Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 223–230
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

An Exact Algorithm for the Steiner Tree Problem with Delays

Valeria Leggieri [1]

Dipartimento di Matematica, Università del Salento, 73100 Lecce, Italy

Mohamed Haouari [2]

Department of Industrial Engineering, Ozyegin University, 34662 Istanbul, Turkey

Chefi Triki [3]

Dipartimento di Matematica, Università del Salento, 73100 Lecce, Italy

Abstract

The Steiner Tree Problem with Delays (STPD) is a variant of the well-known Steiner Tree Problem in which the delay on each path between a source node and a terminal node is limited by a given maximum value. We propose a Branch-and-Cut algorithm for solving this problem using a formulation based on lifted Miller-Tucker-Zemlin subtour elimination constraints. The effectiveness of the proposed algorithm is assessed through computational experiments carried out on dense benchmark instances.

Keywords: Steiner Tree Problem, MTZ subtour elimination constraints, Branch-and-Cut.

[1] Email:`valeria.leggieri@unisalento.it`

[2] Email:`mh6368@yahoo.com`

[3] Email:`chefi.triki@unisalento.it`

doi:10.1016/j.endm.2010.05.029

1 Introduction

The Steiner Tree problem (STP) in graphs is a fundamental $\mathcal{NP}$-hard combinatorial optimization problem. It is defined on an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each edge $e \in E$ has attached a nonnegative cost c_e. Given a node subset $R \subseteq V$ of *terminal nodes*, the STP requires finding a minimum-cost tree T of G that spans all the elements of R, possibly using additional *Steiner nodes* belonging to the set $\bar{R} := V \setminus R$.

In this paper, we address the problem variant where it is required to deliver the same information from a source node towards all the members of a multicast group within a maximum delay limit [7]. This maximum-delay constraint imposes a restriction on an acceptable multicast tree and defines the Steiner Tree Problem with Delays (STPD) (see e.g., [4], [7]). Hence, in the STPD, each edge is assigned a further nonnegative coefficient representing the delay for traversing it, and it is required to find the minimum cost arborescence, rooted at one specific node (called, *source*) and spanning all the terminal nodes within a given maximum delay. To the best of our knowledge, the only approach that optimally solves the STPD has been described in [4]. However, if all the edges are assigned the same delay, then the STPD reduces to a more studied problem namely the hop-constrained STP (see e.g., [2], [3]). In [4], the authors present several reduction procedures, an enhanced polynomial-size formulation, as well as a heuristic. They show that the combination of these features make it feasible to optimally solve large-scale instances that are defined on sparse graphs with up to 1000 nodes. However, their approach failed to solve medium sized instances that are defined on dense graphs.

The main contribution of this paper is to demonstrate that the addition of the so-called *Steiner cuts* significantly tightens the LP relaxation of the aforementioned formulation and makes it possible to optimally solve, using a branch-and-cut (B&C) approach, large-scale instances defined on benchmark complete graphs. The paper is organized as follows. In Section 2, we present a mixed-integer programming formulation for the STPD that both includes lifted Miller-Tuckler-Zimler (MTZ) constraints as well as Steiner cuts, in Section 3, we describe a B&C solution method and finally, computational results are provided in Section 4.

2 Mathematical formulation

2.1 Leggieri et al.'s compact formulation

We formulate the STPD on a bi-directed graph $G = (V, A)$ where $V = \{1, ..., n\}$ is the node set with node $1 \in R$ being the source. Nonnegative weights c_{ij} and delays θ_{ij} are assigned to the arcs $(i, j) \in A$ (with $c_{ij} = c_{ji}$ and $\theta_{ij} = \theta_{ji}$). A dummy node 0 and dummy arcs of the form $(0, j)$ for all $j \in \bar{R} \cup \{1\}$ are added to G. These dummy arcs have zero costs and delays. In [4], Leggieri et al. show that the STPD can be formulated as a shortest spanning arborescence formulation with side-constraints on an expanded graph $G' = (V', A')$ where $V' := V \cup \{0\}$. The STPD amounts to finding a minimum-cost spanning arborescence T' of G' rooted at node 0 and such that the outdegree of a Steiner node adjacent to the root in T' is zero and that the sum of the delays on each path in T' from the source to each terminal node $j \in R^* := R \setminus \{1\}$ is less than or equal to a specified maximum delay Δ.

We denote by $\delta^+_A(i)$ the set of the arcs of A outgoing from i, i.e. $\delta^+_A(i) := \{(i, j) \in A : j \in V\}$ and analogously by $\delta^-_A(i)$ the set of arcs of A incoming in i. Moreover, for each pair of node $i, j \in V$, we indicate by $\theta(i, j)$ the length of the shortest path connecting i to j with the delays as weights.

We introduce two types of variables. We associate with each $i \in V$ a time variable t_i representing the total delay of the path connecting 1 and i, and with each $(i, j) \in A'$ a binary variable x_{ij} that takes value 1 if $(i, j) \in T'$ and 0 otherwise. Variables t_i have zero and Δ as straightforward lower and upper bounds, but the delays on the arcs define a tighter time window within which the communication should be received and forwarded to descendant nodes while satisfying the maximum delay constraints. For each $i \in V$, the length of the shortest path $\lambda_i := \theta(1, i)$ is a tighter lower bound on t_i, and since a Steiner node i reached at time t_i might be included in a feasible arborescence T' only if there exists $j \in R^*$ such that $t_i + \theta(i, j) \leq \Delta$, then $\mu_i := \Delta - \min_{j \in R^*} \theta(i, j)$ is a valid tighter upper bound for t_i. For node 1 we define $\lambda_1 := \mu_1 := 0$ and obviously for each terminal node $i \in R^*$ we set $\mu_i := \Delta$. If $\lambda_i > \mu_i$ and $i \in R$, then the problem is infeasible, while if i is a Steiner node, then i is eliminated from the graph. Analogously, none of the arcs (i, j) such that $\lambda_i + \theta_{ij} > \mu_j$ appears in a feasible solution and, thus they are eliminated (see [4]). Hence, we suppose that $\lambda_i \leq \mu_i$ for each $i \in V$ and we require that $t_i \in [\lambda_i, \mu_i]$.

We define for each $(i, j) \in A$ the set $\varphi_{ji} := \{(k, j) \in A : \; k \in V \setminus \{i\},\ \lambda_k + \theta_{kj} \leq \mu_j \text{ and } \lambda_k + \theta_{kj} + \theta_{ji} > \mu_i\}$, whose elements are all the arcs (k, j) that are incompatible with (j, i), that is both the arcs (k, j) and (j, i) cannot be included in a solution without violating the delay bounds on node

i. For each arc we define the nonnegative coefficients $M_{ij} := \mu_i - \lambda_j + \theta_{ij}$ and $\alpha_{ji} := \mu_i - \lambda_j - \theta_{ji}$, for each $(k,j) \in \varphi_{ji}$, $\beta_{kj} := \lambda_k + \theta_{kj} - \lambda_j$ and for each $(h,i) \in \varphi_{ij}$, $\gamma_{hi} := \max(\mu_i - \mu_h - \theta_{hi}, 0)$. The formulation is the following.

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

$$\text{subject to:} \tag{2}$$

$$\sum_{(i,j)\in\delta^-_{A'}(j)} x_{ij} = 1, \qquad \forall j \in V \tag{3}$$

$$x_{0j} + x_{ij} + x_{ji} \leq 1, \qquad \forall j \in \bar{R}, (i,j) \in \delta^-_A(j) \tag{4}$$

$$\sum_{(i,j)\in\delta^+_A(i)} x_{ij} \geq 1 - x_{0i}, \qquad \forall i \in \bar{R} \tag{5}$$

$$t_i - t_j + M_{ij}x_{ij} + \alpha_{ji}x_{ji} + \sum_{(k,j)\in\varphi_{ji}} \beta_{kj}x_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}x_{hi} \leq M_{ij} - \theta_{ij}, \qquad \forall (i,j) \in A \tag{6}$$

$$\sum_{i:(i,j)\in\delta^-(j)} \max(\lambda_j, \lambda_i + \theta_{ij})x_{ij} \leq t_j, \qquad \forall\, j \in V \setminus \{1\} \tag{7}$$

$$\mu_j - \max(0, \mu_j - \mu_k + \theta_{jk})x_{jk} \geq t_j, \qquad \forall (j,k) \in \delta^+_A(j), j \in V\setminus\{1\} \tag{8}$$

$$t_i \in [\lambda_i, \mu_i], \qquad \forall\, i \in V \tag{9}$$

$$x_{ij} \in \{0,1\}, \qquad \forall\, (i,j) \in A'. \tag{10}$$

The objective function (1) minimizes the total cost and constraints (3) require that on the graph G' the indegree of each node is exactly 1. Constraints (4) enforce each Steiner node adjacent to node 0 to be a leaf and that opposite arcs are not simultaneously selected. Constraints (5) guarantee that a Steiner node not adjacent to 0 has at least one outgoing arc. Constraints (6) are a lifted variant of the so-called Miller-Tucker-Zemlin (MTZ) constraints [5], hence they prevent the solution from including subtours and they enforce (jointly with the domain constraints (9) and (10)) the solution to be delay-feasible [4]. Constraints (7) and (8) are restrictions on the time variables. If $x_{ij} = 1$ for some $j \in V$, then $t_j \geq \lambda_i + \theta_{ij}$ and $t_j \geq \lambda_j$ and in view of (3) it is clear that (7) holds. Constraints (8) are analogously defined.

2.2 Enhancements of the compact formulation

First, it is worth mentioning that an alternative lifting of the MTZ constraints is the following. Define $\alpha'_{ji} := \max(0, \mu_i - \theta_{ij} - \max_{(k,j)\in\delta^-_A(j)}(\lambda_k + \theta_{kj}))$, then

$$t_i - t_j + M_{ij}y_{ij} + \alpha'_{ji}y_{ji} + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}y_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}y_{hi} \leq M_{ij} - \theta_{ij} \tag{11}$$

are valid subtour elimination constraints. Moreover they are not comparable with (6). Indeed, on one hand we have $\alpha_{ji} \geq \alpha'_{ji}$, on the other all the variables corresponding to arcs incoming in j are considered in (11), while in constraints (6) only a subset $\varphi_{ji} \subseteq \delta_A^-(j)$ of them are taken into account.

Moreover, we can append to Model F the valid inequalities

$$\sum_{(i,j)\in\delta_A^+(S)} x_{ij} \geq 1 \qquad \forall S \subset V: \ 1 \in S, \ (V \setminus S) \cap R \neq \emptyset \tag{12}$$

where $\delta_A^+(S) := \{(i,j) \in A : i \in S, \ j \in V \setminus S\}$. These inequalities (introduced by Aneja [1]) are often called Steiner cuts.

In the sequel, we shall refer to the augmented model by F^+.

3 Branch-and-cut solution method

In this section, we provide the details of the implemented solution strategy that uses formulation F^+. Since it has an exponential number of constraints, it is solved by using a B&C algorithm.

- **Separation of the Steiner cuts:** At each node of the B&C tree, we iteratively solve the LP relaxation of the current formulation, and we dynamically append (if it exists) the most violated Steiner cut (12). The separation of Steiner cuts is achieved solving for each source-terminal pair the maximum flow problem (with the current values of x as capacities). If a least one maximum flow values is less than 1, then the minimum capacity cut $(S, V \setminus S)$ is identified and constraint (12) is generated.
- **Additional features:** The strong branching strategy (see e.g., [8]) is used. This technique consists in partially solving a number of subproblems in order to determine the most promising variable to branch. Moreover, the best-bound strategy is chosen, and thus the node with the best objective function value is selected. Finally also constraints (11) are separated. Indeed, if there exists a constraint (11) violated by the current solution, then it is added to the current formulation.
- **Preprocessing and "local" variable setting:** Reduced costs can be used to fix variables at the nodes of the B&C tree (see e.g. [6]). Let (x,t) be the current optimal solution, LLB be its value (it is a local lower bound) and let UB be a global upper bound (the value of the best known feasible solution). For each arc $(i,j) \in A'$, let $\bar{c}_{ij}$ be its reduced cost. If $x_{ij} = 0$ and $LLB + \bar{c}_{ij} > UB$, then x_{ij} can be locally set to zero, whereas if $x_{ij} = 1$ and $LLB - \bar{c}_{ij} > UB$, then x_{ij} can be locally

set to one (where "locally" means that the variables can be fixed to a given value for all the sons of the current node). Setting the values of the variables can impact on the time bounds. If the new value λ_i is strictly greater than its original value, then we locally modify the lower bound in (9), and if i is a Steiner node and the new value μ_i is strictly lower than its original value, then we locally modify the upper bound in (9). Moreover, if i is a Steiner node and for the new values are such that $\lambda_i > \mu_i$, then all variables corresponding to arcs incident in i are locally set to zero, except for the arc $(0, i)$ whose variable is locally fixed to one. Finally, if $\lambda_i + \theta_{ij} > \mu_j$ for an arc (i, j), then x_{ij} is locally fixed to zero. At the beginning of the solution method, the value UB is provided by the heuristic described in [4], then UB is updated every time that better feasible solutions are found in the B&C tree.

- **Synthesis of the B&C algorithm**
  A sketch of the B&C algorithm is the following.

Step 0: Perform the reductions on the original graph (see [4]);
Step 1: Solve the LP relaxation of F, and let (x, t) be its optimal solution;
Step 2: If x is integer then stop. Otherwise if the reduced costs-based reductions eliminate an edge then go to Step 0 else go to Step 3;
Step 3: Perform the $B\&C$ procedure until an optimal integer solution has been found separating constraints (12) and (11), performing the reduced costs preprocessing and the consequent local variable setting and local time bounds strengthening.

Computational experiments have shown that performing the preprocessing at each node of the B&C tree is time consuming. We empirically found that restricting this phase to nodes located at depth 10 and 35 enables to significantly reduce the CPU time.

4 Computational experiments

Experiments were carried out on an Opteron 246 computer with 2 GB RAM memory, using CPLEX 10.2 as solver. Test instances are two classes of complete graphs available at the SteinLib library, namely Berlin52 and Brazil58. These graphs have 52 and 58 nodes and 16 and 25 terminal nodes, respectively. Delays have been randomly generated, so that they are both correlated and non-correlated with the costs. In the first case for each arc (i, j), $\theta_{ij} := r * c_{ij}$ where r is a random number belonging to the interval $[0.8, 1.2]$, whereas in the other the delays are random values belonging to the interval $[1, 100]$. We indicate with Ran the instances where the delays are non-correlated with the costs and with Cor the other case. We have set $\Delta := 1.1 * MP$ in the instances indicated with 0.1 and $\Delta := 1.5 * MP$ in those indicated with 0.5 (where MP is the maximum among the shortest paths with delays as costs between the source and each terminal). For each typology of problem, we have generated 5 instances, solving 40 instances in total. We set the maximum CPU time to 3 hours. We focus on dense instances because Leggieri et al. in [4] found that these instances are very hard to solve. Indeed, they show that formulation F in combination with an aggressive preprocessing phase efficiently solves most of the sparse instances of the classes B, C and D, while it was unable to solve within 1 hour of CPU any of the complete graph instances Brazil58 Cor.

A summary of the results is displayed in Table 1, where we compare two solution strategies: solving F using CPLEX and solving F^+ by $B\&C$. For each solution strategy, we report the mean solution time T (over all the solved instances), the maximum solution time T_M (of the solved instances), the number of unsolved instances NS and the mean gap *gap* (of the unsolved instances) computed as $100 \cdot \frac{UB-LB}{UB}$, where LB is the value of the lower bound obtained upon termination. The B&C approach successfully solved 39 out of 40 instances. The only unsolved instance has a gap of 0.65%. On the contrary, F failed to solve all the 10 instances Brazil58 Cor. Moreover, when a correlated instance is solved by both strategies, the B&C requires significantly shorter CPU times.

Problem	*Solving F with CPLEX*				*Solving F^+ by B&C*		
	T	T_M	US	gap	T	T_M	US
Berlin52 Ran 0.1	0.12	0.18	0		0.11	0.15	0
Berlin52 Ran 0.5	0.29	0.52	0		0.22	0.33	0
Berlin52 Cor 0.1	93.82	290.36	0		4.61	12.09	0
Berlin52 Cor 0.5	820.01	2079.76	0		6.69	9.18	0
Brazil58 Ran 0.1	0.62	1.24	0		0.55	1.22	0
Brazil58 Ran 0.5	5.43	8.76	0		7.85	14.61	0
Brazil58 Cor 0.1	-	-	5	5.28	58.59	193.98	1
Brazil58 Cor 0.5	-	-	5	8.20	964	2837.81	0

Table 1
CPU times (in sec.)

In Table 2, we see the impact of appending the Steiner cuts on the LP relaxation of F. Each entry represents the mean percentage gap (with respect to optimal values) of the LP relaxation of the *preprocessed* instances. The Steiner cuts significantly reduced the integrality gaps of the Cor instances.

In conclusion, we have presented a B&C algorithm for solving the Steiner Tree problem with delay using a formulation based on lifted MTZ subtour

Problem	F	F^+
Berlin 52 0.1 Ran	4.61	4.52
Berlin 52 0.5 Ran	7.87	7.12
Berlin 52 0.1 Cor	16.01	3.39
Berlin 52 0.5 Cor	21.69	3.15
Brazil 58 0.1 Ran	10.88	10.13
Brazil 58 0.5 Ran	20.38	16.76
Brazil 58 0.1 Cor	18.96	6.60
Brazil 58 0.5 Cor	19.15	4.16

Table 2
LP relaxations

elimination constraints and we have provided the evidence that the use of this algorithm results to be effective on correlated dense instances.

References

[1] Y. P. Aneja. An integer linear programming approach to the steiner problem in graphs. *Networks*, 10(2):167–178, 1980.

[2] Alysson M. Costa, Jean-François Cordeau, and Gilbert Laporte. Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks*, 53(2):141–159, 2009.

[3] G. Dahl, L. Gouveia, and C. Requejo. On formulations and methods for the hop-constrained minimum spanning tree problem. In M.G.C. Resende and P.M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, pages 493–515. Springer Science, Business Media, 2006.

[4] V. Leggieri, M. Haouari, S. Layeb, and C. Triki. The steiner tree problem with delays: A tight compact formulation and reduction procedures. Technical report, University of Salento, Lecce, 2007.

[5] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, 7(4):326–329, 1960.

[6] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1988.

[7] C. A. S. Oliveira and P. M. Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers & Operations Research*, 32(8):1953–1981, 2005.

[8] L.A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.