UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision

CNRS UMR 7024

CAHIER DU LAMSADE

254

Juin 2007

Probabilistic models for the STEINER TREE problem

Vangelis Th. Paschos, Orestis Telelis, Vassilis Zissimopulos

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Probabilistic models for the STEINER TREE problem

Vangelis Th. Paschos* Orestis A. Telelis† Vassilis Zissimopoulos‡

June 29, 2007

Abstract

In this paper we consider probabilistic models for STEINER TREE. Under these models, the problem is defined in a two-stage setting over a complete weighted graph whose vertices are associated with a probability of presence in the second stage. A first-stage feasible solution on the input graph might become infeasible in the second stage, when certain vertices of the graph fail (with the specified probability). Therefore, a well defined *modification strategy* is devised which transforms a partial solution to a feasible second-stage solution. The objective is to devise an algorithm for the first-stage solution (sometimes called the *a priori* or *anticipatory* solution) so that the expected second-stage solution cost is minimized. An important feature of this framework is that the modification strategy is essentially a part of the problem, while algorithmic treatment is required in the construction of the anticipatory solution. We essentially provide approximation results regarding two modification strategies. Furthermore, for the first of them we also formulate the associated objective function and prove complexity results for the computation of the anticipatory solution optimizing it.

1 Preliminaries

Given an edge-weighted complete graph $G(V, E, \vec{w})$, where $\vec{w}$ is a metric $|E|$-vector that represents the edge weights, and a subset of "terminal" vertices $T \subseteq V$, the STEINER TREE problem consists of determining a minimum total-cost tree S connecting the vertices of T. We study here the following 2-stage robust optimization model. Consider an instance $G(V, E, \vec{w})$ of STEINER TREE and assume that our problem is not to be necessarily solved on the whole G, but rather on a (unknown a priori) subgraph G'. Suppose that any vertex $v_i \in V$ has a probability p_i, indicating how vertex v_i is likely to be present in the final subgraph G'. Suppose also that any $v_i \in T$ has presence-probability 1 (this, in our context seems natural as we will see in Section 2). Consider finally that once G' is specified, the solver has no opportunity to solve it directly (for example assume that she/he has to react quasi-immediately, so no sufficient time is given her/him). Then, in order to tackle such constraints, the solver computes an *anticipatory Steiner tree* S in G, i.e., a solution for the entire instance G, and once G' becomes known, she/he modifies S by means of a quick algorithm, in order to get a tree S' that remains a Steiner tree with respect to G'. The objective in such approach is to determine an initial Steiner tree S for G such that, for any

*LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine, paschos@lamsade.dauphine.fr. Part of this work was partially carried out while the author was with the Department of Informatics and Telecommunications at the University of Athens on a visiting position supported by the Greek Ministry of Education and Research under the project PYTHAGORAS II. Financial support of the Greek Ministry of Education and Research and hospitality of members of Department of Informatics and Telecommunications are gratefully acknowledged

†Department of Informatics and Telecommunications, University of Athens, telelis@di.uoa.gr. Author partially supported by the Greek Ministry of Education and Research under the project PYTHAGORAS II

‡Department of Informatics and Telecommunications, University of Athens, vassilis@di.uoa.gr. Author partially supported by the Greek Ministry of Education and Research under the project KAPODISTRIAS K.A. 70/4/5821

subgraph G', induced by a subset $V' \subseteq V$ (containing, as assumed just before, all the vertices of T), that will be presented for optimization, the obtained Steiner tree S' respects some pre-defined quality criterion (for example, it is optimal for G', or it achieves, say, approximation ratio of a certain level).

Acquisition, validation and pertinence of input data are tackled in almost any operations research application. Although several well established theoretical models exist for problems arising in real world, direct application of these models may be difficult or even impossible due to incompleteness of data, or due to their questionable validity. Occasionally, one may be asked to produce an optimal operational design even before a complete deterministic picture of input data is provided, but only based on estimations and statistical measures. There are several applications where it might be impossible to obtain a current snapshot of the required information, since this information may be subject to constant high-rate change.

Several optimization frameworks have been introduced by the operations research community for handling these deficiencies, the most known being the *stochastic programming* (see [5, 7, 26] for basics, [27] for latest news, bibliography, and related software and [12, 13, 15, 28, 30, 31, 32] for recent hardness results and approximation algorithms) and the *robust discrete optimization* (see [2, 8, 18, 20, 21] for details). These frameworks constitute a means of structuring uncertainty and taking its existence into account during the optimization process. Robustness of the designed solution from both feasibility and cost perspectives in the presence of uncertainty is the main purpose of devising these frameworks during an operational design process.

As we have already mentioned, the model we deal with in this paper is a 2-stage optimization stochastic model. For graph-problems it can be sketched as follows. Consider an instance $G(V, E)$ of an optimization graph-problem Π and assume that instead of G Π will need to be solved on an unknown a priori subgraph G' of G. Any vertex $v_i \in V$ has a presence probability p_i. Assume also that a quick algorithm $\mathtt{M}$ is given, modifying any solution S for Π on G into a solution S' for Π in any subgraph G' of G (this algorithm is usually called *modification strategy*). The objective is to determine an anticipatory solution S^* for G such that, for any subgraph G', induced by a subset $V' \subseteq V$, that will be presented for optimization, the solution S' respects some pre-defined quality criterion. Such a goal amounts to compute an anticipatory solution S optimizing the expectation of its adaptation to any $V' \subseteq V$, i.e., optimizing the sum of the occurrence probability of V' multiplied by the value of S' over any $V' \subseteq V$. Indeed, under the hypotheses just stated, the objective is to determine some solution S^* optimizing the quantity (called also *functional*):

$$E(G, S, \mathtt{M}) = \sum_{V' \subseteq V} \Pr\left[V'\right] m\left(G\left[V'\right], S'\right) \tag{1}$$

where $m(G[V'], S')$ denotes the cost of S' in $G[V']$ and $\Pr[V']$ the occurrence probability of V' as second-stage portion expressed by: $\Pr[V'] = \prod_{v_i \in V'} p_i \prod_{v_i \in V \setminus V'} (1 - p_i)$. The optimization goal for $E(G, S, \mathtt{M})$ is identical to the one for the (deterministic) original problem Π. The derived stochastic problem will be denoted by PROBABILISTIC Π. It can easily be seen by the discussion just above that for an underlying deterministic problem Π, different modification strategies give rise to different probabilistic models of Π. In other words, for a problem Π any modification strategy induces its own probabilistic counterpart of Π. In order to distinguish these several counterparts we will use in what follows notation PROBABILISTIC $\Pi(\mathtt{M})$ for a probabilistic version of Π derived when using modification strategy $\mathtt{M}$.

This stochastic model is originally introduced in [16, 3] under the name *a priori probabilistic combinatorial optimization* as a means of structuring and handling uncertainty in the validity of input data. Since then, several well-known combinatorial optimization problems have been treated in this framework, such as the *longest path* ([22]), the *maximum independent set* ([23]),

and the *minimum coloring* ([25, 6]). Less recent studies include the *shortest path* ([17]), the *minimum-cost spanning tree* ([4]) and the *travelling salesman* ([16]).

Let us note that a complementary framework to the one of the a priori optimization, is the *reoptimization* consisting of solving ex nihilo and optimally the portion of the instance presented for optimization. Reoptimization is introduced in [16]. The functional for such a process can be expressed as:

$$E^*(G) = \sum_{V' \subseteq V} \Pr\left[V'\right] \operatorname{opt}\left(G'\right) \tag{2}$$

where $\operatorname{opt}(G')$ is the value of an optimal solution for Π in G' and, as previously, $G' = G[V']$. Obviously, for any modification strategy $\mathtt{M}$, denoting by S^* the optimal anticipatory solution of PROBABILISTIC $\Pi(\mathtt{M})$ and assuming that Π is a minimization problem:

$$E^*(G) \leqslant E\left(G, S^*, \mathtt{M}\right) \tag{3}$$

In this paper, we mainly study the approximation of two probabilistic models for STEINER TREE derived by two modifications strategies. Given an anticipatory solution S for PROBABILISTIC STEINER TREE($\mathtt{M}$) its approximation ratio is defined by $E(G, S, \mathtt{M})/E^*(G)$, where $E(G, S, \mathtt{M})$ and $E^*(G)$ are defined by (1) and (2), respectively. In Section 3 we formally introduce the first of them. We study its objective function (also called *functional*) and the complexity of its computation. Let us note that since it carries over all the possible subsets of the input vertex-set, the functional entails an exponential number of additive terms in such a way that the complexity of its numerical computation is not immediately polynomial. Next, we develop approximation issues for this model. In Section 4, we study another model for PROBABILISTIC STEINER TREE induced by a class of modification strategies called *reapproximation* and present approximation results.

2 Motivation and related work

Study of PROBABILISTIC STEINER TREE can be motivated by the following real-world application arising in the design of wireless networks. These networks consist of a set of nodes each transmitting a signal using a certain power level, thus being able to communicate with another network node within a range determined by the level of used power. Therefore the paradigm of multi-hop communication is inherent in such networks. That is if a node u wishes to send a message to some other node v, then this message is forwarded progressively from u to v by intermediate nodes.

Wireless networks lack stable infrastructure, therefore it has been proposed in the literature ([34]), that some nodes of the network should collaborate to simulate the functionality of such an infrastructure by means of a *network backbone* (also called spine). A backbone is a set of interconnected network nodes which handles routing of transmitted messages, monitoring of the network's state, acquisition and cross-processing of sensed data (in the case of sensor networks) and several other application-specific labors. Each node of the network is assigned to a backbone node. Backbone nodes are interconnected efficiently via a tree structure emerging as a result of appropriate tuning of their power levels of transmission (see, for example, [19] for some models concerning optimization of transmission power). Each node wishing to send a message to another node simply sends this message to the backbone node it is assigned to. Subsequently the message is routed through the backbone to the receiver.

Energy-efficient connectivity of the backbone nodes can be achieved through intermediate network nodes, by a solution to the Steiner tree problem (see[19] for graph models related to energy-efficiency optimization). However, nodes of a wireless network are often susceptible to failures caused by environmental factors, energy depletion, etc. Furthermore, it is a common

monitoring process (performed by the backbone nodes) to measure the probability of a wireless network node being available. If we consider wireless architectures with some sort of semi-stable infrastructure (where backbone nodes are of high reliability), then failure of backbone connectivity is due to failures of intermediate nodes connecting the backbone. The probabilistic Steiner tree problem (under the assumption made here on the omnipresence of terminals that represent these backbone nodes) appears to be an appropriate model for restoring connectivity while maintaining low energy consumption.

STEINER TREE is well known to be **NP**-hard. The first approximation algorithm for it appeared in [1] (see also [11, 33]). 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 T. 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 $G[T]$ achieves the same approximation ratio. This result has been improved in [29] 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 can be found in [11]. Recently, a robust optimization model, quite different from the model we tackle here, called *demand-robust Steiner tree* has been introduced and studied in [9]. Finally, let us note that STEINER TREE has also been studied extensively under the paradigm of stochastic programming; for the best known approximation algorithms in this framework see, e.g., [10, 14].

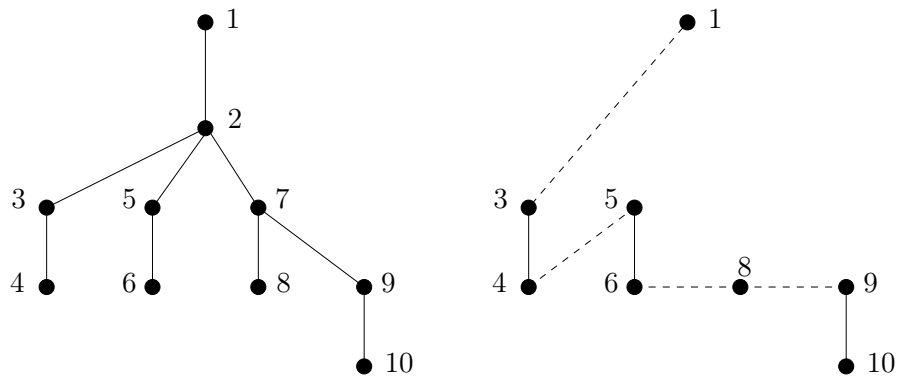
3 A probabilistic version for STEINER TREE with a depth-first-search modification strategy

Here we use the following modification strategy, called `DFS` (following the convention adopted in Section 1 the problem tackled here is denoted by PROBABILISTIC STEINER TREE(`DFS`)). Given an anticipatory Steiner tree $S \subseteq E$ (represented by the set of its edges), `DFS` first orders the edges of S by engaging a depth-first search (DFS) starting from an arbitrary leaf-vertex of S. Each vertex of S is assigned a number equal to the order of its visitation by the DFS. The tree S is thus partitioned into maximal edge-disjoint paths, whose edges are ordered by the numbering of their end-vertices. Let $\mathcal{P}(S) = \{P_1, P_2, \ldots, P_k\}$ be the set of these paths (each such path starting at a `DFS`-backtracking vertex and ending at a leaf vertex of the tree), where the paths are ordered in the order their edges appear in S. We represent S as an ordered multi-set of vertices $\mathcal{L}$ by placing in $\mathcal{L}$ the endpoints of the edges of P_i, $i = 1, \ldots, k$, following the order they appear in P_i and the ordering of $\mathcal{P}(S)$. It is clear that some vertices will appear more than once in $\mathcal{L}$. When a vertex subset $V' \subseteq V$ realizes in second-stage, `DFS` discards from $\mathcal{L}$ all copies of vertices not present in V'. This causes S to break down in components. Let S'' be the surviving edge subset of S in $G[V'] = G'(V', E')$. Then, `DFS` traces the surviving portion $\mathcal{L}'$ of $\mathcal{L}$ and, for any two consecutive vertices v_i and v_j in $\mathcal{L}'$, if v_i is not connected to v_j and $i < j$, then edge (v_i, v_j) is inserted in S''. This process causes reformation of all edge-disjoint paths in $\mathcal{P}(S)$ and results into a feasible Steiner tree S' for G'.

Let us note that the hypothesis that the input graph G is complete is a sine qua non condition for the existence of feasible solutions in any of its induced subgraphs. Indeed, if G is not complete there may exist subsets $V' \subseteq V$ inducing non connected subgraphs in which patching of the components of S'' is impossible.

Example 1. Let as give an example of the functionality of the modification strategy `DFS` described just above. Consider the tree S shown in Figure 1(a). The numbers on vertices stem from the DFS visitation ordering. Notice that label 1 is assigned to a leaf vertex of S. The

DFS ordering partitions S into the following maximal edge-disjoint paths: $P_1 = \{1, 2, 3, 4\}$, $P_2 = \{2, 5, 6\}$, $P_3 = \{2, 7, 8\}$, $P_4 = \{7, 9, 10\}$.



(a) The anticipatory first-stage tree (b) The modified second-stage tree

Figure 1: Functionality of the modification strategy.

Then $\mathcal{P}(S) = \{P_1, P_2, P_3, P_4\}$ and $\mathcal{L} = \{1, 2, 3, 4, 2, 5, 6, 2, 7, 8, 7, 9, 10\}$. Suppose that in second stage the realized vertex set is $V' = V \setminus \{2, 7\}$. Then DFS discards all copies of 2 and 7 from $\mathcal{L}$, thus producing $\mathcal{L}' = \{1, 3, 4, 5, 6, 8, 9, 10\}$. Subsequently, it traces the ordered $\mathcal{L}$ and adds edges as described previously. The result is shown in Figure 1(b). ∎

It can be immediately seen that the complexity of DFS is polynomial with n, the order of the input graph.

3.1 Complexity results for PROBABILISTIC STEINER TREE(DFS)

In what follows in this section, we first formulate the objective function associated with PROBABILISTIC STEINER TREE(DFS) and next we characterize the complexity of computing the anticipatory solution minimizing it. Finally, we study the complexity of "exploiting" the remainders of an anticipatory solution in order to get an optimal second-stage solution.

Proposition 1. PROBABILISTIC STEINER TREE*(DFS) is in* ***NPO****.*

Proof. Recall that **NPO**, the class of the optimization problems that have their decision counterparts in **NP**. For proving the proposition, it suffices to prove that the objective function $E(G, S, \texttt{DFS})$ of PROBABILISTIC STEINER TREE(DFS) is computable in polynomial time.

Let $G(V, E)$ be an instance of PROBABILISTIC STEINER TREE(DFS) and S be an anticipatory Steiner tree on G. Suppose that a set V' inducing a (complete) subgraph $G' = G[V']$ of G is presented for optimization. Assume that S is represented by the set of its edges. Assume finally that S' is the Steiner tree computed as described before. Let $V(S)$ be the set of vertices of S. Denote by $V'' = V(S')$ the subset of V' consisting of the vertices of S' and by E'' the edge-set of $G[V'']$. Let $\mathcal{L}$ be the DFS encoding performed over the anticipatory Steiner tree S by the modification strategy DFS, given as an ordered list of vertices (in the order decided by the DFS) and let $[v_i, v_j]_{\mathcal{L}}$ be the non-empty sublist of $\mathcal{L}$ starting at v_i and ending at v_j (notice that since $\mathcal{L}$ corresponds to a DFS ordering, $i < j$) not including neither v_i nor v_j (it is assumed that if $[v_i, v_j]_{\mathcal{L}} = \emptyset$, or if $i > j$, then $\prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l) = 0$).

We will prove that the objective value (functional) of S is expressed by:

$$E(G, S, \texttt{DFS}) = \sum_{(v_i, v_j) \in S} w(v_i, v_j)\, p_i p_j + \sum_{(v_i, v_j) \in E'' \setminus S} w(v_i, v_j)\, p_i p_j \prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l)$$

The expression for $E(G, S, \texttt{DFS})$ consists of two terms, the former expressing the expected cost of surviving edges of the anticipatory tree S, and the latter expressing the expected cost of edges added by the modification strategy DFS.

For the first term it is straightforward to see that an edge $(v_i, v_j) \in S$ survives if both its endpoints survive, an event occurring with probability $p_i p_j$ (since the two events of survival of each vertex are independent). Hence the expected cost of surviving edges in S is as claimed by the first term of the given expression for $E(G, S, \texttt{DFS})$.

The second term stems by inspection of the functionality of the modification strategy DFS. When the actual second-stage graph realizes, vertices missing from the graph are dropped from the initial DFS encoding $\mathcal{L}$, thus producing an encoding $\mathcal{L}' \subseteq \mathcal{L}$. Then, DFS scans $\mathcal{L}'$ and for any pair of consecutive vertices v_i and v_j it connects them by adding (v_i, v_j) if the three following conditions are satisfied:

- v_i, v_j are actually in $\mathcal{L}'$;
- $i < j$ (by the DFS-produced labelling);
- v_i is not connected to v_j.

Clearly two vertices $v_i, v_j \in \mathcal{L}$ are also in $\mathcal{L}'$ with probability $p_i p_j$. Moreover, an edge (v_i, v_j) is added to the (new) solution S', if v_i is not connected to v_j in second stage. This means (since v_i and v_j are consecutive in $\mathcal{L}'$) that all vertices that existed between v_i and v_j in $\mathcal{L}$ have been dropped from $\mathcal{L}'$. This happens with probability $\prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l)$. Furthermore, neither v_i nor v_j should also appear as intermediates within $[v_i, v_j]_{\mathcal{L}}$ in the original (first-stage) list encoding $\mathcal{L}$, otherwise (since all intermediate vertices are missing from $\mathcal{L}'$), DFS would not encounter them during its scan of $\mathcal{L}'$. Finally, $[v_i, v_j]_{\mathcal{L}}$ should not be empty, otherwise this would entail a surviving edge between v_i and v_j (recall that $i < j$ in the encoding considered), rendering these vertices connected in second-stage. Therefore, the expected cost of added edges is as expressed by the second term of the given expression.

It is easy to see that computation of a single term in the second sum of the functional requires $O(n)$ computations (at most $O(n)$ multiplications). Since we may do this for at most $O(n^2)$ times (the edges in E), it follows that the whole complexity of functional's computation is polynomial (in $O(n^3)$). So, PROBABILISTIC STEINER TREE(DFS) belongs to **NPO**. ∎

Unfortunately, Proposition 1 does not derive a compact characterization for the optimal anticipatory solution for PROBABILISTIC STEINER TREE(DFS). For instance, in [23, 24] probabilistic models as the one used here are studied for MAX INDEPENDENT SET and MIN VERTEX COVER, respectively, under a modification strategy consisting of taking the restriction of an anticipatory solution as solution for the second-stage graph. Under such a strategy, it is shown that an optimal anticipatory solution in a graph is the solution optimizing the total weight of an independent set, or a vertex cover, where the weight of a vertex v_i is either its own probability p_i, if the graph is unweighted or the product $p_i w_i$ if the graph is weighted and the weight of v_i is w_i. Here, the form of the functional provided by Proposition 1 does not imply solution of, say, some well-defined particular version of STEINER TREE (where the weight of an edge could be its initial weight multiplied by the probabilities of its endpoints), or something else of the same order as could be the case if the second term in $E(G, S, \texttt{DFS})$ did not exist. This is due to this second term where the "modified weights" assigned to the edges of G depend on the structure of the anticipatory solution chosen and of the present subgraph of G.

Proposition 2. PROBABILISTIC STEINER TREE*(DFS) is* ***NP****-hard.*

Proof. The inclusion of (the decision version of) PROBABILISTIC STEINER TREE(DFS) in **NP** follows from Proposition 1. Then, just remark that setting occurrence-probability 1 for any vertex

of the input graph, the objective function of PROBABILISTIC STEINER TREE(DFS) coincides with the one of classical STEINER TREE. Hence, the **NP**-hardness of the former in immediately concluded. ∎

We now prove a second hardness result claiming that it is **NP**-hard to polynomially modify the remainders of an anticipatory solution in order to get an optimal second-stage solution for any second-stage graph.

Proposition 3. *The problem of modifying the remainders of an arbitrary anticipatory solution towards optimizing the expected second-stage weight is **NP**-hard, for every modification strategy, even in the case that the anticipatory solution is optimum and the first-stage complete graph has edge-weights in* $\{1, 2\}$.

Proof. The reduction is from STEINER TREE in complete graphs with edge-weights 1 and 2. Consider a complete graph $H(V_H, E_H)$ with edge-weights in $\{1, 2\}$ and an arbitrary subset of vertices $T_H \subseteq V_H$, the terminals that have to be spanned by a minimum-weight tree $\hat{S}$.

We will build a probabilistic Steiner tree instance out of H. Extend H into a complete graph $G(V, E)$ with $V = V_H \cup \{v, x, y\}$ and E being the natural extension of E_H with additional edges so that G is complete. We set $w(v, u) = 1$ for every $u \in V_H \cup \{x, y\}$, $w(x, u) = 2$ for every $u \in V_H \cup \{y\}$ and $w(y, u) = 2$ for every $u \in V_H \cup \{x\}$. Finally, we extend the set of terminals T_H into $T = T_H \cup \{x, y\}$. Let all vertices of G be present with probability 1 apart from v for which the presence-probability is p, for some $p \in (0, 1)$. We so have a probabilistic Steiner tree instance and suppose that we are given an anticipatory solution $S^\star$ which is optimum for G. Notice that this solution is a star graph centered at v, with each $u \in T$ connected to v through an edge (v, u) of weight 1. The weight of this solution is $|T|$, while every other feasible solution in G has weight strictly greater than $|T|$. If $\mathtt{M}$ is a modification strategy, S' is the modified solution in second stage produced by $\mathtt{M}$, and $m(G', S')$ denotes the weight of S' on G' (the materialized second-stage subgraph of G), then we want $\mathtt{M}$ to be such so that the following expression is minimized:

$$E\left(G, S^\star, \mathtt{M}\right) = (1 - p)m\left(G', S'\right) + pw\left(S^\star\right) \tag{4}$$

Let us assume that a graph G (instance of PROBABILISTIC STEINER TREE) as the one described just above is given together with an optimal anticipatory Steiner tree S^* of G. Assume also that there exists a polynomial time algorithm that, if v is present, it returns S^* itself, while, if v is not present, it modifies S^* in order to return a Steiner tree S' for G' (that is G minus v and its incident edges) in such a way that (4) is minimized. Note that in order that this happens, its first term and in particular $m(G', S')$ has to be minimum. Notice also that, in the case of absence of v from G, the anticipatory solution $S^\star$ is completely destroyed. Hence, in second-stage every modification strategy $\mathtt{M}$ will have to reconnect the terminals of T from scratch. The cases that can occur for S' (that as it has just been mentioned minimizes $m(G', S')$) are the four ones illustrated in Figure 2, where a triangle denotes a tree spanning some terminals in H.

If S' is as in Figures 2(a) or 2(b) (let us refer to Figure 2(a), the case of Figure 2(b) being completely analogous), i.e., if x is linked to y that is linked to some vertex of a Steiner tree S of H (both of these edges have, by construction, weight 2) then, obviously, S is an minimum-weight Steiner tree for H, found in polynomial time since S' did so.

Suppose now that S' is as in Figure 2(c), i.e., both x and y are linked to two trees S_1 and S_2 of H, respectively, and these trees are also linked by an edge $(i, j) \in E_H$, then, obviously, $S = S_1 \cup S_2 \cup \{(i, j)\}$ is a Steiner tree of H and on the hypothesis that $m(G', S')$ is minimum so is the cost of S for H.

Finally, suppose that S' is as in Figure 2(d), i.e., x is linked to some trees $S_{x_1}, S_{x_2}, \ldots, S_{x_k}$, vertex y is linked to some trees $S_{y_1}, S_{y_2}, \ldots, S_{y_l}$ of H and x and y are also linked by edge (x, y).

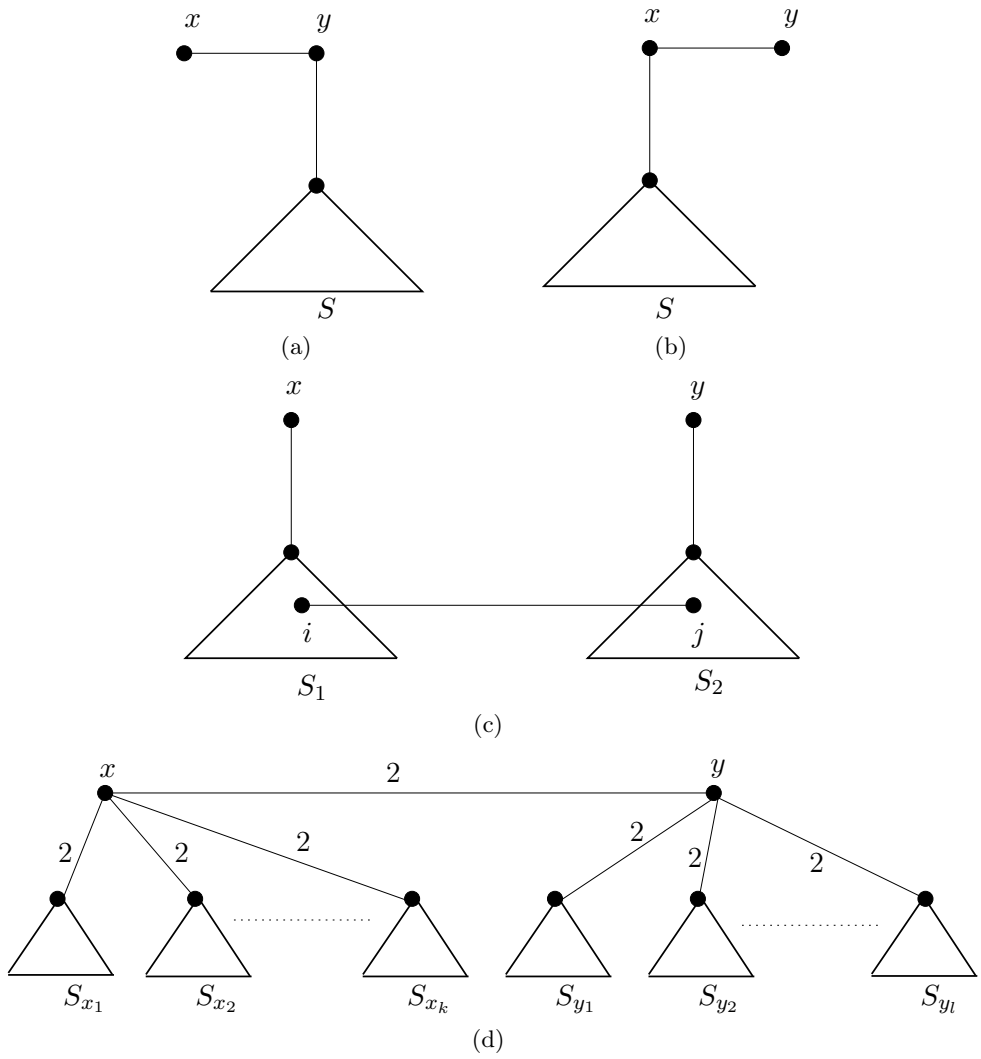


Figure 2: The four possible cases for S'.

Then it is easy to see that S' can be transformed, as in Figure 3, into a Steiner tree S'' of G' with at most the same cost by keeping, say, the edge linking y to S_{y_l}, then linking: S_{y_i} to $S_{y_{i-1}}$ for $i = l$ down to 2, S_{y_1} to S_{x_k} and S_{x_j} to $S_{x_{j-1}}$ for $j = k$ down to 2. The cost of the so obtained Steiner tree S'' is not greater than the one of S' since any edge incident to x or to y has weight 2 and edge-weights in E_H are in $\{1, 2\}$. But then this is exactly the case of Figure 2(a); hence a minimum-cost Steiner tree of H is polynomially built.

So, on the hypothesis that the problem of modifying the remainders of an arbitrary anticipatory solution towards optimizing the expected second-stage weight is polynomial, the tractability of Steiner tree on complete graphs with edge weights 1 and 2 can be derived, a contradiction with the **NP**-hardness of this latter problem. Henceforth, the former one is also **NP**-hard. ∎

3.2 On the approximation of PROBABILISTIC STEINER TREE(DFS)

We now turn to study approximation of PROBABILISTIC STEINER TREE(DFS) for several types of anticipatory solutions. We first show an easy though useful result linking, for any instance G of PROBABILISTIC STEINER TREE, $E^*(G)$ and $\mathrm{opt}(G)$, where $\mathrm{opt}(G)$ is the optimal STEINER TREE-

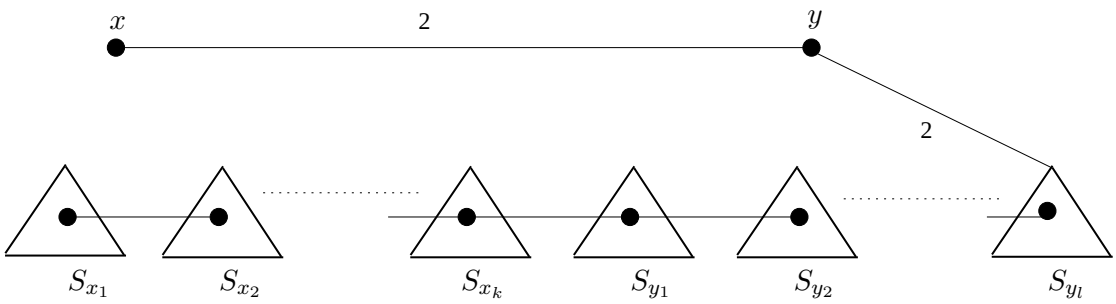


Figure 3: A Steiner tree S'' of total weight at most equal to the cost of S'.

value in G (i.e., the value of an optimal Steiner tree in G by ignoring the vertex-probabilities).

Lemma 1. $E^*(G) \geqslant \mathrm{opt}(G)$.

Proof. Since T is present in any of the sets $V' \subseteq V$ realized in the second stage, an optimal Steiner tree of G has value smaller than, or equal to, the value of an optimal Steiner tree of any second-stage induced subgraph G' of G, i.e., $\mathrm{opt}(G) \leqslant \mathrm{opt}(G')$, for any $G' \subseteq G$. Using it in (2), we get:

$$E^*(G) = \sum_{V' \subseteq V} \Pr\left[V'\right] \mathrm{opt}\left(G'\right) \geqslant \sum_{V' \subseteq V} \Pr\left[V'\right] \mathrm{opt}\left(G\right) = \mathrm{opt}(G) \sum_{V' \subseteq V} \Pr\left[V'\right] = \mathrm{opt}(G) \qquad (5)$$

which concludes the proof. ∎

Based upon Lemma 1 we show another preliminary result linking approximation of (deterministic) STEINER TREE to the reoptimization process.

Proposition 4. *Any ρ-approximate solution for* STEINER TREE *is also ρ-approximate to the functional of reoptimization.*

Proof. Consider a complete edge-weighted graph $G(V, E, \vec{w})$ together with a terminal set $T \subseteq V$ and a vertex-probability vector Pr satisfying $p_i = 1$, for any $v_i \in T$. Let $S \subseteq E$ be a ρ-approximate solution for STEINER TREE in G, i.e.:

$$m(G, S) \leqslant \rho\, \mathrm{opt}(G) \qquad (6)$$

Combination of (6) and (5) yields: $m(G, S)/E^*(G) \leqslant \rho$, as claimed. ∎

Let us note that the results of Lemma 1 and Proposition 4 remain valid for STEINER TREE also when dealing with non-metric or/and non-complete graphs.

3.2.1 Using a minimum spanning tree of $G[T]$ as anticipatory solution

Assume that the classical minimum-cost spanning tree algorithm in $G[T]$ is used to compute an anticipatory solution. Recall that the subgraph induced by the terminals is always present in the second stage.

Proposition 5. *A minimum-cost spanning tree over the subgraph of G induced by T is a* 2*-approximation for* PROBABILISTIC STEINER TREE *regardless of modification strategy chosen.*

Proof. Let H be as G with modified edge-weights (i.e., edge (v_i, v_j) has in H weight $w'(v_i, v_j) = p_i p_j w(v_i, v_j)$). Thanks to the omnipresence of the vertices of T, S remains feasible in second

stage, i.e., no completion is needed. Then, for any modification strategy $\mathtt{M}$, since $\mathtt{M}$ will be never applied in second stage:

$$E(G, S, \mathtt{M}) = m(H, S) = \sum_{(v_i, v_j) \in S} w(v_i, v_j)\, p_i p_j \tag{7}$$

On the other hand, since edge-weights in G are at least as large as the corresponding weights in H, the following holds (also using Lemma 1):

$$E^*(G) \geqslant \mathrm{opt}(G) \geqslant \mathrm{opt}(H) \tag{8}$$

Putting (7) and (8) together and taking into account that S is a 2-approximation for STEINER TREE in H (see Section 2), the result claimed is immediately derived. ∎

3.2.2 More general anticipatory solutions

By the observation made in the beginning of the proof of Proposition 5, a minimum-cost spanning tree S over $G[T]$ (or $H[T]$) is a very particular solution since it is always feasible for second stage. We so turn to study anticipatory solutions that have not a priori such property and could contain also non-terminal vertices of G, i.e., vertices that have a presence-probability less than 1. Let us note that such solutions can be better than a minimum-cost spanning tree S over $G[T]$. This is, for instance, the case of an anticipatory solution computed by [29].

Consider an anticipatory Steiner tree S (no matter how it is produced). Then, the following facts, expressed by Lemmata 2 and 3, hold on any metric graph G. In what follows, we denote by $[v_i, v_j]_S$ the edge-set of a path over a tree S; the vertex subscripts i and j denote the order of visitation of the tree's vertices by $\mathtt{DFS}$.

Lemma 2. *For any three edges (v_i, v_j), (v_k, v_l), (v_q, v_r) inserted by $\mathtt{DFS}$ to S'' the paths $[v_i, v_j]_S$, $[v_k, v_l]_S$ and $[v_q, v_r]_S$ do not share an edge in common.*

Proof. Without loss of generality, assume that $i < j$, $k < l$, $q < r$ and that the corresponding pairs of vertices were found in order $\langle v_i, v_j\rangle$, $\langle v_k, v_l\rangle$, $\langle q, r\rangle$ during the scan of $\mathcal{L}'$. Then, $j < l < r$ also holds. If the corresponding paths intersect on at least one common edge, say (v_s, v_t), it must be that $s, t \leqslant j$, because all edges of $[v_i, v_j]_S$ must have been traversed at least once after visitation of i and right before visitation of j. Thus, $s, t < l$ and $s, t < r$ also holds. In this case, edge (v_s, v_t) was scanned at least three times during the depth-first traversal of the anticipatory tree S: at least once right before visitation of each of the three vertices v_j, v_l and v_r. But this contradicts the fact that a depth-first traversal of a graph scans each edge exactly twice. ∎

From Lemma 2, one can immediately conclude that *any edge of the anticipatory tree S can be shared by at most two paths $[v, u]_S$ that have entailed insertion of edges for completion of S''.*

Lemma 3. *Consider two edges (v_i, v_j) and (v_k, v_l) added by $\mathtt{DFS}$ to S'' during construction of S'. For each edge $(v_s, v_t) \in [v_i, v_j]_S \cap [v_k, v_l]_S$, $(v_s, v_t) \notin S''$.*

Proof. Suppose, a contrario, that there exists $(v_s, v_t) \in [v_i, v_j]_S \cap [v_k, v_l]_S$ with $(v_s, v_t) \in S''$. Without loss of generality, suppose that $i < j$, $k < l$ and that the modification strategy found the pair $\langle v_i, v_j\rangle$ before the pair $\langle v_k, v_l\rangle$ while scanning the list $\mathcal{L}'$. Then, it must be $i < j$, $k < l$ and $j < l$. Since $(v_s, v_t) \in [v_i, v_j]_S \cap [v_k, v_l]_S$, it must be that $s, t \leqslant j$ and, consequently, $s, t < l$. Also, it must hold either that $s, t > k$ or that $s, t > i$ because, otherwise, it must be $s, t < i$ and, given that $s, t \leqslant j$ also, we should conclude that edge (v_s, v_t) has been scanned during depth-first traversal already twice, once right before visitation of v_i and once right before visitation of v_j. In this case, it cannot have been scanned also right before visitation of v_l. But, if either $s, t > k$

or $s, t > i$ hold, then scan of $\mathcal{L}'$ could not have found the vertices v_k, v_l, or the vertices v_i, v_j, respectively, in this order, a contradiction. ∎

Denote by $e_1, e_2, \ldots$, the edges added in S'' during the second stage. As mentioned several times above, these edges correspond to paths of S; denote them by $P_1, P_2, \ldots$ The total weight of S' is bounded by a sum of the weights of $P_1, P_2, \ldots$, plus the weight of edges in S''. It turns out by the previous Lemmata 2 and 3 that each edge of S contributes to the bounding quantity of $w(S')$ at most twice (either by appearing in two paths P_i and P_j and not in S'', or by appearing in S'' and in one path P_i); hence:

$$m\left(G', S'\right) = w\left(S'\right) \leqslant 2w(S) = 2m(G, S) \tag{9}$$

Figure 4, where vertices are named by their DFS ordering, gives an example illustrating what has been just discussed. The anticipatory Steiner tree S is the tree with thin lines. We suppose that white (black) vertices are absent (present) in second stage. Hence, the edges added in this stage in order to reconnect S are edges $(1, 5)$, $(5, 7)$, $(7, 8)$, $(15, 18)$ and $(18, 19)$ (thick lines in Figure 4). Since we assume that the input graph is metric, we have the following for these "new" edges:

$$\begin{aligned}
w(1,5) &\leqslant w(1,2) + w(2,3) + w(3,4) + w(4,5) \\
w(5,7) &\leqslant w(5,6) + w(6,7) \\
w(7,8) &\leqslant w(6,7) + w(6,8) \\
w(15,18) &\leqslant w(11,15) + w(9,11) + w(5,9) + w(4,5) \\
&\quad + w(3,4) + w(3,16) + w(16,17) + w(17,18)
\end{aligned}$$

So, in the so reconstructed tree S', edges $(1, 2)$ and $(2, 3)$ are counted once, due to the insertion of edge $(1, 5)$. Edges $(3, 4)$ and $(4, 5)$ count twice (due to the insertions of $(1, 5)$ and of $(15, 18)$. Edges $(5, 6)$ and $(6, 7)$ count once (for edge $(5, 7)$). Edges $(3, 16)$, $(16, 17)$, $(5, 9)$, $(9, 11)$, and $(11, 15)$ count once, for edge $(15, 18)$. Edge $(17, 18)$ counts twice, once for edge $(15, 18)$ and once for $(18, 19)$. Finally, edge $(17, 19)$ counts once, for edge $(18, 19)$. Edges $(5, 9)$, $(9, 11)$ and $(11, 15)$ count once more in S''. Finally, edges $(9, 10)$, $(11, 12)$, $(11, 14)$ and $(12, 13)$ count only once, i.e., only in S''.

Consider using an arbitrary ρ-approximation algorithm for obtaining a feasible Steiner tree S with cost no more than ρ times the cost of an optimal Steiner tree of G. Then, the following theorem holds.

Theorem 1. *Any ρ-approximation algorithm for* STEINER TREE *yields an anticipatory solution resulting in a 2ρ-approximation for* PROBABILISTIC STEINER TREE*(DFS).*

Proof. Based upon Lemmata 2 and 3, the proof is immediate. Taking into account (4) and that $\mathrm{opt}(G') \geqslant \mathrm{opt}(G)$, we get from (1) using (9):

$$\begin{aligned}
E(G, S, \mathtt{DFS}) &= \sum_{V' \subseteq V} \Pr\left[V'\right] m\left(G', S'\right) \leqslant \sum_{V' \subseteq V} \Pr\left[V'\right] 2m(G, S) \\
&\leqslant 2 \sum_{V' \subseteq V} \Pr\left[V'\right] m(G, S) \leqslant 2\rho \sum_{V' \subseteq V} \Pr\left[V'\right] \mathrm{opt}(G) \\
&\leqslant 2\rho \sum_{V' \subseteq V} \Pr\left[V'\right] \mathrm{opt}\left(G'\right) \leqslant 2\rho E^*(G)
\end{aligned}$$

and the proof of the theorem is complete. ∎

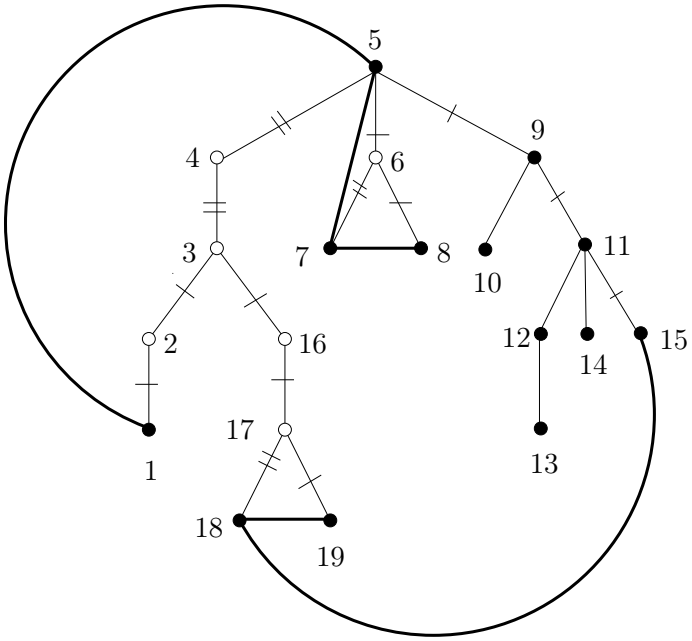


Figure 4: The total weight of a second-stage tree is no more than twice the cost of an anticipatory (first-stage) tree.

Theorem 2. *Ratio 2ρ for* PROBABILISTIC STEINER TREE*(DFS) is tight even when edge costs are 1 and 2 and an optimal* STEINER TREE*-solution is used as anticipatory solution.*

Proof. Consider a complete graph K_{n+1} on $n+1$ vertices numbered by $1, 2, \ldots, n+1$. Assume w.l.o.g. that $n \geqslant 6$ is even and that edge $(1,3)$ and edges $(i, i+1)$, $i = 3, \ldots, n-1$ have cost 2, while the other edges of K_{n+1} have cost 1. Assume, finally, that only vertex 2 is non-terminal and its presence probability is p_2. It is easy to see that, in this graph, $\mathrm{opt}(K_{n+1}) = n+1$ and such a tree is realized in several ways and, in particular, by a star S with center 2, or by a path linking somehow the terminals (for example, using edges $(1,3)$, $(i, i+2)$, for i odd from 1 to $n+1$, edge $(n-2, n+1)$, edges $(j, j-2)$, for j even going from $n-2$ down to 4 and, finally, edge $(4, n)$). Obviously,

$$E^*\left(K_{n+1}\right) = p_2(n+1) + (1-p_2)\,(n+1) = n+1 \tag{10}$$

Assume now that the anticipatory solution computed is just S (of cost $n+1$) and that the DFS ordering of S has produced the original numbering of K_{n+1}, i.e., 1 is the leftmost leaf, 2 the stars'center and $3, \ldots n+1$ the rest of leaves. If 2 is absent, then the completion of the tree will produce the path $(1, 3, 4, \ldots, n+1)$ with cost $2n$. So, the value of $E(K_{n+1}, S, \mathtt{DFS})$ will be:

$$E\left(K_{n+1}, S, \mathtt{DFS}\right) = p_2(n+1) + (1-p_2)\,2n = 2n - p_2 n + p_2 \tag{11}$$

In Figure 5 an illustration of the discussion just above is provided for $n = 6$. The thick edges of K_7 in Figure 5(a) are the ones with cost 2. It is assumed that vertices $1, 3, 4, 5, 6, 7$ are terminals. In Figure 5(b), an optimal Steiner tree of K_7 is shown using non-terminal vertex 2. It is assumed that this tree is also the anticipatory solution. Its vertices' numbers represent also their DFS ordering. In Figure 5(c) is shown the DFS completion of S when 2 is absent. Finally, in Figure 5(d), the optimal Steiner tree of K_7 using only terminals built as described above is shown.

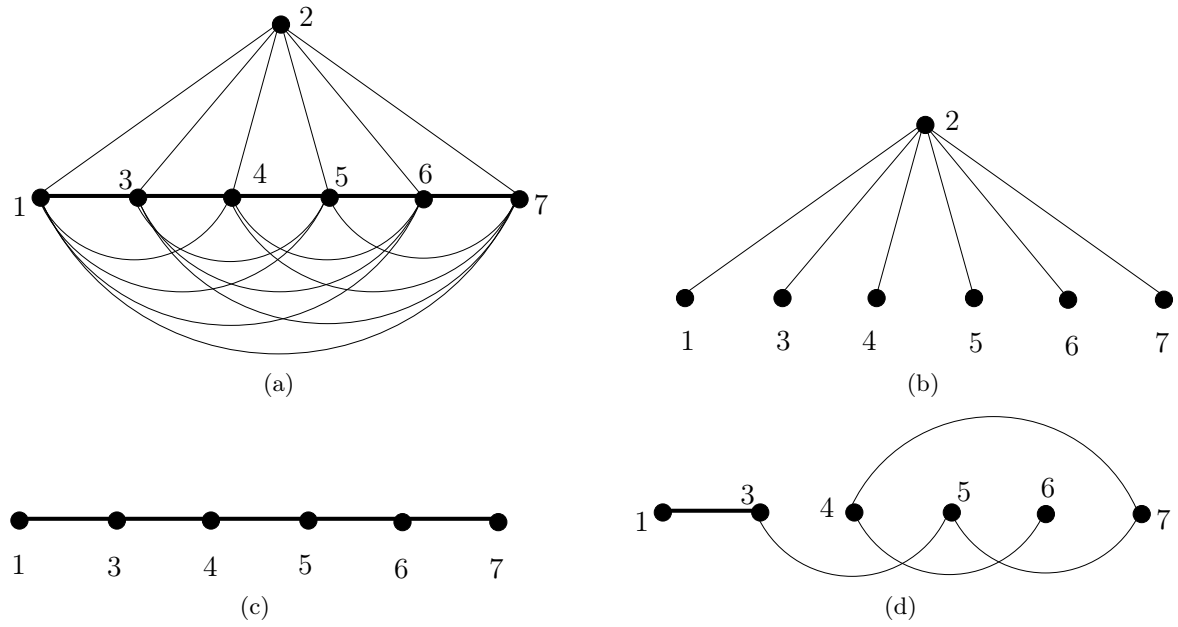


Figure 5: On the tightness of ratio 2ρ.

Dividing (10) by (11) we get a ratio that for small enough values of p_2 tends to 2. ∎

By Theorem 1 if for example the 1.55-approximation algorithm of [29] is used for constructing an anticipatory solution, a 3.1-approximation for PROBABILISTIC STEINER TREE(DFS) is incurred. This result is obviously dominated by Proposition 5. However, Theorem 1 provides a more general structural result linking the approximation of STEINER TREE to the one of PROBABILISTIC STEINER TREE(DFS) and carries over more general solution structures. This, to our opinion, has its own interest. But, if one restricts her/himself to anticipatory solution that are optimal for STEINER TREE in the input-graph, then Theorems 1 and 2 imply a tight approximation ratio 2 for PROBABILISTIC STEINER TREE(DFS). In this case the result incurred is also of practical usefulness since it encompasses more general structures of anticipatory solutions and the approximation ratio yielded is comparable to the one claimed by Proposition 5.

4 A second probabilistic model: PROBABILISTIC STEINER TREE under re-approximation strategy

We consider in this section a different modification strategy for PROBABILISTIC STEINER TREE, to which we refer to as *re-approximation strategy* (REAPX) (following our convention we are going to study PROBABILISTIC STEINER TREE(REAPX)).

Let S be an anticipatory first-stage solution for the Steiner tree problem over the initial weighted graph $G(V, E, \vec{w})$, produced by a ρ-approximation algorithm. Let $V' \subseteq V$ be the realized vertex subset of G in second-stage, and $S'' \subseteq S$ be the surviving portion of the anticipatory Steiner tree S. Obviously, S'' is a forest; denote by C_i, $i = 1, \ldots, q$ its trees (any of its trees can also be a single vertex) and assume w.l.o.g. that any of the first t C_i's, say $C_1, \ldots, C_t$, $t \leqslant q$, contains at least one terminal vertex of G, while any of the $q - t + 1$ last C_i's contains only non-terminal vertices of G.

The REAPX strategy first constructs S'' by discarding absent vertices from S. Then, it replaces any tree C_i, $i = 1, \ldots, t$, by a new terminal vertex u_i and discards any C_i, $i = t+1, \ldots, q$ from V'. Finally, it keeps the rest of vertices in V' (i.e., the non-terminal vertices of V' that have not been

used in S) unchanged. Let us denote by N'' this last set of vertices. It so produces a new vertex-set V'' including t terminals $u_1, \ldots, u_t$ and the non-terminals of N''. Let G'' be the complete graph on V'' and E'' be its edge-set. Any edge $(v_i, v_j) \in E''$ is weighted as follows:

- if $v_i, v_j \in N''$, then $w(v_i, v_j)$ remains as it was in G;
- if both v_i and v_j are some terminals, say u_k and u_l, then $w(v_i, v_j) = \min\{w(v, u) : v \in C_k, u \in C_l\}$;
- analogously, if one of v_i, v_j, say v_i belongs to N'' and the other one, v_j, is some terminal u_k, then $w(v_i, v_j) = \min\{w(v_i, v) : v \in C_k\}$.

`REAPX` runs a ρ'-approximation algorithm `A` for STEINER TREE in G'' to take a tree spanning the terminal set U of G''. Denote by $\bar{S}$ this tree and assume that it is represented as a list of edges. Then, it "unfolds" the vertices of U obtaining so a tree who edge set is $S' = S'' \cup \bar{S}$ that is a tree spanning the initial terminals of V that have survived in V', i.e., a Steiner tree for $G' = G[V']$.

As an example, consider a K_7 and assume, for simplicity, that the weights on its edges are distances in Euclidean space. Suppose that vertices 1, 2, 3, 5 and 6 are located on a line and that $w(1,2) = 1$, $w(2,3) = 2$, $w(3,5) = 3$ and $w(5,6) = 4$. Suppose also that $w(3,4) = w(3,7) = 1$. Assume, finally that vertices 1, 2, 4, 5 and 6 are terminals, while 3 and 7 are not and that the anticipatory tree S of Figure 6(a) has been computed by some ρ-approximation algorithm. Assume that in the second-stage graph G', only vertex 3 is absent (and vertex 7 that does not make part of S survives. Figure 6(b) shows the connected components C_1, C_2 and C_3 of S that survive in G' together with vertex 7 that also survives. Any of the components C_1, C_2 and C_3 contains at least one terminal. So, any of them is contracted into a new vertex u_i, $i = 1, 2, 3$ (recall that if one of them contained no terminal, it would be totally removed). These three vertices, together with vertex 7 are the vertices of the new graph G'' shown in Figure 6(c). Here, $w(u_1, u_2) = \sqrt{5}$ (the weight of edge $(2, 4)$), $w(u_1, u_3) = 5$ (due to edge $(2, 5)$), $w(u_1, u_4) = \sqrt{5}$ (edge $(2, 7)$), $w(u_2, u_3) = \sqrt{10}$ (edge $(4, 5)$), $w(u_2, u_4) = 2$ (edge $(4, 7)$) and $w(u_3, u_4) = \sqrt{10}$ (edge $(5, 7)$). In this new graph, some ρ'-approximation algorithm `A` will be called to compute a new Steiner tree. If, for instance this algorithm is for computing a minimum spanning tree on the the terminals (discussed in Section 3), then the computed Steiner tree for G'' should be as in Figure 6(d) and the "unfolding" should result to a Steiner tree S' for G' that should be as in Figure 6(e) with total cost $10 + \sqrt{5}$.

Noticeably, the `REAPX` strategy does not always incur an asymptotic advantage in comparison to re-evaluating a Steiner tree on G' from scratch, but it exploits remaining parts of an anticipatory Steiner tree, potentially offering some practical time-savings.

Obviously, PROBABILISTIC STEINER TREE(`REAPX`) is a collection of probabilistic variants for PROBABILISTIC STEINER TREE rather than a single one, any variant induced by the particular approximation algorithm `A` used. Furthermore, as it hopefully has become clear from what has been discussed in Section 3, the objective function of a probabilistic problem strongly depends on the modification strategy used. So, it is very unlikely that one could produce an expression for the functional of PROBABILISTIC STEINER TREE capturing any instantiation of `REAPX`.

Dealing with optimal STEINER TREE solutions on G'' (the graph constructed by trees-contraction with terminal vertex-set U) and G' (the finally realized subgraph of G), the following lemma links their optimal solution-values.

Lemma 4. $\mathrm{opt}(G'') \leqslant \mathrm{opt}(G')$.

Proof. Notice that a Steiner tree on G' spanning the initial terminal set T is also feasible for the terminal set U over G'', because for each $v \in U$, either $v \in T$, or v has emerged by contraction of

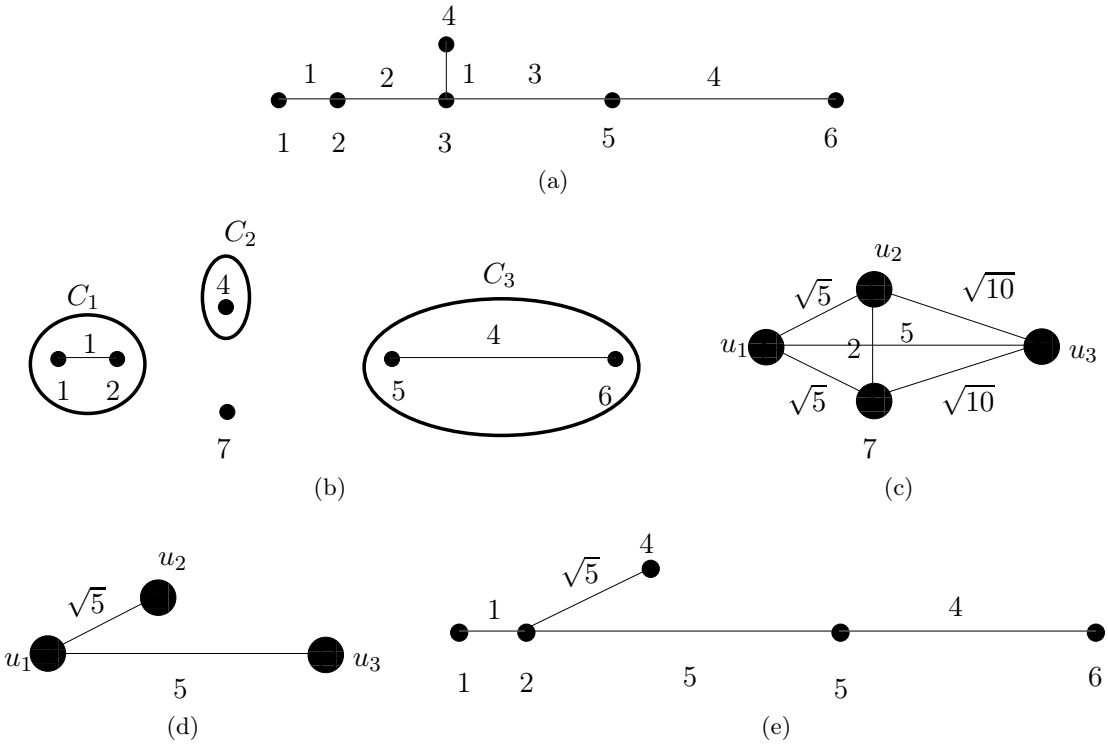


Figure 6: An example of application of REAPX, using a minimum spanning tree computation over the terminals.

a subset of vertices intersecting T. Since $\mathrm{opt}(G'')$ is the value of an optimal Steiner tree in G'', the result follows. ∎

Theorem 3. *Let* A *be a ρ'-approximation algorithm for* STEINER TREE *called by* REAPX *in its second stage. Then an anticipatory Steiner tree S of G that is a ρ-approximation for* STEINER TREE*, is a $\rho + \rho'$-approximation for* PROBABILISTIC STEINER TREE*(*REAPX*).*

Proof. Obviously, $w(S'') \leqslant w(S) = m(G, S) \leqslant \rho\,\mathrm{opt}(G)$. Furthermore, $\mathrm{opt}(G) \leqslant \mathrm{opt}(G')$, since G' lacks some edges of G (see also Lemma 1). Execution of the ρ'-approximation algorithm A on G'' returns an edge set $\bar{S}$ with total weight $w(\bar{S}) \leqslant \rho'\,\mathrm{opt}(G'') \leqslant \rho\,\mathrm{opt}(G')$, by Lemma 4. Then, the returned Steiner tree $S' = S'' \cup \bar{S}$ in G' has total weight:

$$w\left(S'' \cup \bar{S}\right) = m\left(G', S'\right) = w\left(S''\right) + w\left(\bar{S}\right) \leqslant w(S) + \rho'\,\mathrm{opt}\left(G''\right) \tag{12}$$

By the assumption on the approximation achieved by S, Lemma 4 and the fact that $\mathrm{opt}(G) \leqslant \mathrm{opt}(G')$, (12) becomes:

$$m\left(G', S'\right) \leqslant \left(\rho + \rho'\right)\mathrm{opt}\left(G'\right) \tag{13}$$

As previously, let $E(G, S, \mathtt{REAPX})$ denote the functional value incurred by the modification strategy REAPX for PROBABILISTIC STEINER TREE(REAPX). By (1) and (13) we get:

$$E(G, S, \mathtt{REAPX}) = \sum_{V' \subseteq V} Pr\left[V'\right] m\left(G', S'\right) \leqslant \left(\rho + \rho'\right) \sum_{V' \subseteq V} Pr\left[V'\right] \mathrm{opt}\left(G'\right) = \left(\rho + \rho'\right) E^*(G)$$

that proves the proposition. ∎

Note that, from Theorem 3, if S is computed optimally, then the ratio derived from Theorem 3 is $\rho' + 1$. This is a case where the REAPX strategy implies a practical advantage with respect to re-evaluation of a Steiner tree on the second-stage graph from scratch. Also, several other approximation results can be derived depending on the specification of A. For instance:

- if S is the solution computed by the algorithm in [29], then the ratios derived are 2.56 and 3.28, respectively;
- if S is the solution computed by the algorithm in [1], the respective ratios are 3.28 and 4.

To conclude the paper, another way for estimating the approximation quality of an anticipatory solution S for PROBABILISTIC STEINER TREE(M), for any modification strategy M, is by using the approximation ratio $E(G, S, \mathtt{M})/E(G, S^*, \mathtt{M})$ where, as defined in Section 1, S^* is an optimal anticipatory solution. As a final remark, by (3), $E^*(G)$ is a lower bound for $E(G, S^*, \mathtt{DFS})$. Henceforth, the results derived in both Sections 3 and 4 remain valid for this approximation ratio too.

5 Conclusions and further research directions

In this paper we have treated the PROBABILISTIC STEINER TREE problem under the framework of probabilistic combinatorial optimization. We have proposed a fast modification strategy (DFS) for reconstructing a second-stage tree and shown that problem of optimizing the expectation of the second-stage cost by selecting an appropriate first-stage (anticipatory) solution is in NPO under the proposed modification strategy. We have also given approximation results for the probabilistic problem associated with the DFS strategy. For instance, for metric edge-costs and an optimum anticipatory first-stage solution a 2 approximation of the optimum second-stage cost is obtained by DFS. Furthermore, we have given a family of modification strategies (REAPX), each member of which can be instantiated by usage of appropriate approximation algorithms. For the case of general edge costs and an optimum anticipatory first-stage solution, a 2.55 approximation of the optimum second-stage cost can be incurred by REAPX. We note however, that no member of REAPX family of modification strategies can obtain a ratio 2 (as it is the case of DFS) in metric graphs.

Let us note that many of the results of Section 3 (as, for example, Proposition 1, Theorem 1, if we run the DFS modification strategy for every tree of an anticipatory forest) can also be applied directly to the probabilistic Steiner forest problem.

Lowering the approximation ratio for the case of general edge-costs given an optimum anticipatory solution (possibly by devising a novel modification strategy), or refuting such a possibility is a matter of future work. We are also investigating the properties of a different probabilistic model for STEINER TREE, involving probabilistic terminal vertices when all other vertices of the graph are present with probability 1.

Acknowledgment. Many thanks to Cécile Murat for helpful discussions and comments.

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] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Math. Programming, Ser. A*, 90:263–272, 2001.

[3] D. J. Bertsimas. *Probabilistic combinatorial optimization problems.* Phd thesis, Operations Research Center, MIT, Cambridge Mass., USA, 1988.

[4] D. J. Bertsimas. The probabilistic minimum spanning tree problem. *Networks*, 20:245–275, 1990.

[5] J. Birge and F. Louveaux. *Introduction to stochastic programming.* Springer, Berlin, 1997.

[6] F. Della Croce, B. Escoffier, C. Murat, and V. Th. Paschos. A priori optimization for minimum graph-coloring in bipartite and split graphs. In O. Gervasi et al., editor, *Proc. International Conference on Computational Science and its Applications, ICCSA'05*, volume 3483 of *Lecture Notes in Computer Science*, pages 202–211. Springer-Verlag, 2005. Full version available at `http://www.lamsade.dauphine.fr/cahiers/PS/cahier218.ps`.

[7] G. W. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1951.

[8] V. G. Deĭneko and G. J. Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *Operations Research Letters.* To appear.

[9] K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh. How to pay, come what may: approximation algorithms for demand-robust covering problems. In *Proc. FOCS'05*, pages 367–378, 2005.

[10] L. Fleischer, J. Kœnemann, S. Leonardi, and G. Schæfer. Simple cost sharing schemes for multicommodity rent-or-buy and stochastic steiner tree. In *Proc. STOC'06*, pages 663–670, 2006.

[11] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Approximation algorithms for the steiner tree problem in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner trees in industries.* Kluwer Academic Publishers, 2000.

[12] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proc. STOC'04*, pages 417–426, 2004.

[13] A. Gupta, M. Pál, R. Ravi, and A. Sinha. What about wednesday? approximation algorithms for multistage stochastic optimization. In *Proc. Approximation Algorithms for Combinatorial Optimization Problems, APPROX'05*, Lecture Notes in Computer Science. Springer-Verlag, 2005.

[14] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *Proc. FOCS'04*, pages 218–227, 2004.

[15] N. Immorlica, D. R. Karger, M. Minkoff, and V. S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *Proc. Symposium on Discrete Algorithms, SODA'04*, pages 691–700, 2004.

[16] P. Jaillet. Probabilistic traveling salesman problem. Technical Report 185, Operations Research Center, MIT, Cambridge Mass., USA, 1985.

[17] P. Jaillet. Shortest path problems with node failures. *Networks*, 22:589–605, 1992.

[18] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications.* Kluwer Academic Publishers, Boston, 1997.

[19] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and Applications*, 10(1-2):19–34, 2005.

[20] R. Montemanni, L. M. Gambardella, and A. V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Oper. Res. Lett.*, 32:225–232, 2004.

[21] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios. Robust optimization of large-scale systems. *Oper. Res.*, 43(2):264–281, 1995.

[22] C. Murat and V. Th. Paschos. The probabilistic longest path problem. *Networks*, 33:207–219, 1999.

[23] C. Murat and V. Th. Paschos. A priori optimization for the probabilistic maximum independent set problem. *Theoret. Comput. Sci.*, 270:561–590, 2002. Preliminary version available at `http://www.lamsade.dauphine.fr/~paschos/documents/c166.pdf`.

[24] C. Murat and V. Th. Paschos. The probabilistic minimum vertex-covering problem. *Int. Trans. Opl Res.*, 9(1):19–32, 2002. Preliminary version available at `http://www.lamsade.dauphine.fr/~paschos/documents/c170.pdf`.

[25] C. Murat and V. Th. Paschos. On the probabilistic minimum coloring and minimum ***k***-coloring. *Discrete Appl. Math.*, 154:564–586, 2006.

[26] A. Prekopa. *Stochastic programming*. Kluwer Academic Publishers, The Netherlands, 1995.

[27] Stochastic programming community home page. Available at `http://stoprog.org`.

[28] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. In *Proc. International Conference on Integer Programming and Combinatorial Optimization, IPCO'04*, volume 3064 of *Lecture Notes in Computer Science*, pages 101–115. Springer-Verlag, 2004.

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

[30] D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proc. FOCS'04*, pages 228–237, 2004.

[31] C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage and multi-stage stochastic optimization. In *Proceedings of the International Conference on Algorithms for Optimization with Incomplete Information*, 2005.

[32] C. Swamy and D. B. Shmoys. Sampling-based approximation algorithms for multi-stage stochastic pptimization. In *Proc. FOCS'05*, pages 357–366, 2005.

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

[34] P. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.