11]: given a network represented by a graph $G = (V, E)$, with costs associated to edges, let $T \subseteq V$ be the set of terminal nodes and let $Q = \{q = (i, j), \forall i, j \in T\}$ be the set of all pairs of nodes in $T \subseteq V$; the problem is to find a subgraph with minimal cost so that $\forall q \in Q$ at least $r_q \in N$ node -or edge- disjoint paths exist. In the case where $T = V$ and $r_q = k, \forall q \in Q$, with cost associated to the edges and node disjoints paths, the problem is also known as $NCON(G, k)$ [9]

In order to model the HCSNDP, we use a variant of the GSP, that we call "Generalized Steiner Problem with weighted Steiner nodes and diameter constrained (GSPWDC)", which is formulated as follows:

Given a undirected simple graph $G = (V, E)$, with: (i) a set of edge cost or weights $C = \{c_{ij} \in R^+\}, \forall (i, j) \in E$; (ii) a set of terminal nodes $T \subseteq V$; (iii) $Q = \{q\}$ the set of pairs of nodes in $T \subseteq V$, (iv) a matrix with node (or arc) connectivity requirements $R = \{R_q\} \forall q \in Q$; (v) a vector with node weights $A = \{a_i \in R^+, i \in S = V \setminus T\}$; and (vi) a matrix of maximum length of paths allowed (*hop requirements*) $L = \{L_q \geq 0, integer, \forall q \in Q\}$; the GSPWDC consist in finding a minimal cost subgraph $H \subseteq G$ such that it covers T and $\forall q \in Q$ exist at least R_q node or arc disjoint paths linking a pair of terminal nodes $q = (i, j)$ in H, so that each one has no more than L_q hops or arcs.

Calling a pair of terminal nodes q as a *demand*, if all q have a common node, then the *demand* is called *rooted*, otherwise it is *unrooted*. According to Mahjoub [7], when $| Q |= 1$, the HCSNDP can be solved in polynomial time for $L \leq 3$, and it is NP-hard for $L \geq 4$. When $| Q |$ is not constrained, the problem is NP-hard, even in simplest case when Q is *rooted*, R=1 and L=2 $\forall q \in Q$.

The review of related work about models and formulations for the hop-constrained survivable network design problem allows identifying four existing models. The first three models are all variants of the same approach, initially proposed by Gouveia in 1998 [4] and later completed by Botton [1]. All of them were proposed to solve HCSNDP and were also implemented and tested over different networks. The fourth model was presented as an ILP model to solve SNDP in the survey by Kerivin and Mahjoub [6]. This model does not have practical results reported, but it was used to extract some interesting properties of polytopes corresponding to constraints space instead.

All the existing models for hop-constrained survivable networks use the same main idea to represent hop constraints. A set of auxiliary graphs are introduced: an auxiliary graph G^q—also called layered graph due to the method applied for building it— is defined for each $q \in Q$. Each graph G^q contains all existing paths between each pair of terminal nodes $q = (o, d)$ with length not greater than L_q (being L_q the number of hops allowed for $q = (o, d)$). The survivability constraints are formulated over these auxiliary graphs and hop constraints are implicitly considered because in the auxiliary graphs all paths have lengths shorter than L_q.

3. AN ILP FORMULATION FOR THE HCSNDP

The proposed formulation is a variant of the existing models known as "*Hop-indexed formulation*", presented by Gouveia et al. [5] and "*Hop multi-commodity flow formulation* (HOP-MCF)" introduced by Botton [1]. The proposed model also incorporates some concepts used in recent works [6, 7, 4]. The model presented here allows considering constraints with different maximal length of paths between each pair of terminal nodes, as well as different numbers of required node-disjoint paths between each pair of terminal nodes (i.e., allowing *heterogeneous* survivability conditions). The model includes the cost of edges and nodes, too. In order to represent hop constraints, just like in the aforementioned previous works, we use extended layered graphs (one for each $q \in Q$) that implicitly guarantees satisfying the maximum path length constraints.

The main idea behind the proposed formulation is to decompose the problem into $|Q|$ subproblems, one for each pair of terminal nodes $q \in Q$. Let $(o(q), d(q))$ be a pair of origin–destination nodes corresponding to q. Fixed q, each subproblem is modeled with a directed graph composed of $L + 1$ layers (according to the transformation proposed by Gouveia [4]): being $G = (V, E)$ the original undirected graph, the alternative representation is $G^q = (V^q, A^q)$ where $V^q = V_1^q \cup \ldots \cup V_{L+1}^q$ / $V_1^q = o(q)$, $V_{L+1}^q = d(q)$ and $V_l^q \subseteq \{V \setminus \{o(q)\}$ such that there are a simple path between $o(q)$ and each $v \in V_l^q$ with length at most l, with $l = 2 \ldots L$.

Using the same notation proposed by Botton [3], let v_l^q be the copy of $v \in V$ in the l-th layer of graph G^q, then $A^q = \{(i_l^q, j_{l+1}^q)/(i, j) \in E, i_l^q \in V_l^q, j_{l+1}^q \in V_{l+1}^q, l \in \{1, \ldots, L\}\} \cup \{d(q)^l, d(q)^{l+1}, l \in \{2, \ldots L\}\}$. Details and graphical examples of the extended layered graphs were already provided by Botton [3, 1].

An edge in E with end points i and j is denoted as ij, while the arc between $i_l^q \in V_l^q$ and $j_{l+1}^q \in V_{l+1}^q$ in the directed graph is denoted as (i, j, l).

When using the proposed transformation, all paths from $o(q)$ to $d(q)$ in G^q fulfill hop-constraints.

Consider the following set of parameters:

i) a_i denotes the cost associated to each Steiner node i;

ii) c_{ij} is the cost associated to edge $ij, \forall ij \in E$;

iii) R_q is the minimal number of node-disjoint paths required between $o(q)$ and $d(q)$, $\forall q \in Q$;

iv) L_q is the maximum length allowed for paths (hops) between $o(q)$ and $d(q)$, $\forall q \in Q$.

Also, consider the following set of variables:

i) z_{ij} is a binary variable that indicates if edge $ij \in E$ is in the solution;

ii) $x_{ij}^{l,q}$ is the flow through arc (i, j, l), for each q in the layer l of G^q;

iii) N_i is a binary variable that indicates $\forall i \in S$ if the Steiner node i is included or not in the solution. Each N_i is used to allow at most one active outgoing arc from a node i over all layers of G^q, guaranteeing not to repeat nodes in a path between $o(q)$ and $d(q)$.

The ILP-HCSNDP formulation is presented next.

$$\text{(ILP-HCSNDP)} \quad \min \sum_{(i,j)\in E} c_{ij}.z_{ij} + \sum_{i\in V} a_i.N_i \ \text{ with:} \tag{1}$$

$$\sum_{j:(o(q),j,1)\in A^q} x^{1,q}_{o(q),j} = R_q \tag{2}$$

$$\sum_{j:(j,d(q),L_q)\in A^q} x^{L(q),q}_{j,d(q)} = -R_q \tag{3}$$

$$\sum_{j:(j,i,l-1)\in A^q} x^{l-1,q}_{ij} - \sum_{j:(i,j,l)\in A^q} x^{l,q}_{ji} = 0,\ for\ all\ q \in Q,\ l \in \{2,\dots,L_q\},\ i \in V^q_l \tag{4}$$

$$\sum_{l=1,\dots,L_q} x^{l,q}_{ij} + x^{l,q}_{ji} \le z_{ij},\ \ for\ all\ \ (i,j) \in E,\ q \in Q \tag{5}$$

$$z_{ij} \in \{0,1\},\ \ for\ all\ \ (i,j) \in E \tag{6}$$

$$x^{l,q}_{ij} \ge 0\ \ integer\ \ for\ all\ \ (i,j,l) \in A^q,\ q \in Q \tag{7}$$

$$\sum_{l=1,\dots,L_q} \sum_{j\in\delta(i)} x^{lq}_{ij} \le N_i\ for\ all\ q \in Q,\ (i,j) \in E,\ and\ i \in S \tag{8}$$

$$N_i \in \{0,1\},\ for\ all\ i \in S \tag{9}$$

In the ILP-HCSNDP formulation, Equation (1) is the objective function: it proposes minimizing the costs associated to arcs and Steiner nodes. Regarding the constraints, the network flow over G^q (Equations (2), (3), and (4)) assure that there are R_q paths from $o(q)$ to $d(q)$. Equation (5) does not enable using multiple times a given edge ij on a path in G^q, thus guaranteeing the edge-disjointness property, while they also link variables z and flow variables of copies of the same arc in different layers, which means that the total unimodularity property of matrix restriction is lost [1]. As a consequence, a set of constraints (defined in Equation (7)) must be explicitly introduced to obtain a feasible solution.

Equations (2)–(7) are present in the model by Botton; our formulation includes new constraints (defined in Equation (8) and Equation (9)) in order to guarantee the existence of node-disjoint paths and to allow the model to represent the costs associated to Steiner nodes. Note that $x^{l,q}_{ij} \le 1,\ i \ne j$, as stated by Equation (5) and Equation (6).

Constraints (stated in Equations (2), (3), and (4)) represent $|Q|$ independent sets of network flow constraints, one set defined for each $q \in Q$. Then, if constraints defined in Equations (5)–(9) are relaxed, there will be $|Q|$ independent network flow problems to solve, and the solution for each variable x^{lq}_{ij} will be integer. So, the model can take advantage of following a constraint decomposition approach, but in this case, the difficulty is that in the objective function, variables x^{lq}_{ij} have no costs.

A simple idea to easily test the feasibility of constraints related to maximum number of node-disjoint paths allowed, is to consider for each $q\in Q$ a set $CS(q)$ where $CS(q) \subseteq V \setminus \{o(q), d(q)\}$ such that nodes $o(q)$ and $d(q)$ are not connected by a path in subgraph $G' \subseteq G$

induced by $V'=V\backslash CS(q)$. So, CS is a cut set between $o(q)$ and $d(q)$ nodes. Applying Menger's theorem [6], for each q, the minimum size of $CS(q)$ indicates the maximum number of node-disjoint $o(q)$-$d(q)$ paths.

4. LOWER AND UPPER BOUNDS

Due to the intrinsic complexity of ILP-HCSNDP, heuristic and approximate approaches have to be used to cope with general and real-world instances of medium and large dimension. This section proposes and describes a set of procedures to calculate bounds to the exact solution of the problem.

We propose computing two lower bounds (called LB1, LB2) and four upper bounds (called UB1, UB2, UB3,UB4), which are defined and explained below. The proposal for computing upper bounds is based on a general idea: when fixing values of all variables z_{ij} and N_i, then only x_{ij} variables remain, and the original problem can be separated in $\mid Q \mid$ independent subproblems, which can be solved independently, too.

Lower bounds:

- LB1: it is an optimal solution of the ILP-HCSNDP integer relaxation. Here we work over G^q, but it is a linear program working in real variables.
- LB2: it is an optimal solution of the ILP-HCSNDP relaxing constraints of maximal length of paths. Here we work over original graph G, and integer variables.

Upper bounds:

- UB1: it fixes all boolean variables z_{ij} and N_i to one. For each one of $\mid Q \mid$ independent subproblems, we find the optimal solutions. Then, we calculate which arcs we must include in a global feasible solution in order to support all optimal solutions of the subproblems. If some of this subproblems are not feasible then, global problem is not feasible.
- UB2: it fixes all boolean variables z_{ij} and N_i, some to one and other to zero. For each one of $\mid Q \mid$ subproblems, we find optimal solutions and then we calculate which arcs must be included in a global solution in order to support all optimal solutions of the subproblems. If some of this subproblems are infeasible, the solution is discarded.
- UB3: it is an improvement of UB1 and UB2 computed using a Dantzig-Wolfe decomposition (DW) applied over ILP-HCSNDP problem. Equations (2) are included in DW subproblem (network flow problem) and remaining constraints are in the main problem. Integer conditions should be included in the main problem, because the subproblem always has integer solutions. In this work we do not include integer conditions in the main problem, so we used results of DW decomposition only when solutions or upper bounds are integer and improve UB1 or UB2.
- UB4: A greedy heuristic algorithm that builds a feasible solution.

In UB1 and UB2 cases, as z_{ij} and N_i variables are fixed, we can work with separate $| Q |$ independent subproblems, so we could profit this condition working in parallel. In UB3, the subproblem of DW decomposition can be separated, too. For UB1,UB2, and UB3 we work over G^q, then may be hard to calculate a feasible solution due to high dimensions.

Algorithm 1 presents a pseudo-code of the heuristic applied to compute UB4. It is a greedy algorithm that include a diversification phase

The proposed heuristic for computing UB4 starts by calculating a minimum float cost problem over G_d, with capacity arcs constraints and one additional constraint (lines 12-13). In this float problem, the pair of nodes in q are taken as source and destination. All edges have maximal capacity equal to one, except the edge that links sink to source that has minimal capacity R_q. This assures to find R_q node-disjoint paths between nodes in q.

Let $x_{i,j}, \forall (i, j) \in E$ be decision variables. The constraint added in line 13, $\sum_{ij \in E} x_{ij} \leq b$, with $b = R_q{*}L_q$ bounds the total path length. This constraint does not ensure that all paths have length lower than L_q, but it decreases the search domain in order to find a feasible solution. For solving this extended float problem, an integer linear programming can be used. It is even necessary because the last constraint breaks integrality property of the float solution.

After that, the float solution $s(q)$ found, which is a binary vector with $|E|$ elements, is tested in order to determine if hop constraints are satisfied (line 17). If there are some path that do not satisfy hop limits in last finding solution $s(q)$, then b is decreased by one (line 18) and the procedure is repeated. If the procedure does not attain a feasible solution and the extended float problem is infeasible for some value of b, the procedure adds cuts in order to exclude paths that do not satisfy hop limits.

The heuristic procedure finishes when it finds an infeasible problem for some q (line 24), i.e., the global problem does not have a solution, or when it finds a global feasible solution. In effect, as b is an integer and it is decreased by one at each iteration where the feasibility test for L_q fails, the procedure does finish in a finite number of steps. If cuts must be added, they are also performed in a finite number of steps.

Algorithm 1 Proposed heuristic algorithm A:

Require: V, E,T, S, Q,$C = \{c_{ij}, \forall(i, j) \in E\}$, $A = \{a_k, \forall k \in S\}$, p, R, L
Ensure: gs ▷ gs is a binary vector with dimension $|E|$

Build data structures: incidence matrix node-node, incidence matrix node-arcs
Convert undirected graph to directed, where each arc is replaced by a pair of edges with opposite senses.
Replace each Steiner node S by a pair of nodes (S_1, S_2) linked by an edge going from S_1 to S_2, such that all incoming edges to S are incoming to S_1 and all outgoing edges of S are outgoing edges of S_2. Let G_d be this directed graph.
Build incidence matrix node-node and incidence matrix node-arcs NEa for G_d.
Order Q by preferences: Sort list Q according to our R and L values in descendent order (first pairs q of terminal nodes with high $R(q)$ and $L(q)$) that is, we process first those pair of terminal nodes that allow the longest paths, which requires more path linkings between them.
Build an ordered list of Steiner nodes $ListS$, based on preferences calculated according to cost of nodes divided by our degree. Order it by increassing preferences
for all $q \in Q$ **do**
  if $ListS \neq \emptyset$ **then**, ▷ Diversification step
    extract first element r of $ListS$ with probability p
    $c_r = -1$
  end if
  ▷ Search a candidate solution
  Build a minimum float cost problem over G_d, with capacity arcs constraints
  Add a constraint $\sum_{ij \in E} x_{ij} \leq b$, where $b = R_q * L_q$
  while $b > 0$ **do**
    Solve a linear program built in the previous points. Let $s(q)$ be this solution.
    if $s(q) \neq \emptyset$ **then**
      if some path $s(q)$ does not satisfy hop limits R_q **then**
        $b \leftarrow b - 1$
      else
        $b = 0$
      end if
    else
      if $b = R_q * L_q$ **then**
        Display "infeasible problem"
        Stop
      else
        Add a cut excluding paths that do not satisfy hop limits in $s(q)$
      end if
    end if
  end while
  Set to 0 all costs associated with edges in feasible solution corresponding to $s(q)$
  Modify $ListS$, extracting all elements with cost ≤ 0
  $gs = gs \vee s(q)$ ▷ In order to build a global feasible solution
end for

5. COMPUTATIONAL EXPERIMENTS AND RESULTS

This section presents the experimental evaluation of the proposed ILP-HCSNDP model. The analysis is oriented to validate in cases with known solution and to research their behavior on large instances.

5.1. Development and execution platform

The model was implemented in CPLEX 12.51 MIP solver, and the executions were performed on a eight-cores Intel i7 processor at 3.07 GHz having 16GB RAM.

5.2. Problem instances

The proposed formulation was tested over eight graphs, using different values for the size of sets Q, R, and L. This methodology also allows having heterogenous values in matrices R and L.

We decided to work with simple undirected graphs. Table 1 summarizes the main characteristics of the graphs used in the experimental analysis. Column D is the graph density, defined for this type of graphs as $2|E|/|V|(|V|-1)$. Note that the maximal value for D is 1 when solving a complete graph with $\frac{1}{2}|V|(|V|-1)$ edges.

Table 1: Graphs and instances used in the experimental evaluation

graph	$\mid V \mid$	$\mid E \mid$	D	*instance*
FR1	19	43	0.2515	I1
FR2	11	30	0.5454	I2,I3,I4,I10,I11,I12
EON	19	36	0.2105	I5, I13
NFSNET	14	52	0.5714	I6, I14
TA1	24	55	0.1993	I7
B1	50	63	0.0514	I8
B2	50	63	0.0514	I9
RAU2	85	148	0.0415	I15

Instance I1 is built from FR1, a simple graph for which it is easy to find a solution. This problem instance is used to tune the model. Instance FR2 has been studied as a NCON(G, r) instance, with $r_{ij}= 2,\ for\ all\ (i, j) \in E$,

where the edge costs satisfy the triangle inequality. We use this example to test our model in a case with known optimal solution. EON and NFSNET are graphs used in the article by Gouveia et al. [5]. The remaining graphs are taken from libraries of test sets available at Internet: TA1 is from Survivable fixed telecommunication Network Design library (SNDlib, http://sndlib.zib.de), B1 and B2 are instances of the Steiner Tree Problem from SteinLib (http://steinlib.zib.de)

Finally, RAU2 graph is a real-life scenario based on the current Uruguayan academic network (www.rau.edu.uy).

A given graph can be associated with several instances that differ in their parameter values. The last column in Table 1 indicates the problem instances created from each considered graph.

5.3. Numerical results and discussion

Table 2 reports the parameter values and the results obtained for each instance solved. A given value in columns labeled L, R, c_{ij} or a_i means that the respective parameter is constant for all paths between each pair of nodes in T. Otherwise, the "diff" label is used to report when using different values. Instances marked with * are cases with $R = 1$, where optimal solutions or upper bounds are known. Columns labeled *opt*, *const*, *bin*, *int* and *time* report for each instance the optimal value (when attained), the number of constraints, the number of binary variables, the number of integer variables, and the time (in seconds) to solve each instance, without including the time taken to calculate V^q, the set of nodes in the layered graph G^q. The number of constraints and variables before the CPLEX presolve stage are reported; applying a presolve method could significantly reduce that number.

Table 2: Instances details and experimental results

instance	$\mid T \mid$	$\mid Q \mid$	L	R	c_{ij}	a_i	*opt*	*const*	*bin*	*int*	*time (s)*
I1	2	1	5	4	1	1	32	820	60	431	0.04
I2	11	55	7	2	diff	0	25	4497	30	25396	2.58
I3	11	55	8	2	diff	0	24	4992	30	29191	1.45
I4*	11	55	7	1	diff	0	20	4497	30	25396	137.13
I5*	10	45	3	1	1	1	10	16980	45	9887	0.04
I6	6	15	2	2	diff	1	9	7164	60	3169	0.03
I7*	24	396	4	1	1	0	23	24553	51	117201	2.35
I8*	9	36	50	1	diff	0	82	174268	104	238090	1319.00
I9*	13	78	50	1	diff	0	83	339570	79	515729	63177.00
I10	11	55	diff	2	diff	0	24	4713	30	27133	1.55
I11	11	55	diff	2	diff	0	26	4704	30	26983	3.33
I12	11	55	diff	diff	diff	0	25	4704	30	26983	14.28
I13	10	45	diff	diff	diff	diff	12	16909	45	9374	0.06
I14	10	45	diff	diff	diff	diff	18	25519	67	11696	0.14
I15	20	190	diff	diff	diff	diff	6583	1944361	213	582187	8198.33

The experimental evaluation was performed over graphs with up to 85 nodes and 148 arcs. Most instances are solved in a few seconds, only two cases demanded more than an hour: instance I9 (about 20 hours) and instance I15 (about two hours). Instances I8 and I9 took longer to find the optimal solution; these are cases with R=1, and large sets L and T. According to the cases studied, the parameters that most influence the resolution time are $|L|$ and $|T|$; the first is related to the dimensions of G^q and the second to the number of graphs. In effect, the quantity of x variables, which are the most numerous, depends on $|Q|$, $|L|$ and $2\times|E|$, since each G^q is a directed graph. The number of remaining variables (all binary) depends on the number of Steiner nodes and the quantity of undirected edges.

For a given specific instance (I15), we explored several values of $|T|$ in the experiments. The largest value of $|T|$ for which we were able to obtain results in reasonable execution times (less than 24 hours) was $|T|= 20$. This is a relevant result for our research community since all variants of I15 problem instance are built over the real infrastructure of our Uruguayan academic network, allowing to explore different configurations for survibability and quality of service for academic and research projects.

Table 3 reports the lower and upper bounds computed for each instance solved. The column labeled *Bound*, indicates the kind of bound reported. Possible values are: LB1, LB2, UB1, UB2, UB3, or UB4, whose meaning has already been described in 4.

Table 3: Lower and upper bounds

instance	*optimum*	*LB1*	*LB2*	*UB1*	*UB2*	*UB3*	*UB4*
I1	32	32	32				32
I2	25	24.14	24				70
I3	24	24	24				70
I4*	20	12.07	20	84	79	26	42
I5*	10	9	9	21	18	15	25
I6	9	9	6		17	24	19
I7*	23	20.8	22	51	51	43	65
I8*	82	72	82	131	141		131
I9*	83	72.5	83				148
I10	24	24	24				62
I11	26	24.36	24				62
I12	25	22.5	22				60
I13	12	11.5	10				24
I14	18	18	17				20
I15	6583	6278.25	4308				13211

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented and evaluated a formulation for the Hop Constrained Survivable Network Design Problem.

We focused on the node survivability case for networks represented by simple and undirected graphs, not rooted demands, and considering costs in arcs and Steiner nodes. Based on the related previous works, we have developed a new formulation that accounts for specific quality of service and survivability constraints. The proposed model allows a heterogeneous setting for the network by including different values for the length of paths (related to the quality of service) and the number of paths (related to the connectivity demands), between each pair of terminal nodes.

The proposed formulation was evaluated for medium-size instances with up to 85 nodes and 148 arcs. The evaluation accounted for a significantly large number of decision variables. The CPLEX implementation of the proposed formulation was able to effectively solve all but one instance to optimality. Most of the problems were solved in a few seconds. Instance I15, which is based on a real network, was solved in about two hours. No optimal solution was computed for instance I9. The results for instance I15 are relevant, as this case study models the current Uruguayan academic network.

We used relaxation methods to compute lower bounds for the problem. In addition, decomposition techniques and heuristic methods were proposed to find upper bounds. This approach allowed to find accurate lower bounds that are close to the optimal solution for the set of problem instances considered in this article.

In the case of upper bounds, results allowed to conclude that UB1, UB2, UB3 are hard to calculate for some instances. Nevertheless, UB4 can be computed faster and it allows computing results for all instances in less than one hour. In this last case, we find that the proposed heuristic is sensible to preference order in Q, but rather insensitive to chosen diversification technique.

The main lines for the actual and future work are related to: (i) improving UB4, trying to introduce an effective diversification technique or to develop a local search phase that explores neighborhoods of global feasible solutions, (ii) improving the techniques for constructing and managing the graph G^q and applying decomposition algorithms in order to be able to solve significantly larger instances applying the proposed ILP formulation, (iii) try to measure and to assure the distance to optimum of approach solutions, (iv) introducing new instances of graph with special topologies that put to test the algorithms or other instances, particular cases of HCSNDP problems, with known optimal solutions. Being the HCSNDP a NP-hard problem, this kind of (relaxed) exact algorithms may be useful when combined with heuristic methods in order to effectively solve large instances modeling real-life situations.

REFERENCES

[1] Botton, Q., *"Survivable network design with quality of service constraints"*, PhD thesis, Université Catholique de Louvain, Louvain School of Management. 2010.

[2] Botton, Q., and Fortz, B. and Gouveia, L., "On the hop-constrained survivable network design problem with reliable edges", Centro de Investigação Operacional, Faculdade de Ciencias da Universidade de Lisboa, Lisboa, Portugal, 2014. Available at http://www.optimization-online.org/DB_FILE/2012/04/3424.pdf. Accessed on March 2017.

[3] Botton, Q. and Fortz, B. and Gouveia, L. and Poss, M., "Benders decomposition for the hop-constrained survivable network design problem", *INFORMS Journal on Computing*, 25 (1) (2013) 13-26.

[4] Gouveia, L., "Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints", *INFORMS Journal on Computing*, 10 (2) (1998) 180–188.

[5] Gouveia, L. and Patrício, P. and de Sousa, A., "Hop-constrained node survivable network design: An application to MPLS over WDM", *Networks and Spatial Economics*, 8 (1) (2008) 3–21.

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

[7] Mahjoub, R., Simonetti, L., and Uchoa, E., "Hop-level flow formulation for the hop constrained survivable network design problem", Network Optimization, 2011, 176-181.

[8] Raghavan, S., and Anandalingam, G., "The Effect of Hop Limits on Optimal Cost in Survivable Network Design", in: Orlowski, S. and Wessäly, R. (eds.) *Telecommunications Planning: Innovations in Pricing, Network Design and Management*, 2006.

[9] Steiglitz, K. and Weiner, P. and Kleitman, D., "The design of minimum-cost survivable networks", *IEEE Transactions on Circuit Theory*, 16 (4) (1969) 455–460.

[10] Stoer, M., Design of survivable networks, volume 1531 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1992.

[11] Winter, P., "Steiner problem in networks: a survey", *Networks*, 17 (2) (1987) 129-167.