DIMACS Technical Report 2007-01
February 2007

Simple and fast reoptimizations for the Steiner tree problem

by

Bruno Escoffier
LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine
Place du Maréchal De Lattre de Tassigny
75775 Paris Cedex 16, France
E-mail: `escoffier@lamsade.dauphine.fr`

Martin Milanič[1]
RUTCOR, Rutgers University
640 Bartholomew Road, Piscataway
NJ 08854-8003, USA
E-mail: `mmilanic@rutcor.rutgers.edu`

Vangelis Th. Paschos
LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine
Place du Maréchal De Lattre de Tassigny
75775 Paris Cedex 16, France
E-mail: `paschos@lamsade.dauphine.fr`

[1]Part of this work was carried out while the author was with the LAMSADE on a visiting researcher position supported by a common CNRS-NSF research project "Algorithmic Decision Theory" between the LAMSADE and the DIMACS.

ABSTRACT

We address reoptimization issues for the Steiner tree problem. We assume that an optimal solution is given for some instance of the problem and the objective is to maintain a good solution when the instance is subject to minor modifications, the simplest such modifications being vertex insertions and deletions. We propose fast reoptimization strategies for the case of vertex insertions and we show that maintenance of a good solution for the "shrunk" instance, without ex nihilo computation, is impossible when vertex deletions occur. We also provide lower bounds for the approximation ratios of the reoptimization strategies studied.

Keywords: Steiner tree, approximation algorithms, reoptimization

1 Introduction

Given a graph $G = (V, E)$, a subset $R \subseteq V$ of its vertex-set (the so-called *terminal* vertices), and nonnegative integer weights $\{w(e) : e \in E\}$ on the edges of G, the Steiner tree problem consists in finding a lightest *Steiner tree for* (G, R), i.e., a subtree T of G with $R \subseteq V(T)$ (where the weight of a tree is given by the sum of the weights of its edges).

We will assume in the sequel that the graph G is complete, and that the weights on the edges induce a non-negative integer metric on the subsets of size 2 of V, i.e., that for every three vertices $x, y, z \in V$, the *triangle inequality*: $w(xz) \leqslant w(xy) + w(xz)$ holds. Therefore, an instance of the Steiner tree problem is of the form (V, R, w) where $R \subseteq V$ and w is a nonnegative integer metric on V. A *Steiner tree for* (V, R) is a Steiner tree for $((V, \binom{V}{2}), R)$.

The Steiner tree problem is one of the most famous combinatorial optimization problems in network design. The frequency with which it arises in such applications motivates numerous works on this problem, under several assumptions, hypotheses and models. Steiner tree is well known to be **NP**-hard. The first approximation algorithm for it appeared in [1] (see also [7, 11]). This algorithm is a primal-dual generalization of the following simple heuristic: compute the shortest paths between all pairs of terminal vertices and then compute a minimum-cost spanning tree over the shortest-path weighted complete graph with vertex-set R. Removal of redundant edges might be needed in order to transform the tree so computed to a Steiner tree of G. The approximation ratio achieved by this algorithm is bounded above by 2. If G is metric, then a minimum-cost spanning tree on the induced subgraph $G[R]$ (in what follows, given a subset V' of vertices of a graph G, we denote by $G[V']$ the subgraph of G induced by V') achieves the same approximation ratio. This result has been improved in [9] down to 1.55 in complete metric graphs and to 1.28 for complete graphs with edge costs 1 and 2. A survey of approximation results for Steiner tree problem can be found in [7].

In this paper we address the *reoptimization* issue where the following situation is considered. We are given an optimum solution of an initial instance and we wish to maintain a good solution efficiently, when the instance is slightly modified. This working framework has already been adopted for several optimization problems, such as scheduling problems ([4, 5, 10]) for practical applications, and classical polynomial problems, such as the minimum spanning tree, where the goal is to recompute the optimum solution as fast as possible ([6, 8]). It has been also addressed for the minimum traveling salesman problem in [2] and, recently for both minimum and maximum traveling salesman problems in [3].

For the Steiner tree problem handled here, we assume that an optimal solution T_{opt} has already been computed for a metric complete graph G when some minor modification occurs in the graph. This modification may be the arrival of some (one or more) new vertices together with the edges linking them to G (in such a way that the extended graph G' remains complete and metric), or the removal of some vertices of G (together with the edges linking them to the surviving graph). Then, the question is: "can one maintain, or at least modify very quickly the existing solution, in order to obtain a good solution for the modified instance without the need to recompute such a solution thoroughly?". The "goodness" of a solution is measured by computing its approximation ratio. More precisely, if T is a Steiner tree for (V, R), then we say that T is a *ρ-approximation* for (V, R, w) if $w(T) \leqslant \rho w(T_{opt})$ where T_{opt} is a solution to the Steiner tree problem. We say that ρ is the *approximation ratio* achieved by T.

In what follows, we propose a simple reoptimization strategy, called `REOPT`, mainly based upon a minimum spanning tree computation adapted for the case studied (terminal, or nonterminal vertices), that efficiently tackles the case of vertex insertions in the graph. Let us note that most of the approximation algorithms known for the Steiner tree problem seem to be hard to adapt in order to tackle dynamic situations such as the ones handled in this paper. In Section 3, we handle insertion of one vertex x in the initial graph. We provide a tight 3/2-approximation ratio in both cases where x is terminal, or nonterminal. In Section 4, we handle insertions of more than one vertex in the graph. In Section 4.1, we study insertion of $p \geqslant 1$ nonterminals and we prove, for this case also, a tight 3/2-approximation ratio for `REOPT`. On the other hand, in Section 4.2, we assume that p vertices are inserted, k of which being terminals. For this case we show that the ratio of `REOPT` is $2-1/(k+2)$, while its lower bound is $2-2/(k+2)$. In Section 5, we provide a general lower bound on the approximation ratios for a class of solution structures showing, informally, that if one tries to keep a good approximation ratio for the modified solution, she/he must eventually consider vertices that are not contained in the initial optimal solution.

Finally, for the complementary problem of vertex removals from the initial graph, we show in Section 5 that, sometimes, complete recomputation of a new solution for the "shrunk" instance is unavoidable.

2 Some easy preliminary results

We give in this section some preliminary results linking approximation of reoptimization to approximation of the Steiner tree problem. They can be seen as simple reductions from the reoptimization context to the classical approximation one. Their proofs, being fairly similar to the ones in Sections 3 and 4, are omitted.

Proposition 1. *Let T be a ρ-approximation for an instance (V, R, w) of the Steiner tree problem. Suppose that a nonterminal vertex, say x, is added to V together with new edges $\{xy : y \in V\}$ and their weights such that the new instance is again metric. Then, T is a ρ'-approximation for the extended instance such that: $\rho' \leqslant 2(1-(1/|R|))\rho$.*

The result of Proposition 1 is sharp. Indeed, let the weights of edges connecting the new vertex x to the vertices of R be equal to 1, and let all the remaining edge-weights be equal to 2. If we take the current Steiner tree to be an optimal Steiner tree in G, then $w(T) = 2(|R|-1)$, while the new optimal weight is $|R|$.

Proposition 2. *Let T be a ρ-approximation for an instance (V, R, w) of the Steiner tree problem (with $|R| \geqslant 2$). Suppose that a terminal vertex, say x, is added to V together with new edges $\{xy : y \in V\}$ and their weights such that the new instance is again metric.*

Let T' denote the tree T, augmented with a lightest edge connecting a vertex of R with x (ties broken arbitrarily). Then, T' is a ρ'-approximation for the extended instance such that: $\rho' \leqslant (2 - (1/|R|))\rho$.

It is easy to verify that the counterexample of Proposition 1 shows the sharpness of Proposition 2 as well.

3 One vertex is added

In this section, we consider two cases, according to whether the new vertex is terminal or not.

3.1 The new vertex is nonterminal

Let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem. Suppose that a nonterminal vertex, say x, is added to V together with new edges $\{xy : y \in V\}$ and their weights such that the new instance is again metric. Let T_x denote a minimum spanning tree on the vertex-set $R \cup \{x\}$.

We consider the algorithm `REOPT` which consists in computing the best solution between T and T_x (ties broken arbitrarily). Obviously, its complexity is the one of computation of a minimum spanning tree on the clique induced by $R \cup \{x\}$, i.e., $O(|R|^2 \log |R|)$.

Theorem 1. *`REOPT` is a 3/2-approximation algorithm. This bound is tight.*

Proof. Let $\tilde{T}$ denote an optimal Steiner tree in the extended graph, and T' the solution computed by `REOPT`. If $x \notin V(\tilde{T})$, then T is optimum (and so is T'). So we may assume that $x \in V(\tilde{T})$. Let $\{x_1, \dots, x_k\}$ be the set of neighbors of x in $V(\tilde{T})$. Removing x from $\tilde{T}$ results in a forest F consisting of $k \geqslant 1$ trees $T_1, \dots, T_k$ with $x_i \in V(T_i)$ for $i \in [k]$ (in what follows, we denote by $[k]$ the set of integers from 1 to k). Note that $k \leqslant |R|$, since every tree T_i contains at least one terminal vertex. Let T_0 denote the set of edges adjacent to x in $\tilde{T}$. Then:

$$w\left(\tilde{T}\right) = \sum_{i=0}^{k} w\left(T_i\right)$$

Now, link the vertices $x_1, \dots, x_k$ with a path $P = (x_1, \dots, x_k)$. Together with the trees T_i ($i \geqslant 1$), this is a Steiner tree on the initial graph, the value of which is at least $w(T)$. By triangle inequality, we get that $w(P) \leqslant 2w(T_0)$. Therefore:

$$w\left(T'\right) \leqslant w(T) \leqslant 2w\left(T_0\right) + \sum_{i=1}^{k} w\left(T_i\right) \tag{1}$$

On the other hand, let $i \in [k]$. Using an Euler tour on T_i (and triangle inequalities), we can easily find a path $P_i = (x_i, v_i^1, \dots, v_i^{k_i})$ starting in x_i and containing all the terminal vertices $v_i^1, \ldots, v_i^{k_i}$ of T_i such that $w(P_i) \leqslant 2w(T_i)$. Then, using again the triangle inequality ($w(xx_i)+w(x_iv_i^1) \geqslant w(xv_i^1)$), we get that the path $(x, v_i^1, \ldots, v_i^{k_i})$ has value at most $w(xx_i)+2w(T_i)$. Then, the union of these k paths $(x, v_i^1, \ldots, v_i^{k_i})$ if a Steiner tree T'' of value at most $w(T_0) + 2\sum_{i=1}^{k} w(T_i)$. Since this is a spanning tree on $R \cup \{x\}$, we obtain:

$$w\left(T'\right) \leqslant w\left(T_x\right) \leqslant w\left(T''\right) \leqslant w\left(T_0\right) + 2\sum_{i=1}^{k} w\left(T_i\right) \tag{2}$$

The sum of (1) and (2) leads to: $2w(T') \leqslant 3\sum_{i=0}^{k} w(T_i) = 3w(\tilde{T})$, which proves the upper bound claimed.

For the tightness of the bound, consider the following instance (see Figure 1). In the initial graph, there are two groups V_1 and V_2 of n terminal vertices, and one nonterminal vertex v. The weight between v and a vertex in V_1 is equal to 1, as well as the weight between the new vertex x and a vertex in V_2; also, the weight between x and v is 1. All other weights are equal to 2.

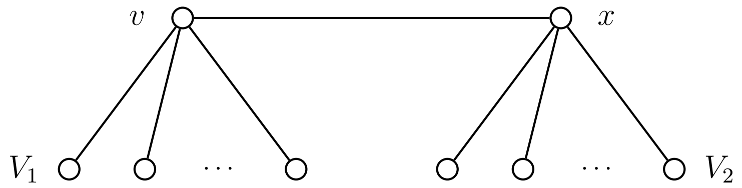

Figure 1: Instance with edges of weight 1.

Then, on the initial instance, an optimum solution T is given by the union of all edges between v and a vertex in V_1 and a path starting in v and containing all the vertices in V_2: $w(T) = 3n$.

Given the symmetry of the final instance, it is easy to see that an optimum spanning tree on $R \cup \{x\}$ has the same value. However, the Steiner tree depicted in Figure 1 has value $2n + 1$. □

Remark 1. *If the number of terminal vertices is small, then one can slightly improve the bound of Theorem 1.* `REOPT` *is a ρ'-approximation algorithm, where:*

$$\rho' \leqslant 2 - \frac{1}{2\left(1 - \frac{1}{|R|}\right)} = \frac{3}{2} - \frac{1}{2(|R|-1)}$$

Indeed, when computing the path P, by triangle inequality we get that $w(P) \leqslant 2w(T_0) - w(xx_1) - w(xx_n)$. Choosing (without loss of generality) xx_1 and xx_n as the two heaviest among the edges xx_i, $w(P) \leqslant 2(1-1/k)w(T_0)$. Plugging this new inequality in the proof of Theorem 1 leads to the result.

Remark 2. *One can wonder what happens when, instead of starting with an optimal solution T, we are given a ρ-approximate solution. A slight modification of the proof of Theorem 1 leads to the fact that the solution computed by* `REOPT` *is a ρ'-approximation, where: $\rho' \leqslant \min\{2, 3\rho/(1+\rho)\}$*

Remark 3. *The example given in the proof of Theorem 1 shows that the result is the same even if we consider instances with all the weights 1 or 2.*

Remark 4. *If we redefine T' to be the shortest among T, T_x, and T'', where T'' denotes a minimum spanning tree on the vertex-set $V(T) \cup \{x\}$, then the so obtained algorithm again has a tight approximation ratio of $3/2 - \Theta(1/|R|)$.*

3.2 The new vertex is terminal

In this subsection, we consider the case when the added vertex is terminal. As previously, let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem. Suppose that a terminal vertex, say x, is added to V together with new edges $\{xy : y \in V\}$ and their weights such that the new instance is again metric. Let T_x denote a minimum spanning tree on the vertex-set $R \cup \{x\}$, and let T'' denote the tree T, augmented with a lightest edge connecting a vertex of R with x.

In this case, `REOPT` computes the best solution T' between T_x and T'' (ties broken arbitrarily). The complexity of its implementation is $O(|R|^2 \log |R|)$.

Theorem 2. *`REOPT` is a 3/2-approximation algorithm. Furthermore, this bound is tight.*

Proof. As in the proof of Theorem 1, let $x_1, \dots, x_k$ denote the neighbors of x in an optimum solution $\tilde{T}$ on the extended graph, let $T_1, \dots, T_k$ be the connected components obtained by removing x from $\tilde{T}$ (with $x_i \in V(T_i)$), and let T_0 be the union of edges xx_i. Then $w(\tilde{T}) = \sum_{i=0}^{k} w(T_i)$, and, as previously, the minimum spanning tree T_x on $R \cup \{x\}$ satisfies:

$$w(T') \leqslant w(T_x) \leqslant w(T_0) + 2\sum_{i=1}^{k} w(T_i) \tag{3}$$

If, as in the proof of Theorem 1, we link the vertices x_i by a path $P_1 = (x_1, \dots, x_n)$, then the union of P_1 and T_i is a tree of value at most $2w(T_0) - w(xx_1) - w(xx_n) + \sum_{i=1}^{k} w(T_i)$. Then, to get a Steiner tree, we have to connect x. Note that each T_i (and in particular T_1) has at least one terminal vertex. If we link x to one terminal vertex of T_1, then this edge has value at most $w(xx_1) + w(T_1)$.

Since T is an optimum solution on the initial instance, $w(T) \leqslant 2w(T_0) - w(xx_1) - w(xx_n) + \sum_{i=1}^{k} w(T_i)$. Moreover, since each terminal vertex is in $V(T)$, the edge used by `REOPT` to connect x has value at most $w(xx_1) + w(T_1)$. Then:

$$w(T'') \leqslant w(T) + w(xx_1) + w(T_1) \leqslant 2w(T_0) + \sum_{i=1}^{k} w(T_i) + w(T_1) - w(xx_n)$$

We can do the same thing choosing, instead of T_1, each of the T_j's:

$$w(T'') \leqslant 2w(T_0) + \sum_{i=1}^{k} w(T_i) + w(T_j) - w(xx_{j-1})$$

Summing up these inequalities leads to:

$$kw(T'') \leqslant (2k-1)w(T_0) + (k+1)\sum_{i=1}^{k} w(T_i) \tag{4}$$

Adding (3) with coefficient $(k-2)$ and (4) with coefficient 1 gives:

$$(2k-2)w(T') \leqslant (3k-3)\sum_{i=0}^{k} w(T_i)$$

The tightness of the bound follows from the instance given in the proof of Theorem 1 (Figure 1, considering now x as a terminal vertex). □

As in Theorem 1, this result can be slightly improved when the number of terminal vertices is small. More precisely, using the fact that we can assume that each nonterminal vertex has degree at least 3 in an optimum solution, one can see that linking x to a terminal vertex of T_i costs at most $w(xx_i)+w(T_i)/2$. Using this inequality, we get a $((3/2)-(1/(8|R|-6)))$-approximate solution.

Moreover, as previously, if we start from a ρ-approximate solution instead of an optimum one, we get a $\min\{2, 3\rho/(\rho+1)\}$-approximate solution.

4 More vertices are added

In this section, we consider two cases, according to whether a set of nonterminal vertices or a set including terminal and nonterminal vertices is inserted to the current graph.

4.1 Nonterminal vertices

Let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem. Suppose that p nonterminal vertices $Y = \{y_1, \ldots, y_p\}$ are added to V together with new edges and their weights such that the new instance is again metric.

We generalize `REOPT` as follows. For $Y' \subseteq Y$, let $T_{Y'}$ denote a minimum spanning tree on the vertex-set $R \cup Y'$. `REOPT` computes the best solution T' among the trees from $\{T\} \cup \{T_{Y'} : Y' \subseteq Y\}$ (ties broken arbitrarily). The complexity of this implementation is at most $O(2^p(|R|+p)^2 \log(|R|+p))$.

Theorem 3. *`REOPT` is a 3/2-approximation algorithm. This bound is tight.*

Proof. Let $\tilde{Y} = V(\tilde{T}) \cap Y$ be the set of new vertices used by an optimum solution $\tilde{T}$. We consider the connected components $T_1, \dots, T_k$ of the subgraph obtained from $\tilde{T}$ when we remove the new vertices. Moreover, let us denote by $X_1, \dots, X_q$ the connected components of the subgraph obtained from $\tilde{T}$ when we remove the initial vertices. Finally, if in $\tilde{T}$ there is an edge between T_i and X_j, we denote this edge by e_{ij}. Note that the bipartite graph $B = [U, L, \tilde{E}]$ where $\tilde{E}$ is the set of these edges e_{ij}, $U = \{T_i, i = 1, \dots, k\}$ and $L = \{X_j, j = 1, \dots, q\}$, is a tree. Obviously:

$$w\left(\tilde{T}\right) = \sum_{i=1}^{k} w\left(T_i\right) + \sum_{j=1}^{q} w\left(X_j\right) + w\left(\tilde{E}\right) \tag{5}$$

First, we bound from above the value of the initial solution T. Starting from $\tilde{T}$, we remove all the X_j's (and edges adjacent to it), in order to get a solution on the initial instance.

Consider X_j and the edges e_{ij} of $\tilde{T}$ adjacent to it. This is a tree; using a Euler tour on this tree (and removing the vertices in X_j), we can connect the T_i's adjacent to X_j using a path P_j of value at most $2\left(w(X_j) + \sum_{i|e_{ij}\in\tilde{E}} w_{ij}\right)$. More precisely, if we note $d_{X_j} = \max_{i|e_{ij}\in\tilde{E}}\{w(e_{ij})\}$, since we compute a path and not a cycle, we can find a path such that:

$$w\left(P_j\right) \leqslant 2\left(w\left(X_j\right) + \sum_{i|e_{ij}\in\tilde{E}} w_{ij}\right) - d_{X_j}$$

Replacing all the X_j's (and edges adjacent to it) by the paths P_j, we get a solution on the initial instance, the value of which is at least the value of T:

$$w(T) \leqslant \sum_{i=1}^{k} w\left(T_i\right) + 2\sum_{j=1}^{q} w\left(X_j\right) + 2w\left(\tilde{E}\right) - \sum_{j=1}^{q} d_{X_j} \tag{6}$$

Now, we bound from above the value of a minimum spanning tree $T_{\tilde{Y}}$ on $R \cup \tilde{Y}$. Starting from $\tilde{T}$, now we have to remove nonterminal vertices from the T_i's. We use the same technique. Consider T_i and the edges e_{ij} of $\tilde{T}$ adjacent to it. Again, this is a tree and using a Euler tour on this tree, we can connect the X_j's adjacent to T_i and the terminal vertices of T_i (if any) using a path P'_i. As previously, if we denote $d_{T_i} = \max_{j|e_{ij}\in\tilde{E}}\{w(e_{ij})\}$, we can find a path such that $w(P'_i) \leqslant 2(w(T_i) + \sum_{j|e_{ij}\in\tilde{E}} w_{ij}) - d_{T_i}$.

Replacing the T_i's (and edges adjacent to it) by the P'_i's, we get a tree on $R \cup Y$, the value of which is at least $w(T_{\tilde{Y}})$:

$$w\left(T_{\tilde{Y}}\right) \leqslant 2\sum_{i=1}^{k} w\left(T_i\right) + \sum_{j=1}^{q} w\left(X_j\right) + 2w\left(\tilde{E}\right) - \sum_{i=1}^{k} d_{T_i} \tag{7}$$

Summing up (6) and (7), we get that the solution T' computed by `REOPT` satisfies:

$$2w\left(T'\right) \leqslant 3\sum_{i=1}^{k} w\left(T_i\right) + 3\sum_{j=1}^{q} w\left(X_j\right) + 4w\left(\tilde{E}\right) - \sum_{i=1}^{k} d_{T_i} - \sum_{j=1}^{q} d_{X_j}$$

To conclude, using (5), we just have to show that $\sum_{i=1}^{k} d_{T_i} + \sum_{j=1}^{q} d_{X_j} \geqslant w(\tilde{E})$. The left hand side corresponds to summing up, for each vertex in the tree B, the heaviest edge adjacent to this vertex. This sum is obviously greater than the total weight $w(\tilde{E})$ of the edges in B: to see this, just consider that $w(\tilde{E})$ can be seen as the sum, for each vertex (root excepted), of the edge linking this vertex to its father. □

4.2 Several terminal and nonterminal vertices are added

Let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem. Suppose that p vertices $Y = \{y_1, \ldots, y_p\}$ are added to V together with new edges and their weights such that the new instance is again metric. Among these p new vertices, $Y_t = \{y_1, \ldots, y_t\}$ are terminal, while the remaining $p - t$ are nonterminal.

As in the case where nonterminal vertices are added, for Y' such that $Y_t \subseteq Y' \subseteq Y$, let $T_{Y'}$ denote a minimum spanning tree on the vertex-set $R \cup Y'$. Also, we consider a minimum spanning tree $T(Y_t)$ on the new terminal vertices, and link this tree to T using a lightest edge (ties broken arbitrarily) between Y_t and $V(T)$. This gives a solution T''.

Then, `REOPT` computes the best solution T' among the trees from $\{T''\} \cup \{T_{Y'} : T_t \subseteq Y' \subseteq Y\}$ (ties broken arbitrarily).

Theorem 4. *`REOPT` is a $(2 - (1/(t+2)))$-approximation algorithm.*

Proof. As previously, let us denote by $\tilde{T}$ an optimum Steiner tree of G. The proof of the theorem is based upon the following two cases:

1. the maximum weight w_{Ter} between two terminal vertices (either new or initial) is greater than $\varepsilon w(\tilde{T})$ (the value of ε will be specified later);
2. $w_{\text{Ter}} \leqslant \varepsilon w(\tilde{T})$.

In the first case, let v_0 and v_1 be two terminal vertices such that $w(v_0 v_1) > \varepsilon w(\tilde{T})$. Consider the tree $\tilde{T}$ rooted at v_0, and consider a depth-first visit of $\tilde{T}$, when v_1 is on the right hand side branch of the tree (the last visited). Then, if we stop this visit when visiting v_1 for the second time, we have a path on all vertices of $V(\tilde{T})$, of value at most $2w(\tilde{T}) - w(v_0 v_1)$ (thanks to triangle inequalities). Hence, a minimum spanning tree on the terminal vertices has value at most $(2 - \varepsilon) w(\tilde{T})$.

In the second case, revisit the proof of Theorem 3, in particular (7) shown there (we use the same notations):

$$w\left(T_{\tilde{Y}}\right) \leqslant 2 \sum_{i=1}^{k} w\left(T_i\right) + \sum_{j=1}^{q} w\left(X_j\right) + 2w\left(\tilde{E}\right) - \sum_{i=1}^{k} d_{T_i} \tag{8}$$

Note that this solution $T_{\tilde{Y}}$ is still feasible, and $w(T') \leqslant w\left(T_{\tilde{Y}}\right)$.

Revisit also (6):

$$w(T) \leqslant \sum_{i=1}^{k} w\left(T_i\right) + 2\sum_{j=1}^{q} w\left(X_j\right) + 2w\left(\tilde{E}\right) - \sum_{j=1}^{q} d_{X_j} \tag{9}$$

Of course, T is not feasible (as soon as $t \geqslant 1$). But since the weight between any two terminal vertices is at most $\varepsilon w(\tilde{T})$, we can connect the t new terminal vertices to an initial one with a path of value at most $t\varepsilon w(\tilde{T})$. In other words, the solution T'' satisfies $w(T'') \leqslant w(T) + t\varepsilon w(\tilde{T})$. Using the fact that $w(\tilde{T}) = \sum_{i=1}^{k} w(T_i) + \sum_{j=1}^{q} w(X_j) + w(\tilde{E})$, we get from (9):

$$w\left(T''\right) \leqslant (1+t\varepsilon)\sum_{i=1}^{k} w\left(T_i\right) + (2+t\varepsilon)\sum_{j=1}^{q} w\left(X_j\right) + (2+t\varepsilon)w\left(\tilde{E}\right) - \sum_{j=1}^{q} d_{X_j} \tag{10}$$

Since T' is better than T'' and $T_{\tilde{Y}}$, we can sum up (8) and (10). Using the fact that $\sum_{j=1}^{q} d_{X_j} + \sum_{j=1}^{q} d_{X_j} \geqslant w(E)$, we obtain:

$$2w\left(T'\right) \leqslant (3+t\varepsilon)\sum_{i=1}^{k} w\left(T_i\right) + (3+t\varepsilon)\sum_{j=1}^{q} w\left(X_j\right) + (3+t\varepsilon)w\left(\tilde{E}\right) = (3+t\varepsilon)w\left(\tilde{T}\right)$$

So, the solution T' is both a $(2-\varepsilon)$- and a $(3+t\varepsilon)/2$-approximation. Letting $\varepsilon = 1/(t+2)$, we obtain the result. □

Remark 5. *The result of Theorem 4 is independent on the number of nonterminal vertices added. When $t = 0$, this is the 3/2-approximation for p new nonterminal vertices. Moreover, this bound is almost tight as shown in Theorem 5 (Section 5).*

Remark 6. *The running time is roughly $O(\max\{2^{p-t}(|R|+p)^2\log(|R|+p), t^2\log t, t|T|\})$. When the number of new nonterminal vertices is small, this is very quick.*

5 Negative results

In the context of reoptimization, it seems natural to reuse the pre-computed solution when the initial instance is subject to modifications. So, we are interested in particular in algorithms that do not perform ex nihilo computations of a new solution but they rather exploit existing ones. Hence, a natural question is to determine whether it is always possible to maintain a good approximation ratio using only vertices of the initial solution plus, eventually, some newly added ones.

The next result shows that this is not the case. Informally, if we wish to keep a good approximation ratio, we have to consider vertices not contained in the current solution as well.

Theorem 5. *Let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem, and X be the set of new vertices (either terminal or not). Let* `A` *be an algorithm for the reoptimization problem that produces a Steiner tree whose vertex-set is contained in $V(T) \cup X$. Then, the following holds:*

1. *if $X = \{x\}$ (one vertex is added, either terminal or not),* `A` *cannot achieve an approximation ratio better than $7/5$; furthermore, if edge-weights are either 1 or 2,* `A` *cannot achieve an approximation ratio better than $4/3$;*

2. *if $X = \{x_1, \ldots, x_t\}$ where, for $i \in [t]$, x_i is terminal (t terminal vertices are added),* `A` *cannot achieve an approximation ratio better than $2 - (2/(t+2))$, even if edge-weights are either 1 or 2.*

Proof. We first deal with item 1. We consider the graph G_i on 5 vertices $\{v_i, u_i^1, u_i^2, u_i^3, t_i\}$ (see Figure 2), with the following weights:

- $w(v_i u_i^k) = 1$, for $k, l \in [3]$;
- $w(u_i^k u_i^l) = 2$, for $k, l \in [3]$, $k \neq l$;
- $w(t_i u_i^k) = 4/3$, for $k \in [3]$;
- $w(v_i t_i) = 7/3$.

The initial graph is composed of n copies $G_1, \ldots, G_n$ of G_i, where $\{u_i^k : k \in [3], i \in [n]\}$ are terminal vertices, with the following weights, for $i, j \in [n]$:

- $w(t_i t_j) = 3 - \epsilon$, for $i \neq j$;
- all the other weights (between vertices of different copies) are equal to 4.

This instance is metric. An optimum solution T on this instance is given by taking edges $t_i u_i^k$, for $k \in [3]$ and $i \in [n]$, and by linking vertices t_i by a path $(t_1, \ldots, t_n)$. Its total weight is $w(T) = 4n + (3 - \epsilon)(n-1) \sim (7 - \epsilon)n$.

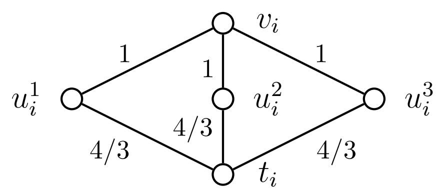

Figure 2: Graph G_i (other weights correspond to shortest paths).

Now, we add the new vertex x, where $w(xv_i) = 2$, $w(xu_i^k) = 3$, and $w(wt_i) = 3 + 4/3 = 13/3$ (for $k \in [3]$ and $i \in [n]$). Assume that x is nonterminal. Then, x is useless to improve the solution T by considering only vertices in $V(T)$. However, the solution $\tilde{T}$ consisting in taking the edges xv_i and $v_iu_i^k$, for $k \in [3]$ and $i \in [n]$, has value $5n$. If, on the other hand, x is terminal, the result is the same.

In the case of weights 1 or 2, we can get a similar result with a bound of 4/3. We use the same kind of graph, but instead of considering G_i, we consider H_i on 4 vertices $\{v_i, u_i^1, u_i^2, t_i\}$ (see Figure 3) with the following weights:

- $w(v_iu_i^k) = 1$, for $k \in [2]$;
- $w(t_iu_i^k) = 1$, for $k \in [2]$;
- all other weights are equal to 2.

The initial graph is composed of n copies $H_1, \dots, H_n$ of H_i, with weight 2 between vertices of different copies.

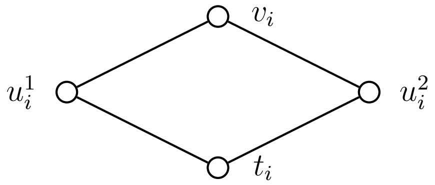

Figure 3: Graph H_i with edges of weight 1.

An optimum solution T for this instance is given by taking edges $t_iu_i^k$, for $k \in [2]$ and $i \in [n]$, and by linking vertices t_i by a path $(t_1, \dots, t_n)$. Its total weight is $w(T) = 2n + 2(n-1) = 4n - 2$.

Now, we add the new vertex x, where $w(xv_i) = 1$, $i \in [n]$, all other weights being equal to 2. Then, x is useless to improve the solution T by considering only vertices in $V(T)$.

However, the solution $\tilde{T}$ consisting in taking the edges xv_i and $v_iu_i^k$, for $k \in [2]$ and $i \in [n]$, has value $3n$. This completes the proof of item 1.

For the proof of item 2, i.e., for the case where t terminal vertices are added, consider that the initial graph has 3 vertices v_1, v_2, v_3, with $w(v_1v_2) = 2$ and $w(v_1v_3) = w(v_2v_3) = 1$. Vertices v_1 and v_2 are terminal. An optimum solution is $T = \{v_1v_2\}$. Then, add t terminal vertices, such that the weights between v_3 and the new vertices are 1, and all other weights are 2.

Then, an optimum solution without considering v_3 has value $2(t+1)$, whereas a star centered in v_3 has value $t+2$. □

We now handle reoptimization when a vertex is removed from the graph. We so have an initial instance (V, R, w) of the Steiner tree problem, and one vertex $x \in V$ is deleted. Of course, the strategy consisting of computing a minimum spanning tree on the set of surviving terminal vertices is a 2-approximation. If we consider, as previously, algorithms operating on some vertex-set contained in $V(T) \setminus \{x\}$, then we cannot improve this ratio.

Theorem 6. *Let T be an optimum solution for an instance (V, R, w) of the Steiner tree problem, and $x \in V$ a vertex deleted from the current graph. Let* `A` *be an algorithm for the reoptimization problem that produces a Steiner tree whose vertex-set is contained in $V(T) \setminus \{x\}$. Then,* `A` *cannot achieve an approximation ratio better than 2, even if edge-weights are either 1 or 2.*

Proof. Let us consider that the initial instance contains n terminal vertices $v_1, \ldots, v_n$, and two nonterminal vertices x and y. Weights between terminal vertices are 2, as well as $w(xy)$, while all other weights are 1. Then a star T on $v_1, \ldots, v_n$ centered in x is an optimum solution of the initial instance. When deleting vertex x, the best solution included in $T \setminus \{x\}$ is a spanning tree on $v_1, \ldots, v_n$, whose value is $2(n-1)$, while a star on $v_1, \ldots, v_n$ centered in y has value n. □

6 Conclusion

We have presented in this paper simple and fast reoptimization algorithms for the Steiner tree problem. We have handled insertion of one vertex x in the initial graph. We have provided reoptimization techniques achieving tight non-trivial approximation ratios for the cases where one or more vertices are inserted in the initial instance. We also have provided lower bounds showing that good approximation ratios cannot always be obtained without considering vertices that are not contained in the initial optimal solution. Finally, we have shown that when handling vertex-removals, complete recomputation of a new solution for the resulting instance is sometimes unavoidable.

The analysis presented in the paper leaves several open questions that, to our opinion, deserve further research.

1. Can one devise a reoptimization with a ratio better than 3/2 in the case where edge-weights are 1 or 2? We feel that a tight approximation ratio of 4/3 should be possible.

2. The second question deals with the matching of the upper and lower bounds of `REOPT` in the case where several terminal and nonterminal vertices are added (Section 4.2). Is it possible to get a lower bound of $2 - 1/(t+2)$, or an upper bound of $2 - 2/(t+2)$, or finally, to cross them somewhere between? Can the negative result of 7/5 in item 1 of Theorem 5 be tightened?

3. Can we find "general" lower bounds when p nonterminal vertices are added? Is it possible, for instance, to get a bound of 3/2 when p nonterminal vertices are added?

References

[1] A. Agrawal, P. N. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] C. Archetti, L. Bertazzi, and M. G. Speranza. Reoptimizing the traveling salesman problem. *Networks*, 42(3):154–159, 2003.

[3] G. Ausiello, B. Escoffier, J. Monnot, and V. Th. Paschos. Reoptimization of minimum and maximum traveling salesman's tours. In L. Arge and R. Freivalds, editors, *Proc. Scandinavian Workshop on Algorithm Theory, SWAT'06*, volume 4059 of *Lecture Notes in Computer Science*, pages 196–207. Springer-Verlag, 2006.

[4] M. Bartusch, R. H. Mohring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Ann. Oper. Res.*, 16:201–240, 1988.

[5] M. Bartusch, R. H. Mohring, and F. J. Radermacher. Design aspects of an advanced model-oriented DSS for scheduling problems in civil engineering. *Decision Support Systems*, 5:321–344, 1989.

[6] D. Eppstein, Z. Galil, G. F. Italiano, and A. Nissenzweig. Sparsification - a technique for speeding up dynamic graph algorithms. *J. Assoc. Comput. Mach.*, 44(5):669–696, 1997.

[7] C. Grpl, S. Hougardy, T. Nierhoff, and H. J. Prmel. Approximation algorithms for the Steiner tree problem in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner trees in industry*. Kluwer Academic Publishers, 2000.

[8] M. R. Henzinger and V. King. Maintaining minimum spanning trees in dynamic graphs. In *Proc. ICALP'97*, Lecture Notes in Computer Science, pages 594–604. Springer-Verlag, 1997.

[9] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proc. Symposium on Discrete Algorithms, SODA'00*, pages 770–779, 2000.

[10] M. W. Schäffter. Scheduling with forbidden sets. *Discrete Appl. Math.*, 72(1-2):155–166, 1997.

[11] V. Vazirani. *Approximation algorithms*. Springer, Berlin, 2001.