Optimization Letters manuscript No.
(will be inserted by the editor)

A Branch-and-Cut Algorithm for the Steiner Tree Problem with Delays

V. Leggieri · M. Haouari · C. Triki

Received: date / Revised version: date

Abstract In this paper, we investigate the Steiner Tree Problem with Delays (STPD), which is a generalized version of the Steiner tree problem applied to multicast routing. For this challenging combinatorial optimization problem, we present an enhanced directed cut-based MIP formulation and an exact solution method based on a branch-and-cut approach. Our computational study reveals that the proposed approach can optimally solve hard dense instances.

Keywords Steiner tree problem · Delay constraints · Branch-and-cut method

1 Introduction

Given a connected undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges and assigned to each edge $e \in E$ a nonnegative cost c_e, the *Steiner Tree Problem* (STP) requires finding a minimum-cost tree of G that spans a node subset $R \subset V$, and possibly additional nodes belonging to the subset $S := V \setminus R$. The elements of set R are called *terminal nodes*, whereas those of S are called *Steiner nodes*. Since R and S are such that $R \cap S = \emptyset$ and $S \cup R = V$, they form a partition of the set V. It is well-known that the STP is an $\mathcal{NP}$-hard problem and since many important network applications can be modeled as STP, it has been extensively studied (see e.g. [2], [9], [14], and [23]).

In this paper, we suppose that each edge of the network is assigned not only a cost, but also a nonnegative integer coefficient that represents the delay in passing through it. Therefore we consider the *Steiner Tree Problem with Delays* (STPD) whose aim is to find a minimum cost arborescence, rooted at one specific node and spanning all the terminal nodes within a given maximal delay. Clearly, the STP is a special case of the

Valeria Leggieri
Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
Dipartimento di Matematica, Università del Salento, 73100 Lecce, Italy

Mohamed Haouari
Ecole Polytechnique de Tunisie, BP 743, 2078, La Marsa, Tunisia

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

STPD with the source (or root) node being any terminal node and the maximum delay being equal to infinity. Thus, the STPD is $\mathcal{NP}$-hard, too.

Satisfying the maximal delay requirement is of crucial importance in all network applications where it is required to spread the same information from a source toward all the members of a multicast group within a specified delay limit ([16], [21], [22]). Heuristic approaches to the STPD have been proposed so far by several authors (see e.g., [16] [25], [10], [11] and [17]). However to the best of our knowledge the work in [18] has been the first significant contribution for solving this problem to the optimality. In [18], indeed, the authors have provided the evidence that combining together reduction procedures with an enhanced polynomial size formulation based on lifted Miller-Tucker-Zemlin subtour elimination constraints allows to solve to optimality sparse instances of the STPD having up to 1000 nodes, but the proposed approach was inappropriate to solve medium size dense instances.

If the delays on all the edges have the same value, then the STPD might be viewed as a generalization the *hop-constrained minimum Steiner tree problem* where a hop-constraint requires that the number of arcs in the path connecting the source node with any terminal node does not exceed a given threshold value (see e.g., [12], [24]). Recently, exact and heuristic approaches for the STP with revenues, budget and hop constraints have been proposed by Costa *et al.* in [7] and [8].

The objective of this paper is to propose a branch-and-cut algorithm for solving to optimality the STPD. To that aim, we provide a directed cut-based formulation whose LP relaxation is compared with that of a previously presented formulation [18]. In addition, we describe several valid inequalities for the STPD. Finally, we provide empirical evidence that solving the enhanced formulation using a branch-and-cut approach makes it possible to solve to optimality hard STPD instances.

The remainder of the paper is organized as follows: in Section 2, we propose a directed cut-based formulation STPD_{DC} that uses lifted Miller-Tucker-Zemlin constraints (MTZ for short) with the purpose of preventing subtours and enforcing delay constraints as well. Section 3 is devoted to the comparison of the linear programming (LP) relaxations of STPD_{DC} with that of an alternative valid formulation that has been proposed in [18]. In Section 4, we present a new possible lifting of the MTZ constraints and several valid inequalities that have been included to enhance the solution procedure. In Section 5, we provide a detailed description of a branch-and-cut solution method. Finally, Section 6 is devoted to the computational results.

2 An enhanced cut-based formulation

Let $G = (V, E)$ be a connected, undirected graph, where $V = \{1, ..., n\}$ is the node set and node 1 is the source node ($1 \in R$). Each edge $e \in E$ is assigned a nonnegative weight c_e and a nonnegative delay θ_e. The STPD consists in finding a minimum-cost subtree T of G that spans all the terminal nodes of R and possibly some of the Steiner nodes and such that the sum of the delays on each path $P(1, j)$ in T from the root node 1 to each terminal node $j \in R^* := R \setminus \{1\}$ is less than or equal to a specified maximum delay Δ. In this paper, the maximal delay Δ is set to the same value for each terminal nodes. However, all our results could easily be extended to the case with node dependent maximal delays. For the sake of convenience, and since the flow between the source and each terminal is naturally directed, we define the STPD on a bi-directed graph $B = (V, A)$. This latter digraph is obtained from G by replacing each edge $e = \{i, j\} \in E$ with two directed arcs (i, j) and (j, i) (with $c_{ij} = c_{ji} = c_e$ and $\theta_{ij} = \theta_{ji} = \theta_e$). However, since all the arc costs and delays are nonnegative, then the arcs that are incident to the source node are not created.

In the sequel, we shall 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 by $\delta_A^-(i)$ the set of arcs of A that are incoming in i, i.e. $\delta_A^-(i) := \{(j, i) \in A : j \in V\}$. Moreover if W is a subset of V, then $\delta_A^+(W)$ is the set of the arcs with tails in W and heads in $W^c := V \setminus W$, i.e. $\delta_A^+(W) = \{(i, j) \in A : i \in W, j \in W^c\}$. Furthermore, for each pair of nodes i and j in V, we denote by $D(i, j)$ the shortest path connecting i to j with the delays as weights and we indicate by $\theta(i, j)$ its length. Since the delays are nonnegative, then the computation of $D(i, j)$ and of $\theta(i, j)$ can be performed in polynomial-time.

In order to formulate the STPD, we define, for each node $i \in V$, a continuous time variable t_i that represents the total delay of the path connecting the root node 1 to node i. Obviously, in any feasible solution, the value of t_i should lie within the interval $[0, \Delta]$. Actually, we can possibly define for each node $i \in V$ a tighter time window $[\lambda_i, \mu_i]$ within which the communication should be received and forwarded by i to its descendant nodes while satisfying the maximum delay constraints. Clearly, the total elapsed time for a message sent from the root node to a node $j \in V \setminus \{1\}$ is larger than or equal to the value of the shortest path $\theta(1, j)$. Thus, we set $\lambda_j := \theta(1, j)$ for each $j \in V \setminus \{1\}$ and $\lambda_1 := 0$. Moreover, a Steiner node $i \in S$ reached at time t_i might be included in a feasible arborescence T only if there exists a terminal node $j \in R^*$ such that $t_i + \theta(i, j) \leq \Delta$. Consequently, we set $\mu_i := \Delta - \min_{j \in R^*} \theta(i, j)$ for each $i \in S$, $\mu_i := \Delta$ for each $i \in R^*$, and $\mu_1 := 0$ for the root node. As noticed in [18] for each node i the values λ_i as well as μ_i are respectively a lower bound and an upper bound on variable t_i. Furthermore, if $\lambda_i > \mu_i$ holds for a terminal node ($i \in R^*$) then the problem is clearly infeasible, while if $\lambda_i > \mu_i$ for a Steiner node ($i \in S$), then this node can be eliminated from the graph since it cannot be included in any feasible solution. Thereby, in the sequel, we suppose that $\lambda_i \leq \mu_i$ for all $i \in V$ and thus, we require that, for each node $i \in V$, $t_i \in [\lambda_i, \mu_i]$. It is noteworthy that if $\lambda_i + \theta_{ij} > \mu_j$ holds for arc $(i, j) \in A$, then this arc can be eliminated, since it would never appear in any feasible solution (see [18]).

Define for each arc $(i, j) \in A$, the set

$$\varphi_{ji} = \{(k, j) \in A : \quad k \in V \setminus \{i\}, \; \lambda_k + \theta_{kj} \leq \mu_j \text{ and } \lambda_k + \theta_{kj} + \theta_{ji} > \mu_i\}, \qquad (1)$$

whose elements are all the arcs (k, j) that are incompatible with arc (j, i). Namely, if both arcs (k, j) and (j, i) are included in a solution, then the delay constraint of node i would be violated. Also, define, for each arc $(i, j) \in A$, the following nonnegative coefficients $M_{ij} := \mu_i - \lambda_j + \theta_{ij}$ and $\alpha_{ji} := \mu_i - \lambda_j - \theta_{ji}$. Moreover, we set $\beta_{kj} := \lambda_k + \theta_{kj} - \lambda_j$ for all $(k, j) \in \varphi_{ji}$ and $\gamma_{hi} := \max(\mu_i - \mu_h - \theta_{hi}, 0)$ for all $(h, i) \in \varphi_{ij}$. Notice that all these coefficients are nonnegative.

2.1 A cut-based formulation

We define for each arc $(i, j) \in A$ a binary variable y_{ij} that takes value 1 if arc (i, j) belongs to the arborescence T and 0 otherwise. We can formulate the STPD as follows:

$$(STPD_{DC}) : \text{Minimize} \quad \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{2}$$

subject to:

$$\sum_{(i,j)\in\delta_A^+(W)} y_{ij} \geq 1 \qquad \forall W \subset V,\ 1 \in W, \quad R^* \cap W^c \neq \emptyset \tag{3}$$

$$t_i - t_j + M_{ij} y_{ij} + \alpha_{ji} y_{ji} + \sum_{(k,j)\in\varphi_{ji}} \beta_{kj} y_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi} y_{hi} \leq M_{ij} - \theta_{ij}, \qquad \forall\ (i,j) \in A \tag{4}$$

$$t_i \in [\lambda_i, \mu_i] \qquad \forall i \in V \tag{5}$$

$$y_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{6}$$

The objective function (2) requires minimizing the total cost. Constraints (3) require that, for each cutset (W, W^c) separating the root node from at least a terminal node, there should be at least one arc outgoing from W (see e.g., [3]). Constraints (4) prevent the solution from including subtours, and also, jointly with constraints (5) and (6), they enforce the solution to be delay-feasible. Constraints (4) are included for the purpose of enforcing that the following requirement (if $y_{ij} = 1 \Rightarrow t_i + \theta_{ij} \leq t_j$) holds for $(i,j) \in A$. They have been first proposed in [18] where it is shown that they can be derived through lifting the so-called Miller-Tucker-Zemlin constraints [19].

To strengthen the LP relaxation of formulation STPD_{DC} we append the following valid constraints. First, we observe that at most one of the two opposite arcs could be contained in a feasible solution. That is, we have:

$$y_{ij} + y_{ji} \leq 1 \qquad \forall\, (i,j) \in A : (j,i) \in A. \tag{7}$$

Also, constraints

$$\sum_{(j,i)\in\delta^-(i)} y_{ji} \leq \sum_{(i,j)\in\delta^+(i)} y_{ij} \qquad \forall\, i \in S \tag{8}$$

enforce each Steiner node with one incoming arc to have at least one outgoing arc (flow-balance constraints see e.g. [23]). Finally, we append to formulation STPD_{DC} the time-bound constraints

$$\sum_{(i,j)\in\delta^-(j)} \max(\lambda_j, \lambda_i + \theta_{ij}) y_{ij} \leq t_j, \qquad \forall\ j \in V \setminus \{1\} \tag{9}$$

and

$$\mu_j - \max(0, \mu_j - \mu_k + \theta_{jk}) y_{jk} \geq t_j, \qquad \forall (j,k) \in \delta^+(j),\ j \in V \setminus \{1\} \tag{10}$$

that are further restrictions on the time variables.

In the sequel, we indicate by STPD_{DC} the directed cut formulation that includes constraints (7), (8), (9) and (10).

3 Comparison of LP relaxations

Now we want to highlight that the optimal value of the linear programming (LP) relaxation of $\mathrm{STPD_{DC}}$ is greater than that of a shortest spanning arborescence formulation with side-constraints (indicated here as $\mathrm{STPD_{LMTZ}}$) proposed and described in [18]. Formulation $\mathrm{STPD_{LMTZ}}$ is constructed on an expanded graph $B' = (V', A')$ obtained from $B(V, A)$ by adding a dummy node 0 as well as dummy arcs of the form $(0, j)$ for all the nodes $j \in S \cup \{1\}$ with zero costs and zero delays. It has, in common with $\mathrm{STPD_{DC}}$, the objective function and the liftings (4) , and then it contains constraints on the degrees of the nodes. Such a formulation is very simple and compact since it contains a number of constraints and variables which is polynomial in the size of the problem.

For the sake of completeness we report formulation $\mathrm{STPD_{LMTZ}}$, where the binary variables x_{ij} take the value 1 if arc $(i, j) \in A'$ belongs to the spanning arborescence and 0 otherwise.

$$(STPD_{LMTZ}) : \text{Minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{11}$$

subject to:

$$\sum_{(i,j)\in\delta^-_{A'}(j)} x_{ij} = 1, \qquad \forall\, j \in V \tag{12}$$

$$x_{0j} + x_{ij} + x_{ji} \leq 1, \qquad \forall\, j \in S,\ (i, j) \in \delta^-_A(j) \tag{13}$$

$$\sum_{(i,j)\in\delta^+_A(i)} x_{ij} \geq 1 - x_{0i}, \qquad \forall\, i \in S \tag{14}$$

$$t_i - t_j + M_{ij}x_{ij} + \alpha_{ji}x_{ji} + \sum_{(k,j)\in\varphi_{ji}} \beta_{kj}x_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}x_{hi} \leq M_{ij} - \theta_{ij}, \qquad \forall\, (i, j) \in A \tag{15}$$

$$t_i \in [\lambda_i, \mu_i], \qquad \forall\, i \in V, \tag{16}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall\, (i, j) \in A'. \tag{17}$$

Constraints (9) and (10) are also valid for formulation $\mathrm{STPD_{LMTZ}}$, and thus appended to it.

In order to compare the LP relaxations, we first provide the following result.

Lemma 1 *Let $(\bar{y}, \bar{t})$ be an optimal solution for the LP relaxation of $STPD_{DC}$, then*

$$\sum_{(k,j)\in\delta^-_A(j)\setminus\{(i,j)\}} \bar{y}_{kj} \geq \bar{y}_{ji} \qquad \forall j \in V \setminus \{1\},\ (j, i) \in A, \tag{18}$$

and

$$\sum_{(i,j)\in\delta^-_A(j)} \bar{y}_{ij} \leq 1 \qquad \forall j \in V \setminus \{1\}. \tag{19}$$

Proof See [23].

□

Denoting by $v(P)$ the optimal value of the LP relaxation of a generic formulation P, we can claim that:

Proposition 1 $v(STPD_{DC}) \geq v(STPD_{LMTZ})$.

Proof In order to compare the LP relaxation of formulations $\mathrm{STPD_{DC}}$ and $\mathrm{STPD_{LMTZ}}$, we need to augment formulation $\mathrm{STPD_{DC}}$ with the variables associated with the arcs $(0,i)$ for $i \in S \cup \{1\}$. Given an optimal solution (y^*, t^*) of the LP relaxation of $\mathrm{STPD_{DC}}$, as in [23], we define $\overline{y}_{ij} := y^*_{ij}$ for each $(i,j) \in A$ and we set $\overline{y}_{0j} := 1 - \sum_{(i,j) \in \delta^-_A(j)} y^*_{ij}$ for the arcs $(0,j)$ with $j \in S \cup \{1\}$. The solution $(\overline{y}, t^*)$ is still an optimal solution for the LP relaxation of $\mathrm{STPD_{DC}}$, since the costs associated with the arcs $(0,j)$ with $j \in S \cup \{1\}$ are zero. Now for each $(i,j) \in A'$ we set $x^*_{ij} = \overline{y}_{ij}$ and we show that (x^*, t^*) is a feasible solution for the LP relaxation of $\mathrm{STPD_{LMTZ}}$.
Let prove that (x^*, t^*) satisfies constraints (12), first for the root node, then for the Steiner nodes and, finally, for the terminal nodes. None of the arcs of A incomes in the root node 1, hence $\delta^-_A(1) = \emptyset$ and then

$$\sum_{(j,1) \in \delta^-_{A'}(1)} x^*_{j1} = x^*_{01} = 1 - \sum_{(j,1) \in \delta^-_A(1)} \bar{y}_{j1} = 1.$$

Let j be e Steiner node ($j \in S$), then

$$\sum_{(i,j) \in \delta^-_{A'}(j)} x^*_{ij} = \sum_{(i,j) \in \delta^-_A(j)} x^*_{ij} + x^*_{0j} = \sum_{(i,j) \in \delta^-_A(j)} \bar{y}_{ij} + 1 - \sum_{(i,j) \in \delta^-_A(j)} \bar{y}_{ij} = 1.$$

Finally, let j be a terminal node ($j \in R^*$), then because of constraints (19) and (3) with $W = V \setminus \{j\}$, it follows that:

$$\sum_{(i,j) \in \delta^-_{A'}(j)} x^*_{ij} = \sum_{(i,j) \in \delta^-_A(j)} x^*_{ij} = \sum_{(i,j) \in \delta^-_A(j)} \bar{y}_{ij} = 1.$$

Thus, constraints (12) are fulfilled by (x^*, t^*). Let consider, now, constraints (13) and let j be in S and (j,i) in $\delta^+_{A'}(j)$. In view of constraints (18) it follows that:

$$x^*_{0j} + x^*_{ij} + x^*_{ji} = 1 - \sum_{(k,j) \in \delta^-_A(j)} \bar{y}_{kj} + \bar{y}_{ij} + \bar{y}_{ji} = 1 - \sum_{(k,j) \in \delta^-_A(j) \setminus \{(i,j)\}} \bar{y}_{kj} + \bar{y}_{ji} \leq 1,$$

and thus constraints (13) are fulfilled by (x^*, t^*). It remains to prove that (x^*, t^*) satisfies constraints (14). Let $i \in S$, since constraints (8) are fulfilled by $\bar{y}$, it holds:

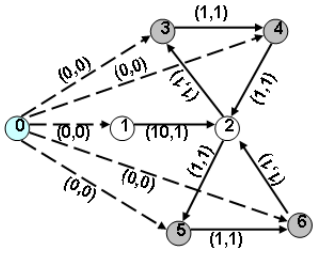
$$\sum_{(i,j) \in \delta^+_A(i)} x^*_{ij} = \sum_{(i,j) \in \delta^+_A(i)} \bar{y}_{ij} \geq \sum_{(j,i) \in \delta^-_A(i)} \bar{y}_{ji} = 1 - x^*_{0i}.$$

The other constraints of $\mathrm{STPD_{LMTZ}}$ are obviously satisfied since they belong also to $\mathrm{STPD_{DC}}$. Therefore, (x^*, t^*) is feasible for the LP relaxation of $\mathrm{STPD_{LMTZ}}$ and hence the thesis follows.

□

The example of Figure 1, reported from [23], can be used to show that there exist Steiner Tree problems with delay constraints in which the optimal solution of the LP relaxation of $\mathrm{STPD_{LMTZ}}$ is not feasible for the LP relaxation of $\mathrm{STPD_{DC}}$.

Fig. 1 Example with $v(STPD_{DC}) > v(STPD_{LMTZ})$



Example 1 Consider the graph in Figure 1, where $R = \{1, 2\}$ and $\Delta = 10$. The delay constraints in this case are redundant for defining any optimal solution. The solution in the variables x is: $x_{03} = x_{04} = x_{25} = x_{56} = x_{62} = \frac{1}{3}$, $x_{23} = x_{34} = x_{42} = x_{05} = x_{06} = \frac{2}{3}$, $x_{01} = 1$. It is optimal for the LP relaxation of $\mathrm{STPD_{LMTZ}}$, but the corresponding solution y is not feasible for the LP relaxation of $\mathrm{STPD_{DC}}$, since if $W = \{1\}$, then $\sum_{(i,j)\in\delta^+(W)} y_{ij} = 0$.

4 Valid inequalities for $\mathbf{STPD_{DC}}$ and a new lifting of the MTZ constraints

In this section, we present classes of valid inequalities that have been used in our implementation, and we propose a different lifting of the MTZ constraints. Before doing this, we append to formulation $\mathrm{STPD_{DC}}$ the inequalities:

$$\sum_{(i,j)\in\delta_A^-(j)} y_{ij} \leq 1 \qquad \forall j \in V \setminus \{1\}. \tag{20}$$

that are fulfilled by any of its optimal solutions.

4.1 A new lifting of the MTZ constraints

Here we present an alternative lifting of the MTZ constraints $t_i - t_j + \theta_{ij} \leq M(1 - y_{ij})$, $\forall (i,j) \in A$ which is incomparable with constraints (4).

First, let consider the results of the following lemma that extends those of the lemma reported in [18].

Lemma 2 *Let $(\bar{y}, \bar{t})$ be a feasible solution for $STPD_{DC}$. Suppose that $\bar{y}_{ij} = 1$ for an arc $(i,j) \in A$. Then:*

i) if $(j,i) \in A$ then $\bar{y}_{ji} = 0$;
ii) $\bar{y}_{kj} = 0$ for any $(k,j) \in \delta_A^-(j) \setminus \{(i,j)\}$;
iii) if $(j,i) \in A$ then $\bar{t}_j = \bar{t}_i + \theta_{ij}$;
iv) $\bar{y}_{hi} = 0$, for any $(h,i) \in \varphi_{ij}$.

Proof The thesis easily follows (see [18]).

□

Constraints (4) are not the only possible lifting of the MTZ constraints, indeed, if we define $\alpha'_{ji} := \max(0, \mu_i - \theta_{ij} - \max_{(k,j)\in\delta^-_A(j)} (\lambda_k + \theta_{kj}))$, then we can state:

Proposition 2 *For each $(i, j) \in A$, the inequality*

$$t_i - t_j + M_{ij}y_{ij} + \alpha'_{ji}y_{ji} + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}y_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}y_{hi} \leq M_{ij} - \theta_{ij} \qquad (21)$$

is valid for $STPD_{DC}$.

Proof Let $(\bar{y}, \bar{t})$ be a feasible integer solution and let (i, j) be an arc of A. We will consider three different cases:

Case 1: $\bar{y}_{ij} = 1$. In view of lemma 2, we should verify that $\bar{t}_i - \bar{t}_j + M_{ij} \leq M_{ij} - \theta_{ij}$, but since $\bar{y}_{ij} = 1$, then the last inequality is obviously fulfilled.

Case 2: $\bar{y}_{ji} = 1$. In view of the points *i)* and *ii)* of lemma 2 applied to the arc (j, i), it holds that $y_{ij} = 0$ and $y_{hi} = 0$ for all $(h, i) \in \varphi_{ij}$, hence we should verify that

$$\bar{t}_i - \bar{t}_j + \alpha'_{ji}\bar{y}_{ji} + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}\bar{y}_{kj} \leq M_{ij} - \theta_{ij}. \qquad (22)$$

Suppose that $\alpha'_{ji} = 0$. If $\bar{y}_{kj} = 0$ for all $(k, j) \in \delta^-_A(j)$ (this situation may occur if j is a Steiner node), since $\bar{t}_i \leq \mu_i$ and $\bar{t}_j \geq \lambda_j$, it holds that $\bar{t}_i - \bar{t}_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$ and thus (22) follows. Else, if there exists an arc $(k', j) \in \delta^-_A(j)$ such that $\bar{y}_{k'j} = 1$, then $\bar{y}_{kj} = 0$ for all $(k, j) \in \delta^-_A(j) \setminus \{(k', j)\}$. In view of constraints (9), it holds that $\bar{t}_j \geq \max(\lambda_j, \lambda_{k'} + \theta_{k'j}) \geq \lambda_{k'} + \theta_{k'j}$, and hence $\bar{t}_i - \bar{t}_j + \alpha'_{ji}\bar{y}_{ji} + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}y_{kj} = \bar{t}_i - \bar{t}_j + \beta_{k'j} = \bar{t}_i - \bar{t}_j + \lambda_{k'} + \theta_{k'j} - \lambda_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$. Suppose on the contrary that $\alpha'_{ji} \neq 0$. If $\bar{y}_{kj} = 0$ for all $(k, j) \in \delta^-_A(j)$, since λ_j is the shortest path value in terms of delay from the root node to j, then $\lambda_j \leq \max_{(k,j)\in\delta^-_A(j)}(\lambda_k + \theta_{kj})$. Furthermore, since $\bar{y}_{ji} = 1$, then $\bar{t}_i = \bar{t}_j + \theta_{ij}$. Therefore $\bar{t}_i - \bar{t}_j + \alpha'_{ji} = \bar{t}_i - \bar{t}_j + \mu_i - \theta_{ij} - \max_{(k,j)\in\delta^-_A(j)}(\lambda_k + \theta_{kj}) \leq \mu_i - \lambda_j = M_{ij} - \theta_{ji}$. Else if there exists an arc $(k', j) \in \delta^-_A(j)$ such that $\bar{y}_{k'j} = 1$, since $\max_{(k,j)\in\delta^-_A(j)}(\lambda_k + \theta_{kj}) \geq \lambda_{k'} + \theta_{k'j}$, and since $\bar{t}_i = \bar{t}_j + \theta_{ij}$, then it holds that: $\bar{t}_i - \bar{t}_j + \alpha'_{ji}\bar{y}_{ji} + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}y_{kj} = \bar{t}_i - \bar{t}_j + \alpha'_{ji} + \beta_{k'j} = \bar{t}_i - \bar{t}_j + \mu_i - \theta_{ij} - \max_{(k,j)\in\delta^-_A(j)}(\lambda_k + \theta_{kj}) + \lambda_{k'} + \theta_{k'j} - \lambda_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ji}$.

Case 3: $\bar{y}_{ij} = 0$ and $\bar{y}_{ji} = 0$. We consider the following subcases:

i) $\bar{y}_{k'j} = 1$, and $\bar{y}_{h'i} = 1$ for the arcs $(k', j) \in \delta^-_A(j)$ and $(h', i) \in \varphi_{ij}$ respectively. Since $\bar{y}_{h'i} = 1$, it holds that $\bar{t}_i \leq \mu_{h'} + \theta_{h'i}$, but $\bar{t}_i \leq \mu_i$ and hence $\bar{t}_i \leq \min(\mu_i, \mu_{h'} + \theta_{h'i})$. Since $\bar{y}_{k'j} = 1$, then $\bar{t}_j \geq \lambda_{k'} + \theta_{k'j}$. Suppose that $\mu_i \leq \mu_{h'} + \theta_{h'i}$, then $\gamma_{h'i} = 0$ and it follows that $\bar{t}_i - \bar{t}_j + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}\bar{y}_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}\bar{y}_{hi} = \bar{t}_i - \bar{t}_j + \beta_{k'j} = \bar{t}_i - \bar{t}_j + \lambda_{k'} + \theta_{k'j} - \lambda_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$. Suppose on the contrary that $\mu_i > \mu_{h'} + \theta_{h'i}$, then $\bar{t}_i \leq \mu_{h'} + \theta_{h'i}$ and thus $\bar{t}_i - \bar{t}_j + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}\bar{y}_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}\bar{y}_{hi} = \bar{t}_i - \bar{t}_j + \beta_{k'j} + \gamma_{h'i} = \bar{t}_i - \bar{t}_j + \lambda_{k'} + \theta_{k'j} - \lambda_j + \mu_i - \mu_{h'} - \theta_{h'i} \leq \mu_{h'} + \theta_{h'i} - \lambda_{k'} - \theta_{k'j} + \lambda_{k'} + \theta_{k'j} - \lambda_j + \mu_i - \mu_{h'} - \theta_{h'i} = \mu_i - \lambda_j = M_{ij} - \theta_{ij}$.

ii) $y_{h'i} = 1$ for an arc $(h', i) \in \varphi_{ij}$ and $y_{kj} = 0$ for all the arcs $(k, j) \in \delta^-_A(j)$. If $\gamma_{h'i} = 0$ then it easily follows that: $\bar{t}_i - \bar{t}_j + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}\bar{y}_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}\bar{y}_{hi} = \bar{t}_i - \bar{t}_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$. Else if $\gamma_{h'i} > 0$ then $\mu_i > \mu_{h'} + \theta_{h'i}$, and $\bar{t}_i \leq \mu_{h'} + \theta_{h'i}$. Hence $\bar{t}_i - \bar{t}_j + \gamma_{h'i} = \bar{t}_i - \bar{t}_j + \mu_i - \mu_{h'} - \theta_{h'i} \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$ and, thus, (21) holds.

iii) $y_{hi} = 0$ for all the arcs $(h,i) \in \varphi_{ij}$ and $y_{k'j} = 1$ for an arc $(k',j) \in \delta^-_A(j)$. As proved above, in this case $\bar{t}_j \geq \lambda_{k'} + \theta_{k'j}$ and hence $\bar{t}_i - \bar{t}_j + \sum_{(k,j)\in\delta^-_A(j)} \beta_{kj}\bar{y}_{kj} + \sum_{(h,i)\in\varphi_{ij}} \gamma_{hi}\bar{y}_{hi} = \bar{t}_i - \bar{t}_j + \beta_{k'j} \leq \bar{t}_i - \bar{t}_j + \lambda_{k'} + \theta_{k'j} - \lambda_j \leq \mu_i - \lambda_j = M_{ij} - \theta_{ij}$.

iv) $y_{hi} = 0$ for all the arcs $(h,i) \in \varphi_{ij}$ and $y_{kj} = 0$ for all the arcs $(k,j) \in \delta^-_A(j)$. Then constraints (21) become $\bar{t}_i - \bar{t}_j \leq M_{ij} + \theta_{ij}$ which is obviously fulfilled.

Since all the possibilities have been taken into account, then the thesis follows.

□

Remark 1 As will become evident from the results of Table 2, constraints (4) and (21) are not comparable. Indeed, on one hand we have $\alpha_{ji} \geq \alpha'_{ji}$, and on the other hand, constraints (21) consider all the variables corresponding to the arcs incoming in node j while in constraints (4) only a subset φ_{ji} of them is taken into account.

4.2 Opposite arcs constraints

Constraints (7) can be strengthened by lifting the variables corresponding to arcs of the set φ_{ij}. Indeed:

Proposition 3 *For each arc $(i,j) \in A$ such that $(j,i) \in A$, the inequality*

$$y_{ij} + y_{ji} + \sum_{(h,i)\in\varphi_{ij}} y_{hi} \leq 1 \tag{23}$$

is valid for $STPD_{DC}$.

Proof Let $(\bar{y},\bar{t})$ be a feasible integer solution of $\mathrm{STPD_{DC}}$ and let $(i,j) \in A$ with $(j,i) \in A$. In view of lemma 2 if $\bar{y}_{ij} = 1$ then $\bar{y}_{ji} = 0$ and $\bar{y}_{hi} = 0$ for all the arcs $(h,i) \in \varphi_{ij}$, but also if $\bar{y}_{ji} = 1$ then $\bar{y}_{ij} = 0$ and $\bar{y}_{hi} = 0$ for all the arcs $(h,i) \in \varphi_{ij}$. By applying again lemma 2, if there exists an arc $(h',i) \in \varphi_{ij}$ such that $\bar{y}_{h'i} = 1$, then $\bar{y}_{ji} = 0$ for all the arcs $(j,i) \in \delta^-_A(i) \setminus \{(h',i)\}$ and, in particular, $\bar{y}_{ji} = 0$ and $\bar{y}_{hi} = 0$ for all the arcs $(h,i) \in \varphi_{ij} \setminus \{(h',i)\}$. Moreover as seen in point *iv)* of the same lemma, if $\bar{y}_{h'i} = 1$ and $\bar{y}_{ij} = 1$, then variable $\bar{t}_j$ violates constraints (5). Consequently, $\bar{y}_{ij} = 0$ since $\bar{y}_{h'i} = 1$. In all the possible cases inequality (23) is fulfilled by $(\bar{y},\bar{t})$.

□

In the sequel, constraints (23) substitute thus constraints (7) in formulation $\mathrm{STPD_{DC}}$.

4.3 Infeasible paths

Clearly, any arc $(i,j) \in A$ such that $\lambda_i + \theta_{ij} > \mu_j$ can be eliminated from the directed graph, since it would never appear in any feasible solution. Actually, we can generalize this condition to paths connecting two non adjacent nodes i and j. Let $D(i,j) := (i = v_1, v_2, v_3, ..., v_{k-1}, v_k = j)$ be the shortest path with delays as weights from node i to j, let $\theta(i,j)$ be its cost and let $|D(i,j)|$ be the number of its arcs.

If $\lambda_i + \theta(i,j) > \mu_j$ then $D(i,j)$ is an infeasible path and then the inequality

$$\sum_{(h,l)\in D(i,j)} y_{hl} \leq |D(i,j)| - 1, \tag{24}$$

can be added to the model.

As proposed in [5] and [6] for the tournament constraints in the context of the TSP with time-windows, inequalities (24) can be strengthened. Indeed it holds that:

Proposition 4 *For all $i, j \in V$, if $\lambda_i + \theta(i,j) > \mu_j$, then*

$$\sum_{h=1}^{k-1} \sum_{l=h+1}^{k} y_{v_h v_l} \leq |D(i,j)| - 1 \tag{25}$$

is a valid inequality.

Proof See [5].

□

4.4 Indegree constraints

For each Steiner node i, we can require not only the fulfillment of the flow-balance constraints (8), but also that whenever there is an outgoing flow, then there should be also an ingoing flow. Let R_i^* be the set of the terminal nodes such that there exists a directed path $P(i,j)$ from node i to j in the graph $B = (V, A)$ such that $\lambda_i + \theta(i,j) \leq \mu_j$.

Proposition 5 *For each Steiner node $i \in S$, the inequality*

$$\min(|R_i^*|, |\delta_A^+(i)|) \sum_{(j,i)\in\delta_A^-(i)} y_{ji} \geq \sum_{(i,j)\in\delta_A^+(i)} y_{ij} \tag{26}$$

is fulfilled by any optimal solution of $STPD_{DC}$.

Proof Let $i \in S$, and let $(\bar{y}, \bar{t})$ be an optimal integer solution for formulation $\mathrm{STPD_{DC}}$. Suppose that $\bar{y}_{ji} = 0$ for all the arcs $(j,i) \in \delta_A^-(i)$; this means that i does not belong to any path connecting the root node to a terminal node. Hence for the optimality of the solution $(\bar{y}, \bar{t})$, it holds that $\bar{y}_{ij} = 0$ for all the arcs $(i,j) \in \delta_A^+(i)$ and then inequality (26) is satisfied by $(\bar{y}, \bar{t})$.
Suppose, on the contrary, that there exists an arc $(k,i) \in \delta_A^-(i)$ such that $\bar{y}_{ki} = 1$. Thus, it is unique, and for the optimality of $(\bar{y}, \bar{t})$, node i belongs to at least one path connecting the root with a terminal node. Moreover, i is the tail node of at most $\min(|R_i^*|, |\delta_A^+(i)|)$ different arcs belonging to paths from 1 to the terminal nodes. Consequently, $\sum_{(i,j)\in\delta_A^+(i)} \bar{y}_{ij} \leq \min(|R_i^*|, |\delta_A^+(i)|)$. In all the cases (26) is fulfilled and hence the thesis.

□

5 A branch-and-cut solution method

Since Formulation $\mathrm{STPD_{DC}}$ includes exponentially many cut constraints, then we have implemented a branch-and-cut (*B&C* for short) solution method for solving it. In a preprocessing step, the solution procedure invokes the preprocessing algorithm that is described in [18]. This preprocessing phase aims at producing equivalent instances of reduced size and also at strengthening the time window intervals. Consequently, the coefficients of the delays constraints are tightened and the resulting LP relaxation is tightened as well.

At each node of the B&C tree, given a tentative solution $(\bar{y}, \bar{t})$ of the relaxed program, the separation of violated inequalities (3) is achieved through solving, for each source-terminal pair, a maximum flow problem on the network G, where the capacity of each arc $(i, j) \in A$ is $\bar{y}_{ij}$. If the corresponding maximum flow value is less than 1, then a minimum capacity cut (W, W^c) is identified and the corresponding constraint (3) is appended to the relaxed master program.

Furthermore, for speeding up the resolution of the LP relaxation, we start first by solving the linear relaxation of formulation $\mathrm{STPD_{LMTZ}}$ with all the costs equal to 1, and we denote by n_a its optimum value. An obvious lower bound on the number of arcs in a feasible STPD solution is given by $\lceil n_a \rceil$. Hence we add to the initial constraint system the inequality:

$$\sum_{(i,j)\in A} y_{ij} \geq \lceil n_a \rceil \,. \tag{27}$$

Actually, we found that the LP relaxation of formulation $\mathrm{STPD_{LMTZ}}$ can be solved in less than a fraction of second for the instances that we have considered and thus we have solved this LP relaxation not only for computing n_a, but also for strengthening the time window intervals $[\lambda_i, \mu_i]$, as explained below.

For each node i that is different from the root, we want to reduce the value of μ_i and, thus, we iteratively solve the LPs of formulation $\mathrm{STPD_{LMTZ}}$ where the lower bound λ_i is substituted by the midpoint of the interval $[\lambda_i, \mu_i]$, that is $\left\lceil \frac{\lambda_i+\mu_i}{2} \right\rceil$. Whenever the new time window interval $[\left\lceil \frac{\lambda_i+\mu_i}{2} \right\rceil, \mu_i]$ produces infeasible problems, we decrease the value of μ_i by updating it: $\mu_i^* := \left\lceil \frac{\lambda_i+\mu_i}{2} \right\rceil - 1$. Moreover, updating the value μ_i for node i may have the following two consequences that are applied recursively each time the value of μ_i is reduced. Firstly, if node j precedes i (that is $(j, i) \in A$) and $\mu_i^* < \lambda_j + \theta_{ij} \leq \mu_i$, then in view of the updated time bound, arc (j, i) can be eliminated from the graph. Secondly, it may occur to tighten the time windows of a Steiner node $j \in V \setminus R$ that precedes i. Indeed if $\max(\mu_i^* - \theta_{ji},\ \max_{l\in\delta^*_A(j)\setminus\{(j,i)\}}(\mu_l - \theta_{jl})) > \mu_j$, then also μ_j can be reduced. We notice that it is useless to try to decrease in the same way the value of λ_i. If the original problem is feasible then, by definition of λ_i, there always exists a path in the graph in which the delay of each node i from the root node is equal to λ_i. The new delay bounds combine to define the final coefficients of the constraints of formulation $\mathrm{STPD_{DC}}$.

The algorithm starts with an initial constraint matrix that is constituted by the inequalities (4), (5), (8), (9), (10), (23) and (27), and by the linear relaxation of constraints (6). In addition, a subset of the cuts (3) are included into the formulation. In particular, we include the cuts obtained by choosing as cutset all the sets of the sequence $(W(1)_i)$ and for each terminal node $r \in R^*$ all the sets of the sequence $(W(r)_i^c)$. Sequence $(W(1)_i)$ starts with $W(1)_0 := \{1\}$, ends if $R^* \subset W(1)_{i+1}$, and has as a generic element the set $W(1)_i := \{k \in V :\ \exists j \in W(1)_{i-1}\ s.t.\ (j, k) \in A\}$, whereas sequence $(W(r)_i^c)$ starts with the set $W(r)_0 := \{r\}$, ends when $1 \in W(r)_{i+1}$, and has as a generic element the set $W(r)_i := \{k \in V :\ \exists j \in W(r)_{i-1}\ s.t.\ (k, j) \in A\}$.

Denote by $\mathrm{STPD'_{DC}}$ the LP relaxation of formulation $\mathrm{STPD_{DC}}$ obtained by considering only the above subset of constraints. We describe now some implementation details of the branch-and-cut procedure.

- **Branching:** The branching is performed on the y variables, using the strong branching strategy ([1], [4], and [26]). It consists in performing a limited number of simplex iterations in order to establish which one of the fractional candidates temporarily fixed to its up and down values gives the best objective value progress before actually branching on it.
  We have also considered SOS1 branching which refers to special ordered set of type I (see e.g. [13], and [26]). Specifically, for the subsets $W := \{r\}$ with $r \in R^*$, constraints (3) can be forced to be satisfied to the equality since clearly only one arc will income in each terminal node in an optimal solution. Thus, for each terminal node $r \in R^*$ the constraint $\sum_{(j,r)\in A} y_{jr} = 1$ can be used for a SOS1 branching. It is, indeed, possible to select a set $T_r \subseteq I_r := \{(j,r) \in A : j \in V\}$ and thus to define two subregions to be identified by the constraints $\sum_{(j,r)\in T_r} y_{jr} = 0$ and $\sum_{(j,r)\in I_r \setminus T_r} y_{jr} = 0$, respectively. We select the SOS1 set T_r on the basis of the delays. Specifically, suppose that during the $B\&C$ phase the current optimal solution $(\bar{y}, \bar{t})$ has fractional variables associated with incoming arcs of a terminal node r. We compute the value $md(r) := \sum_{(j,r)\in I_r} \theta_{jr}\bar{y}_{jr}$ which is the mean delay of the arcs incoming in r and then we set $T_r := \{(j,r) \in I_r : \theta_{jr} \leq md(r)\}$ (in view of this choice, variables associated with arcs having delays which are lower than the mean delay $md(r)$ belong to one subregion).
- **Enumeration strategy:** The best-bound strategy is chosen, and thus the node with the best objective function value is selected.
- **Preprocessing during the B&C phase:** A preprocessing based on the reduced costs (see e.g. [20]) can be performed at each node of the $B\&C$ tree. Let $(\bar{y}, \bar{t})$ be the current optimal solution and LLB be its value (it is a local lower bound), and let UB be a global upper bound and, i.e., the value of the best known feasible solution. Moreover, let $\bar{c}_{ij}$ be the reduced cost of the arc $(i,j) \in A$. It is well known that if $y_{ij} = 0$ and $LLB + \bar{c}_{ij} > UB$, then variable $\bar{y}_{ij}$ can be locally fixed to zero, whereas if $\bar{y}_{ij} = 1$ and $LLB - \bar{c}_{ij} > UB$, then variable $\bar{y}_{ij}$ can be locally fixed to one (by "locally" we mean that the variable can be fixed in all the current node's sons in the $B\&C$ tree). Whenever we fix either to one or to zero the value of some variables, this means that we are changing the topology of the network and we can update the values of λ and μ. If the new value λ_i is strictly greater than its previous value, then we locally modify the lower bound in (5). Moreover if i is a Steiner node and the new value μ_i is strictly lower than its previous value, then we can also locally modify the upper bound in (5). Furthermore, if the new values for a Steiner node are such that $\lambda_i > \mu_i$, then all variables corresponding to arcs incident in i can be locally fixed to zero. Finally, it can be locally fixed to zero the value of the variable y_{ij} if the relation $\lambda_i + \theta_{ij} > \mu_j$ holds.

A synthesis of the algorithm can be formalized as follows:

Step 0: Perform the preprocessing;
Step 1: Solve $\mathrm{STPD}'_{\mathrm{DC}}$, and let $(\overline{y}, \overline{t})$ be its optimal solution;
Step 2: Perform the reduced costs preprocessing; if an edge is eliminated, then go to Step 0 else go to Step 3;
Step 3: If $\overline{y}$ violates a constraint of type (3), then add the most violated constraint to $\mathrm{STPD}'_{\mathrm{DC}}$ and go to Step 1;
Step 4: Start the $B\&C$ procedure until the optimal integer solution has been found:
  a) Separate constraints (3) and (21);
  b) Separate constraints (25) and (26);

c) Perform reduced costs preprocessing and the local variable fixing and time bounds strengthening;
d) Find the terminal node with the highest number of fractional variables associated with its incoming arcs and define the SOS1 set.

Preliminary, yet extensive, computational results have shown that performing steps 4b, 4c and 4d at every node of the *B&C* tree is too expensive from a computational point of view. Nevertheless, we have found that performing these steps on nodes located at depth 10 and 35 reduces not only the number of explored nodes in the B&C tree, but also the CPU time.

6 Computational experiments

All our experiments have been carried out on an Opteron 246 computer with 2 GB RAM memory, by using CPLEX 10.2 and its callbacks within a C code for implementing the B&C procedure.

6.1 Description of the problem instances

We focus our computational study on the *complete graphs Berlin* 52 and *Brazil* 58 from SteinLib library [15]. The approach proposed in [18] solved to optimality all the considered instances *Berlin* 52 within a considerable CPU time, whereas it failed to solve *all* the instances of the class *Brazil* 58. Graphs *Berlin* 52 are composed of 52 nodes, 16 terminals and 1326 edges and graphs *Brazil* 58 are composed of 58 nodes, 25 terminals and 1653 edges. The costs on the edges of these graphs are Euclidean distances. For each of these two graphs, we have generated 20 instances having different delay values associated with the edges. Empirically, we found that challenging instances are obtained if the delays were correlated with the costs. These instances were generated in the following way. First, a random number r is drawn from the interval $[0.8, 1.2]$. Then, for each edge $\{i, j\}$ we set $\theta_{ij} := r * c_{ij}$. Once each arc has been assigned a delay, we have computed the value MP which is the maximum among the $|R^*|$ shortest paths from 1 to each terminal node with the delays as weights, i.e. $MP = \max_{j \in R^*} \theta(1, j)$. In tightly constrained problems indicated with 0.1 we have set $\Delta := 1.1 * MP$, whereas in weakly constrained problems indicated with 0.5 we have set $\Delta := 1.5 * MP$. Hence, we have generated 40 test instances in total.

6.2 $\mathrm{STPD_{DC}}$ based solution methods

Table 1 summarizes the computational results obtained by running the algorithm described in section 5. Column $Ctr(.)$ reports the number of constraints of type (.) generated during the *B&C* procedure, column *sos* reports the number of SOS branching and column N_{nod} displays the number of nodes explored before obtaining an optimal integer solution. In columns Clq and Cvr, the number of clique and of cover inequalities introduced by CPLEX are presented. T indicates the time (in seconds) needed to run Step 4 of the algorithm, while in column T_{tot} we report the total time for solving the instances (involving all the steps of the algorithm). Finally, the last two columns report the performance of the approach that is described in [18], in particular we indicate the CPU time and the remaining $Gap := \frac{UB-LB}{LB} \times 100$ after 3 hours of computation

Problem	STPD$_{DC}$										STPD$_{LMTZ}$	
	$Ctr(3)$	$Ctr(21)$	$Ctr(25)$	$Ctr(26)$	sos	N_{nod}	Clq	Cvr	T	T_{tot}	T_{tot}	Gap
Berlin 52 0.1 01	18	4	0	6	0	2	0	0	1.28	3.96	0.78	0.00
Berlin 52 0.1 02	49	3	1	6	7	85	0	0	9.70	12.38	640.54	0.00
Berlin 52 0.1 03	42	1	0	6	2	46	1	0	3.25	4.54	505.34	0.00
Berlin 52 0.1 04	46	4	1	3	2	32	4	0	2.95	4.53	21.08	0.00
Berlin 52 0.1 05	-	-	-	-	-	-	-	-	-	1.01	1.34	0.00
Berlin 52 0.1 06	11	4	0	6	0	3	0	0	3.17	4.75	70.36	0.00
Berlin 52 0.1 07	7	0	0	3	0	2	0	0	2.12	3.08	136.87	0.00
Berlin 52 0.1 08	11	1	0	8	0	5	5	0	1.88	2.73	6.94	0.00
Berlin 52 0.1 09	15	0	0	11	0	7	1	2	2.07	2.63	30.87	0.00
Berlin 52 0.1 10	16	0	0	8	0	20	2	2	5.13	5.86	514.19	0.00
Berlin 52 0.5 01	27	4	0	0	2	47	1	0	1.65	4.59	3580.30	0.00
Berlin 52 0.5 02	35	5	0	7	5	63	0	2	4.58	7.67	1663.06	0.00
Berlin 52 0.5 03	43	4	0	2	3	60	0	0	5.12	7.14	211.15	0.00
Berlin 52 0.5 04	18	0	0	3	0	12	0	0	3.07	7.31	38.63	0.00
Berlin 52 0.5 05	2	2	0	0	0	10	0	0	0.70	3.14	537.93	0.00
Berlin 52 0.5 06	5	0	0	3	0	9	0	0	3.08	6.28	2082.43	0.00
Berlin 52 0.5 07	79	1	0	9	2	85	1	0	19.14	20.19	393.20	0.00
Berlin 52 0.5 08	62	3	0	7	3	104	0	0	14.62	15.73	3515.70	0.00
Berlin 52 0.5 09	29	3	0	5	0	23	0	0	6.31	7.25	5439.77	0 0.00
Berlin 52 0.5 10	30	5	0	11	2	46	0	0	10.64	12.94	1005.53	0.00
Brazil 58 0.1 01	530	3	0	11	11	421	0	1	2419.67	2433.27	>3h	9.02
Brazil 58 0.1 02	14	4	0	14	0	2	0	0	56.12	75.54	>3h	2.10
Brazil 58 0.1 03	2	2	0	1	0	11	0	0	41.03	56.92	>3h	6.01
Brazil 58 0.1 04	-	-	-	-	-	-	-	-	-	3.79	>3h	1.62
Brazil 58 0.1 05	943	4	1	11	232	17419	0	5	>3h	>3h	>3h	7.32
Brazil 58 0.1 06	77	2	0	5	12	176	1	1	85.23	94.07	>3h	3.31
Brazil 58 0.1 07	4	0	0	3	0	20	3	2	10.70	27.52	>3h	5.85
Brazil 58 0.1 08	22	0	0	3	0	24	0	0	11.90	24.31	>3h	0.93
Brazil 58 0.1 09	3	0	0	5	0	1	0	0	3.14	17.31	>3h	2.03
Brazil 58 0.1 10	18	2	0	7	4	138	3	0	46.67	63.96	>3h	5.41
Brazil 58 0.5 01	145	1	0	12	58	840	0	4	436.78	457.11	>3h	7.44
Brazil 58 0.5 02	115	2	0	15	139	2212	0	1	782.14	802.04	>3h	7.61
Brazil 58 0.5 03	307	0	0	9	20	1952	0	1	1149.26	1173.34	>3h	9.61
Brazil 58 0.5 04	285	0	0	9	19	836	0	2	695.36	716.76	>3h	7.98
Brazil 58 0.5 05	344	7	0	12	18	3811	0	0	2422.78	2434.57	>3h	8.39
Brazil 58 0.5 06	42	7	0	11	3	54	0	0	105.18	122.91	>3h	5.13
Brazil 58 0.5 07	85	1	0	3	5	129	0	0	142.38	161.62	>3h	10.06
Brazil 58 0.5 08	89	0	0	12	6	108	0	0	128.84	148.87	>3h	5.91
Brazil 58 0.5 09	98	3	0	10	7	220	0	0	239.86	272.70	>3h	9.91
Brazil 58 0.5 10	70	0	0	15	4	74	0	0	157.72	181.55	>3h	8.89

Table 1 Computational results of Formulation STPD$_{DC}$

(where UB is the value of the best feasible integer solution and LB is the value of the best lower bound obtained upon termination).

For two instances, namely Berlin 52 0.1 05 and Brazil 58 0.1 04, the LP relaxation of STPD$_{DC}$ produced an optimal integer solution, while there is only one instance, Brazil 58 0.1 05, that is unsolved by the proposed B&C with a remaining gap of 0.07% after 3 hours of computation. It is interesting to emphasize that *none* of the generated instances of class Brazil 58 can be solved with the approach described in [18], and that the gaps upon termination (last column) are still very important in almost all the cases, whereas all the instances except one are solved with the B&C algorithm within at most 2435 seconds.

It is noteworthy that, for the sake of completeness, we have assessed the performance of the proposed B&C algorithm on additional *sparse* instances described in [18], however these results are not reported since they are comparable with those in [18].

6.3 Lifted MTZ constraints and LP relaxation comparison

In Section 3, we have compared formulations STPD$_{DC}$ and STPD$_{LMTZ}$ from a theoretical point of view and in Section 4.1 we have foretold that the two liftings (4) and (21) are not comparable, now we present in Table 2 the computational evidence of our claims. We indicate with "STPD$_{DC}$ (.)" formulation STPD$_{DC}$ where constraints (4) are replaced by constraints (.) and we denote by $v_{LP}(P)$ the optimal value of the

Problem	STPD_{DC} (4)	STPD_{DC} (21)	STPD_{DC} (4)+(21)	$gap(STPD_{LMTZ})$	OPT
Berlin 52 0.1 01	77.70	**77.71**	77.71	17.71	1141
Berlin 52 0.1 02	78.37	**78.38**	78.38	18.05	1319
Berlin 52 0.1 03	87.39	**87.40**	87.40	13.31	1390
Berlin 52 0.1 04	84.91	**84.96**	84.96	17.86	1401
Berlin 52 0.1 05	**100**	**100**	100	13.16	1203
Berlin 52 0.1 06	**73.97**	**73.97**	73.97	19.13	1152
Berlin 52 0.1 07	86.83	**86.95**	86.95	16.98	1354
Berlin 52 0.1 08	**93.47**	**93.47**	93.47	13.74	1272
Berlin 52 0.1 09	86.47	**86.49**	86.49	11.00	1307
Berlin 52 0.1 10	**89.90**	88.94	89.90	16.86	1250
Berlin 52 0.5 01	**96.65**	**96.65**	96.65	19.87	1058
Berlin 52 0.5 02	79.60	**79.62**	79.62	21.59	1097
Berlin 52 0.5 03	81.21	**81.24**	81.24	22.22	1096
Berlin 52 0.5 04	**75.44**	74.98	75.44	23.55	1136
Berlin 52 0.5 05	91.63	**91.64**	91.64	21.23	1070
Berlin 52 0.5 06	**94.46**	**94.46**	94.46	20.44	1063
Berlin 52 0.5 07	**88.92**	**88.92**	88.92	24.74	1120
Berlin 52 0.5 08	89.11	**89.11**	89.11	19.36	1072
Berlin 52 0.5 09	**84.60**	**84.60**	84.60	20.13	1121
Berlin 52 0.5 10	85.35	85.18	**85.36**	20.22	1079
Brazil 58 0.1 01	66.72	67.23	**67.41**	20.49	14530
Brazil 58 0.1 02	55.87	55.93	**56.04**	23.14	16080
Brazil 58 0.1 03	45.03	45.00	**45.03**	22.86	16474
Brazil 58 0.1 04	**100**	**100**	100	10.15	15781
Brazil 58 0.1 05	82.94	**82.94**	82.94	19.10	19679
Brazil 58 0.1 06	47.75	45.30	**47.75**	16.06	23105
Brazil 58 0.1 07	**76.17**	64.47	76.17	26.53	16477
Brazil 58 0.1 08	**99.46**	**99.46**	99.46	11.83	16170
Brazil 58 0.1 09	93.04	**93.07**	93.07	16.20	17350
Brazil 58 0.1 10	96.27	96.27	**96.27**	15.80	21183
Brazil 58 0.5 01	93.73	93.74	**93.74**	15.16	13769
Brazil 58 0.5 02	94.29	94.29	**94.29**	15.01	13758
Brazil 58 0.5 03	**82.08**	82.00	82.08	17.34	14029
Brazil 58 0.5 04	64.54	64.60	**64.67**	22.50	14826
Brazil 58 0.5 05	57.51	57.54	**57.66**	25.75	15035
Brazil 58 0.5 06	46.14	43.38	**46.49**	31.86	16018
Brazil 58 0.5 07	77.15	77.14	**77.15**	18.43	14160
Brazil 58 0.5 08	62.03	61.20	**62.07**	23.53	14860
Brazil 58 0.5 09	**76.34**	76.26	76.34	18.65	14186
Brazil 58 0.5 10	49.90	48.33	**50.85**	28.34	15638

Table 2 Comparison of the liftings of the MTZ constraints

LP relaxation of a generic formulation P. Last column of table 2 reports the optimal value OPT of each instance, and in column $gap(STPD_{LMTZ})$ we display the $gap := \frac{OPT - v_{LP}(STPD_{LMTZ})}{v_{LP}(STPD_{LMTZ})} \times 100$. The remaining columns report the percentage $\frac{v_{LP}(P) - v_{LP}(STPD_{LMTZ})}{OPT - v_{LP}(STPD_{LMTZ})} \times 100$, where formulation P is respectively STPD_{DC} (4), STPD_{DC} (21) and finally STPD_{DC} (4) + (21). These values show how closer the LP relaxation of formulations STPD_{DC} are to the optimal value with respect to the LP relaxation of $\text{STPD}_{\text{LMTZ}}$.

We have highlighted in bold which lifting (4) or (21) produces the best LP bound and when both entries are in bold, it means that formulations STPD_{DC} (4) and STPD_{DC} (21) have the same LP relaxation values. When column STPD_{DC} (4)+(21) is displayed in bold, then the combination of the two lifted constraints produced a strictly better LP relaxation value. The instance with the worst gap for all the formulations is Brazil 58 0.5 06, where the gap for STPD_{DC} (4) is 14.96, for STPD_{DC} (21) is 15.85 and using both the constraints the gap is 14.85. If we compute the mean gap over all the considered instances, we obtain the value 19.25 for formulation $\text{STPD}_{\text{LMTZ}}$, 3.81 for STPD_{DC} (4) and 3.93 for STPD_{DC} (21), while the combination of the two liftings leads to a mean gap of 3.80.

From the computational results, it appears evident that the use of constraints (4) produces on the average a better LP relaxations than using constraints (21) and this justifies our choice of considering constraints (4) in formulation STPD_{DC} and to generate constraints (21) only if necessary during the B&C solution method.

7 Conclusion

In this paper, we have proposed an enhanced directed cut-based formulation for the STPD together with an exact branch-and-cut algorithm. The proposed approach includes several enhanced algorithmic features. In particular, it incorporates a new lifting of the delay/subtour elimination constraint, an effective preprocessing procedure, and an SOS branching, while including additional valid inequalities. Computational results attest the efficacy of the proposed algorithm, which can solve to optimality hard dense STPD instances. Furthermore, we have provided empirical evidence that the proposed approach consistently outperforms a previously proposed compact formulation-based solution procedure. As a topic for future research, we recommend the derivation of facet-defining inequalities that might prove useful for accelerating the convergence of the algorithm. The literature on the exact solution of the STPD is very scant and its in depth investigation is still in its infancy. We hope that the results presented in this paper will stimulate further research in this topic.

References

1. T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42 – 54, 2005.
2. E. Althaus, T. Polzin, and S. V. Daneshmand. Improving linear programming approaches for the Steiner tree problem. In *Experimental and efficient algorithms*, volume 2647 of *Lecture Notes in Comput. Sci.*, pages 1–14. Springer, Berlin, 2003.
3. Y. P. Aneja. An integer linear programming approach to the Steiner problem in graphs. *Networks*, 10(2):167–178, 1980.
4. D. Applegate, R. Bixby, and W. Cook. Finding cuts in the tsp (a preliminary report), 1995.
5. N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Math. Program.*, 90(3, Ser. A):475–506, 2001.
6. N. Ascheuer, M. Fischetti, and M. Grötschel. A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks*, 36(2):69–79, 2000.
7. A. M. Costa, J.-F. Cordeau, and G. Laporte. Fast heuristics for the Steiner tree problem with revenues, budget and hop constraints. *European Journal of Operational Research*, 190(1):68–78, 2008.
8. A. M. Costa, J.F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks*, 53(2):141–159, 2009.
9. D. Z. Du, B. Lu, H. Ngo, and P. M. Pardalos. Steiner tree problems. In: C. Floudas and P. Pardalos (eds.), *Encyclopedia of Optimization*, 5:227–290, 2001.
10. N. Ghaboosi and A. T. Haghighat. Tabu search based algorithms for bandwidth-delay-constrained least-cost multicast routing. *Telecommunication Systems*, 34(3-4):147–166, 2007.
11. A. Ghanwani. Neural and delay based heuristics for the Steiner problem in networks. *European Journal of Operational Research*, 108(2):241–265, 1998.
12. L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with Hop constraints. *Computers & Operations Research*, 22(9):959–970, 1995.
13. E. L. Johnson, G. L. Nemhauser, and M.W.P. Savelsbergh. Progress in linear programming-based algorithms for integer programming: An exposition. *INFORMS Journal on Computing*, 12, 2000.
14. T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.
15. T. Koch, A. Martin, and S. Voβ. SteinLib. *http://elib.zib.de/steinlib.*
16. V. P. Kompella, J. Pasquale, and G. C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3):286–292, 1993.

17. Z. Kun, W. Heng, and F. Y. Liu. Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing. *Computer Communications*, 28(11):1356–1370, 2005.
18. V. Leggieri, M. Haouari, S. Layeb, and C. Triki. The Steiner tree problem with delays: A tight compact formulation and reduction procedures. Technical report.
19. 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.
20. 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.
21. 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.
22. C. A. S. Oliveira and P. M. Pardalos. Construction algorithms and approximation bounds for the streaming cache placement problem in multicast networks. *Cybernetics and System Analysis*, 41(6):898–908, 2005.
23. T. Polzin and S. V. Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, 112(1-3):241–261, 2001.
24. M. Santos, L. M.A. Drummond, and E. Uchoa. A distributed dual ascent algorithm for the hop-constrained Steiner tree problem. *Operations Research Letters*, 38(1):57 – 62, 2010.
25. R. Sriram, G. Manimaran, and C. S. Ram Murthy. Algorithms for delay-constrained low-cost multicast tree construction. *Computer Communications*, 21(18):1693–1706, 1998.
26. L.A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.