A robust 2-stage version for the STEINER TREE problem

Vangelis Th. Paschos, Orestis Teletis, Vassilis Zissimopoulos

A robust 2-stage version for the STEINER TREE problem

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

Résumé

Dans cet article, nous considérons une version robuste pour l'arbre de Steiner. Sous cette version, le problème est défini dans un cadre en deux étapes sur un graphe complet à arêtes pondérées dont les sommets sont associés avec des probabilités de présence en seconde étape. Une solution réalisable pour la première étape sur le graphe d'entrée peut devenir non-réalisable pour la seconde étape, quand quelques sommets disparaissent. Dans ce cas, une « stratégie de modification » est conçue qui transforme une solution partielle en une solution réalisable pour la seconde étape. L'objectif est de concevoir un algorithme qui calcul une solution pour la première étape (cette solution s'appelle solution *a priori* ou « d'anticipation ») de qui minimise la fonction objectif du problème robuste. Une caractéristique importante de ce modèle robuste est que la stratégie de modification est partie du problème. Nous recherchons de résultats de complexité et d'approximation en utilisant une stratégie de modification basée sur l'algorithme du parcours en profondeur.

Mots-clefs : Approximation, Arbre de Steiner, Complexité, Graphe, Optimisation probabiliste, Robustesse

Abstract

In this paper we consider a robust version for STEINER TREE. Under it, 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

*LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine. paschos@lamsade.dauphine.fr. Work partially performed while the author was with the Division of Theoretical Computer Science of the Department of Informatics and Telecommunications at the University of Athens

†Division of Theoretical Computer Science, Department of Informatics and Telecommunications, University of Athens. {telelis,vassilis}@di.uoa.gr

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 provide complexity and approximation results regarding a modification strategy based upon the well known depth-first-search algorithm.

Key words : Approximation, Complexity, Graph, Probabilistic optimization, Robustness, Steiner tree

1 Introduction and motivation

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 optimization stochastic 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. 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 an efficient 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 subgraph G', induced by a subset $V' \subseteq V$, 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 consistency 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 [6, 8, 27] for basics, [28] for latest news, bibliography, and related software and [13, 14, 16, 29, 31, 32, 33] for recent hardness results and approximation algorithms) and the *robust discrete optimization* (see [3, 9, 19, 21, 22] 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.

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 (vertex-probabilities are independent). Assume also that a more or less efficient 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*):

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

where $m(G[V'], S')$ denotes the cost of S' in $G[V']$[1] 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[2].

[1] Quantity $m(G[V'], S')$ depends indeed also on S since it is the fitting of S in $G[V']$; in this sense, $m(G[V'], S')$ should be rather written as $m(G[V'], S', S)$; but, for simplicity, this dependance will be omitted.

[2] In this sense, the correct way to denote a robust version of a combinatorial optimization problem Π

This model is originally introduced in [17, 4] under the name *a priori probabilistic combinatorial optimization* as a means of structuring and handling uncertainty in the validity of input data and of tackling robustness requirements for the solutions computed on such data. Since then, several well-known combinatorial optimization problems have been treated in this framework, such as the *longest path* ([23]), the *maximum independent set* ([24]), and the *minimum coloring* ([25, 7]). Less recent studies include the *shortest path* ([18]), the *minimum-cost spanning tree* ([5]) and the *travelling salesman* ([17]). Several examples of natural situations concerning planning, timetabling, etc., that can be modelled as probabilistic combinatorial optimization problems are given in [25, 26].

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 [17]. The functional for such a process can be expressed as:

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

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 Π (under $\mathtt{M}$) and assuming that Π is a minimization problem:

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

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 ([35]), 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, [20] for some models concerning optimization of

(under the settings described) is by PROBABILISTIC $\Pi(\mathtt{M})$; however, since throughout the paper the modification strategy used is fixed, we simplify things by just using PROBABILISTIC Π instead.

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.

Efficient connectivity of the backbone nodes can be achieved by a solution to the STEINER TREE problem ([20]). However, nodes of a wireless network are often subject 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. Therefore the PROBABILISTIC STEINER TREE problem seems in this case a more appropriate model as a means of providing connectivity to the backbone nodes.

STEINER TREE is well known to be **NP**-hard. The first approximation algorithm for it appeared in [1] (see also [12, 34]) and consists first of a shortest path computation for all pairs of terminal vertices and next of a computation of a minimum-cost spanning tree over the shortest-path weighted complete graph with vertex set T. The approximation ratio achieved by this algorithm is bounded above by 2.

This result has been improved in [30] for metric complete graphs, down to 1.55 in complete metric graphs and to 1.28 for complete graphs with edge costs 1 and 2. Recently, a robust optimization model, quite different from the model we tackle here, called *demand-robust Steiner tree* has been introduced and studied in [10].

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., [11, 15].

In what follows, we mainly study the complexity and approximation of a robust model for STEINER TREE derived by a modification strategy based upon the depth-first-search algorithm ([2]). 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 2 we formally introduce the model tackled, we study its objective function (also called *functional*), 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) and we discuss approximation issues for this model.

2 Robustness for STEINER TREE under the depth-first-search modification strategy

2.1 The derived problem and its complexity

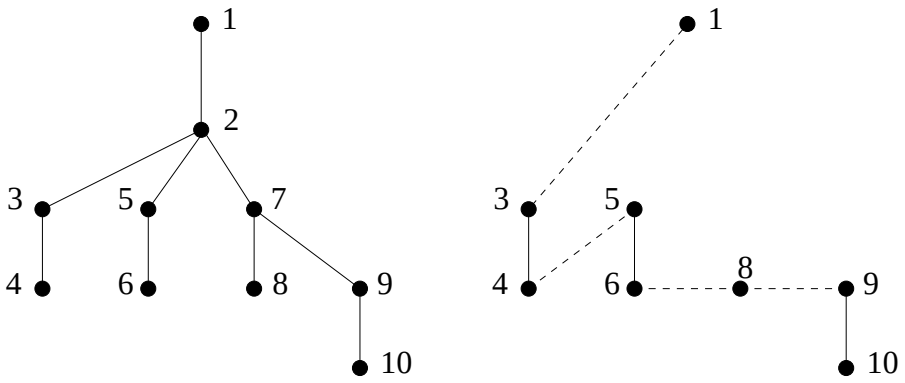
We use the following modification strategy, called DFS. Given an anticipatory Steiner tree $S \subseteq E$ (represented by the set of its edges), DFS first orders the edges of S by engaging a depth-first search (DFS) starting from an arbitrary leaf-vertex of S. Each vertex of S is assigned a number equal 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, 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'. It can be immediately seen that the complexity of DFS is linear with n (the order of the input graph).

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: $P_1 = \{1, 2, 3, 4\}$, $P_2 = \{2, 5, 6\}$, $P_3 = \{2, 7, 8\}$, $P_4 = \{7, 9, 10\}$.

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). ∎

Proposition 1. PROBABILISTIC STEINER TREE *is* ***NP****-hard.*



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

Figure 1: Application of the DFS modification strategy.

Proof. The most technical part of the proof is the inclusion of PROBABILISTIC STEINER TREE in **NPO**[3]. Then, **NP**-hardness of PROBABILISTIC STEINER TREE immediately follows from the fact that setting occurrence-probability 1 for any vertex of the input graph, PROBABILISTIC STEINER TREE coincides with STEINER TREE.

For proving inclusion of PROBABILISTIC STEINER TREE in **NPO** it suffices to prove that the objective function $E(G, S, \texttt{DFS})$ of the problem is computable in polynomial time.

Let $G(V, E)$ be an instance of PROBABILISTIC STEINER TREE 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.

[3] **NPO** is the class of the optimization problems the decision counterparts of which are in **NP**.

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: (i) v_i, v_j are actually in $\mathcal{L}'$ (ii) $i < j$ (by the DFS-produced labelling) and (iii) 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 in $\mathcal{L}'$. 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 it would entail a surviving edge between v_i and v_j (recall that $v_i < v_j$ in the encoding), 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 $n + 1$ 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 belongs to **NPO** and, with the remark in the beginning of the proof, its **NP**-hardness is also concluded. ∎

Proposition 1 does not derive a compact characterization for the optimal anticipatory solution for PROBABILISTIC STEINER TREE. This is due to the second term of the functional $E(G, S, \texttt{DFS})$ that depends on the structure of the anticipatory solution chosen and of the present subgraph of G.

2.2 On the approximability of PROBABILISTIC STEINER TREE

We now turn to study approximation of PROBABILISTIC STEINER TREE 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-value in G (i.e., the value of an optimal Steiner tree in G by ignoring the vertex-

probabilities).

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

Indeed, 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: $E^*(G) \geqslant \sum_{V' \subseteq V} \Pr[V'] \,\mathrm{opt}(G) = \mathrm{opt}(G) \sum_{V' \subseteq V} \Pr[V'] = \mathrm{opt}(G)$, q.e.d.

2.2.1 When anticipatory solution is obtained by minimum spanning tree computation

We now turn to the classical minimum-cost spanning tree algorithm on $G[T]$ (sketched in Section 1) and show the following result.

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

Proof. Let H be an isomorphic of G with modified edge-weights, i.e., where $(v_i, v_j) \in E$ instead of weight $w(v_i, v_j)$, it has in H weight $w'(v_i, v_j) = p_i p_j w(v_i, v_j)$. Then,

$$E(G, S, \texttt{DFS}) = m(H, S) = \sum_{(v_i, v_j) \in S} w\,(v_i, v_j)\, p_i p_j \tag{4}$$

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 Fact 1):

$$E^*(G) \geqslant \mathrm{opt}(G) \geqslant \mathrm{opt}(H) \tag{5}$$

Putting (4) and (5) together and taking into account that S is a 2-approximation for STEINER TREE in H (see Section 1), the result claimed is immediately derived. ∎

A minimum-cost spanning tree S over $G[T]$ (or $H[T]$) is a very particular solution since, thanks to the omnipresence of the vertices of T, S remains feasible in second stage, i.e., no completion is needed.

2.2.2 More general anticipatory solutions

We so turn to study anticipatory solutions that have not a priori such property (i.e., that need completion in the second stage in order to become feasible Steiner trees) and could

contain also non-terminal vertices of G, i.e., vertices that have a presence-probability less than 1. More precisely, we consider using an arbitrary ρ-approximation algorithm for obtaining a feasible Steiner tree with cost no more than ρ times the cost of an optimal Steiner tree of G. We show the following proposition.

Proposition 3. *Any ρ-approximation algorithm for* STEINER TREE *yields an anticipatory solution resulting in a 2ρ-approximation for* PROBABILISTIC STEINER TREE. *This ratio is tight even when edge costs are 1 and 2 and an optimal* STEINER TREE*-solution is used as anticipatory (first-stage) solution.*

Proof. The proof is based upon the following emphasized claim. *The total cost of edges added to the partial tree S'' in order to produce S' is at most $2m(G, S)$, where S is the anticipatory Steiner tree computed in G.*

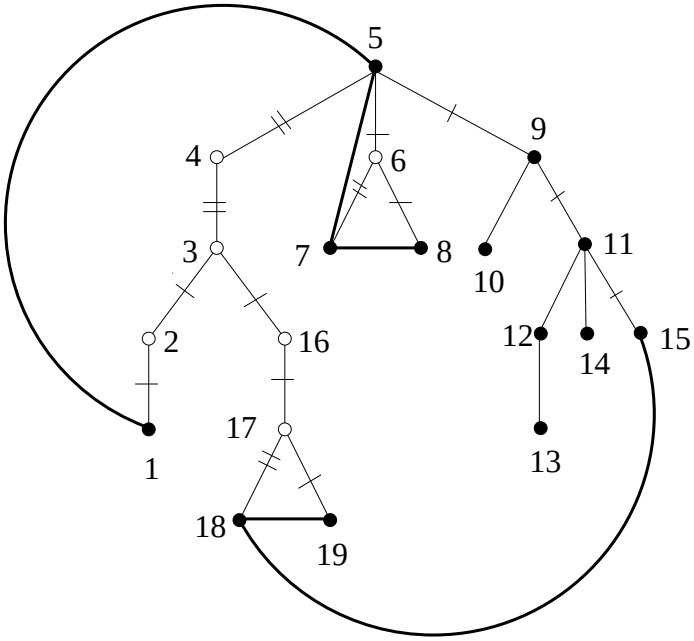


Figure 2: The total cost of edges added to the partial tree S'' in order to produce S' (the Steiner tree in G') is at most $2m(G, S)$.

Figure 2, where vertices are named by their DFS ordering, gives an example illustrating the central idea in the proof of the claim that follows. The anticipatory Steiner tree S is the tree with thin lines. We suppose that white vertices are absent in second stage. Hence, the edges added in this stage in order to reconnect S are (the thick) edges $(1, 5)$, $(5, 7)$, $(7, 8)$, $(15, 18)$ and $(18, 19)$. In the so reconstructed tree S', and since we assume that G is metric, 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 $(18, 19)$.

In order to formally prove the claim, "rehang", for clarity, the tree S from its leaf numbered by 1 (that becomes the root of S). Then, any vertex has a DFS number less than the numbers of the vertices on its subtree.

Consider an edge $(v_i, v_j) \in E$, absent from G' (i.e., either v_i or v_j are absent from V'). Then, the anticipatory tree will be completed by an edge (v_k, v_l) such that $k \leqslant i$ and $l \geqslant j$. In other words, v_k is a predecessor of v_i and v_j (or v_i itself) and v_l a successor of them (or v_j itself). Obviously, by the fact that G is metric, $w([v_k, v_l]_{\mathcal{L}}) \leqslant \sum_{(v_r,v_s)\in[v_k,v_l]_{\mathcal{L}}} w(v_r, v_s)$. Hence, the cost of $(v_i, v_j) \in [v_k, v_l]_{\mathcal{L}}$ will count once "forwardly" when (v_k, v_l) is added to S''.

Assume that an edge (v_p, v_q), $p \geqslant l$ is added later. Obviously $q \geqslant p \geqslant l$. Edge (v_i, v_j) will count once more "backwardly" if v_p is in the subtree rooted at v_j and v_q is in some subtree rooted at a predecessor of v_i and v_j, this subtree being different from the one in which lie v_i and v_j. Once more, $w([v_p, v_q]_{\mathcal{L}}) \leqslant \sum_{(v_r,v_s)\in[v_p,v_q]_{\mathcal{L}}} w(v_r, v_s)$.

It is easy to see that edge (v_i, v_j) will not count another time forwardly. Indeed, if it counted once more in this way due, say, to an edge (v_a, v_b), $a < k$ and $b > l$, then, edges (v_a, v_k), (v_k, v_l) and $v_l, v_b)$ should have been added for completion of S'' instead of only (v_a, v_b) and (v_k, v_l). Hence (v_i, v_j) would have counted only once.

Assume now that (v_i, v_j) counts a second time backwardly due to an edge (v_c, v_d) added after (v_p, v_q). Obviously, $p < q < c < d$. Then, as mentioned above, v_c is on a subtree rooted at v_j. Since $c > q$, there is enough vertices in this subtree in order that one of them takes the DFS number q, i.e., v_q should not been located where it has been assumed to be previously (in some subtree rooted at a predecessor of v_i and v_j, this subtree being different from the one in which lie v_i and v_j). So, v_i, v_j will backwardly count at most once and the claim is proved.

Observe also that the edges of S that will count twice in the completion are absent from S''. Indeed, an edge of S that survives in G', so in S'', will count in a completion edge only "backwardly".

Denote by S_1 the set of edge of S that will count once for evaluating the completion edges and by S_2 the (absent) edges of S that will count twice. Denote also by S' the Steiner tree of G' resulting from the completion of S''. Then:

$$m\left(G', S'\right) = w\left(S'\right) \leqslant w\left(S''\right) + w\left(S_1\right) + 2w\left(S_2\right) \tag{6}$$

Sets S_1 and S_2 are obviously disjoint and $S_1 \cup S_2 \subseteq S$. Hence, $w(S_1) + w(S_2) \leqslant w(S) = m(G, S)$. On the other hand, since as noticed just above, S_2 is not present in S'', S_2

and S'' are also disjoint, both sets being subsets of S. Hence, again $w(S'') + w(S_2) \leqslant m(G, S)$. Putting so these things together with (6), we get: $m(G', S') \leqslant 2m(G, S)$. Taking finally into account that since the vertices of T are, by assumption, always present in V', $\mathrm{opt}(G') \geqslant \mathrm{opt}(G)$, we get from (1):

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

In order to prove tightness, 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,

$$E^*(K_{n+1}) = p_2(n+1) + (1-p_2)(n+1) = n+1 \tag{7}$$

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:

$$E(K_{n+1}, S, \mathtt{DFS}) = p_2(n+1) + (1-p_2)\, 2n = 2n - p_2 n + p_2 \tag{8}$$

In Figure 3 an illustration of the discussion just above is provided for $n = 6$. The thick edges of K_7 in Figure 3(a) are the ones with cost 2. It is assumed that vertices $1, 3, 4, 5, 6, 7$ are terminals. In Figure 3(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 3(c) is shown the DFS completion of S when 2 is absent. Finally, in Figure 3(d), the optimal Steiner tree of K_7 using only terminals built as described above is shown.

Dividing (7) by (8) we get a ratio that for small enough values of p_2 tends to 2. ∎

3 Some comments

By Proposition 3, if for example the 1.55-approximation algorithm of [30] is used for constructing an anticipatory solution, a 3.1-approximation for PROBABILISTIC STEINER

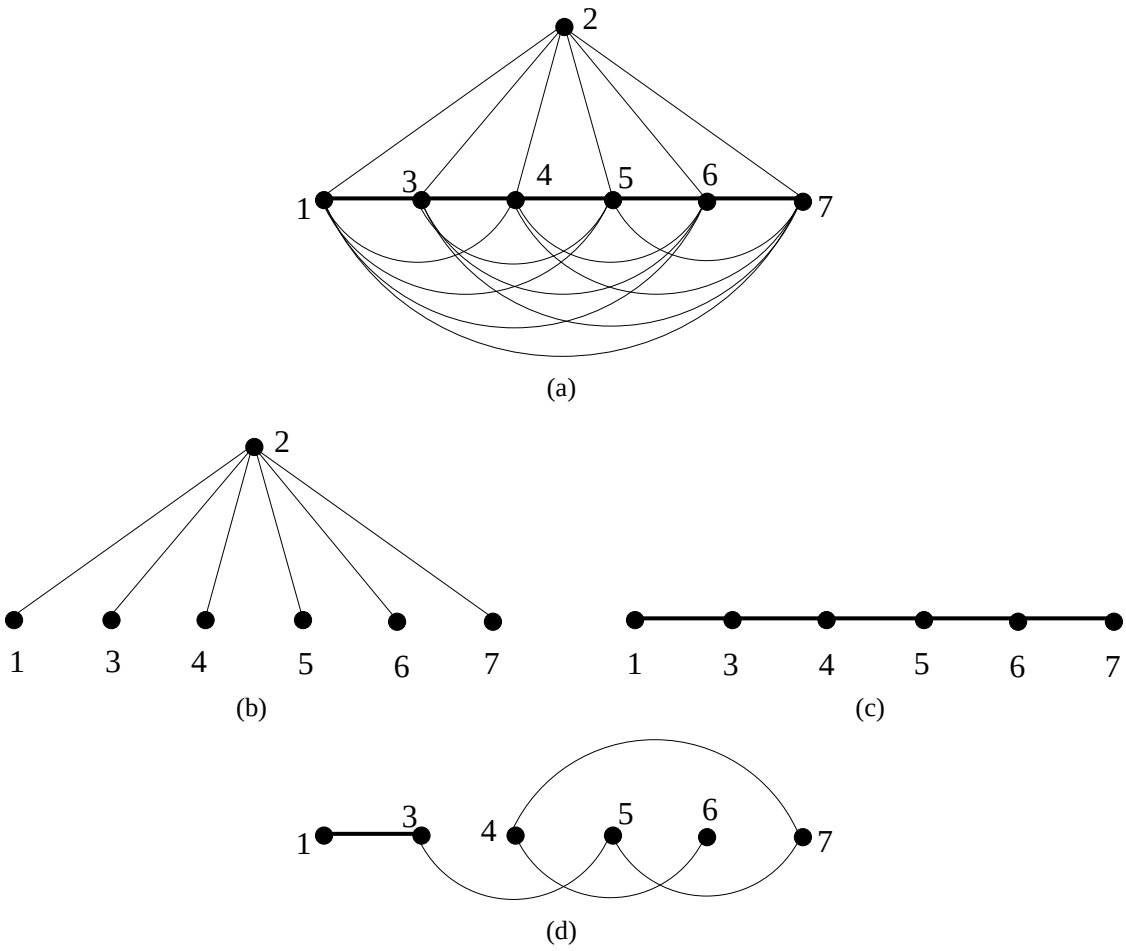


Figure 3: On the tightness of ratio 2ρ.

TREE is incurred. This result is obviously dominated by Proposition 2. However, Theorem 3 provides a more general structural result linking the approximation of STEINER TREE to the one of PROBABILISTIC STEINER TREE and carries over more general solution structures. This, to our opinion, has its own interest.

But, if one restricts to anticipatory solutions that are optimal for STEINER TREE in the input-graph, then Proposition 3 implies a tight approximation ratio 2 for PROBABILISTIC STEINER TREE. In this case the result incurred is also of practical usefulness since it encompasses more general structures of anticipatory solutions and the approximation ratio obtained is comparable to the one claimed by Proposition 2.

Before concluding the paper let us note that 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 noticed in Section 1, S^* is an optimal anticipatory solution. By (3), $E^*(G)$ is a lower bound for $E(G, S^*, \mathtt{DFS})$. Henceforth, the approximation results derived in this section remain valid when using approximation ratio $E(G, S, \mathtt{M})/E(G, S^*, \mathtt{M})$ also.

Acknowledgment. Many thanks to Cécile Murat for helpful discussions and comments. The comments and suggestions of an anonymous referee 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 J. Comput.*, 24(3):440–456, 1995.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, Reading, MA, 1975.

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

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

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

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

[7] 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`.

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

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

[10] 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.

[11] 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.

[12] 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 industry*. Kluwer Academic Publishers, 2000.

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

[14] 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.

[15] 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.

[16] 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.

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

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

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

[20] 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.

[21] 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.

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

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

[24] 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`.

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

[26] C. Murat and V. Th. Paschos. *Probabilistic combinatorial optimization on graphs*. ISTE and Hermès Science Publishing, London, 2006.

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

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

[29] 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.

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

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

[32] 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.

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

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

[35] 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.