Hop-constrained tree-shaped networks [1]

Marta S.R. Monteiro [a] Dalila B.M.M. Fontes [a,2] Fernando A.C.C. Fontes [b]
[a] *Faculdade de Economia da Universidade do Porto and LIAAD/INESC TEC*
[b] *Faculdade de Engenharia da Universidade do Porto and ISR-Porto*

Abstract. Hop constraints are used to limit the number of links between two given points in a network, this way improving the quality of service by increasing the availability and reliability of the network. They have been applied to a limited number of problems, although their application can be of the greatest importance both from the academical and practical points-of-view. In this work, we survey relevant and recent works on hop-constrained problems focusing on problems with tree shaped solutions.

Keywords. Survey, Hop constraints, Minimum Spanning Trees, Lower Bounds, Heuristics, Flows

1. Introduction

Tree topology problems are well researched for many reasons. Firstly, they frequently appear in practical situations, e.g., a spanning tree is the best topology for many telecommunication network designs, consisting of finding the best way to link telecommunication devices at different locations to a central host. Secondly, they have a large number of applications within distribution networks, for example, oil, gas, water, electricity, and in many other problems, such as community detection (for instance in social networks), taxonomy, clustering, handwriting recognition, etc. Thirdly, trees are solutions for other problems, such as the minimum concave cost network flow problem. Finally, trees can be components of the solution to other problems: for approximating the travelling salesman problem [1], the multi-terminal minimum cut problem [2], the minimum-cost weighted perfect matching [3], just to mention but a few.

Constraints limiting the number of arcs (hops) in the unique path between a source node and any other node in a network are called hop constraints. Whenever a network optimization problem includes hop constraints, it is known as a Hop-constrained problem. Hop constraints have been applied to a small, though diversified, number of problem classes. Examples of 2-hop constraints applied to problems in transportation, statistics, and plant location are provided and discussed in [4].

A well-known hop-constrained problem for which the solution is a tree is the Minimum Spanning Tree (MST). In the MST problem, the objective is to find a tree span-

[1]This work is partially funded by the ERDF through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, projects PTDC/EGE-GES/099741/2008 and PTDC/EEA-CRO/116014/2009.

[2]Corresponding author: Faculdade de Economia da Universidade do Porto and LIAAD/INESC TEC, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal; Email: fontes@fep.up.pt

ning all nodes in a network while minimizing the total costs incurred. This combinatorial problem is frequently used to model problems in the area of telecommunications, consisting of a central device, for example a hub, that must be linked to a set of remote terminals. It has been shown that by including hop constraints it is possible to generate trees with a much better service and only with a moderate increase on the total cost [5].

In this work, we describe relevant and recent literature on hop-constrained optimization problems, mainly for problems for solutions are tree-shaped.

The remainder of this work is organized as follows. Section 2 provides a discussion on the main reasons for applying hop constraints. In Section 3, we introduce the Hop-constrained Minimum Spanning Tree problem (HMST) along with a possible mathematical formulation for the problem and we review exact and heuristic solution methods that have been used to solve it. A new generalization of the HMST problem that considers flows other than the unit is presented and reviewed in Section 4. In Section 5, we give a brief account of other problems in which hop constraints have been introduced. Finally, Section 6 summarizes and concludes this work.

2. Hop constraints: what are they for?

In this work, we discuss problem with hop constraints in the context of tree shaped networks. In these problems, hop constraints limit the number of arcs between the source node and every other node in the tree. The addition of a maximum number of arcs in each path is usually related to reliability and availability issues.

In a problem in which a central hub is sending packages of information to terminal computers, availability is usually associated with the probability of perfect functioning of all links between the central hub and each one of the terminals. Whereas reliability is the probability that the transmission of the package will not be interrupted by any external factor resulting in the fail of a link [5]. For instance, if a link failure is associated with some probability p, and failures are independent, then, the probability that a path with h arcs is operational is given by $(1-p)^h$. Therefore, we can easily conclude that the increase on the number of arcs between the central hub and the terminal computer will result in a decrease on the reliability and on the availability of the network. Although imposing a maximum limit on the number of hops on a network may increase the total costs, the overall gain in terms of service quality may compensate.

Hop constraints can also be used to prevent the degradation of signal quality. For example, many telecommunication networks impose a maximum of three hops for data paths and two hops for voice paths [6]. In addition, hop constraints can also be associated with lower time delays, for instance in a multi-drop lines network, in which packages of information may have to queue before reaching their destination. The total delay time is dependent on the number of arcs between the origin and destination nodes [7], therefore delay times increase with the number of hops in the network.

Having introduced and discussed the importance of hop constraints, in the following sections we review solution methods for some hop-constrained network problems. The solution methods range from proposing alternative mathematical programming formulations to metaheuristic methods such as Genetic Algorithms (GAs) and Ant Colony Optimization (ACO) algorithms.

3. The Hop-Constrained Minimum Spanning Tree Problem

In the hop-constrained minimum spanning tree problem, besides finding a tree spanning all nodes at minimum cost, the number of arcs in each path from the source node to every other node must not exceed a given integer H value, which is called the hop parameter.

Dahl [4] studies the HMST problem for $H = 2$ and compares the polyhedra of the models with directed and undirected arcs. Alfandari and Paschos [8] have shown that the HMST with $H = 2$ cannot be approximated by polynomial time approximation schemes unless $P = NP$. The HMST with $H = 2$ is equivalent to a version of the Simple Uncapacitated Facility Location (SUFL) problem where the potential facility sites coincide with the locations of the clients to be served. It is well known that the SUFL problem is NP-hard which implies that the HMST is also NP-Hard. Furthermore, Manyem and Stallmann [9] have shown that the HMST problem is not in the class of problems for which it is possible to have polynomial time heuristics with a constant-factor approximation bound.

3.1. A formulation for the HMST problem

Several mathematical programming formulations have been proposed for the HMST problem in an attempt to improve the performance of the solution methods, most of which are relaxations that produce lower bounds. In one of the earliest formulations, Gouveia [10] reported several formulations involving node variables. Starting from a basic model incorporating the Miller-Tucker-Zemlin subtour elimination constraints [11], Gouveia proposes strengthened models by lifting these constraints. The lifting is performed using the strong liftings introduced in [12] and also by proposing a new class. Gouveia also proposes a different modeling approach. In [13] an alternative formulation based on directed or undirected multicommodity flows is presented. A shaper model is obtained by lifting the hop-constraints. Another modeling approach is taken in [14], since in this work the problem is formulated in a directed graph. It is expected that the formulation thus obtained is more compact, as is shown in several works (see e.g. [15]) that deal with modeling aspects of related network design problems. In addition, and based on a variable redefinition proposed in [16], an extended formulation is also proposed. A hop indexed formulation is presented in [17] using four indexed decision variables. Since an arc may be in the path from the root node to more than one other node, the authors use the indices for the arc (i, j), the index for the path it is being used on, i.e. to which node, and the number of hops (q) in the path. In [18] a modeling approach that views the whole problem as defined in an appropriate layered directed graph has been proposed and later Gouveia et al [19] have shown that the HMST problem is equivalent to a Steiner tree problem [20, 21] in an adequate layered graph. Moreover, they provided a number of results explaining why this extended HMST formulation is significantly stronger than previous formulations. Akgun and Tansel [22] develop new formulations for HMST. The formulations are based on Miller-Tucker-Zemlin (MTZ) subtour elimination constraints, MTZ-based liftings in the literature offered for HMST, and a new set of topology-enforcing constraints. New sets of constraints to strengthen the MTZ subtour elimination constraints, as well as hop-related topology-enforcing constraints. Akgun [7] presents three new models that use the Asymmetric Travelling Salesman Problem (ATSP) model of Sherali and Driscoll [23] as the basis to formulate HMST. The first is obtained by adapting the aforementioned ATSP; while the other two models are obtained by incorporating the topology-enforcing and MTZ-related constraints offered in [22] into the first model.

In this section, for the sake of simplicity, we give a formulation for the directed HMST problem by Gouveia [13] which is based on a multicommodity network flow formulation.

Consider a directed network $G = (N, A)$, where N is a set of $n+1$ nodes and A is a set of m arcs (i, j). Each arc (i, j) has associated a cost c_{ij}. In the HMST one wishes to find the least cost tree such that all vertices are connected. In addition, the maximum number of arcs on the unique path from the source node to every other node cannot be greater than a hop parameter, H.

Two decision variables are defined for this formulation:

$$x_{ij} = \begin{cases} 1, \text{ if arc } (i,j) \text{ is in the solution tree,} \\ 0, \text{ otherwise,} \end{cases}$$

$$y_{ijk} = \begin{cases} 1, \text{ if arc } (i,j) \text{ is in the path from root node } t \text{ to} \\ \quad \text{node } k,\ k \neq i, \\ 0, \text{ otherwise.} \end{cases}$$

Model 1 A multicommodity flow based formulation for the HMST problem.

$$\text{min:} \quad \sum_{(i,j)\in A} c_{ij} x_{ij}, \tag{1}$$

$$\text{s.t.:} \quad \sum_{i\in N} x_{ij} = 1, \quad \forall_{j\in N\setminus\{t\}}, \tag{2}$$

$$\sum_{i\in N} y_{ijk} - \sum_{i\in N\setminus\{t\}} y_{jik} = 0, \forall_{k,j\in N\setminus\{t\}, j\neq k}, \tag{3}$$

$$\sum_{i\in N} y_{ijj} = 1, \quad \forall_{j\in N\setminus\{t\}}, \tag{4}$$

$$\sum_{(i,j)\in A} y_{ijk} \leq H, \quad \forall_{k\in N\setminus\{t\}}, \tag{5}$$

$$y_{ijk} \leq x_{ij}, \quad \forall_{(i,j)\in A}, \forall_{k\in N\setminus\{t\}}, \tag{6}$$

$$x_{ij} \in \{0,1\}, \quad \forall_{(i,j)\in A}, \tag{7}$$

$$y_{ijk} \in \{0,1\}, \quad \forall_{(i,j)\in A}, \forall_{k\in N\setminus\{t\}}. \tag{8}$$

Equation (1) represents the cost function to be minimized, whiled equation (2) guarantees that every node is in the solution and has only one arc entering it. Decision variables y_{ijk} use a third index as in multicommodity network flow models and constraints (3) refer to the usual flow conservation constraints. The feasibility of the solution regarding the hop constraints is guaranteed by equations (4) and (5). The coupling constraints, which are given by equations (6), state that an arc (i, j) can only be in the path between root node t and node k if the arc is included in the solution. Constraints (7) and (8) state the nature of the decision variables.

3.2. *Solution Methods for the HMST*

We start this section by reviewing solution methods providing lower bounds for the HMST problem, since it has been the most popular method used to approach it. Lower bounds are usually obtained based on the solution of different relaxation techniques applied to the mathematical programming models developed. The second part of the section is devoted to heuristic methods, which are still scarce for this problem.

3.2.1. *Lower Bounds for the HMST*

Gouveia [10] provides models for the HMST problem involving node variables, instead of the usual arc variables, and using extensions of Miller-Tucker-Zemlin subtour elimination constraints. The latter ones are used in order to prevent the solution to incorporate cycles. In order to derive stronger formulations, the author applies Linear Programming (LP) relaxation to the binary variables and then incorporates MTZ constraints. The MTZ subtour elimination constraints are lifted by using Desrocher and Laporte [12] liftings. In addition, subtour elimination constraints earlier developed in [24] are also considered. These constraints are then relaxed in a Lagrangean Relaxation (LR) fashion and the model is solved by a subgradient method. The lower bounds provided by LR models are spanning trees, that may not satisfy the hop constraints. Upper bounds for the HMST problem are then obtained by converting the lower bounds into feasible solutions as follows: all nodes which stand at a distance from the root node t greater than H are disconnected from their parent node and are directly connected to the root node. This is repeated while there are paths exceeding H. Problems with up to 40 nodes, in fully connected graphs, are solved considering linear costs. However, since the results are not as good as expected, stronger formulations are still in need.

Gouveia [13] gives a Multi-Commodity Flow (MCF) formulation for the HMST problem and suggests lower bounding schemes based on LP relaxation and on Lagrangean Relaxation, improving upon his previous work. The author develops a formulation for the undirected case of the HMST problem, from which a formulation for the directed case is derived by replacing each undirected arc $[i,j]$ with two directed arcs (i,j) and (j,i) and where $c_{ij} = c_{ji}$. The hop constraints are lifted deriving two new sharper models. The first one is due to the observation that if an arc (k,v) is in the solution tree, then the number of arcs in the path between the root node t and node k cannot be equal to H. The second one, developed to decrease the number of variables in the model, uses only the subset of the previous constraints involving the lowest cost arc (k,v_k) outgoing from node k. An arc elimination test is developed and it removes any arc (i,j) for which $c_{ij} \leq c_{tj}$. Lagrangean relaxations are applied to the models. The heuristic previously defined in [10] is improved by allowing nodes violating the hop constraint to be linked to another node j as long as the sum of the position of node j and the depth of node i is not greater than H; otherwise, violating nodes are linked to the root node. Although the results show that the LP bounds for these models are considerably better than those of MTZ formulations, MCF formulations lead to very large integer programming models whose LP relaxations require excessive solution times and core storage as the network size and H get larger [25].

Based on the variable redefinition method in [16], Gouveia [14] develops a new lower bounding method given by the LP relaxation of a multicommodity flow formulation (MCF_L). Another heuristic is constructed in several steps. Firstly, a lagrangean relaxation of the MCF formulation is obtained by associating nonnegative multipliers of the form λ_{ijk} to the coupling constraints (6) which are then dualized in the usual lagrangean way (MCF_λ). This new relaxed problem can be divided into two subproblems. The first subproblem involves only the x_{ij} variables and can be easily solved by inspection. The second one involves the y_{ijk} variables and can be further separated into $|N-1|$ subproblems, one for each $k \in N \setminus \{t\}$. Each of these subproblems corresponds to a shortest path problem between node t and node k with at most H hops. Following on the Dynamic Programming (DP) approach proposed in [16], the shortest-path problems can be easily solved by the DP. Furthermore, an extended hop-path formulation can be derived for the original hop-constrained problem. The new formulation is given by considering the binary hop-indexed arc variables z_{ijh}, which assume the value 1 if from state i, which is in position $h-1$, we move to state j in position h. The combination of the extended hop-path formulation with the MCF formulation results in a new model ($HDMCF$) involving the usual x_{ij} variables and new variables z_{ijhk}, the latter ones with an extra index when compared with y_{ijk}, indicating whether an arc (i,j) is in position h of the path from node t to node k. Finally, the binary constraints of the model are relaxed and the new relaxed model ($HDMCF_L$) is solved. The results demonstrate that the bounds of the $HDMCF_L$ model are close to the optimal value. Computational times for solving the MCF_L are larger than those of $HDMCF_L$. This, in part, due to the fact that MTZ formulations are more compact both in the numbers of variables and constraints.

The Lagrangean relaxation proposed in [14] produces very tight bounds for small values of h, however it is very time consuming to obtain bounds that are close to the theoretically possible best bounds. To overcome this problem Gouveia and Requejo [17] develop a new LR to be applied to the hop-indexed MultiCommodity Flow formulation ($HDMCF$) of the HMST problem (explained earlier in this section). To derive the new LR of the model, ($HDMCF_\lambda$), the flow conservation constraints are dualized in the usual lagrangean way. The problem is further simplified, for a given value of the lagrangean multipliers by observing that the optimal values for the binary variables z_{ijhk} can be obtained by dropping the linking constraints $\sum_{h=1}^{H} z_{ijhk} \leq x_{ij}$ and by relaxing the binary variables z_{tj1k} and z_{ijhk} from $HDMCF_\lambda$. This modified relaxation may be separated into two simple inspection subproblems: one involving only the x_{ij} binary variables, and the other involving only the z_{ijhk} binary variables. Once the optimal values for the x_{ij} and z_{ijhk} are obtained, an approximation of the optimal multipliers is also obtained, using a subgradient optimization method. The authors compare the results obtained for three models: the $HDMCF_\lambda$, the $HDMCF_L$, and the MCF_λ. The arc elimination procedure proposed in [13] is applied before solving each problem instance using models $HDMCF_\lambda$ and $HDMCF_L$, which, in some cases, considerably reduces the number of arcs. Problem instances with 20, 40, and 60 nodes, defined over complete graphs are solved. The new lagrangean relaxation $HDMCF_\lambda$ has a better performance than the one developed in [14], both in terms of the bounds found and in terms of the computational time required to find them.

Dahl et al [26] introduce a new formulation for the HMST problem using only natural design variables and an exponential number of constraints composed of the so-called jump inequalities that are shown to be facet-defining. Their proposed formulation uses fewer variables but has weaker LP bounds than MCF formulations. Due to the exponential number of constraints, the authors propose a Lagrangean-based bounding scheme. Computational results indicate that the LP bounds are better as h increases than those reported in previous studies.

More recently, approaches modeling the HMST problem as a Steiner tree in a layered directed graph have been proposed in [18, 27]. A layer is established for each hop value h, where $h = 1, 2, \ldots, H$. Nodes respecting each h value are copied into the corresponding h layer. Therefore, layer 0 only has the source node assigned to it, layer 1 contains all nodes reachable from the source node within one arc, an so on. Each node in layers 1 and $H-1$ is linked to their corresponding node in layer H and have a 0 cost associated. The other arcs of the problem are represented by links between nodes in sequential layers until layer H is reached, creating a graph where a spanning tree in the original graph now corresponds to a Steiner tree. Lower and upper bounds are computed by using a dual ascent heuristic, closely related to the one provided in [28], and a primal heuristic SPH-Prim given in [29], respectively. Then the cutting plane algorithm developed by Hao and Orlin [30] is applied. The overall approach is proved to be very efficient, solving problems with up to 160 nodes in reasonable times.

Akgun and Tansel [22] revisit the node based formulation in [10] and provide new MTZ based formulations, by proposing a new set of MTZ constraints and a new set of topology enforcing constraints that improve the former LP relaxation bounds and solution times. The topology enforcing constraints and the MTZ constraints are based on the distinction between leaf-nodes and central-nodes (nodes with more than one incident arc). The results obtained show that the new constraints are competitive with earlier proposed liftings to MillerTuckerZemlin constraints in [10], some of which are based on the well-known strong liftings introduced in [12]. Also recently, other MTZ based formulations have been proposed by Akgun [7]. The first model $HMST/SD$ incorporates an adaptation of Sherali and Driscoll [23] constraints, developed for the asymmetric TSP, where a linearization of the nonlinear product terms is performed and a more dedicated coupling constraints are added to the model. The second and third models, $HMST/SD1$ and $HMST/SD2$, are obtained by incorporating, respectively, the new MTZ constraints and topology enforcing constraints, previously developed in [22], into the first model. Models $HMST/SD1$ and $HMST/SD2$ dominate the MTZ-based models with the best LP bounds in the literature. Nonetheless, solution times are not improved for optimally solved problems. On the other hand, the results imply that $HMST/SD2$ is likely to produce better solution times for the harder, large-size instances. Comparison of $HMST/SD2$ with flow-based and hop-indexed formulations indicates that $HMST/SD2$ is inferior with respect to LP bounds. However, it can produce good feasible solutions in a very short time.

The interested reader is referred to the comprehensive survey by Dahl et al [25], and to the references therein for a discussion on formulations, including alternative formulations for the HMST problem based on natural design variables and on an exponential sized set of constraints. Solution methods comprising lower bounds for the hop-constrained MST problem, including techniques such as lagrangean relaxation or column generation, are also discussed.

3.2.2. Heuristic Methods for the HMST

It has been argued by Dahl et al [25] that in order to solve realistic sized HMST problems heuristic methods are much more adequate. Nevertheless, although heuristic methods are very often used to solve many optimization problems, there has not been much work on heuristics to solve the HMST problem. As far as the authors are aware of, regarding the development of good heuristic methods to solve the HMST problem, the only existing works are the ones reviewed bellow.

In [31] five heuristic procedures are developed to solve the HMST problem based on ideas proposed for the capacitated minimum spanning tree, taking advantage of the problem similarities. Firstly, a savings heuristic, which belongs to the class of constructive heuristics, is given. In this heuristic the initial solution tree is such that all nodes are linked to the source node. Then, arcs in the solution tree are swapped with arcs not present in the solution tree that represent the best savings. Secondly, a second order algorithm is developed. This algorithm uses different starting solutions generated by the savings heuristic. To generate different initial solutions, the savings heuristic is given a set of allowed arcs S_1, as well as a set of prohibited arcs S_2. Each time a new solution is attempted, different sets S_1 and S_2 are given. Then, in each iteration the savings heuristic is used to find a solution, based on S_1 and on S_2. An improvement is also proposed by transforming S_1 and S_2 into permanent included and excluded arcs, whenever a solution improves the incumbent solution. The other four heuristics proposed are also based on the use of sets S_1 and S_2 and differ only on the rules used to create these sets. The first couple of heuristics do not use set S_2. Heuristic I1 considers simple inhibitions, that is, it defines S_2 as the set of arcs present in the previous solution, provided that they are not linked to the source node. Then, the savings heuristic is run considering the exclusion of one arc (in S_2) at a time. Heuristic I2 is similar to I1 but instead it considers prohibiting two arcs at a time. The second couple of heuristics do not consider S_2. Heuristic J incorporates into S_2, for each node, the cheapest arcs incident to the node and the arcs with the minimum cost which are closer to the source node. The savings heuristic is run forcing one arc at a time to enter the solution. Heuristic J4 uses as candidate arcs to be included in the next solution, that is S_1, the set consisting of the four cheapest arcs incident to each node, provided that they are not linked to the source node and they are not yet in the solution. The fifth and last heuristic is called ILA and is a modified version of I2, where at each iteration a new solution is calculated for each arc to be prohibited. Then, all arcs of that solution are possible candidates for prohibition. Problems with 40 and 80 nodes, defined on complete graphs, have been generated both with the root node at the center (TC) and at the corner of a grid (TE). The use of the elimination test given in [13] substantially reduces the size of the problem instances in TC problems and also of some in TE. The comparison of the results obtained for the described heuristics demonstrates that ILA provides the best solutions, although with higher time requirements. The results previously reported in the literature have been improved.

Correia [32] develops a Genetic Algorithm (GA) to solve HMST problems and tests combinations of different population generation methods, chromosome encodings, and crossover and mutation operators. Two methods have been considered to generate solutions: random and heuristic. The random method does not guarantee that solutions are feasible. Since unfeasible solutions are discarded and many such solutions are generated, computational times for this method are too large (in some cases, no population was generated within the 24 hours allowed). Solutions generated by the heuristic method are feasible and obtained as follows. A solution tree has H levels. At level 0 the tree consists of the root node only, while the remaining levels have a random number of nodes which is at least one. All nodes not in previous levels are associated with the last level. Arcs exist only between nodes in adjacent levels. Two different chromosome codifications are tested: one uses the well-known Prüfer number encoding [33], while the other one uses arcs-set encoding [34]. The Prüfer encoding is associated with the one point crossover and with the single point mutation operators. The arcs-set encoding uses three crossover operators named PrimRST, KruskalRST, and RandomWalkRST. The first two operators are based on Prim's and Kruskal's algorithms for MSTs and the third one is based on a random walk to generate random spanning trees. Regarding the mutation operators, the first operator starts by adding an arc to the solution thus forming a cycle and then removes one other arc from the cycle. The second operator eliminates an arc from the solution and then adds another arc such that the graph remains connected. Child chromosomes resulting from the crossover and mutation operators that do not respect the hop constraint are discarded. The results obtained allowed to conclude that the Prüfer sequence had a better performance for large sized problems although it never found an optimum solution. The arcs-set encoding allowed smaller running times and better solutions for small sized problems (up to 21 nodes). In comparison with literature results, the results obtained provide much worst solutions. In [31] the optimality gap, i.e. $(UB-Opt)/OPT$, is always within 1.5% and 2.7%, for problem instances in TC-1 with 40 and 80 nodes, while in [32] the optimality gaps for the same problems are within 30% and 60%, respectively. Nevertheless, in [32] the computational times, for the larger problems, are much smaller (less that 100 seconds against over 30 minutes).

In [19] a Dynamic Programming (DP) model is developed, based on a node level representation similar to the one proposed in [35]. The size of the state space of the DP grows exponentially with problem size. However, when polynomially restricted, it allows for searching neighborhood structures based on node-level exchanges [35]. The level of a node is defined as the maximum number of arcs a node can have between itself and the source node. Given a positive integer value d, where d can take any value up to the number of nodes n, the state-space restriction rule allows the move of at most d nodes between any consecutive levels. The objective of this rule is to eliminate some states of the original formulation to make the DP more tractable. A restriction on the state-transition rule is also included in order to reduce the computational effort of computing the cost of an optimal tree associated with the states with a depth of at most k. Parallel shift moves starting and ending at different pairs of overlapping levels are prohibited, as well as path moves. Three neighbourhood constructions are defined based on the restricted DP formulation: shift, swap, and shift/swap neighbourhoods, where the latter one is the set of all neighbour solutions obtained either with a swap or a shift move. A standard arc-exchange neighbourhood is also developed, as well as a method combining arc-exchange and swap and shift moves. These neighbourhoods are used to construct five distinct heuristics and the one combining arc-exchange, swap and shift moves has been found to be the best one. The computational experiments have been performed on a set of benchmark instances with 40 and with 80 nodes. The authors have compared their results with the best currently known ones [31], which have been outperformed, both in solution quality and in computational time requirements. In addition, the best performing heuristic produces heuristic solutions that are within 1.2% of optimality always in less that 1 minute.

4. The Hop-Constrained Minimum cost Spanning Tree Problem with Flows

Recently, in [36] a new problem has been defined: the Hop-constrained Minimum cost Spanning Tree problem with Flows (HMFST), which is an extension of the HMST problem. The extension i the fact that, in addition to find the arcs to be used, we also must find the flows to be routed through each arc. Therefore, the main difference between the HMST and the HMFST problem is that the latter problem allows nodes to have different flow demands. Furthermore, in this new problem, the costs to be minimized are nonlinear and made of two components: arc setup costs, as usual, and flow dependent routing costs.

The need for such a problem arises when we think of a transportation network that can be highly compromised if one or several of its links are interrupted. For example, if a section of a railway track blocks the passage of the train transporting some highly degradable commodity, high costs may be incurred by the owning company as time goes by. Therefore, the reliability and availability of the network is of a major importance, and limiting the number of arcs in the network is a way of assuring minimal damage. Another example is the benefit that a water supply network can have from hop constraints by decreasing the number of affected sectors in case of a disruption.

4.1. A formulation for the HMFST problem

Formally, the HMFST problem can be defined as follows. Consider a directed network $G = (N,A)$, where N is the set of $n+1$ nodes, with n demand nodes and one single source node t, and $A(\subseteq N \times N \setminus \{t\})$ is the set of m available arcs (i,j). In the HMFST one wishes to minimize the total costs f_{ij} incurred with the network while satisfying the nodes demand d_j. The total demand of the network, D, is given by the summation of all node demands. The commodity flows from the source node t to the n demand nodes $i \in N \setminus \{t\}$. In addition, the maximum number of arcs on a path from the source node to each demand node is constrained by a hop parameter, H. The position of an arc in the solution tree, counting from the source node t, is represented by an extra index h. The mathematical programming model that is given next for the HMFST problem is an adaptation of the model provided in [37]. Considering the two following decision variables, the model can be written as given in Model 2:

$$
\begin{array}{lcl}
x_{ijh} & - & \text{flow on arc } (i,j) \text{ which is in position } h, \\
y_{ijh} & = & \begin{cases} 1, \text{ if } x_{ijh} > 0, \\ 0, \text{ if } x_{ijh} = 0. \end{cases}
\end{array}
$$

The objective is to minimize the total costs incurred, i.e. costs associated with setting up the arcs and the costs associated with routing the flow through the used arcs, as given in equation (9). Equations (10) guarantee that every node is in the solution in exactly one position. Equations (11) are the balance equations, which in this case also state that if the flow enters a node through an arc in position h, then the flow leaving this node must do it through an arc in position $h+1$. Equations (12) are the coupling constraints and constraints (13) and (14) state the nonnegative and binary nature of the decision variables. It is assumed that the commodity available at the source node t matches the total demand.

Model 2 A mathematical formulation for the HMFST problem.

$$\text{min:} \sum_{i \in N} \sum_{j \in N \setminus \{t\}} \sum_{h=1}^{H} f_{ij}(x_{ijh}, y_{ijh}), \tag{9}$$

$$\text{s.t.:} \sum_{i \in N} \sum_{h=1}^{H} y_{ijh} = 1, \quad \forall_{j \in N \setminus \{t\}}, \tag{10}$$

$$\sum_{i \in N} x_{ijh} - \sum_{i \in N \setminus \{t\}} x_{ji,h+1} = d_j \sum_{i \in N} y_{ijh}, \quad \forall_{j \in N \setminus \{t\}}, \forall_{h \in \{1,\ldots,H-1\}}, \tag{11}$$

$$y_{ijh} \leq x_{ijh} \leq D \times y_{ijh}, \quad \forall_{i \in N}, \forall_{j \in N \setminus \{t\}}, \forall_{h \in \{1,\ldots,H\}}, \tag{12}$$

$$x_{ijh} \geq 0, \quad \forall_{i \in N}, \forall_{j \in N \setminus \{t\}}, \forall_{h \in \{1,\ldots,H\}}, \tag{13}$$

$$y_{ijh} \in \{0,1\}, \quad \forall_{i \in N}, \forall_{j \in N \setminus \{t\}}, \forall_{h \in \{1,\ldots,H\}}. \tag{14}$$

4.2. Solution Methods for the HMFST

The HMFST problem is very recent and, to the moment, there are only three works approaching it: one is focused on an exact method, dynamic programming, and the other two are based on metaheuristic methods.

Fontes [36] uses dynamic programming to solve the HMFST problem. States are defined by three state variables, the set of demand nodes S to be considered, the node x acting as a source node to demand nodes in S, and the hop parameter value h, thus a state is represented as (S, x, h). Stages are defined by the number of demand nodes under consideration in set S. The problem instances solved have 10, 12, 15, 17 and 19 nodes, hop parameter ranging from 3 up to 10, and three distinct nonlinear cost functions with discontinuities, depending on a percentage of the total demand, other than at the origin. The cost functions considered involve two components: a setup or fixed cost incurred by using the arc and a routing cost both linearly and nonlinearly dependent on the flow being routed through the arc. In total, the author solves 4050 problem instances to optimality being able to demonstrate that the computational performance is independent of cost function type.

Fontes and Gonçalves [38] use a Multi-Population hybrid biased random key Genetic Algorithm (MPGA), with three populations evolving separately, in order to solve the HMFST problem. These populations, which are randomly generated, are let to evolve independently and then, after every 15 generations the two best chromosomes are included in all the other populations. The encoding of the solution tree is made recurring to random keys. Therefore, a chromosome is made of $3n$ genes, where n represents the number of nodes in the tree. The first $2n$ genes of the chromosome are used by a decoder procedure called Tree-Constructor to construct a solution tree. (The decoder uses the first n genes to provide an order in which the nodes are considered to enter the tree, and the other n nodes are used to select an antecessor for the corresponding node). The remaining n genes are used later by a local search procedure. To construct the solution tree, the algorithm starts with the highest priority node not yet supplied and searches within the set of the remaining nodes, by priority order, for a feasible supplier, i.e., a node not creating a cycle. The hop-constraint is handled a posteriori, by penalizing infeasible solutions with more than h arcs in the path from the source node. The penalty of a solution is proportional to the number of extra hops in each violated path. Local search is performed by replacing a node in the tree with another node not in the tree, as long as no cycle is formed. The order in which the nodes are considered for the improvement of the solution is given by the last n genes of the chromosome. In order to evolve from a generation to another, a set consisting of the 25% best (top) solutions is directly copied onto the next generation. Then, a biased uniform crossover is performed, between a parent chosen from the top solutions and a parent chosen from the set of all solutions, creating a new chromosome where each gene has a higher (biased) probability to be taken from the best parent. Finally, the mutation operator used, which is called the immigration operator, randomly generates new solutions, as in the first population. The newly generated chromosomes substitute the bottom (worst) 15% chromosomes of the next generation. The results were obtained for 2880 problem instances, and were compared with the exact solutions obtained by the aforementioned DP procedure and with CPLEX, proving that the heuristic is very efficient and effective.

Monteiro et al [39] improve upon the above results by using a hybrid between an Ant Colony Optimization algorithm and a Local Search (LS) procedure, which is named HACO. The HACO algorithm is based on the min-max ant system [40] in the sense that it uses pheromone bounds to avoid the fast convergence of the pheromone trail. The algorithm identifies the five best solutions found by the ants at the current iteration and the LS procedure is applied to them. The LS is based on swaps between arcs already in the solution tree and arcs not in the solution tree, provided that the hop constraint is not violated and that cycles are not created. The algorithm incorporates an extra mechanism for dealing with stagnation and cycling of solutions. Whenever the best solution has not been improved for 200 iterations the pheromone matrix values are reinitialized. Problem instances with up to 80 nodes were solved. One of the advantages of HACO is that the algorithm is always able to find a feasible solution, when there is one, whereas CPLEX and the MPGA are not. The optimality gaps obtained by the HACO algorithm are always as good or better than the ones reported in the literature [38], except for 13 problem instances out of 2798 solved. The computational time requirements of the ACO algorithm were much lower, even when compared with the ones obtained with CPLEX for the larger problem instances.

5. Other problems with hop constraints

In the previous sections, we have reviewed two types of problems with similar characteristics, namely: the solution has the form of a directed tree, there are hop constraints limiting the number of arcs (or nodes) in the path between a source (root) node and every other node in the tree. However, we can find in the literature other problems incorporating hop constraints but that do not have necessarily a tree shaped solution. In this section, we give an account of a few of them.

Voss [41] extends the mathematical formulation of the HMST given in [10] for the Steiner Tree Problem (STP) with hop constraints, and develops a tabu search heuristic based on an edge exchange neighbourhood. Later on, Santos et al [42] propose a heuristic joining together the Dual Ascent algorithm defined in [43] with the graph transformation given in [44].

More recently, hop-constrained STPs with multiple root nodes have been introduced [45, 46]. In this problem, the number of hops between each relevant node and an arbitrary root does not exceed a given hop limit h, where the set of relevant nodes may be equal to the set of terminals or to the union of terminals and root nodes. While in [45] the authors compare flow-based and path-based mixed integer programming models and implement branch-and-price algorithms to solve it, in [46] the problem is solved by using branch-and-cut algorithms for layered graph formulations of the problem, where each root node is associated with one layered graph.

The Hop-Constrained Network Design (HCND) problem, and its variants, are very well studied problems. In it, the objective is to minimize the total costs incurred with the establishment of a network for distributing a given set of commodities, provided that each commodity does no use more than h arcs between its source node and its destination node [47]. In Balakrishnan and Altinkemer [48] the k-arc-disjoint HCND problem was studied when $k = 1$ within the framework of a more general model for backbone networks. The authors gave a mixed-integer programming formulation and developed a Lagrangean-based algorithm to obtain upper and lower bounds, respectively. Other versions have also been studied, see e.g. the case of the 1 arc-disjoint HCND problem with $h = 2$ [49] and the 2 arc-disjoint HCND problem with $h = 3$ [50]. Other very closely related problems have been addressed: the k edge-disjoint 2-hop-constrained paths problem, has been addressed by Dahl et al [51], where an integer programming formulation is proposed and the associated polytope is characterized; the two-edge connected HCND version is considered in [52]. The authors give an integer programming formulation of the problem in the space of the design variables when $H = 2,3$, study the associated polytope, and develop a branch-and-cut.

The Two Level Network Design (TLND) problem has already been addressed in [53, 54, 55], although with other names. This problem is concerned with reliability requirements in a centralized network design problem where two different cable technologies are available. In [53], the problem is shown to be NP-hard and lower and upper bounds are obtained. For the former a Lagrangean relaxation, of the multicommodity flow LP model, together with subgradient optimization is used; while for the latter a Lagrangean heuristic is developed. The problem studied in [55] is an extension of the TLND in which additional transition costs need to be paid for intermediate facilities placed at the transition nodes, i.e., nodes where the change of technology takes place. The authors show how to model the problem in an extended graph based on node splitting and provide a new family of inequalities that implies, and even strictly dominates, all previously described cuts. In addition, a polynomial time separation algorithm is provided.

More recently, in [56] Secondary Hop constraints (SH) have been introduced for the first time. The problem is usually associated with fibre optics lines (primary) versus copper cables (secondary) technology. Since secondary links are less reliable there is the need to impose a maximum number of arcs between the root and each of the secondary nodes. The authors give three MIP formulations: a first formulation considers a directed graph and includes a generalization of the jump constraints given in [57], which are called cut-jump inequalities; the other two formulations model the TLNDSH as a Steiner Arborescence problem with facility and node-degree constraints in a layered graph model, one with and another without node splitting. The model with node splitting provides a stronger formulation.

The hop-constrained path problem consists of finding a minimum cost path with no more than h links between two distinct nodes. Dahl [57] studies the dominant of the convex hull and provides a linear description for $h \leq 3$ and facet defining inequalities for $h \geq 4$. Later, Ben-Ameur [58] defined some classes of 2-connected graphs satisfying path (and cycle)-length constraints. Bley [59] addresses approximation and computational issues for the node-disjoint and edge-disjoint hop-constrained path problems. In particular, he showed that the problem of computing the maximum number of edge-disjoint paths between two given nodes of length equal to 3 is polynomial. This result answered an open question in [60]. In [61] valid inequalities for the directed hop-constrained shortest path problem are discussed. The authors give complete linear characterizations of the hop-constrained path polytope when the maximum number of hops is of 2 or 3. Integer programming formulations are given in [62] for the two 4-hop-constrained paths problem in both the edge and node cases. Coullard et al [63] investigated the structure of the polyhedron associated with the directed st-walks having exactly h arcs of a directed graph, where a directed walk is a directed path that may go through the same node more than once. They presented an extended formulation of the problem and gave a linear description of the associated polyhedron. In [26] it has been considered the case of the directed st-walks with at most h arcs. They present a linear description of that polytope and describe generalized valid inequalities that define facets for the dominant of that polytope, which, quite surprisingly, shows that obtaining a complete description for the dominant of the st-walk polytope when $h = 4$.

6. Conclusion

In this article, we have addressed hop-constrained optimization problems, mainly problems for which the solution is tree-shaped. Hop-constrained trees have much applicability in practical problems specially in the telecommunication area, where the hop constraint is usually associated to the availability and reliability of the network.

We surveyed optimization techniques for the h-constrained minimum spanning tree (HMST). Several relaxations have been proposed over the years for the HMST, mainly based on Lagragean relaxation, with which lower bounds have been obtained. Some upper bound techniques have also been proposed, typically using as a starting point the lower bound solutions. More recently, heuristic solution methods have been developed. The current best results for the existing benchmark problems are due to Gouveia at al [19].

We have also reviewed other network problems involving hop constraints. Amongst these, we focused on the Hop-constrained Minimum cost Spanning Tree problem with Flows (HMFST), which is an extension of the HMST problem. For this problem, up until now only three works addressing it have been reported. One proposing an exact method (dynamic programming) and the other two proposing metaheuristics (a genetic algorithm and an ant colony optimization algorithm). Although the DP approach provides optimal solutions, it can only be used to solve small size problem instances due to the exponential computational requirements. Regarding the metaheuristics, the ant colony optimization algorithm provides better results, both regarding the solution quality and the computational time required to find it.

Regarding future work, we believe that much can still be done in the development of solution approaches, in particular using meta-heuristics, given the usual NP-hard nature of the hop-constrained problems. In addition, there is still room and interest in considering more complex and realistic cost functions.

References

[1] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Report 388, Graduate School of Industrial Administration, CMU, 1976.

[2] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

[3] K.J. Supowit, D. A. Plaisted, and E. M. Reingold. Heuristics for weighted perfect matching. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, STOC '80, pages 398–419, New York, NY, USA, 1980. ACM.

[4] G. Dahl. The 2-hop spanning tree problem. *Operations Research Letters*, 23:21–26, 1998.

[5] K. Woolston and S. Albin. The design of centralized networks with reliability and availability constraints. *Computers & Operations Research*, 15:207–217, 1988.

[6] L.J. LeBlanc and R. Reddoch. Reliable link topology/capacity and routing in backbone telecommunication networks. Working Paper 90-08, Owen Graduate School of Management, Vanderbilt University, 1990.

[7] I. Akgün. New formulations for the hop-constrained minimum spanning tree problem via Sherali and Driscoll's tightened Miller-Tucker-Zemlin constraints. *Computers & Operations Research*, 38:277–286, 2011.

[8] L. Alfandari and V. Th. Paschos. Approximating minimum spanning tree of depth 2. *International Transactions in Operational Researchs*, 6:607–622, 1999.

[9] P. Manyem and M. Stallmann. Some approximation results in multicasting. Working Paper, North Carolina State University, 1996.

[10] L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers and Operations Research*, 22:959–970, 1995.

[11] C. Miller, A. Tucker, and R. Zemlin. Integer programming formulations and travelling salesman problems. *Journal of the ACM*, 7:326–329, 1960.

[12] M. Desrocher and G. Laporte. Improvements to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36, 1991.

[13] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95:178–190, 1996.

[14] L. Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and steiner trees with hop constraints. *Informs Journal on Computing*, 10:180–188, 1998.

[15] T.L. Magnanti and L. Wolsey. *Optimal Trees*, pages 503–615. North-Holland, 1995.

[16] R. Martins. Generating alternative mixed integer programming models using variable redefinition. *Operations Research*, 35:820–831, 1987.

[17] L. Gouveia and C. Requejo. A new lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research*, 132:539–552, 2001.

[18] L. Gouveia, L. Simonetti, and E. Uchoa. Modelling the hop-constrained minimum spanning tree problem over a layered graph. In *In Proceedings of the International Network Optimization Conference, Spa, Belgium,*, 2007.

[19] L. Gouveia, A. Paias, and D. Sharma. Restricted dynamic programming based neighborhoods for the hop-constrained minimum spanning tree problem. *Journal of Heuristics*, 17:23–37, 2011.

[20] F. Hwang, D. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics. North-Holland, Amsterdam, 1992.

[21] N. Maculan. The steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.

[22] I. Akgün and B. Ç. Tansel. New formulations of the hop-constrained minimum spanning tree problem via miller-tucker-zemlin constraints. *European Journal of Operational Research*, 212(2):263–276, 2011.

[23] H.D. Sherali and P.J. Driscoll. On tightening the relaxations of millertuckerzemlin formulations for asymmetric traveling salesman problems. *Operations Research*, 50:656–669, 2002.

[24] J. Edmonds. Optimum branchings. In *Mathematics of the Decision Sciences, Lectures in Applied Mathematics*. G. B. Dantzig and A. F. Veinott, Jr, 1968.

[25] G. Dahl, L. Gouveia, and C. Requeijo. On formulations and methods for the hop-constrained minimum spanning tree problem. In M.G.C. Resende and P.M. Pardalos, editors, *Handbooks of Telecommunications*, pages 493–515. Springer, 2006.

[26] G. Dahl, N. Foldnes, and L. Gouveia. A note on hop-constrained walk polytopes. *Operations Research Letters*, 32(4):345–349, 2004.

[27] 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(1-2):123–148, 2011.

[28] E. Uchoa, M. Poggi de Arago, and C.C. Ribeiro. Dual heuristics on the exact solution of large steiner problems. *Electronic Notes in Discrete Mathematics*, 7:150–153, 2001.

[29] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

[30] J. Hao and J.B. Orlin. A faster algorithm for finding the minimum cut of a graph. In *Proceedings of the 3rd ACM-SIAM Symposium on Discrete Algorithms*, pages 165–174, 2002.

[31] M. Fernandes, L. Gouveia, and S. Voß. Determining hop-constrained spanning trees with repetitive heuristics. *Journal of Telecommunications and Information Technology*, 4:16–22, 2007.

[32] P.M.H. Correia. Problema da árvore de suporte de custo mínimo com restrições de salto. Master's thesis, Universidade de Aveiro, 2010.

[33] H. Prüfer. Neuer bewis eine saizes uber permutationen. *Archiv für Mathematik und Physik*, 27:742–744, 1918.

[34] G.R. Raidl. An effcient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, 2000.

[35] M. Gruber, J. van Hemert, and G.R. Raidl. Neighbourhood searches for the bounded diameter minimum spanning tree problem embedded in a VNS, EA and ACO. In *Proceedings of GECCO06*, Seattle, Washington, 2006.

[36] D. B. M. M. Fontes. Optimal hop-constrained trees for nonlinear cost flow networks. *Infor*, 48, 2010.

[37] L. Gouveia and P. Martins. The capacitated minimal spanning tree problem: an experiment with a hop-indexed model. *Annals of Operations Research*, 86:271–294, 1999.

[38] D. B. M. M. Fontes and J. F. Gonçalves. A multi-population hybrid biased random key genetic algorithm for hop-constrained trees in nonlinear cost flow networks. *Optimization Letters*, 7:1303–1324, 2013.

[39] M.S.R. Monteiro, D.B.M.M. Fontes, and F.A.C.C. Fontes. Hop-constrained minimum spanning tree problems in nonlinear cost network flow: an ant colony optimization approach. *Opt Lett*, submitted, 2013.

[40] T. Stützle and H. Hoos. Max-min ant system and local search for the traveling salesman problem. In *IEEE International Conference On Evolutionary Computation (ICEC'97)*, pages 309–314, Piscataway,NJ, 1997. IEEE Press.

[41] S. Voß. The steiner tree problem with hop constraints. *Annals of Operations Research*, 86:321–345, 1999.

[42] L. Santos, J. Coutinho-Rodrigues, and J.R. Current. An improved ant colony optimization based algorithm for the capacitated arc routing problem. *Transportation Research Part B: Methodological*, 44(2):246 – 266, 2010.

[43] L. Drummond, M. Santos, and E. Uchoa. A distributed dual ascent algorithm for steiner problems in multicast routing. *Networks*, 53(2):170–183, 2009.

[44] L. Gouveia, L. Simonetti, and E. Uchoa. Modelling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs. *On-Line First*, 2009.

[45] L. Gouveia, M. Leitner, and I. Ljubić. On the hop constrained steiner tree problem with multiple root nodes. In A.Ridha Mahjoub, Vangelis Markakis, Ioannis Milis, and VangelisTh. Paschos, editors, *Combinatorial Optimization*, volume 7422 of *Lecture Notes in Computer Science*, pages 201–212. Springer Berlin Heidelberg, 2012.

[46] L. Gouveia, M. Leitner, and I. Ljubić. Hop constrained steiner trees with multiple root nodes. Technical Report TR-186-1-13-02, Technische Universität Wien, 2013.

[47] H. Pirkul and S. Soni. New formulations and solution procedures for the hop constrained network design problem. *European Journal of Operational Research*, 148(1):126–140, 2003.

[48] A. Balakrishnan and K. Altinkemer. Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, 4(2):192–205, 1992.

[49] G. Dahl and B. Johannessen. The 2-path network design problem. *Networks*, 43:190–199, 2004.

[50] D. Huygens, A.R. Mahjoub, and P. Pesneau. Two edge-disjoint hop-constrained paths and polyhedra. *SIAM Journal of Discrete Math*, 18:287–312, 2002.

[51] G. Dahl, D. Huygens, A.R. Mahjoub, and P. Pesneau. On the k edge-disjoint 2-hop-constrained paths polytope. *Operations Research Letters*, 34:577–582, 2006.

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

[53] L. Gouveia and E. Janssen. Designing reliable tree networks with two cable technologies. *European Journal of Operational Research*, 105(3):552–568, 1998.

[54] L. Gouveia and J. Telhada. Distance-constrained hierarchical networks. In *Proceedings of International Network Optimization Conference, INOC 2005*, pages 416–421, 2005.

[55] S. Gollowitzer, L. Gouveia, and I. Ljubic. A node splitting technique for two level network design problems with transition nodes. In *Network Optimization - 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*, volume 6701 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2011.

[56] S. Gollowitzer, L. Gouveia, and I. Ljubic. The two level network design problem with secondary hop constraints. In *Network Optimization - 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*, volume 6701 of *Lecture Notes in Computer Science*, pages 71–76. Springer, 2011.

[57] G. Dahl. Notes on polyhedra associated with hop-constrained paths. *Operations research letters*, 25(2):97–100, 1999.

[58] W. Ben-Ameur. Constrained length connectivity and survivable networks. *Networks*, 36:17–33, 2000.

[59] A. Bley. On the complexity of vertex-disjoint length restricted path problems. *Computational Complexity*, 12:131–149, 2003.

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

[61] G. Dahl and L. Gouveia. On the directed hop-constrained shortest path problem. *Operations Research Letters*, 32(1):15–22, 2004.

[62] D. Huygens and A.R. Mahjoub. Integer programming formulations for two 4-hop-constrained paths problem. *Networks*, 49:135–144, 2007.

[63] C.R. Coullard, A.B. Gamble, and J. Liu. The k-walk polyhedron. In D.-Z. Du and J. J. Sun, editors, *Advances in optimization and approximation*, pages 9–29. Kluwer Scientific Publishers, Norwell, 1994.