Probabilistic Models for the STEINER TREE Problem

Vangelis Th. Paschos[1]* Orestis A. Telelis[2] †‡

Vassilis Zissimopoulos[3]§

[1]LAMSADE, CNRS UMR 7024, Université Paris-Dauphine, France
paschos@lamsade.dauphine.fr

[2]Computer Science Department, Aarhus University, Denmark
telelis@cs.au.dk

[3]Department of Informatics and Telecommunications, University of Athens, Greece
vassilis@di.uoa.gr

Abstract

We consider a probabilistic model for the Steiner Tree problem. Under this model the problem is defined in a 2-stage setting over a first-stage complete weighted graph having its vertices associated with a probability of presence (independently each from another) 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. Therefore, a well defined *modification strategy* is devised for modifying the remainders of a first-stage solution to render it second-stage feasible. The objective is to minimize the expected weight of the second-stage solution over the distribution of all possible second-stage materializable subgraphs of the input graph. We recognize two complementary computational problems in this setting, one being the a priori computation of first-stage decisions given a particular modification strategy, and the second being the cost-efficient modification of a first-stage feasible solution. We prove that both these problems are **NP**-hard for the Steiner Tree problem under this setting. We design and analyze probabilistically an efficient modification strategy, and derive tight approximation results for both aforementioned problems. We show that our techniques can be extended to the case of the more general Steiner Forest problem in the same probabilistic setting.

Keywords: *Steiner Tree, Forest, Approximation, Graph, Complexity*

1 Introduction

Given an edge-weighted graph $G(V, E)$, with positive edge weights $w : E \rightarrow \Re^+$, and a subset of vertices $T \subseteq V$ (called the "terminal" vertices) the Steiner Tree problem requires selecting a subset of edges $S \subseteq E$ of minimum total weight, that interconnect vertices of T (possibly spanning also vertices other that the ones in T). We study the Steiner Tree problem in a

*Part of this work was 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.

†The Center for Algorithmic Game Theory, funded by the Carlsberg Foundation, Denmark

‡Work partially carried out while the author was at IBISC CNRS FRE 3190, Université d'Evry Val d'Essonne, France.

§Author partially supported by the Greek Ministry of Education and Research under the project KAPODISTRIAS K.A. 70/4/5821.

2-stage robust optimization setting, where we are uncertain of the availability of vertices in $V \setminus T$, in that each such vertex is present with some probability independently of all other vertices (this has also been called *independent activation model* in the literature). Vertices of T are always present with probability 1. The input graph G is considered as a first-stage input, while in the second stage a vertex-subgraph G' of G materializes, in which a vertex v is present with the specified probability. Our objective is to take some a priori (first-stage) decisions regarding the layout of the tree, so as to be able to come up with a feasible tree for every possibly materializable subgraph $G'(V', E')$ of G with $V' \subseteq V$, and minimize the expected total weight of edges used over the distribution of all such subgraphs. We will mostly consider a priori decisions that constitute a feasible solution of the problem on the first-stage input graph. We refer to this solution as *anticipatory* solution. This is the probabilistic version of the Steiner Tree problem that we consider.

Note that an anticipatory first-stage feasible solution for G may become infeasible in G' (since certain vertices may be missing, along with incident edges). A brute-force way to cope with this problem is to pre-compute a feasible and approximate (or maybe optimum) solution for every possible subgraph of G that may materialize, and apply an appropriate solution when the subgraph actually appears. In principle there need not be a constraint on the computational effort applied for taking a priori decisions, as long as they can support a fast response strategy to the actually materialized data. In this light however, we require that such a response should be of strictly lower complexity compared to the a priori computational effort. A straightforward pattern for implementing this setting is for example to compute an optimum anticipatory solution over G, and if this solution is not feasible for the materialized subgraph G', use a polynomial-time approximation algorithm to obtain a completely different feasible solution for G'. On the other hand, a natural challenge is to design such an efficient response strategy for modifying a priori decisions (called *modification strategy*), that can be supported by polynomial-time computable a priori decisions. In our setting we recognize two complementary computational problems:

- **A priori optimization:** Given a fixed modification strategy, find an efficient way of taking a priori decisions, leading to optimization of the expected minimum weight.
- **Modification:** Given a feasible anticipatory solution, design a modification strategy that can produce second-stage feasible solutions optimizing the expected minimum weight.

The first problem requires taking into account the existence of a specific modification strategy when taking a priori decisions, so as to optimize the final (second-stage) outcome, while the second problem amounts to designing a modification strategy for optimizing the second-stage outcome, given "interesting" a priori decisions: such interesting decisions are feasible anticipatory solutions for the first-stage graph G that are of optimum or approximately optimum weight. Consideration of feasible anticipatory solutions is natural, especially when it is not known in advance whether a transition from first to second stage will actually occur and when. Therefore one might be interested that his/her operational design is for example optimal for as long as first-stage input data remain valid.

The problem model we consider finds natural application in networks, where uncertain availability of intermediate nodes requires fast adjustments of traffic forwarding, as for example in enhancing fault-tolerance in wireless multi-hop communication networks. Certain high-fidelity nodes in such networks are responsible for collecting messages from other nodes, and forwarding them through multiple hops to their destination. These high-fidelity nodes are expected to be always present, and their interconnection through a minimum weight tree (possibly spanning also other nodes) provides a virtual backbone to the network, functioning at low energy consumption (see e.g. [15] for such graph models regarding energy consumption). When intermediate nodes of the backbone fail, it is essential that the backbone is reconstructed efficiently to operate in low energy.

In this paper we show that both problems of *a priori optimization* and *modification* are **NP**-hard (Section 3). We design a modification strategy and derive an analytic expression of the expected value of second-stage solutions it produces (Section 4). Subsequently we derive non-trivial polynomial-time approximation results for metric graphs (Section 5) and generalize our results to the case of the more general Steiner Forest problem on 2-stage probabilistic metric graphs (Section 6). We show that the analysis of the proposed modification strategy is tight (Section 7). Finally we consider implementation of modification strategies by usage of known approximation algorithms (Section 8) and conclude. Followingly we discuss related work, and introduce formal problem definitions and notation (Section 2).

Related Work The Steiner Tree problem is a well known **NP**-hard (even in metric graphs) network design problem (see Garey & Johnson [8] problem [ND12]). It is a special case of the Steiner Forest problem, which requires selecting a subset of edges of minimum total weight, that connects simultaneously given pairs of vertices (when all pairs share a vertex in common, an instance of the Steiner Tree problem is obtained). The primal-dual algorithm by Agrawal et al. [1] (see also [9, 29]) achieves a factor 2 approximation of the optimum weight for both problems. For the Steiner Tree problem in particular, the algorithm described in [25] achieves factor 1.55 approximation in general weighted graphs, and 1.28 on metric graphs with edge weights $1, 2$. In metric graphs the simple heuristic of computing a minimum spanning tree on the subgraph induced by terminal vertices is a factor 2 approximation algorithm [29].

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 well developed being *Stochastic Programming* (see [5, 6, 22] for basics, [23] for latest news, bibliography, and related software and [10, 11, 12, 24, 26, 27, 28] for recent hardness results and approximation algorithms) and *Robust Discrete Optimization* (see [2, 7, 14, 16, 17] for details). These frameworks constitute a means of structuring uncertainty of input, 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.

Our work is mostly related to the framework of *Probabilistic Combinatorial Optimization*, introduced in [13, 3], where modification strategies as the one described previously are analyzed probabilistically, so that the expected value of their outcome can be computed efficiently (this ensures that the problem of taking a priori decisions for a particular strategy belongs to class **NPO**). Several network design problems have been treated in the probabilistic combinatorial optimization framework, including minimum coloring [21], maximum independent set [19], longest path [18], and minimum spanning tree [4]. Apart from probabilistic analysis of modification strategies, results in [21, 19] also include derivation of approximability properties.

2 Definitions and Preliminaries

We deal with a 2-stage stochastic optimization model involving a first-stage complete weighted graph $G(V, E)$ on n vertices, with edge weights given by a function $w : E \to \Re^+$. We are given a set of terminal vertices $T \subseteq V$, and each vertex $v \in V$ is associated to a real value in $[0,1]$ given by a function $\pi : V \to [0,1]$, with $\pi(v_i) = p_i$ and $\pi(v_i) = 1$ for all $v_i \in T$. We assume that in second stage a subgraph $G'(V', E')$ of G materializes as the outcome of n independent Bernoulli trials, one per vertex $v \in V$: $v \in V'$ with probability $\pi(v)$. Then $E' = \{(u,v) \in E | u \in V', v \in V'\}$.

Assume a subset of edges $S \subseteq E$ constituting *a priori* first-stage decisions (anticipatory solution), and let $S_1 \subseteq S \cap E'$ denote the subset that remains valid in G'. Let $\mathtt{M}$ denote an efficient algorithm, called *modification strategy*, that will augment S_1 into a feasible Steiner Tree for the set of terminals T on G'. We denote the expected weight of the second-stage outcome by $E_\pi(G, S, \mathtt{M})$. Let $opt(G', T)$ refer to the weight of the optimum Steiner Tree for T on G' for every subgraph $G'(V', E')$ of G such that $T \subseteq V'$. The expected minimum weight over the distribution of subgraphs of G is:

$$E^*_\pi(G) = \sum_{V' \subseteq V} \Pr[V'] opt(G', T) \tag{1}$$

where $\Pr[V'] = \prod_{v \in V'} \pi(v) \prod_{\in V \setminus V'} (1 - \pi(v))$ is the distribution describing probability of occurrence of a specific subset of V in second stage. Under these definitions we study the following two problems:

- **A priori optimization** of PROBABILISTIC STEINER TREE($\mathtt{M}$): Given a fixed modification strategy $\mathtt{M}$, find an algorithm for taking a priori decisions $S \subseteq E$, that optimize $E_\pi(G, S, \mathtt{M})$.
- **Modification** for PROBABILISTIC STEINER TREE: Given a feasible optimum or approximate anticipatory solution S, find a modification strategy $\mathtt{M}$ for modifying the valid remainders S_1 of these decisions on G' towards a feasible solution S' for G', so that $E_\pi(G, S, \mathtt{M})$ is optimized.

In what follows we show that both these problems are **NP**-hard, and we will study approximation techniques. The approximation ratio is defined as $E_\pi(G, S, \mathtt{M})/E^*_\pi(G)$. Given an edge subset S that is a tree and two vertices v_i, v_j spanned by S, we will refer to the unique path connecting v_i to v_j in S with $[v_i \cdots v_j]_S$, and denote the sum of weight of its edges by $w([v_i \cdots v_j]_S)$. In analysis developed within subsequent sections we are going to make use of the following lemmas:

Lemma 1 *Given an edge-weighted a priori input graph G, an arbitrary subgraph G' of G (induced by a subset of vertices of G) and a set of terminal vertices T that exist also in G', if $opt(G, T)$, $opt(G', T)$ are the values of the optimum Steiner trees connecting T on G and G' respectively, we have $opt(G', T) \geq opt(G, T)$.*

Proof. Every feasible solution for T on G' is also feasible on G. □

Lemma 2 *$E^*(G) \geqslant opt(G)$.*

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., $opt(G) \leqslant opt(G')$, for any $G' \subseteq G$. Using it in (1), we get:

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

which concludes the proof. □

3 Complexity

In this section we prove **NP**-hardness results for both problems of a priori optimization and modification for the PROBABILISTIC STEINER TREE problem, by reduction from the deterministic Steiner Tree problem.

Proposition 1 *A priori optimization of the* PROBABILISTIC STEINER TREE *problem is* ***NP****-hard for every modification strategy.*

Proof. The proof stems from **NP**-hardness of the Steiner Tree problem: by setting $\pi(v) = 1$ for all $v \in V$ yields a deterministic Steiner Tree instance. □

As mentioned previously regarding the modification problem, our focus is on feasible anticipatory solutions on the first-stage graph G. For this particular case we show that:

Proposition 2 *The problem of modifying the remainders of an arbitrary anticipatory solution towards optimizing the expected second-stage weight is* ***NP****-hard, even in the case that 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 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 $\mathtt{M}$, then we want $\mathtt{M}$ to be such so that the following expression is minimized:

$$E_\pi(G, S^\star, \mathtt{M}) = (1 - p)w(S') + pw(S^\star) \tag{3}$$

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' in such a way that (3) is minimized. Note that in order that this happens, its first term and in particular $w(S')$ has to be minimum. Notice also that, in the case of absence of v from G, the anticipatory solution $S^\star$ is completely destroyed, i.e., $S_1 = \emptyset$. Hence, in second-stage every modification strategy $\mathtt{M}$ will have to reconnect the terminals of T from scratch on G'. The cases that can occur for S' are the four ones illustrated in Figure 1, where a triangle denotes a tree spanning some terminals in H.

If S' is as in Figures 1(a) or 1(b) (let us refer to Figure 1(a), the case of Figure 1(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 1(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,

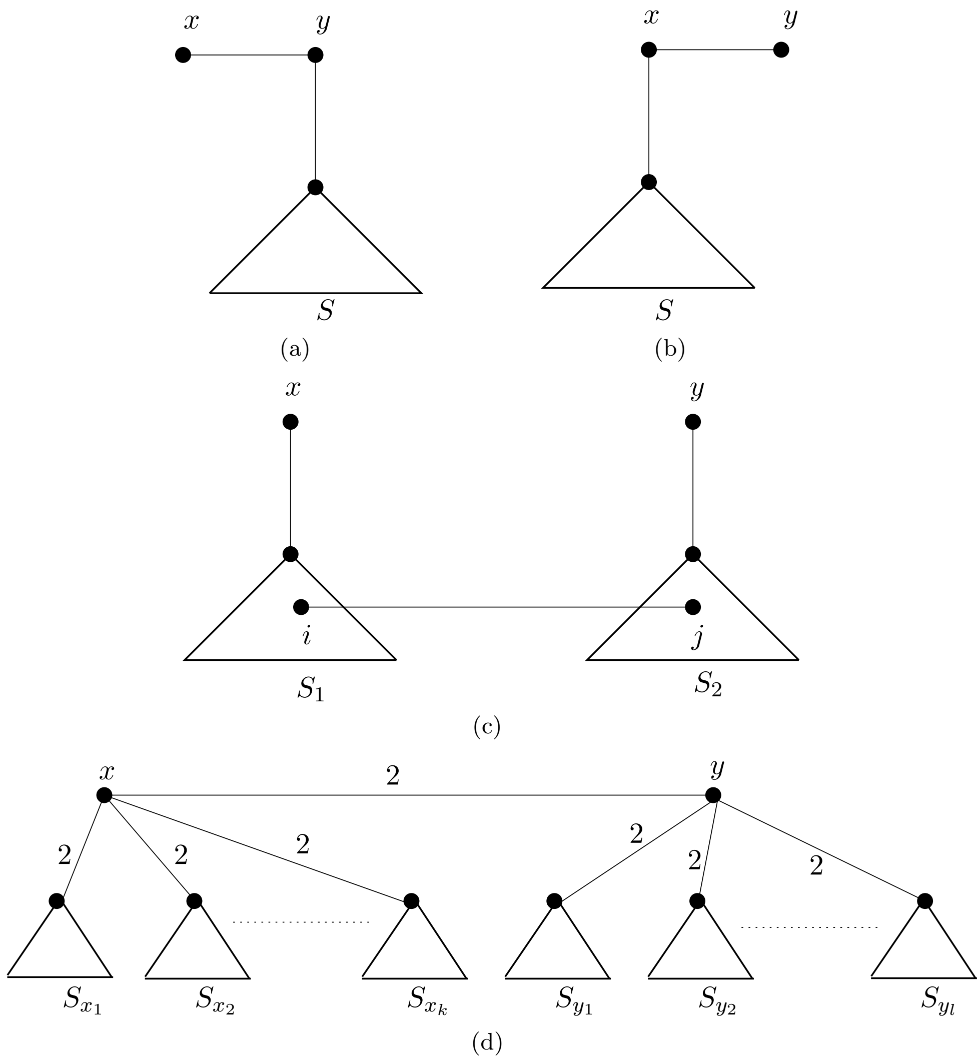


Figure 1: The four possible cases for S'.

$S = S_1 \cup S_2 \cup \{(i,j)\}$ is a Steiner tree of H and on the hypothesis that $w(S')$ is minimum, so is the weight of S for H.

Finally, suppose that S' is as in Figure 1(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). Then it is easy to see that S' can be transformed, as in Figure 2, into a Steiner tree S'' of G' with at most the same weight 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 weight 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 1(a); hence a minimum-weight Steiner tree of H is polynomially built.

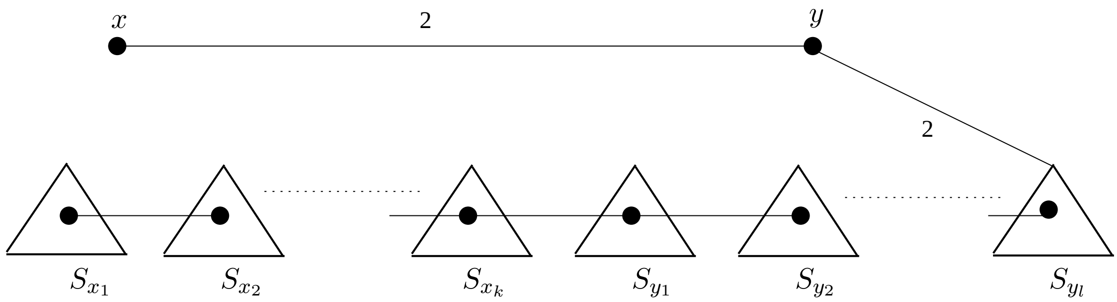


Figure 2: A Steiner tree S'' of total weight at most equal to the weight of S'.

So, on the hypothesis that the problem of modifying the remainders of an arbitrary an anticipatory feasible 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. □

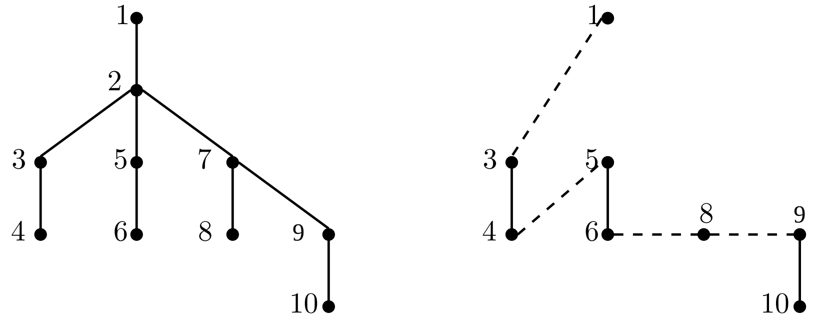
4 A Depth-First-Search Modification Strategy

In this paragraph we design and analyze probabilistically a modification strategy (also referred to as `DFS`) for an anticipatory solution S. When the subgraph G' materializes, the modification strategy `DFS` reconnects the valid remainders of the anticipatory tree using edges from E', so as to render it feasible for G'. We explain the strategy followed for reconnecting an anticipatory tree S that has been disconnected in G'.

Consider the first-stage tree solution S, and let $S_1 \subseteq S$ denote its valid remainders on the materialized second-stage subgraph G'. The strategy orders the vertices in $V(S)$ using a Depth-First-Search, starting from an arbitrary leaf-vertex the tree. Vertices of S are inserted in an ordered list $\mathcal{L}$ in order of visitation by `DFS` in the following way: if v_i and v_{i+1} are two distinct vertices visited by `DFS` consecutively for the first time, but no (v_i, v_{i+1}) edge exists in S, then they are appended to $\mathcal{L}$ along with the parent vertex u of v_{i+1}, in the order v_i, u, v_{i+1}. Thus $\mathcal{L}$ may contain some vertices more than once (in fact, as many times as their children in S). However, $|\mathcal{L}| = O(|S|)$.

When the actual second-stage subgraph G' materializes, the modifying algorithm sets $S' = S_1$. Then it removes from $\mathcal{L}$ every copy of vertex $v \in V \setminus V'$ thus producing the list $\mathcal{L}'$. It scans $\mathcal{L}'$ in order and for every two consecutive vertices v_i, v_j it inserts in S' an edge (v_i, v_j) if $i < j$ and v_i, v_j are not already connected in S'. We illustrate the functionality of the modification algorithm over a particular tree by an example.

Example Figure 3(a) depicts an anticipatory tree solution numbered according to `DFS` vis-



(a) An anticipatory solution.

(b) The modified tree in absence of vertices v_2, v_7. Dashed edges were added by the modification strategy.

Figure 3: Functionality of the modification strategy over a particular tree.

itation starting from a leaf-vertex. The corresponding ordered list produced in this way is $\mathcal{L} = \{1, 2, 3, 4, 2, 5, 6, 2, 7, 8, 7, 9, 10\}$. Assuming that vertices 2 and 7 are absent from the vertex set of the actually materialized subgraph, all occurrences of these vertices are dropped from $\mathcal{L}$ and $\mathcal{L}' = \{1, 3, 4, 5, 6, 8, 9, 10\}$ emerges. The modifying algorithm scans $\mathcal{L}'$ in order and adds edges $(1, 3)$, $(4, 5)$, $(6, 8)$, $(8, 9)$, so as to reconnect the remainders of the anticipatory tree, as shown in Figure 3(b).
We prove the following:

Proposition 3 *The* `DFS` *modification strategy produces a connected second-stage tree solution out of an anticipatory tree solution.*

Proof. For every vertex v_j in $\mathcal{L}'$ there is an appearance of v_j in $\mathcal{L}'$ after a vertex v_i with $i < j$, so that v_j is connected to v_i by the end of execution of `DFS`. This holds for all vertices, apart from the one appearing first in $\mathcal{L}'$. This implies that all vertices are connected into one component by the end of execution of `DFS` for S. Furthermore the emerging construction cannot contain cycles for two reasons: S did not have cycles and in order for a cycle to occur in the modified solution S', insertion of at least one edge (v_i, v_j) is required while its endpoints have been already connected. This cannot happen by functionality of the modifying algorithm. □

The complexity of the modification strategy is almost linear in the number of vertices of G. Indeed, a depth-first search over an anticipatory tree S is of $O(|S|)$ complexity, while by using `UNION-FIND` disjoint sets representation for maintaining connected components during the scan of $\mathcal{L}'$, an $O(|S|\alpha(|S|)) = O(n\alpha(n))$ time is spent.

Theorem 1 *Given an arbitrary feasible anticipatory solution S, the expected weight of a modified solution S' produced by modification strategy* `DFS` *is:*

$$\begin{aligned} E_\pi(G, S, \texttt{DFS}) &= \sum_{(v_i, v_j) \in S} p_i p_j w(v_i, v_j) + \\ &+ \sum_{(v_i, v_j) \in E(V(S)) \setminus S} w(v_i, v_j) p_i p_j \times \prod_{\substack{v_l \in [v_i, v_j]_{\mathcal{L}}: \\ i<j \;,\; v_i, v_j \notin [v_i, v_j]_{\mathcal{L}}}} (1 - p_l) \end{aligned}$$

where $V(S)$ is the set of vertices incident to edges of S, and $E(V(S))$ is the set of all edges induced by vertices in $V(S)$. Furthermore, $[v_i, v_j]_{\mathcal{L}}$ the sublist of $\mathcal{L}$ starting at v_i and ending in v_j not including these two vertices. For all sublists not satisfying the specified restrictions we define the product to be 0.

Proof. The stated expression consists of two terms, the first one expressing the expected weight of surviving edges in the materialized subgraph (that is the expected weight of S_1), while the second expresses the expected weight of edges added to S_1 by the modification strategy, so that S_1 is augmented into a feasible tree S' for G'. The first term is justified by the fact that $(v_i, v_j) \in S$ survives in S_1 if and only if both its endpoints survive. This happens with probability $p_i p_j$, since these two events are independent.

The second term emerges by inspection of the functionality of the modification strategy DFS. When G' materializes, missing vertices (in $V \setminus V'$) are dropped from the ordered list encoding $\mathcal{L}$ and the modified list $\mathcal{L}'$ emerges. The modification strategy scans $\mathcal{L}'$ and for every pair of consecutive vertices v_i, v_j it connects them using an edge (v_i, v_j) if and only if $i < j$ and v_i is not connected to v_j already.

Vertices $v_i, v_j \in \mathcal{L}$ both survive in $\mathcal{L}'$ with probability $p_i p_j$. Vertices v_i and v_j are not connected to each other if all vertices between v_i and v_j in $\mathcal{L}$ are missing from $\mathcal{L}'$, and 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 appear as intermediates in the sublist $[v_i, v_j]_{\mathcal{L}}$, otherwise they should also be missing, and would not be encountered by the modification strategy. Finally, the sublist $[v_i, v_j]_{\mathcal{L}}$ should not be empty, otherwise a surviving edge (v_i, v_j) is implied, rendering v_j connected to v_i. □

Clearly the expression given in Theorem 1 is computable in polynomial-time. Thus:

Corollary 1 *The problem of a priori optimization for* PROBABILISTIC STEINER TREE*(*DFS*) belongs to the class **NPO**.*

Unfortunately, Theorem 1 does not derive a compact characterization for the optimal anticipatory solution for PROBABILISTIC STEINER TREE(DFS). For instance, in [19, 20] 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 Theorem 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 $E(G, S, \mathtt{DFS})$ 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.

5 Approximation on Metric Graphs

In this paragraph we derive approximation results for the PROBABILISTIC STEINER TREE problem on metric graphs. We show that the PROBABILISTIC STEINER TREE problem is easily approximable within a factor of 2 regardless of modification strategy. We then turn to show that the previously introduced modification strategy can in fact modify a variety of first-stage feasible solutions so that an approximation of the expected minimum weight is obtained. We apply these results in the next section to the PROBABILISTIC STEINER FOREST problem on metric graphs, so as to obtain approximation results. We show at first that:

Proposition 4 *There is a polynomial-time* 2 *factor approximation algorithm for the* PROBABILISTIC STEINER TREE *problem on metric graphs, that is independent of the chosen modification strategy.*

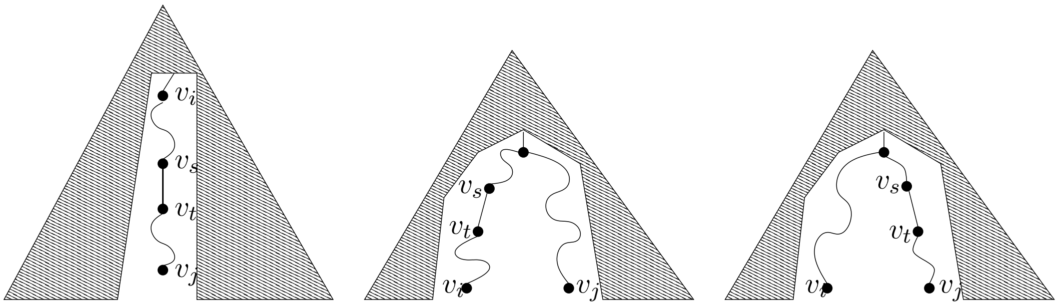


Figure 4: Three cases that may happen for edge (v_s, v_t) with respect to v_i, v_j (proof of Lemma 4).

Proof. This algorithm is the well known *minimum spanning tree* heuristic used for metric graphs (see e.g. [29]), which returns as a feasible first-stage solution S the minimum spanning tree of the subgraph induced by the terminal vertices. Notice that for the PROBABILISTIC STEINER TREE problem, such a tree remains connected and feasible also in the second-stage materialized subgraph G', since terminal vertices are present with probability 1.

It remains to show that it is also 2-approximate to $E^\star_\pi(G)$. Indeed, if $opt(G)$, $opt(G')$ denote the optimum Steiner Tree values for the first-stage graph G and the second-stage materialized subgraph G', respectively, by Lemma 4, $opt(G') \geq opt(G)$ for every possible G' containing the set T of terminal vertices. Then, using also Lemma 2:

$$w(S) \leq 2opt(G) \leq 2E^\star_\pi(G)$$

which concludes the proof. □

We now turn to show that the proposed modification algorithm `DFS` is in fact able to modify every α-approximate anticipatory solution, so as to produce a feasible solution that is at most 2α-approximate to the expected minimum weight. The heart of our results is the following theorem:

Theorem 2 *If S' is a modified feasible solution produced by* `DFS` *modification strategy for the* PROBABILISTIC STEINER TREE *problem on a metric graph, given an anticipatory feasible solution S, then $w(S') \leq 2w(S)$.*

In the following we denote by S_m the subset of edges added by the modification strategy to S_1. We prove first some lemmas that will be combined towards the proof of the theorem.

Lemma 3 *For every edge $(v_i, v_j) \in S_m$ we have $w(v_i, v_j) \leq w([v_i \dots v_j]_S)$.*

Proof. Immediate by the triangle inequality holding for the weight function $w : E \rightarrow \Re^+$. □

According to Lemma 3 we can express the weight of the modified tree S' as follows:

$$w(S') = w(S_1) + w(S_m) \leq \sum_{e \in S_1} w(e) + \sum_{(v_i,v_j) \in S_m} w([v_i \dots v_j]_S) \tag{4}$$

Lemma 4 *For every three distinct edges (v_i, v_j), (v_k, v_l), (v_q, v_r) in S_m the paths $[v_i \cdots v_j]_S$, $[v_k \cdots v_l]_S$, $[v_q \cdots v_r]_S$, do not share an edge in common.*

Proof. By functionality of the modification strategy we have that $i < j$, $k < l$, $q < r$. Furthermore, if we assume without loss of generality that the vertex pairs were encountered in the order $\langle i, j\rangle$, $\langle k, l\rangle$, $\langle q, r\rangle$ during scanning of $\mathcal{L}'$, then we deduce that $j < l < r$. If the paths intersect in some common edge (v_s, v_t), then it must be $s, t \leq j$ (Figure 4 depicts all possible cases), thus $s, t < l$ and $s, t < r$. In this case edge (v_s, v_t) must have been scanned at least three

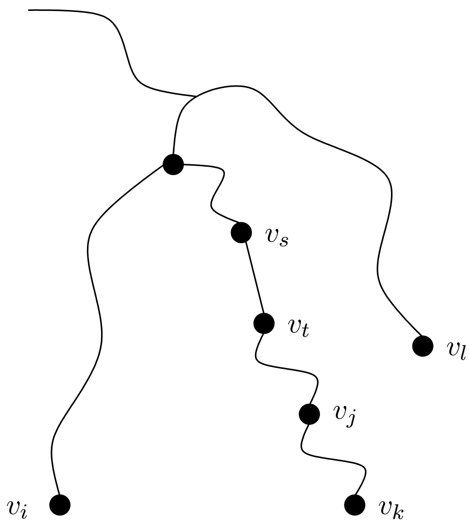


Figure 5: Case $s, t > i$ examined in the proof of Lemma 5: $(v_s, v_t) \in [v_i \cdots v_j]_S \cap [v_k \cdots v_l]_S$ and suppose that $(v_s, v_t) \in S_1$. Then obviously v_s, v_t survive in $\mathcal{L}'$ and appear as intermediates in the two pairs $\langle v_i, v_j \rangle$ and $\langle v_k, v_l \rangle$.

times during depth-first search: once before visitation of each of the vertices v_j, v_l, v_r. But this contradicts the fact that a depth-first search scans each edge of a graph exactly twice. □

The following lemma will help us to complete the proof of the theorem:

Lemma 5 *Consider two edges (v_i, v_j), (v_k, v_l) in S_m. For every edge (v_s, v_t) with $(v_s, v_t) \in [v_i \cdots v_j]_S \cap [v_k \cdots v_l]_S$ it holds $(v_s, v_t) \notin S_1$.*

Proof. The proof is by contradiction. Suppose that $(v_s, v_t) \in [v_i \cdots v_j]_S \cap [v_k \cdots v_l]_S$ and $(v_s, v_t) \in S_1$. Without loss of generality we assume that the modification strategy encountered first the pair $\langle v_i, v_j \rangle$ and afterwards the pair $\langle v_k, v_l \rangle$ in $\mathcal{L}'$. It must be $i < j$, $k < l$ and $j < l$ (v_j may coincide with v_k). Since $(v_s, v_t) \in [v_i \cdots v_j]_S \cap [v_k \cdots v_l]_S$, then we must have $s, t \leq j$ and, consequently, $s, t < l$. Furthermore, it must hold either that (i) $s, t > k$ or that (ii) $s, t > i$, otherwise it should be $s, t < i$ and, given that $s, t \leq j$ also, we would deduce that (v_s, v_t) would have been scanned twice during DFS, once before visitation of v_i and once before visitation of v_j. In this case it could not have been scanned again right before visitation of v_l. Now, (i) cannot hold because $k \geq j$ and $s, t < j$. If (ii) holds, i.e., $s, t > i$, it is implied that the modification strategy did not encounter in $\mathcal{L}'$ vertices v_k, v_l and v_i, v_j consecutively (Figure 5), which is a contradiction. □

The proof of Theorem 2 can now be completed as follows:

Proof. Relation (4) can be written:

$$
\begin{aligned}
w(S') &\leq \sum_{e \in S_1} w(e) + \sum_{(v_i, v_j) \in S_m} w([v_i \ldots v_j]_S) \\
&= \sum_{e \in S_1} w(e) + \sum_{(v_i, v_j) \in S_m} \sum_{e \in [v_i \ldots v_j]_S} w(e) \\
&= \sum_{e \in S_1} w(e) + \sum_{(v_i, v_j) \in S_m} \Big(\sum_{e \in [v_i \ldots v_j]_S : e \in S_1} w(e) + \sum_{e \in [v_i \ldots v_j]_S : e \notin S_1} w(e) \Big) \qquad (5)
\end{aligned}
$$

By Lemmas 4 and 5 the following are implied:

$$\sum_{(v_i,v_j)\in S_m} \sum_{e\in[v_i\ldots v_j]_S : e\in S_1} w(e) \leq \sum_{e\in S_1} w(e) \tag{6}$$

$$\sum_{(v_i,v_j)\in S_m} \sum_{e\in[v_i\ldots v_j]_S : e\notin S_1} w(e) \leq 2 \sum_{e\in S\setminus S_1} w(e) \tag{7}$$

By replacing the relations (6) and (7) in the expression (5) we obtain:

$$w(S') \leq 2\sum_{e\in S_1} w(e) + 2\sum_{e\in S\setminus S_1} w(e) \leq 2w(S) \tag{8}$$

which concludes the proof. □

Theorem 2 leads to the following result regarding modification and the variety of anticipatory feasible solutions that `DFS` can handle effectively:

Corollary 2 *There is an $O(n\alpha(n))$ time modification strategy that can modify an anticipatory feasible and α-approximate solution for the* PROBABILISTIC STEINER TREE *problem on metric graphs, to yield a 2α-approximate solution of the minimum expected weight. This leads to factor 2 approximation for an optimum anticipatory solution and to factor 3.1 approximation for an anticipatory solution given by the algorithm of [25].*

Proof. By Theorem 2 for an α-approximate anticipatory solution S we get $w(S') \leq 2\alpha opt(G)$, where $opt(G)$ denotes the optimum solution value on G. Using Lemma 1 we obtain $w(S') \leq 2\alpha opt(G')$. Taking expectation over the distribution of materializable subgraphs G' and using Lemma 2, yields the stated result: $E_\pi(G, S, \mathtt{DFS}) \leq 2\alpha E^\star(G)$. □

6 An Application: Probabilistic Steiner Forests

In this section we show that the DFS modification strategy can be extended for the case of the more general PROBABILISTIC STEINER FOREST problem on metric graphs. The Steiner Forest problem concerns connecting simultaneously pairs of terminal vertices (and not necessarily interconnecting all terminals), while minimizing the total weight of used edges. Therefore a feasible solution to the problem is generally a collection of trees. Given such a feasible anticipatory solution F, we consider the PROBABILISTIC STEINER FOREST(`DFS`) problem, where `DFS` is executed independently on each tree of the anticipatory feasible forest, so as to reconnect it on the actually materialized subgraph G'. The resulting forest F' is obviously feasible on G', since it connects the same pairs of terminals that F did on G. Furthermore, both F and F' consist of the same number of trees (because DFS is executed on each disconnected tree of S independently in the second stage). We note that, **NP**-hardness results regarding a priori optimization, and modification, trivially carry to the case of PROBABILISTIC STEINER FOREST.

The complexity of this extended version of `DFS` is still $O(n\alpha(n))$, because if the anticipatory forest F consists of k trees S_i, $i = 1\ldots k$, then at most $O(|S_i|\alpha(|S_i|))$ time is spent per tree. Since $|S_i| = O(n)$ and $\sum_{i=1}^{k} |S_i| = |F| = O(n)$ the complexity is as stated. We obtain the following result regarding modification:

Theorem 3 *There is an $O(n\alpha(n))$ time modification strategy for the* PROBABILISTIC STEINER FOREST *problem on metric graphs that, when applied to an α-approximate anticipatory solution, produces feasible solutions that are 2α-approximate to the optimum expected cost.*

Proof. Assume an anticipatory feasible forest F and a modified solution F', each consisting of k trees S_i, S'_i, $i = 1\ldots k$ respectively. Then:

$$w(F) = \sum_{i=1}^{k} w(S_i), \quad w(F') = \sum_{i=1}^{k} w(S'_i)$$

By Theorem 2 we have that $w(S'_i) \leq 2w(S_i)$, thus: $w(F') \leq 2w(F)$. Let $opt(G)$ and $opt(G')$ be the weights of an optimum Steiner forest on G and G' respectively for the given terminal pairs, and $w(F) \leq \alpha opt(G)$. We have $opt(G) \leq opt(G')$ for every possible subgraph G' of G, by a similar version of Lemma 1 for the Steiner forest problem. Thus $w(F') \leq 2\alpha opt(G')$. Taking expectation the distribution of materializable second-stage subgraphs G' and using Lemma 2, yields $E_\pi(G, F, \texttt{DFS}) \leq 2\alpha E^\star_\pi(G)$. □

Corollary 3 *There is a polynomial-time a priori approximation algorithm for the* PROBABILISTIC STEINER FOREST(DFS) *on metric graphs (the algorithm of [1]) yielding factor* 4 *approximation of the expected minimum weight. Furthermore, there is an* $O(n\alpha(n))$ *time modification strategy that can modify an optimum anticipatory solution for the* PROBABILISTIC STEINER FOREST *problem, so as to yield factor* 2 *approximation of the expected minimum weight.*

Notably, for the PROBABILISTIC STEINER FOREST problem on metric graphs, a trivial polynomial-time a priori approximation algorithm, that is independent of chosen modification strategy (as the one exhibited by Proposition 4 for PROBABILISTIC STEINER TREE) is not applicable. This makes the PROBABILISTIC STEINER FOREST(DFS) model extremely useful in practice, if not a unique choice. We also note that in both cases mentioned in the corollary as much as in cases considered earlier for PROBABILISTIC STEINER TREE, the proposed modification strategy is faster than the algorithm used for a priori decisions, and is far more efficient than the trivial practices discussed in the introduction: in fact, any approximation algorithm used for taking a priori decisions (including the one of [1, 9]) will incur $\Omega(n^2)$ complexity.

7 Tightness of Analysis

In this section we show that the result proved in Theorem 2 is in fact tight, i.e., given an arbitrary anticipatory feasible and ρ-approximate solution S, the modification strategy DFS produces a second-stage modified solution S' that can have expected weight arbitrarily close to $2\rho E^\star_\pi(G)$.

Consider a complete graph K_{n+1} on $n+1$ vertices numbered by $1, 2, \ldots, n+1$. Assume w.l.o.g. that $n \geq 6$ is even and that edge $(1, 3)$ and edges $(i, i+1)$, $i = 3, \ldots, n-1$ have weight 2, while the other edges of K_{n+1} have weight 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, $opt(K_{n+1}) = n+1$ 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,

$$E^*(K_{n+1}) = p_2(n+1) + (1-p_2)(n+1) = n+1 \tag{9}$$

Assume now that the anticipatory solution computed is just S (of weight $n+1$) 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 star's 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 weight $2n$. So, the value of $E(K_{n+1}, S, \texttt{DFS})$ will be:

$$E(K_{n+1}, S, \texttt{DFS}) = p_2(n+1) + (1-p_2)2n = 2n - p_2 n + p_2 \tag{10}$$

In Figure 6 an illustration of the discussion just above is provided for $n = 6$. The thick edges of K_7 in Figure 6(a) are the ones with weight 2. It is assumed that vertices $1, 3, 4, 5, 6, 7$ are terminals. In Figure 6(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 6(c) is shown the DFS completion of S when 2 is absent. Finally, in Figure 6(d), the optimal Steiner tree of K_7 using only terminals built as described above is shown. Dividing (9) by (10) we get a ratio that for small enough values of p_2 tends to 2.

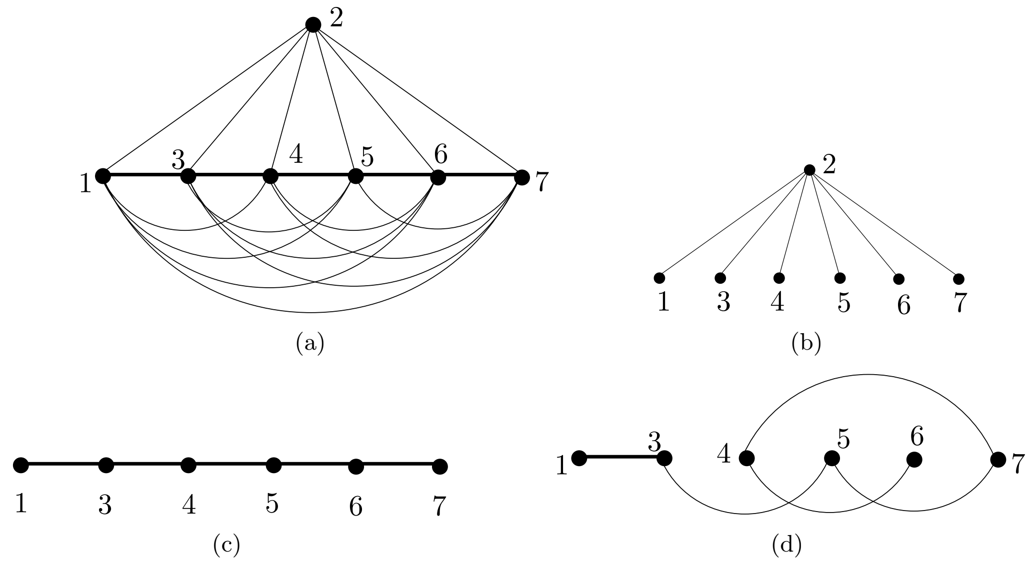


Figure 6: On the tightness of ratio 2ρ.

8 Approximation Algorithms as Modification Strategies

In this section we consider usage of known approximation algorithms for the Steiner Tree problem as components of a modification strategy. We refer to the emerging modification strategy as REAPX (for re-approximation). Following our convention we are going to study PROBABILISTIC STEINER TREE(REAPX). Throughout this section we consider arbitrarily weighted input graphs.

Let S be an anticipatory feasible first-stage solution for the Steiner Tree problem over the initial weighted graph $G(V, E)$, produced by a ρ-approximation algorithm. Let $V' \subseteq V$ be the realized vertex subset of G in second-stage, and $S_1 \subseteq S$ be the surviving portion of the anticipatory Steiner tree S. Obviously, S_1 is generally a forest; denote by C_i, $i = 1, \dots, q$ its trees (any of its trees can also be a single vertex). The REAPX strategy first constructs S_1 by discarding absent vertices from S. Then, it contracts and replaces by a single vertex u_i each tree C_i, $i = 1, \dots, t$, that spans at least a terminal vertex from T. Due to contractions several pairs of vertices may be connected through multiple edges. REAPX retains and considers the least weighted edge in this case. Let $G_1(V_1, E_1)$ be the resulting graph after contractions and proper removal of multiple edges. Let T_1 denote the set of terminal vertices where vertices of T that were contracted are substituted in T_1 by the single vertex to which they were contracted.

REAPX runs a ρ'-approximation algorithm A for the Steiner Tree problem with terminals set T_1 on G_1. Denote by $\bar{S}$ this tree and assume that it is represented as a list of edges. Then, it "unfolds" contracted vertices thus obtaining a feasible tree for T on G_1 with edge set $S' \supset \bar{S}$, also containing edges of contracted trees.

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. We show that:

Lemma 6 *$opt(G'') \leq opt(G')$.*

Proof. Notice that a Steiner tree on G' spanning the initial terminal set T is also feasible for the terminal set T_1 over G_1, because for each $v \in T_1$, either $v \in T$, or v has emerged by contraction of a subset of vertices intersecting T. Since $opt(G_1)$ is the value of an optimal Steiner tree in G_1, the result follows. □

Theorem 4 *Let* `A` *be a ρ'-approximation algorithm for the Steiner Tree problem called by* `REAPX` *in its second stage. Then an anticipatory feasible and ρ-approximate Steiner tree S on G, yields a $\rho + \rho'$-approximation for* PROBABILISTIC STEINER TREE*(*`REAPX`*).*

Proof. Obviously, $w(S_1) \leq w(S) \leq \rho opt(G)$. Furthermore, as already discussed, $opt(G) \leq opt(G')$ because every feasible tree for G' is also feasible for G. Execution of the ρ'-approximation algorithm `A` on G_1 returns an edge set $\bar{S}$ with total weight $w(\bar{S}) \leq \rho' opt(G_1) \leq \rho' opt(G')$, by Lemma 6. Then, the returned Steiner tree $S' = S_1 \cup \bar{S}$ in G' has total weight at most:

$$w(S') \leq w(S_1) + w(\bar{S}) \leq \rho opt(G) + \rho' opt(G') \leq (\rho + \rho') opt(G') \qquad (11)$$

Taking expectation over all possible materializable subgraphs G' and using Lemma 2, yields the stated result. □

Note that, from Theorem 4, if S is computed optimally, then the ratio derived from Theorem 4 is $\rho' + 1$. This is a case where the `REAPX` strategy implies an asymptotic 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:

- For metric graphs with edge weights $1, 2$ if S is the solution computed by the algorithm in [25], ratios 2.56 and 3.28 are derived when the algorithms of [25] and [1] are used respectively in implementing `REAPX`.
- if S is the solution computed by the algorithm in [1], ratios are 3.28 and 4 are derived by using the algorithms of [25] (for metric graphs with edge weights $1, 2$) and [1] (for general graphs) respectively.

9 Conclusions

In this paper we have treated the PROBABILISTIC STEINER TREE problem under the framework of 2-stage probabilistic combinatorial optimization. We have shown that both problems, of a priori optimization, and modification of arbitrary anticipatory feasible solutions are **NP**-hard. Subsequently we 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. For metric graphs we have shown that the modification strategy at most doubles the weight of an anticipatory solution and thus obtained approximation results that we could extend to the case of the more general PROBABILISTIC STEINER FOREST problem. We have shown that our analysis is tight.

Notably, 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 $E_\pi(G, S, \mathtt{M})/E_\pi(G, S^*, \mathtt{M})$ where S^* is an optimal anticipatory solution. Notice that, by (1), $E^*_\pi(G)$ is a lower bound for $E_\pi(G, S^*, \mathtt{M})$ (since solutions produced by any modification strategy are at least optimal on the materialized subgraph). Therefore, all of our approximation results remain valid for this approximation ratio too.

Study of the general edge-costs case given an optimum anticipatory solution (possibly by devising a novel modification strategy) 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.

Acknowledgements. Many thanks to Cécile Murat for helpful discussions and comments. The very useful comments and suggestions of three anonymous referees are also gratefully acknowledged.

References

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

[2] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming (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] G. W. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1951.

[7] V. G. Deĭneko and G. J. Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *Operations Research Letters*, 34(2):175–179, 2006.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[9] M. X. Goemans and D. P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[10] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the ACM Annual Symposium on Theory of Computing (STOC)*, pages 417–426, 2004.

[11] A. Gupta, M. Pál, R. Ravi, and A. Sinha. What about wednesday? approximation algorithms for multistag stochastic optimization. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2005.

[12] 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 *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 691–700, 2004.

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

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

[15] 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.

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

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

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

[19] C. Murat and V. Th. Paschos. A priori optimization for the probabilistic maximum independent set problem. *Theoretical Computer Science*, 270:561–590, 2002.

[20] C. Murat and V. Th. Paschos. The probabilistic minimum vertex-covering problem. *International Transactions on Operational Research*, 9(1):19–32, 2002.

[21] C. Murat and V. Th. Paschos. On the probabilistic minimum coloring and minimum $\boldsymbol{k}$-coloring. *Discrete Applied Mathematics*, 154:564–586, 2006.

[22] A. Prekopa. *Stochastic programming.* Kluwer Academic Publishers, 1995.

[23] Stochastic Programming. Community Homepage. *Available at* `http://stoprog.org`.

[24] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. In *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization (IPCO), Springer LNCS 2064*, pages 101–115, 2004.

[25] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 770–779, 2000.

[26] D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 228–237, 2004.

[27] 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.

[28] C. Swamy and D. B. Shmoys. Sampling-based approximation algorithms for multi-stage stochastic optimization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 357–366, 2005.

[29] V. Vazirani. *Approximation algorithms.* Springer, Berlin, 2003.