ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The Steiner Tree Problem with Delays: A compact formulation and reduction procedures

Valeria Leggieri [a,b,*], Mohamed Haouari [c,d,e], Chefi Triki [b]

[a] Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
[b] Dipartimento di Matematica, Università del Salento, 73100 Lecce, Italy
[c] Combinatorial Optimization Research Group, ROI, Ecole Polytechnique de Tunisie, BP 743, 2078 La Marsa, Tunisia
[d] Department of Industrial Engineering, Faculty of Engineering, Ozyegin University, Istanbul, Turkey
[e] Princess Fatimah Alnijris' Research Chair for AMT, College of Engineering, King Saud University, Saudi Arabia

ARTICLE INFO

Article history:
Received 13 May 2010
Received in revised form 20 June 2011
Accepted 6 July 2011
Available online xxxx

Keywords:
Steiner tree problem
MTZ subtour elimination constraints
Reduction techniques

ABSTRACT

This paper investigates the Steiner Tree Problem with Delays (STPD), a variation of the classical Steiner Tree problem that arises in multicast routing. We propose an exact solution approach that is based on a polynomial-size formulation for this challenging NP-hard problem. The LP relaxation of this formulation is enhanced through the derivation of new lifted Miller–Tucker–Zemlin subtour elimination constraints. Furthermore, we present several preprocessing techniques for both reducing the problem size and tightening the LP relaxation. Finally, we report the results of extensive computational experiments on instances with up to 1000 nodes. These results attest to the efficacy of the combination of the enhanced formulation and reduction techniques.

1. Introduction

The Steiner Tree Problem (STP) is defined on a connected undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, with cost c_e on each edge $e \in E$. The STP requires finding a minimum-cost tree of G that spans a subset of nodes $R \subset V$, whose elements are called terminal (or, required) nodes, making possibly use of additional nodes (so-called *Steiner nodes*) from $S := V \setminus R$. The STP is a very fundamental combinatorial optimization problem with a long history (see e.g., [6,13]). During the last two decades, this $\mathcal{NP}$-hard problem has been intensely investigated and several effective exact solution strategies have been proposed so far (see e.g., [1,16,24]). These research efforts are mainly motivated by the pertinence of the STP to a wide range of complex network design applications. Recently, and prompted by the dramatic growth of the telecommunication sector, several new variants of the STP have been proposed and studied. For instance, some of these new variations include Steiner problems with profits, where besides the costs associated with edges, there are also revenues associated with nodes (see e.g. [4]). In addition, alternative variations aim at enriching the classical STP model by including a Quality of Service (QoS) requirement. The importance of this latter concept stems from its relevance to several real-time applications, where it is required to deliver the same information from a source toward all the members of a multicast group within a specified delay limit [17,23]. Naturally, the QoS requirement and, specifically, the maximum-delay constraint impose a restriction on an acceptable multicast tree. To this aim, in this paper, we consider the following generalization of the STP. We are given a connected, undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the node set, with node 1 being a source node, and E is the edge set, along with a nonnegative edge weight c_e and delay θ_e associated

* Corresponding author at: Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy.
E-mail address: valeria.leggieri@unibz.it (V. Leggieri).

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.07.008

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

with each edge $e \in E$. We assume that both weights and delays are nonnegative integers (see Section 4.1). The node set is partitioned into a subset R of terminals (with $1 \in R$) and a subset S of Steiner nodes. The problem consists in finding a minimum-cost subtree T of G that spans all terminals of R (possibly involving Steiner nodes) such that the sum of the delays on each path P_j in T from the source node to each terminal node $j \in R^* := R\backslash\{1\}$ is less than or equal to a specified maximum delay Δ. We refer to this problem as the *Steiner Tree Problem with Delays* (STPD). Clearly, the Steiner tree problem is a special case of the STPD with the source node being any terminal node and $\Delta = +\infty$. Thus, the STPD is $\mathcal{NP}$-hard.

Many heuristics for solving the STPD arising in the telecommunication context have been proposed so far. Kompella et al. present in [17] greedy heuristics where they find a spanning tree of the closure graph of the constrained shortest path between the source and the terminal nodes. Sriram et al. in [28] propose two-phase algorithms for sparse and static communication groups where in Phase 1, all the possible shortest paths from the source to each terminal, respecting the maximum delay requirement, are computed and then, in Phase 2, these paths are used for constructing the multicast tree. Zhu et al. [31] propose a heuristic based on a feasible search optimization method that starts with the minimum delay tree and, then, decrease the cost of the delay-bounded tree. Standard approaches have been used to solve the Steiner tree problem with delay constraints. These approaches include, for instance, tabu search [7] and simulated annealing [18].

Furthermore, it is noteworthy to observe that when all arcs have unit delays and the set R coincides with V, then the STPD problem reduces to the so-called *hop-constrained minimum spanning tree problem* (HMSTP). This latter model is a well-studied network optimization problem and important contributions pertaining to its solution have been presented so far. In particular, in [8] Gouveia presented several formulations that are based on liftings of MTZ subtour elimination constraints. Stronger multicommodity flow formulations are presented in [9] together with some computational results on graphs having up to 40 nodes.

Moreover in [25], the authors introduced a distributed heuristic that produced better solutions with respect to previously known algorithms. Recently, Gouveia et al. presented in [11] a directed cut model on a layered graph in order to solve the HMSTP and they adapted the proposed approach to the diameter-constrained minimum spanning tree problem. In [11] a branch-and-cut algorithm has been used for solving the presented problems and this method is shown to be significantly better than previously known methods for both problems.

Finally, interesting results on the rooted distance constrained minimum spanning tree problem have been presented in [10]. This problem consists in finding a spanning tree such that the path from root node to any other node has a constrained total delay and generalizes the HMSTP, since the edges are assigned any delay value. In this paper Gouveia et al. presented several equivalent modeling approaches and results for complete graph instances with up to 40 nodes. In the test instances the mean of the delays was at most 5 and total delay was at most 25. The proposed approach might be adapted for the STPD with small delay values.

Our main contributions in this paper are the following:

(i) We develop an exact approach for the STPD.
(ii) We propose a valid compact formulation for the STPD that is based on new lifted Miller–Tucker–Zemlin subtour elimination constraints.
(iii) We describe effective reduction techniques.
(iv) We provide the results of extensive computational experiments that attest the strong evidence that the combination of the aforementioned contributions allow to solve to optimality STPD instances, having up to 1000 nodes for some sparse graphs.

The remainder of this paper is organized as follows. In Section 2, we present an enhanced polynomial-length mixed-integer programming formulation for the STPD. In Section 3, we describe several tailored reduction techniques. In Section 4, we report the results of extensive computational experiments. Finally, some concluding remarks are provided in the last section.

2. Compact MIP formulations

Instead of formulating the STPD on the undirected graph G, we consider the bi-directed graph $B = (V, A)$ obtained from G by replacing each edge $e = \{i, j\} \in E$ with two directed arcs (i, j) and (j, i) (with corresponding costs $c_{ij} = c_{ji} = c_e$ and delays $\theta_{ij} = \theta_{ji} = \theta_e$) with one exception: since all the costs and all the delays are nonnegative, the incoming arcs to the source node are not created.

In this section, we investigate compact formulations (i.e., formulations involving a polynomial number of constraints and variables) for the STPD.

2.1. A multicommodity flow-based formulation

To begin with, we provide a first formulation that is an extension of the so-called Multicommodity Flow Formulation of the STP.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Denoting by y_{ij} the binary variable that takes value 1 if arc $(i, j) \in A$ belongs to the arborescence and 0 otherwise, the formulation is:

$$(\text{STPD}_{\text{MCF}}) : \text{Minimize} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{1}$$

subject to:

$$\sum_{(1,i)\in A} x_{1i}^k = 1, \quad \forall k \in R^*, \tag{2}$$

$$\sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = 0, \quad \forall k \in R^*,\ j \in V \setminus \{1, k\}, \tag{3}$$

$$\sum_{(i,k)\in A} x_{ik}^k = 1, \quad \forall k \in R^*, \tag{4}$$

$$0 \le x_{ij}^k \le y_{ij}, \quad \forall k \in R^*,\ (i, j) \in A, \tag{5}$$

$$\sum_{(i,j)\in A} \theta_{ij} x_{ij}^k \le \Delta, \quad \forall k \in R^*, \tag{6}$$

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

where, for each terminal node $k \in R^*$ and arc $(i, j) \in A$, the variable x_{ij}^k represents the quantity of commodity k flowing through (i, j). Constraints (2)–(4) are the flow conservation constraints that guarantee that there is a flow of one unit outgoing from the source and incoming in each node of R^*. Constraints (5) enforce that, for each commodity $k \in R^*$, the flow in the arc $(i, j) \in A$ is bounded by y_{ij} and finally, (6) are the delay constraints.

Another well-known equivalent (exponential-size) multicommodity flow model is the so-called path flow formulation, in which path-flow variables are used in lieu of arc-flow variables. The resulting formulation would be similar to the one proposed by Gouveia et al. in [10] for the rooted distance-constrained minimum spanning tree problem.

2.2. A new MTZ-based compact formulation

The STP can be also formulated as a *shortest spanning arborescence problem with side-constraints* (note that other types of tree-based formulations for the STP have been investigated for instance in [15]). Toward this aim, a dummy node 0, as well as dummy arcs of the form $(0, j)$, for all $j \in S \cup \{1\}$, are added to B. Each dummy arc is assigned a zero cost and a zero delay. Let $\bar{B} = (\bar{V}, \bar{A})$ denote the resulting expanded digraph. Now, consider a spanning arborescence $\vec{T} = (\bar{V}, \bar{A}(\vec{T}))$ of $\bar{B}$ that is rooted at node 0. If an arc $(0, j)$ belongs to arborescence $\vec{T}$ and j is a Steiner node, then j cannot have any outgoing arcs and hence the outdegree of a Steiner node adjacent to 0 in $\vec{T}$ is zero. Conversely, if a Steiner node j is not adjacent to 0 in $\vec{T}$, then node j belongs to a path in $\vec{T}$ connecting the root node with a required node.

Proposition 2.1. *If $\vec{T}$ is delay-feasible in $\bar{B}$, then it corresponds to a feasible STPD solution T in B having the same cost. Conversely, given T, then it corresponds to a delay-feasible spanning arborescence $\vec{T}$ in $\bar{B}$.*

In the sequel, we shall denote by $\delta^+(i)$ the set of the arcs of $\bar{A}$ outgoing from i, i.e., $\delta^+(i) := \{(i, j) \in \bar{A} : j \in \bar{V}\}$ and by $\delta^-(i)$ the set of arcs of $\bar{A}$ that are incoming in i, i.e., $\delta^-(i) := \{(j, i) \in \bar{A} : j \in \bar{V}\}$.

Extending to the arcs of $\bar{A}$, the definition of the y variables and defining for each $j \in V$ a (continuous) time variable t_j which represents the total delay of the path connecting 0 to j, we construct the following model for the STPD:

$$(\text{STPD}_{\text{MTZ}}) : \text{Minimize} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{8}$$

subject to:

$$\sum_{(i,j)\in\delta^-(j)} y_{ij} = 1, \quad \forall j \in V, \tag{9}$$

$$y_{0j} + y_{ij} + y_{ji} \le 1, \quad \forall j \in S,\ (i, j) \in \delta^-(j), \tag{10}$$

$$t_i - t_j + \theta_{ij} \le M_{ij}(1 - y_{ij}), \quad \forall (i, j) \in A, \tag{11}$$

$$t_1 = 0, \tag{12}$$

$$t_j \in [0, \Delta], \quad \forall j \in V \setminus \{1\}, \tag{13}$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \bar{A}, \tag{14}$$

where $M_{ij} := \Delta + \theta_{ij}$ for every $(i, j) \in A$.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

The objective function (8) is to minimize the total cost, where we restricted the sum to the arcs of A since the arcs of $\bar{A} \setminus A$ have zero costs. Constraints (9) require that the indegree of each node is exactly 1. Constraints (10) enforce the solution to satisfy the condition that each Steiner node j adjacent to node 0 should be a leaf. Moreover, these latter constraints require that, for any pair of opposite arcs, at most one arc should be contained in the solution. The role of constraints (11) is twofold. First, they prevent the solution from including subtours in the same way as the well-known Miller–Tucker–Zemlin constraints (or, MTZ for short) act in the context of the much studied traveling salesman problem [20,5]. Second, they enforce (jointly with constraints (12)–(14)) the solution to be delay-feasible. Indeed, if $y_{ij} = 1$, then (11) reads $t_i + \theta_{ij} \leq t_j$, otherwise, it turns out to be the redundant constraint $t_i - t_j \leq \Delta$. Finally, constraints (14) require that the arc variables are binary valued.

The distinctive feature of formulation STPD_{MTZ} is its simplicity and compactness. If n is the cardinality of V and m those of E, then the STPD_{MTZ} formulation involves $O(m)$ binary variables, $O(n)$ continuous variables, and $O(m)$ constraints. However, it is well-documented in the literature that the linear programming (or LP for short) relaxations of similar tree-based formulations for the STP are very weak (see for example [24]) and, therefore, one could easily realize that formulation STPD_{MTZ} might only be useful for solving small-sized STPD instances. In the sequel, we present an enhanced version of formulation STPD_{MTZ} that exhibits a significantly tighter LP relaxation.

2.3. Enhancements

2.3.1. Valid inequalities

There exists at least one arc outgoing from the root and, thus the constraint:

$$\sum_{(1,j)\in\delta^+(1)} y_{1j} \geq 1 \tag{15}$$

can be added to the model. Also, we can include the trivial constraints

$$y_{ij} + y_{ji} \leq 1, \quad \forall j \in R, (i,j) \in A, \tag{16}$$

that are redundant for defining any optimal solution of the linear relaxation of the formulation as the objective value coefficients are non-negative.

Moreover, we notice that a Steiner node that is not adjacent to node 0 cannot be a leaf in any optimal solution, thus the constraints

$$\sum_{(i,j)\in\delta^+(i)} y_{ij} \geq 1 - y_{0i}, \quad \forall i \in S \tag{17}$$

are valid and can, therefore, be appended to the model. We point out that the addition of (17) improves the LP relaxation of the proposed formulation.

2.3.2. Bounds on the time variables

Actually, we can compute for each t_j ($j \in V \setminus \{1\}$) a lower bound λ_j, as well as an upper bound μ_j, and thus, we can strengthen the bounding constraints (13). Indeed, the delays on the arcs define, for each node $i \in V$, a time window within which the communication should be received and forwarded to descendant nodes, while satisfying the maximum delay constraints. To this aim, given a pair of nodes i and j, we denote by $\theta(i,j)$ the value of the minimum delay path in B from i to j. Since the delays are positive, then the computation of $\theta(i,j)$ can be performed in polynomial-time. Clearly, the total elapsed time for a message sent from the source node to a node $j \in V \setminus \{1\}$ is larger than or equal to $\theta(1,j)$. Thus, we set $\lambda_j := \theta(1,j)$ for each $j \in V \setminus \{1\}$. Obviously, if for some terminal node $i \in R$ we have $\lambda_i > \Delta$, then the instance is infeasible. Moreover, a Steiner node $i \in S$ 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 $i \in S$. If $i \in R^*$, then obviously $\mu_i := \Delta$.

If $\lambda_i \leq \mu_i$ holds for each node $i \in V \setminus \{1\}$, then constraints (13) can be replaced by

$$\lambda_i \leq t_i \leq \mu_i, \quad \forall i \in V \setminus \{1\}. \tag{13'}$$

Otherwise, the problem is either infeasible, or a Steiner node can be eliminated (see Section 3.2). In the sequel, we suppose that $\lambda_i \leq \mu_i$ for all $i \in V \setminus \{1\}$.

It is noteworthy that we can strengthen constraints (11) by setting

$$M_{ij} := \mu_i - \lambda_j + \theta_{ij}. \tag{18}$$

Actually, also constraints (13′) might be strengthened by observing that, if $y_{ij} = 1$ for some $j \in V$, then $t_j \geq \max(\lambda_j, \lambda_i + \theta_{ij})$. Consequently, constraints

$$t_j \geq \sum_{i:(i,j)\in\delta^-(j)} \max(\lambda_j, \lambda_i + \theta_{ij}) y_{ij}, \quad \forall j \in V \setminus \{1\}, \tag{19}$$

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

are valid. It is noteworthy that a similar inequality has been previously proposed by Desrochers and Laporte in [5] in the context of the traveling salesman problem with time windows. Moreover, we observe that if $y_{jk} = 1$, then we have that $t_j \leq \mu_j - \max(0, \mu_j - \mu_k + \theta_{jk})$. Thus, the constraints

$$t_j \leq \mu_j - \max(0, \mu_j - \mu_k + \theta_{jk}) y_{jk}, \quad \forall (j,k) \in \delta^+(j), j \in V \setminus \{1\} \tag{20}$$

are valid.

2.3.3. Lifting the MTZ constraints

The proposed lifting might be viewed as a non-trivial generalization of previous liftings of MTZ constraints (see e.g., [8] or [5]). To this aim, we define for each arc $(i,j) \in A$ the subset

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

which is formed by the arcs (k,j) of A that are incompatible with arc (j,i), namely if both (k,j) and (j,i) were included in a solution, then the delay bounds on node i would be violated.

For every arc $(i,j) \in A$, we define $\alpha_{ji} := \mu_i - \lambda_j - \theta_{ji}$. 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, these latter coefficients are nonnegative and not always identical to zero.

Proposition 2.2. *The constraints*

$$t_i - t_j + M_{ij} y_{ij} + \alpha_{ji} y_{ji} + \sum_{(k,j) \in \varphi_{ji}} \beta_{kj} y_{kj} + \sum_{(h,i) \in \varphi_{ij}} \gamma_{hi} y_{hi} \leq M_{ij} - \theta_{ij}, \quad \forall (i,j) \in A \tag{21}$$

are valid inequalities.

In the sequel, we denote by $\text{STPD}_{\text{LMTZ}}$ the model that is obtained from STPD_{MTZ} by appending constraints (15), (17), (19) and (20) and by substituting constraints (11) with (21) and constraints (13) with (13′). Notice that $\text{STPD}_{\text{LMTZ}}$ has $O(m)$ variables and $O(m)$ constraints.

Let $v_{\text{LP}}(P)$ denote the optimal value of the LP relaxation of a mixed-integer program P.

Proposition 2.3. $v_{\text{LP}}(\text{STPD}_{\text{LMTZ}}) > v_{\text{LP}}(\text{STPD}_{\text{MTZ}})$.

3. Reduction techniques

Preprocessing plays a very useful role in solving combinatorial optimization problems. This technique, indeed, reduces the size of the problems by means of logical implications, producing equivalent instances. The preprocessing performed in our problem is based on an adaptation of the known preprocessing techniques for the Steiner Tree problem, on the fulfillment of the time window requests, and on the reduced costs based reductions.

3.1. Graph-based reductions

The graph-based reductions are performed on the undirected graph $G = (V, E)$ and the goal of this process is to reduce the size of the problem contracting or deleting nodes and/or edges in order to obtain an equivalent, but reduced, graph $G' = (V', E')$ [16,2,3,30]. Quite simple reduction tests for the STP are the degree tests applied recursively to each reduced graph, until no more reduction can be performed. For each $i \in V$, we denote by $\delta(i)$ the set of the edges of E that are incident to i. Because of the presence of the delay on the arcs, if we want to contract certain edges, then we need to store the delays. For this reason, we introduce a variable $m_i \in \mathbb{R}$ for each $i \in V$ and initially we set m_i to zero for all $i \in V$. Storing m_i enables to reduce the value of the maximum delay μ_i for node i, if some edges that are incident to i are contracted. At the end of the preprocessing procedure, m_i takes the value of the maximum of the delays of the edges that have been contracted in i.

Degree one test. For every node $i \in V$

(i) If $i \in S$ and $|\delta(i)| = 1$, then i and its incident edges are eliminated.
(ii) If $i \in R$, and $\delta(i) = \{\{i,j\}\}$, then $\{i,j\}$ is contracted and the cost c_{ij} added to the optimal solution value. Furthermore if $\theta_{ij} > m_j$, the values of μ_j and m_j are updated: $\mu_j := \mu_i + m_j - \theta_{ij}$ and $m_j := \theta_{ij}$, respectively.

Contracting an edge $\{i,j\}$ incident to a node $i \in R$ means: if $j \in R$, eliminate the edge $\{i,j\}$ and reduce the cardinality of R, or if $j \in S$, eliminate the edge $\{i,j\}$.

The degree two test is analogous to the test of the Steiner Tree problem, but a further condition on the delays must be considered in order to take heed of the maximum delay restriction at the terminal nodes.

Degree two test. Let $i \in S$ be a Steiner node with $\delta(i) = \{\{i,k\}, \{j,i\}\}$:

(i) If $\{k,j\} \notin E$, then the edges $\{k,i\}$ and $\{i,j\}$ are substituted by a new edge $\{k,j\}$ with cost $c_{kj} = c_{ki} + c_{ij}$ and delay $\theta_{kj} = \theta_{ki} + \theta_{ij}$ and i can be eliminated together with its incident edges.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

(ii) If $\{k, j\} \in E$, $c_{ki} + c_{ij} > c_{kj}$ and $\theta_{ki} + \theta_{ij} > \theta_{kj}$, then i can be eliminated from the graph together with the incident edges in i.
(iii) If $\{k, j\} \in E$, $c_{ki} + c_{ij} \leq c_{kj}$ and $\theta_{ki} + \theta_{ij} \leq \theta_{kj}$, then node i is removed from the graph together with its incident edges and the edge $\{k, j\}$ is given the cost $c_{kj} = c_{ki} + c_{ij}$ and the delay $\theta_{kj} = \theta_{ki} + \theta_{ij}$.

3.2. Delay-based reductions

A necessary condition for a node $i \in V$ to be included in a feasible solution is that the total delay in i belongs to the interval $[\lambda_i, \mu_i]$. This restriction can be used to perform a first delay based preprocessing. Indeed, if a time window $[\lambda_i, \mu_i]$ is empty, then the time required to reach node i from node 1 is greater than the residual time to reach the nearest (in terms of delays) terminal node.
Non-empty time windows. For every node $i \in V$

(i) If $\lambda_i > \mu_i$ and $i \in S$, then i and its incident edges are eliminated.
(ii) If $\lambda_i > \mu_i$ and $i \in R^*$, then the instance is infeasible.

In addition, the delays can be used to eliminate edges and, after having considered the digraph, can be used to eliminate arcs.
Adjacent time request. For every edge $\{i, j\} \in E$, if $\lambda_i + \theta_{ij} > \mu_j$ and $\lambda_j + \theta_{ji} > \mu_i$, then edge $\{i, j\}$ can be eliminated from the graph.
Direct arcs test. All the directed arcs $(i, j) \in A$ such that $\lambda_i + \theta_{ij} > \mu_j$ can be eliminated from the directed graph.

3.3. Reduced cost-based reductions

We denote by v_{LP} the optimal value of the LP relaxation of the problem and by v_{UB} an upper bound on the optimal solution value. Let (y, t) be an optimal solution of the LP relaxation of the problem and $\bar{c}_{ij}$ be the reduced cost of each arc $(i, j) \in \bar{A}$. Using the results in [22], we observe that if $y_{ij} = 0$ and $v_{\mathrm{LP}} + \bar{c}_{ij} > v_{\mathrm{UB}}$, then variable y_{ij} cannot belong to an optimal solution and hence, arc (i, j) can be eliminated from the graph. Moreover, if $y_{ij} = 1$ and $v_{\mathrm{LP}} - \bar{c}_{ij} > v_{\mathrm{UB}}$, then the arc (i, j) is always in an optimal solution.

We can use these considerations in order to achieve a further reduction of the problem size. For a "*preliminary reduced cost-based reduction*", we can consider the following cases:

(i) If $y_{0i} = 1$ for $i \in S$ and $v_{\mathrm{LP}} - \bar{c}_{0i} > v_{\mathrm{UB}}$, then node i can be discarded from the graph since it would never be included in an optimal solution.
(ii) If $y_{ij} = y_{ji} = 0$, with $v_{\mathrm{LP}} + \bar{c}_{ij} > v_{\mathrm{UB}}$ and $v_{\mathrm{LP}} + \bar{c}_{ji} > v_{\mathrm{UB}}$, then both arcs (i, j) and (j, i) can be eliminated.
(iii) If $y_{ij} = 1$ and $v_{\mathrm{LP}} - \bar{c}_{ij} > v_{\mathrm{UB}}$, then we should distinguish three cases:
Case 1: $i, j \in R$. Then, add to the digraph all the arcs (i, h) with $h \in V$ such that $(j, h) \in A$, with costs $c_{ih} = c_{jh}$ and delays $\theta_{ih} = \theta_{ij} + \theta_{jh}$, eliminate j and all its incident arcs, add the cost c_{ij} to the optimal solution value and decrease of one unit the cardinality of R. If $\theta_{ij} > m_i$, the values of μ_i and m_i are updated: $\mu_i := \mu_j + m_i - \theta_{ij}$ and $m_i := \theta_{ij}$, respectively.
Case 2: $i \in V$ and $j \in S$. Then, add to the digraph all the arcs (i, h) with $h \in V$ such that $(j, h) \in A$, with costs $c_{ih} = c_{jh}$ and delays $\theta_{ih} = \theta_{ij} + \theta_{jh}$, eliminate j and all its incident arcs and add the cost c_{ij} to the optimal solution value. If $\theta_{ij} > m_i$, the values of μ_i and m_i are updated: $\mu_i := \mu_j + m_i - \theta_{ij}$ and $m_i := \theta_{ij}$, respectively.
Case 3: $i \in S$ and $j \in R$. Then, add to the digraph all the arcs (i, h) with $h \in V$ such that $(j, h) \in A$, with costs $c_{ih} = c_{jh}$ and delays $\theta_{ih} = \theta_{ij} + \theta_{jh}$, substitute node j in R by node i, eliminate j and all its incident arcs and add the cost c_{ij} to the optimal solution value. If $\theta_{ij} > m_i$, the values of μ_i and m_i are updated: $\mu_i := \mu_j + m_i - \theta_{ij}$ and $m_i := \theta_{ij}$, respectively.
For a "*final reduced cost-based reduction*", we consider the further cases:
(iv) If $y_{ij} = 0$, $v_{\mathrm{LP}} + \bar{c}_{ij} > v_{\mathrm{UB}}$ and $0 < y_{ji} < 1$, then eliminate the arc (i, j).
(v) If $y_{ij} = 0$, $v_{\mathrm{LP}} + \bar{c}_{ij} > v_{\mathrm{UB}}$ and $y_{ji} = 0$, but $v_{\mathrm{LP}} + \bar{c}_{ji} \leq v_{\mathrm{UB}}$, then eliminate the arc (i, j).

3.4. Preprocessing framework

We summarize here the steps of the preprocessing procedure that is repeated in a recursive way until no more reduction can be achieved. The first four steps are performed on the undirected graph G:

Step 1: Perform degree one test.
Step 2: Perform non-empty time windows test.
Step 3: Perform adjacent time request test.
Step 4: Perform degree two test.
Step 5: If at least one contraction or elimination has been executed go to Step 1, otherwise go to Step 6 (the subsequent tests are performed on the digraph B obtained from the reduced undirected graph G).

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Step 6: Perform the direct arc elimination.
Step 7: Solve the LP relaxation of the formulation on B, obtaining v_{LP}.
Step 8: Invoke a heuristic and compute an upper bound value v_{UB}.
Step 9: Perform the preliminary reduced costs based reduction.
Step 10: If at least one contraction or elimination has been executed, consider the undirected graph constructed using the current digraph B and go to Step 1, else perform the final reduced cost based reduction, and Stop.

At the end of the preprocessing procedure, the MIP solver is invoked in order to solve the problem on the reduced graph with formulation $\mathrm{STPD}_{\mathrm{LMTZ}}$. In the sequel, we refer to this solution strategy by $\mathrm{STPD}_{\mathrm{LMTZ}} + P$.

3.5. A heuristic approach for the STPD

In this section we briefly describe a heuristic algorithm H_1 that we have implemented in order to obtain feasible near-optimal solutions for the STPD. Basically, the heuristic approach is an adaptation of the so-called *Shortest Path Heuristic* that has been proposed for the standard STP by Takahashi and Matsuyama [29] and by Costa et al. in [4] for the STP problem with revenue budgets and hop constraints.

Given a tree T and a terminal node j not belonging to T, we denote by $\mathcal{P}(T, j)$ the shortest delay-constrained path between T and j (i.e., the shortest path between a node k in T and j and such that the sum of the delays from source 1 to j is less than or equal to the threshold value μ_j) and by $c(\mathcal{P}(T, j))$ its length.

The idea for constructing a feasible solution of the STPD is to start with an initial delay-feasible tree T that is constituted by the path $\mathcal{P}(\{1\}, j)$ satisfying $c(\mathcal{P}(\{1\}, j)) = \max_{k\in R} c(\mathcal{P}(\{1\}, k))$ (thus, j is the most distant terminal node from the source node), and then, iteratively insert in T a node $j \in R \setminus V(T)$ such that

$$c(\mathcal{P}(T, j)) = \min_{k\in R\setminus V(T)} c(\mathcal{P}(T, k)).$$

The algorithm stops when T spans all terminal nodes. Clearly, since the initial tree is feasible and, at each iteration a feasible path is appended, the delivered tree is necessarily feasible (unless the instance is infeasible).

It is noteworthy that the computation of $\mathcal{P}(T, k)$ requires solving a *single* shortest path problem with delay constraints (SPPD) that is known to be $\mathcal{NP}$-hard. However, it can be solved by dynamic programming in pseudopolynomial time [14]. The approach that we have implemented is based on Lagrangian relaxation [12] and on an iterative dichotomous search [21]. For the sake of clarity, we briefly describe this approach. Let x be the incidence vector of a $1 - t$ path in G and let X be the set of $1 - t$ paths. The SPPD can be formulated as follows

$$\text{SPPD: Minimize } \{cx : x \in X \text{ and } \delta x \le \Delta\}. \tag{22}$$

Dualizing the delay constraint in a Lagrangian fashion yields the Lagrangian dual

$$\text{LD} : \text{Maximize}_{u\ge 0}\{\min_{x\in X}(c + u\delta)x\}. \tag{23}$$

Minoux [21] shows that LD can be solved very simply using an iterative dichotomous search. At each iteration, a (standard) shortest path problem with modified costs $c + u\delta$ is solved. In our experiments, we found that this search procedure converges very quickly and very often requires less than four iterations. Interestingly, after solving LD we get not only an optimal Lagrangian variable u^* and a lower bound L_1 on the optimal path but *also* an upper bound UB_1 corresponding to a feasible delay-constrained path. In case where $L_1 < \mathrm{UB}_1$, then Handler and Zang in [12] show that it suffices to use a ranking procedure for successively generating the kth shortest path with respect to the modified cost $c + u^*\delta$. The algorithm stops when a proven optimal path is generated. In our implementation, the kth shortest path problem is generated using the general ranking procedure which is described in [19].

Since the exact solution of SPPD might require for some large instances an excessive computing time, we have designed an additional variant of our heuristic which we refer to as H2. In H2 instead of computing an exact shortest path, we content ourselves with an approximate solution that is delivered when a maximal number of q shortest paths have been computed or a feasible solution with a preset gap ϵ_1 has been obtained. In order to get high-quality solutions, the q shortest paths have been computed $n_$iter times using a perturbation strategy. This strategy requires setting at each iteration a perturbed cost $\hat{c}_e = \mathrm{rand} * c_e$ for each edge e, where rand is randomly drawn from the uniform distribution $U[1 - r, 1 + r]$. The range parameter r is set empirically. The iterative process is prematurely halted if a solution having a gap less than a specified gap ϵ_2 has been found. After having performed extensive preliminary experiments, we found that "good" performance is achieved by setting $q = 5$, $\epsilon_1 = 0.05$, $r = 0.5$, $n\_\mathrm{iter} = 50$ and $\epsilon_2 = 0.01$.

4. Computational experiments

In order to assess the performance of the proposed reduction techniques and the effectiveness of the derived compact formulation, we have carried out extensive computational experiments on a large set of randomly generated instances. All our experiments were carried out on an Opteron 246, 2 GHz computer with 2 GB RAM memory. We have used CPLEX 10.2 for solving the LP/MIP programs. We have set a maximum CPU time limit of one hour per instance.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Table 1
Comparison of the LP relaxations bounds.

Problem	$\text{STPD}_{\text{LMTZ}} + P$		$\text{STPD}_{\text{MTZ}} + P$		$\text{STPD}_{\text{LMTZ}}$		STPD_{MTZ}	
	Gap	*Max_Gap*	*Gap*	*Max_Gap*	*Gap*	*Max_Gap*	*Gap*	*Max_Gap*
B Ran 0.1	**4.91**	13.01	14.26	28.63	20.64	58.76	32.82	104.86
B Ran 0.5	**4.82**	14.36	13.70	33.78	5.75	20.81	16.20	64.16
B Cor 0.1	**2.90**	9.35	11.88	30.94	5.72	30.91	16.02	51.65
B Cor 0.5	**2.09**	11.43	11.64	27.37	2.10	11.52	12.40	46.24
C Ran 0.1	**8.27**	28.13	19.22	41.73	25.01	55.23	31.54	55.88
C Ran 0.5	**7.87**	16.32	15.98	33.74	12.21	35.29	18.44	35.46
C Cor 0.1	**5.54**	10.51	14.17	22.73	10.43	33.33	17.09	33.50
C Cor 0.5	**1.12**	3.51	9.81	21.48	3.84	15.69	10.07	21.97
D Ran 0.1	**4.90**	6.53	13.85	17.71	10.99	17.85	16.88	21.50
D Ran 0.5	**4.10**	9.28	10.08	11.83	4.92	9.28	10.68	14.83
D Cor 0.1	**0.37**	0.99	7.65	13.24	8.51	19.59	12.28	21.47
D Cor 0.5	**1.10**	4.95	8.62	12.49	2.68	9.28	9.08	15.60

4.1. Description of the problem instances

As benchmark instances, we have considered the problems proposed in the *SteinLib library* (http://elib.zib.de/steinlib) for the standard STP. In particular the 18 instances of Class B, the first 10 instances of Class C, and the first 5 instances of Class D have been used as test problems. The graphs of the classes B, C and D are randomly generated sparse graphs with edge weights between 1 and 10. The number of vertices and of edges ranges from 50 to 1000 and from 63 to 1250, respectively.

We have randomly generated the delays on the edges. In the instances denoted by *Ran* delays assume a uniform random value drawn from the interval [1, 100] and thus they are non-correlated with costs. In instances denoted by *Cor*, delays are correlated with the costs, indeed a random number r is drawn from the interval [0.8, 1.2] and, for each edge $\{i, j\}$, θ_{ij} is set to $r * c_{ij}$. In this latter case, the delay values range from 1 to 15 for instances B, C and D. On the basis of the generated delays, we have computed the value MP which is the maximum among the shortest paths with the delays as costs between the source node and each terminal node, i.e., $\text{MP} = \max_{j \in R^*} \theta(1, j)$. In the problems indicated with 0.1, we have set Δ to the value $\Delta := 1.1 * \text{MP}$ and, in the problems indicated with 0.5, we have set Δ to $\Delta := 1.5 * \text{MP}$. With these choices, we have generated tightly as well as loosely delay-constrained instances. Moreover, none of the generated instances is infeasible. We refer, for example, as "B Ran 0.1" to the set of instances of the class B with delays that are not correlated with the costs and with $\Delta = 1.1 * \text{MP}$. In so doing, we generated a testbed including 132 STPD instances (all the test instances are available at http://poincare.unile.it/chefi/STPD/).

4.2. Comparison of the LP relaxations bounds

First, we present the results of a comparison of the following optimal values:

- $v_{\text{LP}}(\text{STPD}_{\text{LMTZ}} + P)$: this value is obtained after performing a preprocessing and solving the LP relaxation of formulation $\text{STPD}_{\text{LMTZ}}$;
- $v_{\text{LP}}(\text{STPD}_{\text{MTZ}} + P)$: this value is obtained after performing a preprocessing and solving the LP relaxation of formulation STPD_{MTZ};
- $v_{\text{LP}}(\text{STPD}_{\text{LMTZ}})$: this value is obtained after solving the LP relaxation of formulation $\text{STPD}_{\text{LMTZ}}$;
- $v_{\text{LP}}(\text{STPD}_{\text{MTZ}})$: this value is obtained after solving the LP relaxation of formulation STPD_{MTZ}.

For each value $v_{\text{LP}}(.)$, we compute the percentage gap with respect to the optimal integer solution (i.e., $100 \times \frac{v^* - v_{\text{LP}}(.)}{v_{\text{LP}}(.)}$) (if v^* is unknown then we replace it with the value of the heuristic solution). In Table 1, we report for each class of instances the values of the average gap (*Gap*) and maximum gap (*Max_Gap*).

Looking at Table 1 we see that:

- The combination of the preprocessing procedure and the enhanced formulation $\text{STPD}_{\text{LMTZ}}$ consistently yields a reduced average gap for all problem classes.
- The preprocessing positively impinges on the tightness of both formulations.
- Formulation STPD_{MTZ} has a weak relaxation, since for some instances we see that the gap is very large and reaches nearly 105%.
- The average percentage gap is larger for tightly delay-constrained instances and for non-correlated delays.

In order to get a more accurate picture of the effectiveness of the proposed lifted constraints (13), we compare the performance of $\text{STPD}_{\text{LMTZ}} + P$ with the performance of $\text{STPD}_{\text{LMTZ}'} + P$ that is obtained by substituting (14) with

$$t_i - t_j + M_{ij} y_{ij} + \alpha_{ji} y_{ji} \leq M_{ij} - \theta_{ij}, \quad \forall (i, j) \in A. \tag{14'}$$

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Table 2
Comparison of the liftings.

Problem	$STPD_{LMTZ} + P$		$STPD_{LMTZ'} + P$		*OBJ*
	Gap	*Max_Gap*	*Gap*	*Max_Gap*	
B Ran 0.1	**4.91**	13.01	8.83	20.69	18
B Ran 0.2	**7.08**	17.86	7.87	19.18	17
B Ran 0.3	**6.07**	13.56	6.99	22.55	16
B Ran 0.4	**6.11**	14.82	6.55	15.89	14
B Ran 0.5	**4.82**	14.36	5.77	14.85	13
B Cor 0.1	**2.90**	9.35	4.04	17.14	15
B Cor 0.2	**3.21**	19.39	3.34	20.95	12
B Cor 0.3	**2.80**	14.45	7.08	17.14	8
B Cor 0.4	**2.20**	12.47	2.20	12.47	7
B Cor 0.5	**2.09**	11.43	2.61	12.85	7

Table 3
Mean CPU times (in s).

Problem	H2	$STPD_{LMTZ} + P$		
	T2	T1	TT	US
B Ran 0.1	1.35	0.16	1.51	0
B Ran 0.5	1.19	0.62	1.81	0
B Cor 0.1	1.11	0.19	1.30	0
B Cor 0.5	1.04	0.26	1.29	0
C Ran 0.1	217.23	174.26	332.57	1
C Ran 0.5	193.29	76.35	181.76	3
C Cor 0.1	184.89	166.37	284.74	2
C Cor 0.5	170.66	92.24	262.89	0
D Ran 0.1	1272.24	12.06	424.54	2
D Ran 0.5	1125.44	21.84	376.95	2
D Cor 0.1	1162.99	288.71	1451.70	0
D Cor 0.5	1039.67	96.03	1135.70	0

These latter constraints correspond to the lifted MTZ constraints that have been proposed by Desrochers and Laporte [5]. Table 2 displays a summary of the results on class B (for the sake of brevity, similar results that were obtained on classes C and D are omitted).

We see from this table that constraints (14) produce tighter LP bounds than using constraints (14′). This attests the efficacy of the lifting (14) and, in particular the influence of the last two terms of the left-hand-side of (14) in improving the gaps.

However, if no reduction procedure is performed on the instances, then constraints (14) have the same impact as constraints (14′) on the LP relaxations.

Moreover, column *OBJ* reports the number of instances whose optimal value is different from the optimal value of the pure STP. Therefore, instances indicated with 0.1 (when $\Delta = 1.1 * \mathrm{MP}$) can be considered highly delay constrained while instances indicated with 0.5 can be considered weakly delay constrained.

4.3. *Performance of the* $STPD_{LMTZ}$ *formulation and the reduction procedure*

In this section, we present the results that we have obtained after combining the proposed reduction procedure with formulation $STPD_{LMTZ}$ for solving the set of 132 instances using a general-purpose MIP solver. The results are reported in Table 3. For each problem class, we report:

- T1: mean CPU time for solving the MIP formulations (computed over *all solved* instances);
- T2: mean CPU time of the heuristic H2 (computed *over all* instances);
- TT: mean total CPU time (computed over *all solved* instances and including the time for the preprocessing procedures);
- US: number of unsolved instances.

We see from Table 3, that the proposed approach LMTZ+P is very effective since 92% of the instances (122 out of 132) are solved to optimality within the specified time limit. The non-correlated problems require in general longer CPU times since the quality of the corresponding LP relaxation bounds is, generally, worse than those of the correlated instances. Actually, this remarkable performance has been made possible by the *combination* of the preprocessing and the lifted formulation. Indeed, we have investigated the effectiveness of the following alternative solution strategies:

- MTZ + P: we perform the preprocessing, and then, we solve formulation MTZ,
- LMTZ: we solve formulation LMTZ without performing any preprocessing.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Table 4
Comparison of the CPU times required by the alternative solution strategies.

Problem	$STPD_{LMTZ} + P$ TT	$STPD_{MTZ} + P$ TT	$STPD_{LMTZ}$ TT
C 0.1 Ran 01	5.70	**5.67**	80.68
C 0.1 Ran 02	**12.41**	13.44	918.70
C 0.1 Ran 03	**310.37**	US	US
C 0.1 Ran 04	**241.64**	US	US
C 0.1 Ran 05	**443.44**	US	US
C 0.1 Ran 06	11.22	**11.19**	US
C 0.1 Ran 07	**28.01**	28.72	US
C 0.1 Ran 08	**308.63**	US	US
C 0.1 Ran 09	**1631.72**	US	US
C 0.1 Ran 10	US	US	US

Table 5
Effectiveness of the reduction procedures.

Problem	DP			Preprocessing		
	%n	*%r*	*%a*	*%n*	*%r*	*%a*
B Ran 0.1	45.85	9.91	49.97	51.85	16.19	56.47
B Ran 0.5	36.67	7.82	30.24	45.41	13.34	43.08
B Cor 0.1	46.37	9.48	47.67	54.96	20.34	57.68
B Cor 0.5	37.38	7.82	31.75	52.30	16.59	51.94
C Ran 0.1	61.94	7.60	59.51	62.92	11.61	60.15
C Ran 0.5	51.32	6.16	45.06	57.58	6.16	51.98
C Cor 0.1	60.08	6.33	56.00	60.48	6.33	56.54
C Cor 0.5	51.94	6.16	45.49	60.88	7.16	54.90

We have used these two solution strategies for solving the ten instances of the subclass C 0.1 Ran. We have chosen this class because of the significance of the resulting data reported in Table 4. We see that LMTZ and MTZ + P exhibit a very poor performance since they failed to solve 8 and 6 instances of the class C 0.1 Ran, respectively. By contrast, LMTZ + P failed to solve just one instance within the same subclass.

4.4. Effectiveness of the reduction procedures

In Table 5, we present the mean percentage of reduction of the number of nodes, terminal nodes and arcs performed using first only the degree-delay based preprocessing (DP) and, then, using the entire preprocessing procedure presented in Section 5. If n is the original number of nodes and n' is the number of nodes in the reduced problem, in column *%n* we report the mean of the values $100(n-n')/n$ over all the instances belonging to the same class of problems. Similarly, we indicate by *%r* and *%a* the percentages of reduction of terminal nodes and of arcs respectively. We see that the implemented reduction procedures perform extremely well on these sparse graphs and that, for all the classes, the problem sizes are significantly reduced.

4.5. Performance on complete graphs

For the sake of completeness, we have solved additional STPD instances that are defined on Euclidean complete graphs. These Euclidean instances are derived from the benchmark STP instance Berlin52 and Brazil58 of Steinlib. These latter instances have 52 nodes and 16 terminals, and 58 nodes and 25 terminals, respectively. We have generated 5 different instances for each problem type, and we have created the delays as described in Section 4.1.

For Berlin52 and Brazil58 with delays correlated with the costs, the delays range from 3 to 560 and from 100 to 7800, respectively, while Δ ranges from 158 to 242, and from 3500 to 5751, respectively. The results are displayed in Table 6.

Looking at this table, we see that the percentage of arc reduction that is achieved on these dense graphs is very important. We observe that a small percentage of nodes reduction implies a deterioration of the gaps. All the instances Berlin52 have been optimally solved, even though longer CPU times are required when the delays are correlated with the costs. This behavior is illustrated by instance Brazil58. If the delays are non-correlated with the costs all the generated instances of Brazil58 have been solved to the optimality within at most 5 s, while if the delays are correlated with the costs none of the generated instances has be solved within one hour of CPU time.

4.6. Comparison of $STPD_{LMTZ}$ with $STPD_{MCF}$

Finally, we compare our formulation with the Multicommodity Flow Formulation $STPD_{MCF}$ which is known to exhibit a tight LP relaxation for the STP.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Table 6
Mean reduction percentage, Gap and CPU times (in s).

Problem	$STPD_{LMTZ} + P$					
	%n	*%r*	*%a*	*Gap*	T1	US
Berlin52 Ran 0.1	26.15	0.00	94.34	5.06	0.05	0
Berlin52 Ran 0.5	11.54	0.00	89.04	7.87	0.32	0
Berlin52 Cor 0.1	8.08	0.00	67.35	16.02	83.43	0
Berlin52 Cor 0.5	2.69	0.00	47.69	21.69	814.78	0
Brazil58 Ran 0.1	12.07	0.80	92.63	11.06	0.49	0
Brazil58 Ran 0.5	2.41	0.00	85.81	20.38	4.72	0
Brazil58 Cor 0.1	1.03	2.40	36.28	–	–	5
Brazil58 Cor 0.5	0.00	0.00	19.27	–	–	5

Table 7
Comparison of $STPD_{LMTZ} + P$ and $STPD_{MCF} + P$.

Problem	$STPD_{LMTZ} + P$		$STPD_{MCF} + P$	
	Gap	*T*	*Gap*	*T*
C 0.1 Cor 01	7.69	0.04	**1.30**	**0.02**
C 0.1 Cor 02	2.69	0.13	**1.19**	**0.12**
C 0.1 Cor 03	3.97	**11.26**	**1.99**	>3600
C 0.1 Cor 04	1.79	**24.98**	**1.44**	>3600
C 0.1 Cor 05	1.03	**45.59**	**0.68**	>3600
C 0.1 Cor 06	10.44	**0.02**	**0.90**	0.03
C 0.1 Cor 07	4.05	**0.26**	**4.38**	1.07
C 0.1 Cor 08	3.10	**1248.71**	*	>3600
C 0.1 Cor 09	–	>3600	*	>3600
C 0.1 Cor 10	–	>3600	*	>3600
C 0.5 Cor 01	0.79	0.20	**0.00**	**0.20**
C 0.5 Cor 02	2.27	0.68	**0.00**	**0.63**
C 0.5 Cor 03	1.74	**27.88**	**0.11**	336.77
C 0.5 Cor 04	0.46	**31.66**	**0.31**	>3600
C 0.5 Cor 05	0.57	**23.31**	**0.13**	>3600
C 0.5 Cor 06	3.51	**0.21**	**0.00**	0.22
C 0.5 Cor 07	0.00	**1.19**	**0.00**	2.27
C 0.5 Cor 08	0.39	**13.93**	**0.06**	2116.19
C 0.5 Cor 09	**0.96**	**480.62**	*	>3600
C 0.5 Cor 10	**0.50**	**342.67**	*	>3600

In Table 7, we report the gaps and the CPU times for $STPD_{LMTZ} + P$ and for $STPD_{MCF} + P$ over 20 instances of Class C with correlated delays and after having performed the reduction procedures. In this table, the symbol "∗" means that the LP relaxation of $STPD_{MCF} + P$ has not been solved within one hour of CPU time and the symbol "-" means that a proven optimal integer solution has not been obtained.

We see from Table 7 that $STPD_{MCF} + P$ provides, in all the cases, gaps which are smaller than those provided by $STPD_{LMTZ} + P$. Interestingly, we observe that, despite the fact that $STPD_{MCF} + P$ exhibits a tight LP relaxation, it fails to solve 10 of the 20 instances. By contrast, $STPD_{LMTZ} + P$ fails to solve only 2 over 20 instances. Moreover, when an instance is solved by both formulations, $STPD_{LMTZ} + P$ requires significantly shorter CPU time than $STPD_{MCF} + P$ does.

5. Conclusion

In this paper, we have proposed an exact approach for the STPD. Our approach is based on a new tightened compact formulation for the Steiner Tree Problem with Delays. The proposed formulation is based on a new lifted Miller–Tucker–Zemlin subtour elimination constraint. Furthermore, we have described several tailored preprocessing techniques for both reducing the problem size and tightening the LP relaxation.

The relevance of our research stems from the well-known fact that the classical MTZ subtour elimination constraints consistently yield the weakest LP relaxation of all proposed formulations. Consequently, even though the MTZ constraints-based formulations are both elegant and compact, they are often disregarded because of their hopeless ineffectiveness. Hence, an interesting issue that was investigated in this paper is to develop new enhanced variants of the MTZ constraints that prove useful for solving instances of practical size. Our primary contribution is to demonstrate the efficacy of the combination of the new enhanced MTZ constraints and reduction techniques. Indeed, we have reported the results of comprehensive computational experiments that provide strong evidence that the proposed compact formulation-based approach can consistently solve medium to large-scale STPD instances to optimality. These results were obtained on a large set of STPD instances that includes both sparse and dense graphs as well as weakly and tightly constrained instances. To the best of our knowledge, this is the first significant contribution with regard to the optimal solution of nontrivial instances of an NP-hard problem using MTZ constraints-type constraints.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008

Considering the computational benefits of deriving tightened compact formulations, it is of interest to investigate the application of the Reformulation-Linearization Technique (RLT) to the STPD (and to similar tree optimization problems as well). This technique is a generic strategy for constructing tight polyhedral relaxations of the feasible set of 0-1 mixed integer linear programs [26]. During the last few years, it has been successfully applied to derive tightened compact formulations for the asymmetric traveling salesman [27]. Research into this direction is recommended for future studies. Nevertheless, the study of computationally effective compact formulations is still in its infancy, and one could reasonably expect that tailored state-of-the-art exact approaches (that often require significant analysis and design efforts) would prove very useful to solve very large scale STPD instances. Thus, a second issue worthy of future investigation, is the study of the facial structure of the STPD. We expect that the description of facets together with strong valid inequalities would be useful to tighten the model representation. This would lead to the development of an effective branch-and-cut algorithm.

Acknowledgments

The authors would like to thank the anonymous reviewers for their precious suggestions provided.

Appendix

In this section, we collect the proofs of the results presented in Section 2.

Proof of Proposition 2.1. If the dummy arcs are removed and isolated Steiner nodes are discarded from $\vec{T}$, then we get an arborescence T rooted at node 1 and spanning all terminals and, possibly, a subset of Steiner nodes. Since the dummy arcs have a zero cost, then $\vec{T}$ and T have the same cost. Moreover, if $\vec{T}$ is delay-feasible, then T is delay-feasible and, thus a feasible STPD solution. Conversely, if T is a feasible STPD solution, then adding to T the arc $(0, 1)$ and the arcs $(0, j)$ for each j isolated Steiner nodes in T yields a delay-feasible spanning arborescence $\vec{T}$ of $\bar{B}$ rooted at node 0 having the same cost as T. □

Lemma 5.1. *Let $(\bar{y}, \bar{t})$ be a feasible solution for* STPD_{MTZ}. *For each $(i, j) \in A$, if $\bar{y}_{ij} = 1$, then:* (i) $\bar{y}_{ji} = 0$; (ii) $\bar{y}_{kj} = 0$, *for any* $(k, j) \in \varphi_{ji}$; (iii) $\bar{y}_{hi} = 0$, *for any* $(h, i) \in \varphi_{ij}$.

Proof. Let $(i, j) \in A$ with $\bar{y}_{ij} = 1$.

(i) In view of constraints (10) or (16), it holds that $\bar{y}_{ji} = 0$.
(ii) Let $(k, j) \in \varphi_{ji}$; since constraints (9) are fulfilled, then $\bar{y}_{kj} = 0$.
(iii) Suppose on the contrary that $\bar{y}_{hi} = 1$ for an arc $(h, i) \in \varphi_{ij}$. Then from (9), it follows that $\bar{y}_{li} = 0$ for all $(l, i) \in A$ with $l \neq h$. Expressing constraints (11) for the arcs (h, i) and (i, j), we have $\bar{t}_j \geq \bar{t}_i + \theta_{ij}$ and $\bar{t}_i \geq \bar{t}_h + \theta_{hi}$, respectively, and thus $\bar{t}_j \geq \bar{t}_h + \theta_{hi} + \theta_{ij}$. Combining the fact that $\bar{t}_h \geq \lambda_h$, and that $\lambda_h + \theta_{hi} + \theta_{ij} > \mu_j$ since $(h, i) \in \varphi_{ij}$, it follows that $\bar{t}_j \geq \bar{t}_h + \theta_{hi} + \theta_{ij} \geq \lambda_h + \theta_{hi} + \theta_{ij} > \mu_j$, which is a contradiction since $\bar{t}_j \leq \mu_j$ and hence the thesis. □

Proof of Proposition 2.2. Let $(\bar{y}, \bar{t})$ be a feasible integer solution. We will consider three different cases:

Case 1: $\bar{y}_{ij} = 1$. Applying Lemma 5.1, we get $\bar{t}_i - \bar{t}_j \leq -\theta_{ij}$ which is true.
Case 2: $\bar{y}_{ji} = 1$. Considering (21) for arc (j, i) we get

$$\bar{t}_j - \bar{t}_i \leq -\theta_{ji}. \tag{24}$$

Moreover, we have $\bar{y}_{kj} = 0$ for any $(k, j) \in \varphi_{ji}$, and $\bar{y}_{hi} = 0$ for any $(h, i) \in \varphi_{ij}$. Hence, considering (21) for the arc (i, j), we get

$$\bar{t}_i - \bar{t}_j \leq M_{ij} - \theta_{ij} - \alpha_{ji}. \tag{25}$$

The right-hand side of (25) is equal to θ_{ji} thus, combining (24) and (25) we get

$$\theta_{ji} \leq \bar{t}_i - \bar{t}_j \leq \theta_{ji}. \tag{26}$$

Hence, we have $\bar{t}_i - \bar{t}_j = \theta_{ji}$. This latter inequality holds because $\bar{y}_{ji} = 1$.
Case 3: $\bar{y}_{ij} = 0$ and $\bar{y}_{ji} = 0$. If i and j do not have any predecessors different from the dummy node 0, then constraint (21) is verified. Suppose that h^* and k^* are the predecessors of i and j, respectively (the case in which either i or j has a predecessor in V can be easily obtained from this case). We consider the following subcases:

(i) $(h^*, i) \in \varphi_{ij}$ and $(k^*, j) \in \varphi_{ji}$. In this case (21) reads

$$\bar{t}_i - \bar{t}_j \leq M_{ij} - \theta_{ij} - \gamma_{h^*i} - \beta_{k^*j} = \mu_i - \lambda_j - \gamma_{h^*i} - \beta_{k^*j}. \tag{27}$$

Since $\bar{y}_{h^*i} = 1$ and $\bar{y}_{k^*j} = 1$, then $\bar{t}_i \leq \min(\mu_i, \mu_{h^*} + \theta_{h^*i})$ and $\bar{t}_j \geq \lambda_{k^*} + \theta_{k^*j}$. If $\mu_i \leq \mu_{h^*} + \theta_{h^*i}$, then $\gamma_{h^*i} = 0$ and $\bar{t}_i - \bar{t}_j \leq \mu_i - \gamma_{h^*i} - (\lambda_{k^*} + \theta_{k^*j}) = \mu_i - \lambda_j - \gamma_{h^*i} - \beta_{k^*j}$, else if $\mu_i > \mu_{h^*} + \theta_{h^*i}$, then $\bar{t}_i \leq \mu_{h^*} + \theta_{h^*i}$ and thus $\bar{t}_i - \bar{t}_j \leq \mu_{h^*} + \theta_{h^*i} - (\lambda_{k^*} + \theta_{k^*j}) = \mu_i - \mu_i + \mu_{h^*} + \theta_{h^*i} - \lambda_j - \beta_{k^*j} = \mu_i - \lambda_j - \gamma_{h^*i} - \beta_{k^*j}$. Hence, inequality (27) is verified.

(ii) $(h^*, i) \notin \varphi_{ij}$ and $(k^*, j) \notin \varphi_{ji}$. In this case (21) becomes $\bar{t}_i - \bar{t}_j \leq M_{ij} - \theta_{ij}$ which is obviously correct.
(iii) $(h^*, i) \notin \varphi_{ij}$ and $(k^*, j) \in \varphi_{ji}$. In this case (21) reads

$$\bar{t}_i - \bar{t}_j \leq M_{ij} - \theta_{ij} - \beta_{k^*j} = \mu_i - \lambda_j - \beta_{k^*j} \tag{28}$$

which is obviously correct since $\bar{t}_i \leq \mu_i$ and $\bar{t}_j \geq \lambda_{k^*} + \theta_{k^*j} = \beta_{k^*j} + \lambda_j$.
(iv) $(h^*, i) \in \varphi_{ij}$ and $(k^*, j) \notin \varphi_{ji}$. This case is similar to (iii) and concludes the proof. □

References

[1] E. Althaus, T. Polzin, S.V. Daneshmand, Improving linear programming approaches for the Steiner tree problem, in: Experimental and Efficient algorithms, in: Lecture Notes in Comput. Sci., vol. 2647, Springer, Berlin, 2003, pp. 1–14.
[2] A. Balakrishnan, N.R. Patel, Problem reduction methods and a tree generation algorithm for the Steiner network problem, Networks 17 (1) (1987) 65–85.
[3] S. Chopra, E. Gorres, M.R. Rao, Solving the Steiner tree problem on a graph using branch and cut, ORSA Journal on Computing 4 (3) (1992) 320–335.
[4] A.M. Costa, J.-F. Cordeau, G. Laporte, Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints, Networks 53 (2) (2009) 141–159.
[5] M. Desrochers, G. Laporte, Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints, Operations Research Letters 10 (1) (1991) 27–36.
[6] D.Z. Du, B. Lu, H. Ngo, P.M. Pardalos, Steiner tree problems, in: C.A. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization, second ed., Springer, 2009, pp. 3723–3736.
[7] N. Ghaboosi, A.T. Haghighat, Tabu search based algorithms for bandwidth-delay-constrained least-cost multicast routing, Telecommunication Systems 34 (3–4) (2007) 147–166.
[8] L. Gouveia, Using the Miller–Tucker–Zemlin constraints to formulate a minimal spanning tree problem with hop constraints, Computers & Operations Research 22 (9) (1995) 959–970.
[9] L. Gouveia, Multicommodity flow models for spanning trees with hop constraints, European Journal of Operational Research 95 (1) (1996) 178–190.
[10] L. Gouveia, A. Paias, D. Sharma, Modeling and solving the rooted distance-constrained minimum spanning tree problem, Computers & Operations Research 35 (2) (2008) 600–613.
[11] L. Gouveia, L. Simonetti, E. Uchoa, Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs, Mathematical Programming 128 (2011) 123–148.
[12] G.Y. Handler, I. Zang, A dual algorithm for the constrained shortest path problem, Networks 10 (4) (1980) 293–309.
[13] F.K. Hwang, D.S. Richards, Steiner tree problems, Networks 22 (1) (1992) 55–89.
[14] H.C. Joksch, The shortest route problem with constraints, Journal of Mathematical Analysis and Applications 14 (1966) 191–197.
[15] B.N. Khoury, P.M. Pardalos, An exact branch and bound algorithm for the Steiner problem in graphs, in: Computing and Combinatorics, in: Lecture Notes in Comput. Sci., Springer, Berlin, 1995, pp. 582–590.
[16] T. Koch, A. Martin, Solving Steiner tree problems in graphs to optimality, Networks 32 (3) (1998) 207–232.
[17] V.P. Kompella, J. Pasquale, G.C. Polyzos, Multicast routing for multimedia communication, IEEE/ACM Transactions on Networking 1 (3) (1993) 286–292.
[18] Z. Kun, W. Heng, F.Y. Liu, Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing, Computer Communications 28 (11) (2005) 1356–1370.
[19] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Reinhart, and Winston, 1976.
[20] C.E. Miller, A.W. Tucker, R.A. Zemlin, Integer programming formulation of traveling salesman problems, Journal of the Association for Computing Machinery 7 (4) (1960) 326–329.
[21] M. Minoux, Plus court chemin avec contraintes: algorithmes et applications, Annales des Télécommunications 30 (1975) 383–394.
[22] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons Inc., New York, 1988.
[23] C.A.S. Oliveira, P.M. Pardalos, A survey of combinatorial optimization problems in multicast routing, Computers & Operations Research 32 (8) (2005) 1953–1981.
[24] T. Polzin, S.V. Daneshmand, A comparison of Steiner tree relaxations, Discrete Applied Mathematics 112 (1–3) (2001) 241–261.
[25] M. Santos, L.M. Drummond, E. Uchoa, A distributed dual ascent algorithm for the hop-constrained Steiner tree problem, Operations Research Letters 38 (1) (2010) 57–62.
[26] H.D. Sherali, W.P. Adams, A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems, Discrete Applied Mathematics 52 (1) (1994) 83–106.
[27] H.D. Sherali, P.J. Driscoll, On tightening the relaxations of Miller–Tucker–Zemlin formulations for asymmetric traveling salesman problems, Operations Research 50 (4) (2002) 656–669.
[28] R. Sriram, G. Manimaran, C.S.R. Murthy, Algorithms for delay-constrained low-cost multicast tree construction, Computer Communications 21 (18) (1998) 1693–1706.
[29] H. Takahashi, A. Matsuyama, An approximate solution for the Steiner problem in graphs, Mathematica Japonica 24 (6) (1979–1980) 573–577.
[30] E. Uchoa, M. Poggi de Aragão, C.C. Ribeiro, Preprocessing Steiner problems from VLSI layout, Networks 40 (1) (2002) 38–50.
[31] Q. Zhu, M. Parsa, J.J. Garcia-Luna-Aceves, A source-based algorithm for delay-constrained minimum-cost multicasting, in: INFOCOM'95, Washington, DC, USA, 1995, pp. 377–385.

Please cite this article in press as: V. Leggieri, et al., The Steiner Tree Problem with Delays: A compact formulation and reduction procedures, Discrete Applied Mathematics (2011), doi:10.1016/j.dam.2011.07.008