# On the PROBABILISTIC MIN SPANNING TREE problem[*]

Nicolas Boria[1]        Cécile Murat[1]        Vangelis Th. Paschos[1,2]

[1]LAMSADE, CNRS and Université Paris-Dauphine, France
{boria,murat,paschos}@lamsade.dauphine.fr
[2]Institut Universitaire de France

March 28, 2011

### Abstract

We study a probabilistic optimization model for MIN SPANNING TREE, where any vertex $v_i$ of the input-graph $G(V, E)$ has some presence probability $p_i$ in the final instance $G' \subset G$ that will effectively be optimized. Suppose that when this "real" instance $G'$ becomes known, a spanning tree $T$, called *anticipatory* or *a priori* spanning tree, has already been computed in $G$ and one can run a quick algorithm (quicker than one that recomputes from scratch), called *modification strategy*, that modifies the anticipatory tree $T$ in order to fit $G'$. The goal is to compute an anticipatory spanning tree of $G$ such that, its modification for any $G' \subseteq G$ is optimal for $G'$. This is what we call PROBABILISTIC MIN SPANNING TREE problem. In this paper we study complexity and approximation of PROBABILISTIC MIN SPANNING TREE in complete graphs under two distinct modification strategies leading to different complexity results for the problem. For the first of the strategies developed, we also study two natural subproblems of PROBABILISTIC MIN SPANNING TREE, namely, the PROBABILISTIC METRIC MIN SPANNING TREE and the PROBABILISTIC MIN SPANNING TREE 1,2 that deal with metric complete graphs and complete graphs with edge-weights either 1, or 2, respectively.

## 1 Introduction

The basic problematic of probabilistic combinatorial optimization (in graphs) is the following. We are given a graph $G(V, E)$ on which we have to solve some optimization problem $\Pi$. But, for some reasons depending on the reality modeled by $G$, $\Pi$ is only going to be solved for some subgraph $G'$ of $G$ (determined by the vertices that will finally be present) rather than for the whole of $G$. The measure of how likely it is that a vertex $v_i \in V$ will belong to $G'$ (i.e., will be present for the final optimization) is expressed by a probability $p_i$ associated with $v_i$. How we can proceed in order to solve $\Pi$ under this kind of uncertainty?

A first very natural idea that comes to mind is that one waits until $G'$ is specified (i.e., it is present and ready for optimization) and, at this time, one solves $\Pi$ in $G'$. This is what is called *reoptimization*. But what if there remains very little time for such a computation? In this case, another way to proceed is the following. One solves $\Pi$ in the whole of $G$ in order to get a feasible solution (denoted by $S$), called *a priori* or *anticipatory solution*, which will serve her/him as a kind of benchmark for the solution on the effectively present subgraph $G'$. One has also to be provided with an algorithm that modifies $S$ in order to fit $G'$. This algorithm is called *modification strategy* (let us denote it by M) and produces from $S$ a solution $S'$ that is feasible for $\Pi$ in $G'$. The objective now becomes to compute an anticipatory solution that, when

1

modified by M, remains "good" for any subgraph of $G$ (if this subgraph is the one where $\Pi$ will be finally solved). This amounts to computing a solution that optimizes a kind of expectation of the value of the modification of $S$ over all the possible subgraphs of $G$, i.e., the sum of the products of the probability that $G'$ is the finally present graph multiplied by the value of $S'$. Obviously, the presence-probability of $G'$ is the probability that all of its vertices are present and the other vertices outside $G'$ are absent. This expectation, depending on both the instance of the deterministic problem $\Pi$, the vertex-probabilities, and the modification strategy adopted, is called the *functional* and is expressed by:

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

where $\Pr[V'] = \prod_{v_i \in V'} p_i \prod_{v_i \in V \setminus V'} (1 - p_i)$ is the distribution describing probability of occurrence of a specific subset $V' \subseteq V$, i.e., of the graph $G[V']$ and $m(G', S', \mathtt{M})$ is the value of the solution $S'$ produced by application of M on the anticipatory solution $S$.

Seen in this way, the probabilistic version PROBABILISTIC $\Pi$ of a (deterministic) combinatorial optimization problem $\Pi$ becomes another equally deterministic problem $\Pi'$, the solutions of which have the same feasibility constraints as those of $\Pi$ but with a different objective function where vertex-probabilities intervene. In this sense, probabilistic combinatorial optimization is very close to what in the last couple of years has been called "one stage optimisation under independent decision models", an area very popular in the stochastic optimization community. It is hopefully clear from (1) and from the whole discussion above, that $m(G', S', \mathtt{M})$ strongly depends on the modification strategy M used to adapt the anticipatory solution $S$ to the present graph $G'$. So, both $E(G, S, \mathtt{M})$ and the a priori solution $S^*$ optimizing it, also strongly depend on M. In other words, as we will see later, for a fixed instance $I$ of a deterministic problem $\Pi$, *two distinct modification strategies induce two distinct probabilistic problems* having $I$ and $\Pi$ as common deterministic supports. These two distinct problems may have very different functionals and, consequently, different optimal anticipatory solutions that induce very different complexities for computing them.

What are the main mathematical problems dealing with probabilistic consideration of a problem $\Pi$ in the sense discussed above? One can identify at least five interesting mathematical and computational problems dealing with probabilistic combinatorial optimization:

1. write the functional down in an analytical closed form;

2. if such an expression of the functional is possible, prove that its value is polynomially computable (this amounts to proving that the decision version of the modified problem $\Pi'$ belongs to **NP**);

3. determine the complexity of the computation of the optimal *a priori* solution, i.e., of the solution optimizing the functional (in other words, determine the computational complexity of $\Pi'$);

4. if $\Pi'$ is **NP**-hard, study polynomial approximation issues;

5. always, under the hypothesis of the **NP**-hardness of $\Pi'$, determine its complexity in the special cases where $\Pi$ is polynomial, and in the case of **NP**-hardness, study approximation issues.

Let us note that, although curious, point 2 in the above list is neither trivial nor senseless. Simply consider that the summation for the functional includes, in a graph of order $n$, $2^n$ terms (one for

each subgraph of $G$). So, polynomiality of the computation of the functional is, in general, not immediate.

Several optimization frameworks have been introduced by the operations research community for handling data uncertainty, the most well developed being *Stochastic programming* (see [7, 14, 30] for basics) and *Robust discrete optimization* (see, for example, [22]). The framework of *Probabilistic Combinatorial Optimization* where our work is located was introduced by [17, 2]. In [1, 2, 3, 4, 5, 17, 18, 19, 20], restricted versions of routing and network-design probabilistic minimization problems (in complete graphs) have been studied under the robustness model dealt here (called *a priori optimization*). In [6], the analysis of the probabilistic minimum travelling salesman problem, originally performed in [2, 17], has been revisited and refined. Several other combinatorial problems have been recently treated in the probabilistic combinatorial optimization framework, including minimum coloring ([27, 8]), maximum independent set and minimum vertex cover ([25, 26]), longest path ([24]), Steiner tree problems ([28, 29]). Note also that probabilistic minimum spanning tree has also studied by [4] but under a very different probabilistic model.

## 2  Preliminaries

We apply in this paper the probabilistic combinatorial optimization setting just described in the minimum spanning tree problem. Given an edge-weighted graph $G(V, E)$, with positive edge weights $d : E \to \mathbb{Q}^+$, the minimum spanning tree problem (MIN SPANNING TREE) consists of determining a minimum total edge-weight tree spanning $V$. MIN SPANNING TREE is a celebrated problem, very frequently modeling several kinds of networks in transports, communications, energy, logistics, etc.

The most popular algorithm for MIN SPANNING TREE is the seminal Kruskal's Algorithm ([23]) which consists of sorting edges in non-decreasing order with respect to their weights, and of inserting them in the current solution provided that they do not create cycles with those already inserted. The complexity of this algorithm is $O(m \log n)$ where $m = |E|$ and $n = |V|$. Several improvements have been performed upon this result, for instance, the $O(m \log \log n)$ result given independently by [32] and [11]. More recently, [10] proposes an algorithm for MIN SPANNING TREE (the best known until now, to our knowledge) with running time $O(m\alpha(m, n))$, where $\alpha$ is the functional inverse of Ackerman's function defined by [31]. We also quote an implementation of Prim's algorithm (this is another famous MIN SPANNING TREE-algorithm) using Fibonacci heaps that works in $O(\max\{m, n \log n\})$ ([13]).

MIN SPANNING TREE has been actively studied under several optimization models like on-line computation, dynamic optimization, etc. Its study always motivates numerous researchers in theoretical computer science and in operational research.

The formal definition of PROBABILISTIC MIN SPANNING TREE under modification strategy M is as follows. Consider a complete weighted graph $G(V, E)$ on $n$ vertices, with edge weights given by a function $d : E \to \mathbb{Q}^+$. Set $V = \{v_1, v_2, \ldots, v_n\}$. Each vertex $v_i \in V$, is associated with a presence probability $p_i \in \mathbb{Q}^+$ measuring, as already mentioned, how likely is that $v_i$ will be present in the instance where PROBABILISTIC MIN SPANNING TREE will really be solved. We assume that subgraph $G'(V', E')$ of $G$ materializes as the outcome of $n$ independent Bernoulli trials, one per vertex $v_i \in V$: $v_i \in V'$ with probability $p_i$. Then, $E' = \{(u, v) \in E : u \in V' \text{ and } v \in V'\}$. Let us note that it seems to be natural that, for a fixed modification strategy M and for some anticipatory spanning tree $T$, some basic properties of its structure must be preserved in any tree $T'$ built when M adjusts $T$ to $G'(V', E')$, for any $V' \subseteq V$. Such a basic property is, for instance, the relation "predecessor-successor" in $T$. In order that this relation is preserved in any $T'$, we assume that there exists a vertex, denoted by $v_1$ with $p_1 = 1$. The assumption that the input-graph is complete is made in order to ensure connectivity of the tree $T'$ for any

subgraph $G' \subseteq G$; in any case if the real-world problem at hand implies non-complete graphs, one can complete them by "appropriately heavy" edges.

Consider an application of PROBABILISTIC MIN SPANNING TREE coming from message broadcasting in a computer network ([9]). We use there a tree spanning the vertices of this network. Assuming that the weight of an edge of the network represents the transmission-cost on the link represented by this edge, we look for a minimum-cost tree spanning all the vertices of the underlying network. In distributed computation, the structure of a tree allows broadcasting from the root to the leaves, and vice-versa, as well as resolution of several problems dealing with synchronization, or mutual exclusion ([9]). Assume that each vertex $v_i$ of the network can fail or be unavailable with some probability $p_i$, except some vertex, denoted by $v_1$, that represents the central server that "never" fails (or very rarely). In such case, we can consider that the presence probability of $v_1$ is 1 and of any other vertex $v_i$ is $1 - p_i$. Given some modification strategy, motivated by the particular application to be handled, the objective is to determine some spanning tree with minimum expectation. Other applications dealing with PROBABILISTIC MIN SPANNING TREE can be found in [2].

In what follows, we design two modification strategies for PROBABILISTIC MIN SPANNING TREE and study the two distinct probabilistic optimization problems derived. In Section 4 we design strategy `CLOSEST_ANCESTOR`, and derive an analytic expression of the functional of the solutions $T^*$. We next show that, under this strategy the problem of the *a priori optimization*, i.e., the problem of determining an anticipatory solution minimizing the functional, is **NP**-hard in general complete graphs (Section 4.1). Subsequently, we study complexity of the PROBABILISTIC MIN SPANNING TREE problem when dealing with particular cases of vertex-probabilities values and/or edge weights (Section 4.2.1) and particular cases of anticipatory solutions (Section 4.2.2). We next derive polynomial-time approximation results for metric graphs (Section 4.3.1) and for graphs where edge-weights are either 1 or 2 (Section 4.3.2). Next, in Section 5, we devise another modification strategy for PROBABILISTIC MIN SPANNING TREE, called `ROOT`, and show that, under this strategy, PROBABILISTIC MIN SPANNING TREE is polynomial.

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]. Let $\mathrm{opt}(G')$ refer to the weight of the optimum spanning tree on $G'$ for every subgraph $G'(V', E')$ of $G$. The expected minimum weight over the distribution of subgraphs of $G$, i.e., the functional of reoptimization is:

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

Obviously, denoting by $T_{\texttt{M}}^*$ the optimal anticipatory solution of PROBABILISTIC MIN SPANNING TREE, under any modification strategy `M`, it holds that:

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

In Section 3, we perform further study of reoptimization and exhibit some relations between it and (deterministic) MIN SPANNING TREE.

For simplicity, in Sections 4 and 5 we simplify notations and omit strategies' names from the expressions for the functionals. However, we underline once more the fact that the probabilistic versions of PROBABILISTIC MIN SPANNING TREE studied in these sections are indeed two distinct optimization problems.

## 3  Reoptimization and MIN SPANNING TREE

We study in this section relations between the value of the expectation $E^*(G)$ in the reoptimization setting and the optimal value of the (deterministic) MIN SPANNING TREE. Our goal is to

prove the following result.

**Proposition 1.** *Consider a complete edge-weighted graph $G$ defined on a set $V$ of $n$ vertices $V = \{v_1, \ldots, v_n\}$ associated with a system of vertex probabilities $p_1 = 1$, $p_i = p$, $i = 2, \ldots, n$. Denote by $\mathrm{opt}(G)$ the value of an optimal solution $T^*$ for (deterministic)* MIN SPANNING TREE. *Then, $E^*(G) \geqslant p\,\mathrm{opt}(G)$.*

**Proof.** Since vertex $v_1$ (assumed to be the root of every tree solution of MIN SPANNING TREE in every subgraph of $G$) is always present, setting $G' = G[V']$, (2) can be written as:

$$E^*(G) = \sum_{k=2}^{n} p^{k-1}(1-p)^{n-k} \times \sum_{\substack{V' \subseteq V \\ |V'|=k}} \mathrm{opt}\left(G'\right) \tag{4}$$

Set $D_k = \sum_{V' \subseteq V, |V'|=k} \mathrm{opt}(G')$. Then, (4) becomes:

$$E^*(G) = \sum_{k=2}^{n} p^{k-1}(1-p)^{n-k} D_k \tag{5}$$

We first show that:

$$D_k \geqslant \binom{n-2}{k-2} \mathrm{opt}(G) \tag{6}$$

Denote by $E[T^*]$ the edge-set of the tree $T^*$. Then, for every $i \neq 1$, set $V_i = V \setminus \{v_i\}$ and denote by $f(i)$, the father of $i$ in $T^*$. Setting $G_i = G[V_i]$, we have for every $(v_i, v_{f(i)}) \in E[T^*]$, $i \neq 1$:

$$\mathrm{opt}\left(G_i\right) + d_{i,f(i)} \geqslant \mathrm{opt}(G) \tag{7}$$

since a minimum spanning tree of $G_i$ plus some edge connecting $v_i$ to it is a tree of $G$. Summing (7) over $i = 2, \ldots, n$, we get:

$$\sum_{i=2}^{n} \mathrm{opt}\left(G_i\right) + \sum_{i=2}^{n} d_{i,f(i)} \geqslant (n-1)\,\mathrm{opt}(G) \tag{8}$$

Moreover, obviously, it holds that

$$\sum_{i=2}^{n} d_{i,f(i)} = \mathrm{opt}(G) \tag{9}$$

From (8) and (9) the following is deduced:

$$D_{n-1} = \sum_{i=2}^{n} \mathrm{opt}\left(G_i\right) \geqslant (n-2)\,\mathrm{opt}(G) \tag{10}$$

Let us now consider the $t = C_{n-1}^{k-1}$ subsets of $V$ of size $k$ (recall that $v_1$ is omnipresent), $A_1, A_2, \ldots, A_t$. For any $i$, set $A_{i,j} = A_i \setminus \{j\}$. With a reasoning completely similar to the above, setting $G_{i,j} = G[A_{i,j}]$, we can derive:

$$\sum_{j \in A_i} \mathrm{opt}\left(G_{i,j}\right) \;\geqslant\; (k-2)\,\mathrm{opt}\left(G_i\right) \tag{11}$$

$$\sum_{i=1}^{t} \sum_{j \in A_i} \mathrm{opt}\left(G_{i,j}\right) \;\geqslant\; (k-2)\sum_{i=1}^{t} \mathrm{opt}\left(G_i\right) \;=\; (k-2)D_k \tag{12}$$

5

where (12) is obtained summing (11) for every $i$. Since in the lefthand side term each of the $C_{n-1}^{k-2}$ subsets of $V$ (of cardinality $k-1$) is taken into account $n-k+1$ times, this term can be written:

$$\sum_{i=1}^{t} \sum_{j \in A_i} \mathrm{opt}\,(G_{i,j}) = (n-k+1)D_{k-1} \tag{13}$$

Indeed, w.l.o.g. let us assume that $A_1 = \{v_1, v_2, \ldots, v_{k-1}, v_k\}$, $A_2 = \{v_1, v_2, \ldots, v_{k-1}, v_{k+1}\}$, $A_3 = \{v_1, v_2, \ldots, v_{k-1}, v_{k+2}\}$, ..., $A_{n-k+1} = \{v_1, v_2, \ldots, v_{k-1}, v_n\}$. Let us consider set $A_{1,k}$ obtained from $A_1$ by removing vertex $v_k$:

$$\overbrace{\underbrace{v_1, v_2, \ldots, v_{k-1}, v_k}_{A_1}}^{A_{1,k}}, \ldots, v_n$$

We build in the same way pairs $(A_2, A_{2,k+1})$, $(A_3, A_{3,k+2})$, ..., $(A_{n-k+1}, A_{n-k+1,n})$:

$$\overbrace{\underbrace{v_1, v_2, \ldots, v_{k-1}, v_{k+1}}_{A_2}}^{A_{2,k+1}}, v_k, v_{k+2}, \ldots, v_n$$

$$\vdots$$

$$\overbrace{\underbrace{v_1, v_2, \ldots, v_{k-1}, v_n}_{A_{n-k+1}}}^{A_{n-k+1,n}}, v_k, \ldots, v_{n-1}$$

We can easily see that $A_{1,k} = A_{2,k+1} = A_{3,k+2} = \ldots = A_{n-k+1,n}$; hence, any of the $C_{n-1}^{k-2}$ distinct subsets of $V$ of cardinality $k-1$ can be obtained using $n-k+1$ different ways, that proves (13).

Combining (12) and (13) we get:

$$D_{k-1} \geqslant \frac{k-2}{n-k+1}D_k \tag{14}$$

By successively applying (14) and using (10) we derive:

$$\begin{aligned}
D_k &\geqslant \frac{k-1}{n-k}D_{k+1} \geqslant \frac{(k-1)k}{(n-k)(n-(k+1))}D_{k+2} \\
&\vdots \\
&\geqslant \frac{(k-1)k\ldots(n-3)}{(n-k)(n-(k+1))\ldots(n-(n-2))}D_{n-1} \\
&\geqslant \frac{(k-1)k\ldots(n-3)(n-2)}{(n-k)(n-(k+1))\ldots(n-(n-1))}\mathrm{opt}(G) \\
&= \frac{(n-2)!}{(k-2)!(n-k)!}\mathrm{opt}(G) = \binom{n-2}{k-2}\mathrm{opt}(G)
\end{aligned}$$

as claimed in (6). Combining it with (5), we get:

$$E^*(G) \geqslant \sum_{k=2}^{n} p^{k-1}(1-p)^{n-k}\binom{n-2}{k-2}\mathrm{opt}(G) = p\,\mathrm{opt}(G)$$

completing so the proof of the proposition. ∎

Combining (3) and Proposition 1, the following holds, for any modification strategy M:

$$E\,(G, T_{\mathtt{M}}^*) \geqslant p\,\mathrm{opt}(G) \tag{15}$$

Note that equality in (15) is attained for $p=1$.

6

## 4   PROBABILISTIC MIN SPANNING TREE under CLOSEST_ANCESTOR

Consider a tree $T$ spanning $V$ and number vertices in $T$ in a left-to-right breadth-first-search (bfs) way. Consider a subgraph $G'(V', E') = G[V']$ of $G$ induced by a set $V' \subseteq V$. The modification strategy (adjusting $T$ to a spanning tree $T'$ of $G'$ and denoted by CLOSEST_ANCESTOR in what follows) that will be analyzed in this section works as follows:

- remove the vertices of $V \setminus V'$ and the edges of $E$ incident to these vertices; let $F(V') = \{T_1, T_2, \ldots, T_k\}$ be the so-obtained forest and assume that $v_1 \in V(T_1)$ and that, for $i, j = 2 \ldots, k$, $i < j$ if the index of the root of $T_i$ is smaller than that of the root of $T_j$;

- for $i = 2, \ldots, k$ add as father of the root of $T_i$ its largest-index ancestor that is still present in $V(T)$.

Note that, given two vertices $v_j$ and $v_l$ with $j < l$ (in the bfs numbering of $T$), if $v_j$ is not an ancestor of $v_l$ in $T$, then edge $(v_j, v_l)$ will never belong to any $T'$ modification of $T$ for any $V' \subseteq V$. Clearly, the complexity of CLOSEST_ANCESTOR is $O(n)$. Note also that, with respect to the application sketched in Section 1, this strategy allows to maintain the same order of broadcasting to the several internal nodes of the anticipatory tree, independently on the sub-instance that will finally be realized.
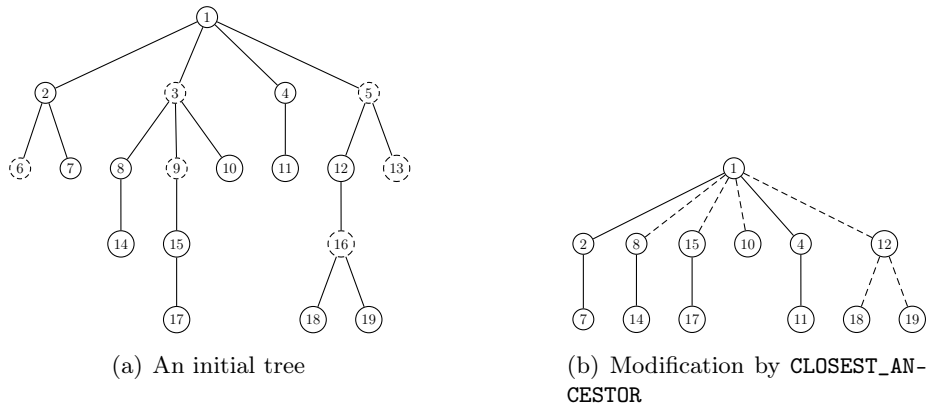


(a) An initial tree

(b) Modification by CLOSEST_AN-CESTOR

Figure 1: An anticipatory spanning tree $T$ and its adjustment.

Figure 1(b) gives an example of how strategy CLOSEST_ANCESTOR works starting from an initial tree $T$ shown in Figure 1(a) and assuming that vertices 3, 5, 6, 9, 13 and 16 are absent from $V'$.

Our goal in this section is to study the following problem: *find an algorithm for taking "good" a priori decisions, i.e., that determines a spanning tree $T^*$, that optimizes $E(G, T)$ under modification strategy CLOSEST_ANCESTOR*; this is PROBABILISTIC MIN SPANNING TREE (under strategy CLOSEST_ANCESTOR). In what follows, we show that this problem is **NP**-hard in general complete graphs with $p_1 = 1$. We then study approximation of this problem in metric graphs as well as in a particular subclass of them where edge-weights are either 1 or 2. The approximation ratio is defined as $E(G, T)/E(G, T^*)$.

### 4.1   The complexity of PROBABILISTIC MIN SPANNING TREE

The following result holds for the functional $E(G, T)$ associated with an anticipatory spanning tree $T$ and the modification strategy CLOSEST_ANCESTOR.

**Proposition 2.** *Consider a complete graph $G(V, E)$, provided with a vertex-probability system $(p_i)_{i=1,\ldots,n}$ with $p_1 = 1$, any edge $(v_i, v_j)$ of which has weight $d_{ij}$ and a spanning tree $T$ of $G$. Then, the expectation associated with* `CLOSEST_ANCESTOR` *can be expressed by:*

$$E(G, T) = \sum_{(v_i, v_j) \in T} p_i p_j d_{ij} + \sum_{v_i \in V} \sum_{\substack{v_j \in D(v_i) \\ (v_i, v_j) \notin T}} p_i p_j \prod_{v_k \in \mu[v_i, v_j]} (1 - p_k) \, d_{ij} \quad (16)$$

*where $D(v_i)$ is the set of successors of $v_i$ in $T$ and $\mu[v_i, v_j]$ is the set of vertices in the (unique) path of $T$ from $v_i$ to $v_j$ not including neither of them.*

**Proof.** Following `CLOSEST_ANCESTOR`, if $(v_i, v_j) \in T$, then this edge will be in $T'$ iff $v_i, v_j \in V'$; on the other hand, if $(v_i, v_j) \notin T$, then edge $(v_i, v_j)$ will be added in $T'$ iff $v_i, v_j \in V'$ and $v_j \in D(v_i)$ and every vertex in $\mu[v_i, v_j]$ is not in $V'$. From these observations we derive using (1):

$$
\begin{aligned}
E(G, T) &= \sum_{V' \subseteq V} \Pr[V'] \, m(G', T') = \sum_{V' \subseteq V} \Pr[V'] \sum_{(v_i, v_j) \in T'} d_{ij} \\
&= \sum_{V' \subseteq V} \Pr[V'] \sum_{(v_i, v_j) \in T' \cap T} d_{ij} + \sum_{V' \subseteq V} \Pr[V'] \sum_{(v_i, v_j) \in T' \cap (E \setminus T)} d_{ij} \\
&= \sum_{(v_i, v_j) \in T} \sum_{V' \subseteq V} \Pr[V'] \, \mathbf{1}_{\{(v_i, v_j) \in T'\}} d_{ij} + \sum_{(v_i, v_j) \in (E \setminus T)} \sum_{V' \subseteq V} \Pr[V'] \, \mathbf{1}_{\{(v_i, v_j) \in T'\}} d_{ij} \\
&= \sum_{(v_i, v_j) \in T} p_i p_j d_{ij} + \sum_{v_i \in V} \sum_{\substack{v_j \in D(v_i) \\ (v_i, v_j) \notin T}} p_i p_j \prod_{v_k \in \mu[v_i, v_j]} (1 - p_k) \, d_{ij}
\end{aligned}
$$

as claimed. ∎

Clearly, (16) can be computed in polynomial time. So, the following corollary holds.

**Corollary 1.** PROBABILISTIC MIN SPANNING TREE $\in$ **NPO**.

As already mentioned, PROBABILISTIC MIN SPANNING TREE consists of determining an anticipatory spanning tree $T^*$ of $G$ minimizing $E(G, T)$. Unfortunately, Proposition 2 does not derive a compact combinatorial characterization for the optimal anticipatory solution of PROBABILISTIC MIN SPANNING TREE. In particular, the form of the functional does not imply solution, for instance, of some well-defined weighted version of the (deterministic) MIN SPANNING TREE. This is due to the second term of the expression for $E(G, T)$ in (16). There, the "costs" assigned to the edges depend on the structure of the anticipatory solution chosen and of the present subgraph of $G$.

The decision version of PROBABILISTIC MIN SPANNING TREE under `CLOSEST_ANCESTOR`, denoted by PROBABILISTIC MIN SPANNING TREE($K$) can be stated as follows: "given an edge-weighted complete graph $G(V, E)$, provided with a vertex-probability system $(p_i)_{i=1,\ldots,n}$ with $p_1 = 1$ and a constant $K$, does there exist a tree $T$ such that $E(G, T) \leqslant K$?", where $E(G, T)$ is given by (16).

**Proposition 3.** PROBABILISTIC MIN SPANNING TREE($K$) *is* **NP**-complete.

**Proof.** It is easy to see that PROBABILISTIC MIN SPANNING TREE $\in$ **NP** (Corollary 1). In order to show completeness, we reduce 3 EXACT COVER to PROBABILISTIC MIN SPANNING TREE. 3 EXACT COVER is defined as follows: given a ground set $X$ of size $3q$ and a collection $\mathcal{E}$ of $3q$ subsets of $X$ each of size 3, does there exist a sub-collection $\mathcal{E}' = \{S_1, \ldots, S_q\}$ of $\mathcal{E}$ such that $\bigcup_{i=1}^q S_i = X$?" (obviously, $\mathcal{E}'$ is a partition of $X$).

Consider an arbitrary instance $I(X, \mathcal{E})$ of 3 EXACT COVER; we construct the following instance for PROBABILISTIC MIN SPANNING TREE:

- the vertex-set $V$ is a set of $6q + 2$ vertices built by associating a vertex $x_i$ with an element $x_i \in X$, a vertex $y_j$ with a set $S_j \in \mathcal{E}$ and by adding a vertex $r$ (playing the role of the omnipresent root) and a vertex $s$ (representing the solution); for some positive fixed constant $p < 1/2$, vertices $x_i$ are provided with probability $p$, vertices $y_j$ with probability $1 - p$ and vertices $r$ and $s$ with probability 1;

- edge-weights are defined as follows:
    - for every $S_j = \{x_{i_1}, x_{i_2}, x_{i_3}\}$, $j = 1, \ldots, 3q$, $d_{i_1 j} = d_{i_2 j} = d_{i_3 j} = 1$;
    - edges linking $s$ to vertices $y_j$ have weight $M > 3p^2/2(1 - p)$ and those linking $s$ to vertices of $x_i$ have weight $M/p + 2$;
    - edges linking $r$ to vertices $y_j$ as well as edge $(r, s)$ have all weight 0, while edges linking $r$ to vertices $x_i$ have weight $M/p^2 + 1$;
    - all the other edges have arbitrarily large weight $B \gg M/p^2 + 1$;

- $K = q(M(1 + 2p) + 3p(p + 1))$.

It is easy to see that this reduction is polynomial. It is illustrated in Figure 2 where, for readability, some edges, in particular those of weight $B$ are omitted.
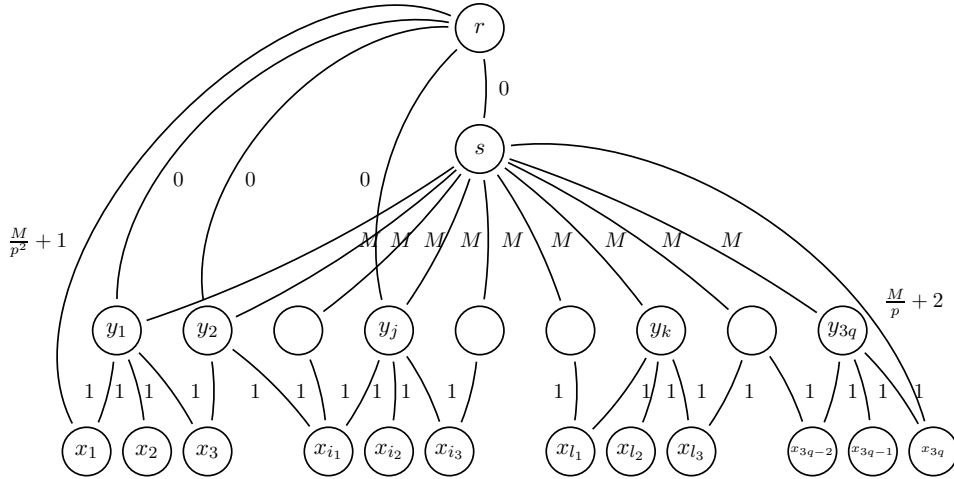


Figure 2: An example for the reduction from 3 EXACT COVER to PROBABILISTIC MIN SPANNING TREE.

We now prove that if 3 EXACT COVER admits a solution $\mathcal{E}^*$, then $G$ has a minimum spanning tree $T^*$ the shape of which is as in Figure 3 and whose value is:

$$
\begin{aligned}
E(G, T^*) &= 0 + 2q \times 0 + q\left[(1 - p)M + 3p(1 - p) \times 1 + 3p(1 - (1 - p)) \times \left(\frac{M}{p} + 2\right)\right] \\
&= q\left[M - pM + 3p - 3p^2 + 3pM + 6p^2\right] \\
&= q\left[M(1 + 2p) + 3p(p + 1)\right] = K
\end{aligned}
$$

In other words, we are going to show that there exists a tree spanning $V$ with expectation at most $K$, iff $X$ admits a partition among the sets of $\mathcal{E}$.

The $\Rightarrow$ part of the proof is easy. Given a solution $\mathcal{E}^*$ for 3 EXACT COVER in $I(X, \mathcal{E})$, a solution $T^*$ for PROBABILISTIC MIN SPANNING TREE is immediately built in $G$ as shown in Figure 3.
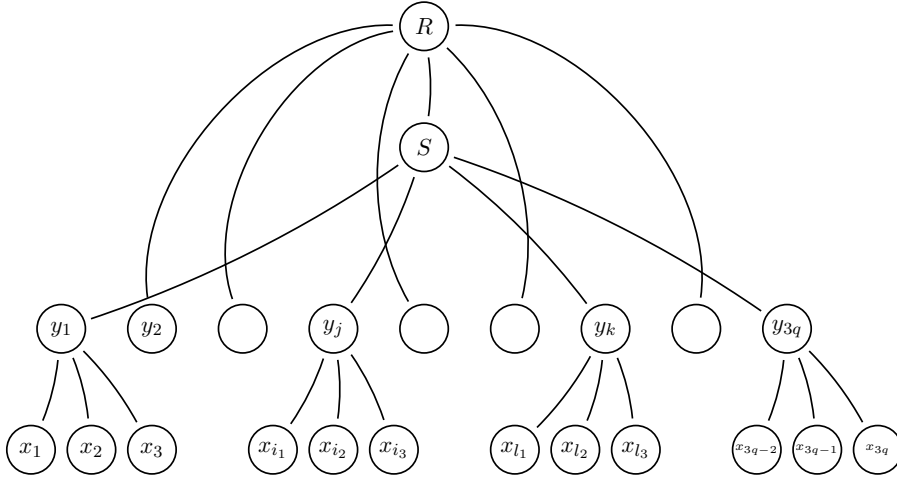
9

Figure 3: The shape of $T^*$.

We prove the $\Leftarrow$ part by inspection of all the possible trees spanning $V$. Let us do two easy preliminary remarks:

- any tree using an edge of weight $B$ has expectation greater than $K$;

- any tree where absent edges are replaced (following strategy CLOSEST_ANCESTOR) by edges of weight $B$ also has expectation greater than $K$.

As a consequence, the trees that are to be inspected are those that any reconnection will not use edges of weight $B$, i.e., those that span the subgraph shown in Figure 4. There exist 16 such trees illustrated in Figures 5 to 8. For reasons of economy, in all these figures, only one out of the three children of each $y_i$ is displayed. Note that tree $T_1$ in Figure 5 corresponds to the optimal solution, whose full shape is shown in Figure 3.
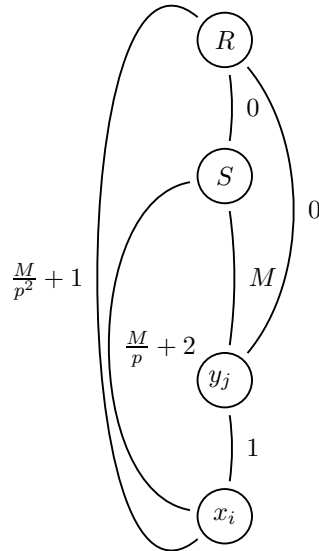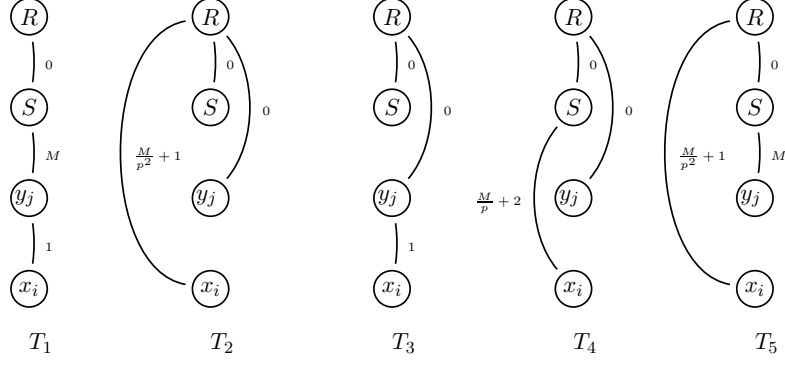


Figure 4: The "interesting" subgraph of $G$.

10

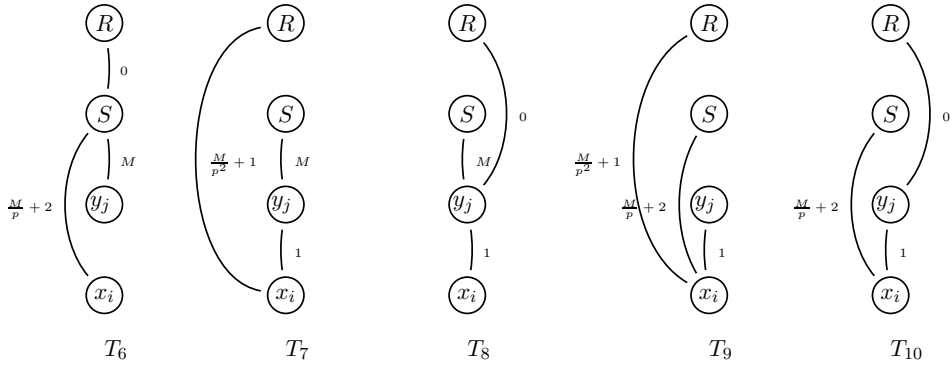Figure 5: 5 out of 16 trees spanning component of Figure 4 ...



Figure 6: ...other 5 out of 16 trees spanning component of Figure 4 ...

In all, trees in Figures 5 to 8 have the following functionals (computed over their full shapes, i.e., considering all the three children of each $y_i$):

$$
\begin{aligned}
E\left(G, T_1\right) & = 1 \times 1 \times 0 + 1 \times (1-p) \times M + 3 \times (1-p) \times p \times 1 \\
& \quad + 3 \times p \times (1 - (1-p)) \times \left(\frac{M}{p} + 2\right) \\
& = M(1-p) + 3p(1-p) + 3pM + 6p^2 \quad = \quad M(1+2p) + 3p(p+1) \\
E\left(G, T_2\right) & = 3p\left(\frac{M}{p^2} + 1\right) \quad = \quad \frac{3}{p} \times M + 3p \\
E\left(G, T_3\right) & = 3(1-p)p + 3p^2\left(\frac{M}{p^2} + 1\right) \quad = \quad 3M + 3p \\
E\left(G, T_4\right) & = 3p\left(\frac{M}{p} + 2\right) \quad = \quad 3M + 6p \\
E\left(G, T_5\right) & = (1-p)M + 3p\left(\frac{M}{p^2} + 1\right) \quad = \quad M\left(1 - p + \frac{3}{p}\right) + 3p \\
E\left(G, T_6\right) & = (1-p)M + 3p\left(\frac{M}{p} + 2\right) \quad = \quad M(4-p) + 6p \\
E\left(G, T_7\right) & = (1-p)M + (1-p)p + 3p\left(\frac{M}{p^2} + 1\right) + p^2\left(\frac{M}{p} + 2\right) \quad = \quad M\left(\frac{3}{p} + 1\right) + 4p + p^2
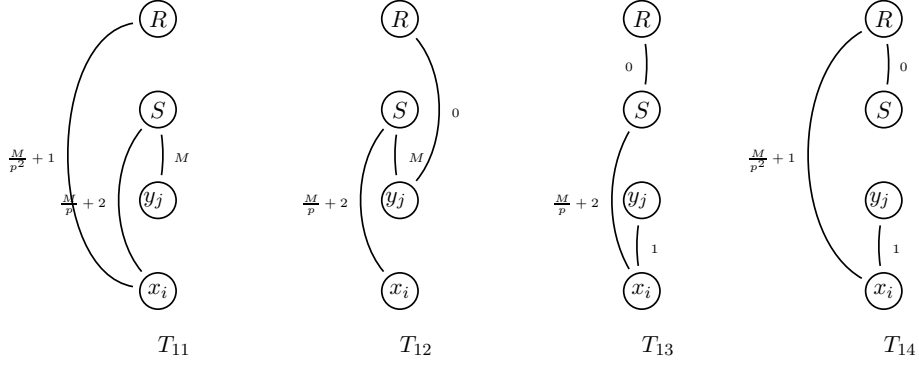\end{aligned}
$$

11

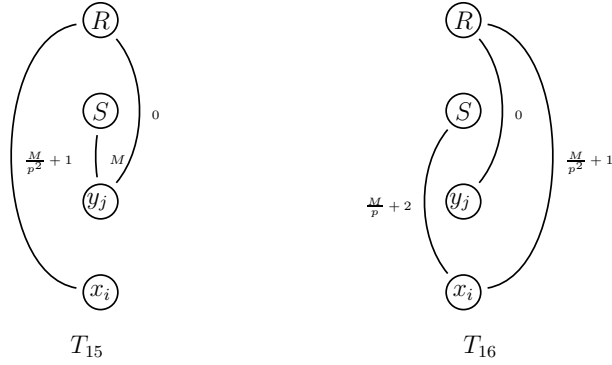Figure 7: ...other 4 out of 16 trees spanning component of Figure 4.



Figure 8: ...and the last 2 out of 16 trees spanning component of Figure 4.

$$E\left(G,T_8\right) = (1-p)M + 3p(1-p) + 3p^2\left(\frac{M}{p^2}+1\right) = M(4-p) + 3p$$

$$E\left(G,T_9\right) = 3p\left(\frac{M}{p^2}+1\right) + p\left(\frac{M}{p}+2\right) + p(1-p) = M\left(1+\frac{3}{p}\right) + 6p - p^2$$

$$E\left(G,T_{10}\right) = p\left(\frac{M}{p}+2\right) + 3p(1-p) + 3p^2\left(\frac{M}{p^2}+1\right) + (1-p)^2M = M\left(5-2p+p^2\right) + 5p$$

$$E\left(G,T_{11}\right) = (1-p)M + p\left(\frac{M}{p}+2\right) + 3p\left(\frac{M}{p^2}+1\right) = M\left(2-p+\frac{3}{p}\right) + 5p$$

$$E\left(G,T_{12}\right) = (1-p)M + 3p\left(\frac{M}{p}+2\right) = M(4-p) + 6p$$

$$E\left(G,T_{13}\right) = p(1-p) + 3p\left(\frac{M}{p}+2\right) + (1-p)^2M = M\left(3+(1-p)^2\right) + 7p - p^2$$

$$E\left(G,T_{14}\right) = p(1-p) + 3p\left(\frac{M}{p^2}+1\right) = M\frac{3}{p} + 4p - p^2$$

$$E\left(G,T_{15}\right) = (1-p)M + 3p\left(\frac{M}{p^2}+1\right) = M\left(\frac{3}{p}+1-p\right) + 3p$$

$$E\left(G,T_{16}\right) = 3p\left(\frac{M}{p^2}+1\right) + p\left(\frac{M}{p}+2\right) = M\left(\frac{3}{p}+1\right) + 5p$$

12

Trees $T_2$, $T_3$, $T_4$, $T_9$, $T_{10}$, $T_{13}$ and $T_{14}$ correspond to the case where $y_j$, that is, the set $S_j$ of $\mathcal{E}$ is not part of a 3 EXACT COVER-solution (in other words, the element $x_i$ of the ground set $X$ is not covered by $S_j$). In this case, the best one can do for the corresponding spanning tree is to take edges $(r, s)$ and $(r, y_j)$. The rest of the trees sketched in Figures 5 to 8 correspond to the case where $S_j$ covers $x_i$ in some 3 EXACT COVER-solution.

In all, $E(G, T_1)$ is minimum over every other $E(G, T_i)$, $i = 2, \ldots, 16$ it is the only that is smaller than $K$. Tree $T_1$ corresponds to $q$ subtrees as the first one in Figure 5 but having three leaves plus $2q$ subtrees that are simple edges of the form $(r, y_j)$. Obviously, given $T_1$ a solution for 3 EXACT COVER is immediately computed by considering the $q$ sets corresponding to $y$-vertices of $T_1$ that are not leaves. ∎

## 4.2 Stability of the solutions

We address in this section the question of "stability of the solutions" that can be stated as follows: "under which conditions and for what kinds of instances and/or particular structures of anticipatory solutions, an optimal solution for MIN SPANNING TREE remains optimal for PROBABILISTIC MIN SPANNING TREE?". We study in this section some particular but natural cases carrying over assumptions either on the values of vertex-probabilities and/or edge-weights, or on the form of the anticipatory solution.

### 4.2.1 Dealing with vertex-probabilities and edge-weights

Revisit functional's expression (16). For a vertex $v_i$, denote by $f(v_i)$ its father in $T$, by $p_{f(i)}$ the presence probability of $f(v_i)$ and by $A(v_i)$ the set of its ancestors in $T$. Then, (16) can be rewritten as:

$$
\begin{aligned}
E(G, T) &= \sum_{v_i \in V \setminus \{v_1\}} p_i p_{f(i)} d_{i f(v_i)} + \sum_{v_i \in V \setminus \{v_1\}} \sum_{v_j \in A(v_i) \setminus \{f(v_i)\}} p_i p_j \prod_{v_k \in \mu[v_i, v_j]} (1 - p_k)\, d_{ij} \\
&= \sum_{v_i \in V \setminus \{v_1\}} C_i
\end{aligned}
\tag{17}
$$

where:

$$
C_i = p_i p_{f(i)} d_{i f(v_i)} + \sum_{v_j \in A(v_i) \setminus \{f(v_i)\}} p_i p_j \prod_{v_k \in \mu[v_i, v_j]} (1 - p_k)\, d_{ij}
\tag{18}
$$

and can be seen as the contribution of vertex $v_i$ in $E(G, T)$.

We assume now that edge-weights in $G$ are identical. We can show that, if $d_{ij} = d$, $(v_i, v_j) \in E$, then $C_i = d \times p_i$, $v_i \in V$. In order to give some intuition about it let us consider the anticipatory tree of Figure 9, assume that edge weights in the input-graph are identical and equal to $d$ and take, say, vertex 7. The contribution of it in (16) is:

$$
C_7 = d \times p_7 \left[ p_5 + p_4 (1 - p_5) + p_2 (1 - p_5)(1 - p_4) + (1 - p_5)(1 - p_4)(1 - p_2) \right] = d \times p_7
\tag{19}
$$

The same holds for the contribution of any other vertex in the tree.

Indeed, consider some vertex $v_i \in V$ and assume, for simplicity, that vertices in the path of $T$ from $v_1$ to $v_i$ are numbered from 1 to $i$. By writing down $C_i$ and by some algebra as previously in (19) we derive $C_i = d \times p_i$. Hence, (17) becomes:

$$
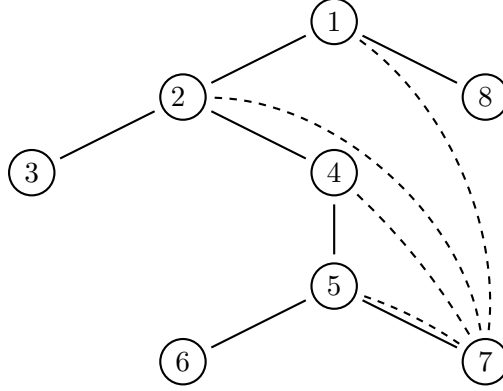E(G, T) = d \times \sum_{i=2}^{n} p_i
$$

Figure 9: About a reinterpretation of functional's expression (16).

**Proposition 4.** *If edge-weights are all identical, $E(G, T) = d \times \sum_{i=2}^{n} p_i$. In this case all the anticipatory solutions have the same value.*

**Corollary 2.** *If $p_1 = 1$, $p_i = p$, $i = 2, \ldots, n$ and $d_{ij} = d$, $(v_i, v_j) \in E$, $i \neq j$, then, for any tree $T$ spanning $V$, $E(G, T) = dp(n - 1)$.*

Discussion above can be directly generalized for deriving a general upper bound for $E(G, T)$ and for every anticipatory solution $T$. Set $D = \max\{d_{ij} : (v_i, v_j) \in E\}$. Then, (20) leads to $C_i \leqslant D \times p_i$ and (17) to $E(G, T) \leqslant D \times \sum_{i=2}^{n} p_i$ and the following corollary that will be used later holds.

**Corollary 3.** *If $D = \max\{d_{ij} : (v_i, v_j) \in E\}$ then, for any anticipatory solution $T$ of* PROBA-BILISTIC MIN SPANNING TREE, *$E(G, T) \leqslant D \times \sum_{i=2}^{n} p_i$.*

### 4.2.2 Particular cases of anticipatory solutions: stars and paths

In what follows we deal with two particular structures of trees, the star (rooted at $v_1$) and the path. We first consider the star rooted at (the omnipresent) vertex $v_1$. The following result holds.

**Proposition 5.** *Let $T$ be a star rooted at $v_1$. If $T$ is an optimal solution for* MIN SPANNING TREE, *then it is also an optimal anticipatory solution for* PROBABILISTIC MIN SPANNING TREE.

**Proof.** Recall that by (17), $E(G, T) = \sum_{v_i \in V \setminus \{v_1\}} C_i$ where $C_i$ is given by (18). Observe now that, if the vertices of $T$ are numbered in a dfs order (starting from the root) and if the set $A(v_i)$ of the ancestors of a vertex $v_i$ in $T$ is exactly the set $A(v_i) = \{v_1, v_2, \ldots, v_{i-1}\}$, then $C_i$ can be written as:

$$C_i = \sum_{j=1}^{i-1} p_i p_j d_{ij} \prod_{l=j+1}^{i-1} (1 - p_l) \tag{20}$$

Since the star $T$ is a minimum spanning tree, it holds that, for any vertex $i$, $d_{1i} \leqslant d_{ij}$, for every $j \neq 1, i$. Hence, $C_i \geqslant p_i \times d_{1i} \times \sum_{j=1}^{i-1} p_j \prod_{l=j+1}^{i-1} (1 - p_l)$.

If we denote by $C_i^T$ the contribution of vertex $v_i$ in the functional $E(G, T)$ of the star $T$, then, for every $i$, $C_i^T = p_i \times d_{1i}$. So, in order to complete the proof of the proposition, we have

to show that, for any $v_i \in V$, $C_i^T \leqslant C_i$, where $C_i$ refers to every other spanning tree of $G$. For this, it suffices to prove that:

$$\sum_{j=1}^{i-1} p_j \prod_{l=j+1}^{i-1} (1 - p_l) \geqslant 1 \tag{21}$$

We show (21) by induction on $i$. For $i = 2$, the lefthand side of (21) is equal to $p_1 = 1$, so the inequality claimed is true. Suppose it true for $i = n$, i.e.:

$$\sum_{j=1}^{n-1} p_j \prod_{l=j+1}^{n-1} (1 - p_l) \geqslant 1 \tag{22}$$

Then, at range $n + 1$ it holds:

$$
\sum_{j=1}^{n} p_j \prod_{l=j+1}^{n} (1 - p_l) = \sum_{j=1}^{n-1} p_j \prod_{l=j+1}^{n} (1 - p_l) + p_n = \sum_{j=1}^{n-1} p_j \prod_{l=j+1}^{n-1} (1 - p_l) \times (1 - p_n) + p_n
$$

$$
= (1 - p_n) \times \sum_{j=1}^{n-1} p_j \prod_{l=j+1}^{n-1} (1 - p_l) + p_n \overset{(22)}{\geqslant} (1 - p_n) + p_n = 1
$$

The proof of the proposition is now completed. ∎

Unfortunately, in the case where optimal solution for MIN SPANNING TREE is a path, optimality of such a solution for PROBABILISTIC MIN SPANNING TREE cannot be derived as previously in the case of stars.

|        | $v_1$ | $v_2$ | $v_3$ | $\cdots$ | $v_{n-2}$ | $v_{n-1}$ | $v_n$ |
|--------|-------|-------|-------|----------|-----------|-----------|-------|
| $v_1$     | 0     | 1     | 2     | $\cdots$ | 2         | 2         | 2     |
| $v_2$     | 1     | 0     | 1     | $\cdots$ | 2         | 2         | 2     |
| $v_3$     | 2     | 1     | 0     | $\cdots$ | 2         | 2         | 2     |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $v_{n-2}$ | 2     | 2     | 2     | $\cdots$ | 0         | 1         | $K$   |
| $v_{n-1}$ | 2     | 2     | 2     | $\cdots$ | 1         | 0         | 1     |
| $v_n$     | 2     | 2     | 2     | $\cdots$ | $K$       | 1         | 0     |

Table 1: The adjacency matrix of a graph $G$ where optimal solutions for MIN SPANNING TREE and PROBABILISTIC MIN SPANNING TREE do not coincide.

Indeed consider a complete graph $G$, the adjacency matrix of which is given in Table 1 and its vertex-probability system is $(1, p \ldots, p)$, with $p < (K - 2)/(K - 1)$ and $K \geqslant n$. Optimal MIN SPANNING TREE-solution in $G$ is unique and is the path $P = (1, 2, \ldots, n - 1)$ with value $n - 1$. The functional $E(G, P)$ of path $P$ is $E(G, P) = (2n - 3)p + (K - n)p^2 - (K - 2)p^3$. On the other hand, the unique optimal anticipatory solution for PROBABILISTIC MIN SPANNING TREE is the tree $T^*$ of Figure 10 with functional's value $E(G, T^*) = (2n - 3)p + (2 - n)p^2 + p^3 < E(G, P)$, when $p < (K - 2)/(K - 1)$.

### 4.3  PROBABILISTIC METRIC MIN SPANNING TREE

In this section, we study PROBABILISTIC METRIC MIN SPANNING TREE problem, that is PROBABILISTIC MIN SPANNING TREE in metric complete graphs, i.e., in complete graphs whose edge-weights satisfy the triangular property that can be expressed as follows: if $(v_i, v_j, v_k)$ is a $K_3$ in $G$, then $d_{ab} \leqslant d_{ac} + d_{bc}$, for any permutation $(a, b, c)$ of $\{i, j, k\}$.
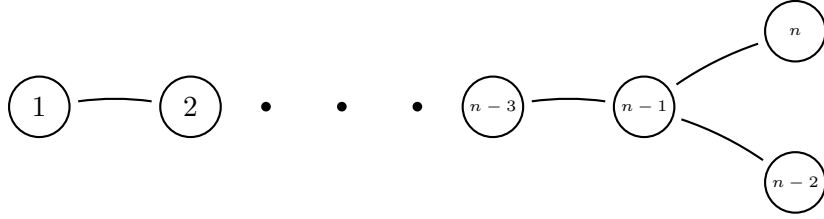
Figure 10: The optimal solution of the graph of Table 1.

Note that, as it can be seen from the proof of Proposition 3, it does not apply in the case of metric graphs. Indeed, the complexity status of PROBABILISTIC METRIC MIN SPANNING TREE remains open, even if we feel that this variant is also **NP**-complete. In what follows we give approximation results for the general case of PROBABILISTIC METRIC MIN SPANNING TREE as well as for a natural subcase, namely PROBABILISTIC MIN SPANNING TREE 1,2, where edge weights are either 1, or 2.

We denote by $\hat{T}$ and $T^*$ a tree computed by Kruskal's algorithm (i.e., a minimum spanning tree of $G$) and an optimal anticipatory solution for PROBABILISTIC METRIC MIN SPANNING TREE (under `CLOSEST_ANCESTOR`), respectively, and we assume that they are represented as sets of edges. Observe also that, by the metric property the weight of any edge of $G$ is smaller than the weight of any spanning tree of $G$ and a fortiori than the weight $m(G, \hat{T}, \texttt{CLOSEST\_ANCESTOR}) = \text{opt}(G)$ of $\hat{T}$. Indeed, let $(v_i, v_j)$ be any edge of $G$, $T$ be some spanning tree of $G$ and $P(v_i, v_j)$ be the unique path from $v_i$ to $v_j$ in $T$. According to the metric hypothesis, $d_{ij} \leqslant w(P(v_i, v_j)) \leqslant m(G, T)$, where $w(P(v_i, v_j))$ is the weight of the path $P(v_i, v_j)$. In what follows, given a set $Q$ of (weighted) edges, we denote by $w(Q)$ its total weight.

### 4.3.1 General metric case

#### 4.3.1.1 The average approximation ratio of a minimum spanning tree

For any $V_i \subseteq V$, we set $G_i(V_i, E_i) = G[V_i]$. Let $T_i^*$ and $\hat{T}_i$ be the spanning trees on $G_i$ resulting from the application of strategy `CLOSEST_ANCESTOR` on $T^*$ and $\hat{T}$, respectively. Set $r(G_i) = m(G_i, \hat{T}_i)/m(G_i, T_i^*)$ and $E(r) = \sum_{V_i \subseteq V} \Pr[V_i] r(G_i)$. Quantity $E(r)$ is indeed the average approximation ratio of a minimum spanning tree for PROBABILISTIC METRIC MIN SPANNING TREE.

**Proposition 6.** $E(r) \leqslant (n+2)/4$.

**Proof.** Fix an induced subgraph $G_i$ of $G$ and let $\hat{T} \cap \hat{T}_i = S$. Edges of $S$ are part of an optimal spanning tree on $G$ and thus, they are also part of an optimal spanning tree of $G_i$. Indeed, revisiting the proof of optimality of Kruskal's Algorithm, one can see that a tree $T$ is a minimum spanning tree on $G$, iff all the edges of $T$ are of minimum weight in at least one cut of $G$. Applying this to PROBABILISTIC MIN SPANNING TREE, any edge $e$ belonging to $S$ is of minimum weight in at least one cut of $G$; thus, $e$ is also of minimum weight in one cut in any subgraph $G_i$ (provided that $e$ appears in $G_i$) and, therefore, it belongs to a minimum spanning tree in all the subgraphs of $G$ where it is present.

Discussion just above leads to:

$$w(S) \leqslant m(G_i, T_i^*) \tag{23}$$

The edge-set $\hat{T}_i \setminus S$ is the set of the edges used by `CLOSEST_ANCESTOR` to reconnect the $S$. As observed in the beginning of Section 4.3, the weight of each edge of $\hat{T}_i \setminus S$ is smaller than, or

16

equal to, $m(G_i, T_i^*)$, so:

$$w\left(\hat{T}_i \setminus S\right) \leqslant \left|\hat{T}_i \setminus S\right| m\left(G_i, T_i^*\right) \tag{24}$$

Combining (23) and (24), we get:

$$r_i = \frac{m\left(G_i, \hat{T}_i\right)}{m\left(G_i, T_i^*\right)} \leqslant 1 + \left|\hat{T}_i \setminus S\right| \tag{25}$$

The quantity $|\hat{T}_i \setminus S|$ is, as mentioned above, the number of edges inserted by CLOSEST_ANCESTOR to reconnect $S$, but it also represents the number of vertices present in $G_i$, but whose fathers in $\hat{T}$ (assumed rooted at $v_1$) are absent from $G_i$. For each vertex of $\hat{T}$ except for those directly connected to the root $v_1$, the probability to be present in $G_i$ but not its father is $p(1-p)$. Obviously, for the vertices directly connected to the root, this probability is 0. In order to count the number of edges in $\hat{T}_i \setminus S$, one can consider a set of $n - 1 - X$ Bernoulli trials (where $X$ is the number of vertices directly connected to $v_1$ in $\hat{T}$), with a probability of success $p(1-p)$, each success adding an edge to $\hat{T}_i \setminus S$. In this way, $|\hat{T}_i \setminus S|$ is a random variable following a binomial law, so one can directly compute its expectation:

$$\left|\hat{T}_i \setminus S\right| \sim B\left(n - 1 - X, p(1-p)\right)$$
$$E\left(\left|\hat{T}_i \setminus S\right|\right) = (n - 1 - X)p(1-p) \tag{26}$$

Summing (25) for each $G_i$, we derive: $E(r) \leqslant 1 + E(|\hat{T}_i \setminus S|)$ and combining it with (26), we can easily get:

$$E(r) \leqslant 1 + (n - 1 - X)p(1-p) \overset{X \geqslant 1}{\leqslant} \frac{n}{4} + \frac{1}{2} = \frac{n+2}{4}$$

as claimed. ∎

#### 4.3.1.2   The approximation of PROBABILISTIC METRIC MIN SPANNING TREE

Revisit for a while the proof of Proposition 6 in Section 4.3.1.1. Notice that $w(S) \leqslant \text{opt}(G)$ and that the weight of the edge-set $\hat{T}_i \setminus S$ satisfies $w(\hat{T}_i \setminus S) \leqslant |\hat{T}_i \setminus S| \, \text{opt}(G)$. So, performing the same analysis and replacing $T_i^*$ with $T^*$, we finally get:

$$\frac{E\left(G, \hat{T}\right)}{\text{opt}(G)} \leqslant 1 + (n - 2)p(1-p) \tag{27}$$

Combination of (27) and (15) immediately leads to:

$$\frac{E\left(G, \hat{T}\right)}{E\left(G, T^*\right)} \leqslant 1 + (n - 2)(1-p) \tag{28}$$

In the same spirit of the analysis of Christofides' algorithm ([12]), consider now a left-to-right dfs ordering of the vertices of $\hat{T}$, the non-elementary path produced by this dfs traversal of $\hat{T}$ and observe that in this traversal any edge is encountered at most twice. This path can be transformed into an elementary Hamiltonian path $T'$ by using suitable shortcuts as shown in Figure 11(b) where these shortcuts are the boldfaced lines (edges) $(5,6), (8,9), (9,10), (10,11), (12,13)$ and the whole path $T'$ is simply the path $(i, i+1)$, $i = 1, \ldots, 12$ (vertices of $\hat{T}$ in Figure 11(a) are numbered in a left-to-right dfs order). By the metric property and the dfs numbering of the vertices of the tree, the weight of a shortcut $(i,j)$, $i < j$, is bounded above by the total weight

of the path of $\hat{T}$ that starting from $i$ "climbs" the tree up to the lowest common ancestor $k$ of $i$ and $j$ and then goes down to $j$. With respect to this shortcut, edges of the path from $k$ to $i$ will count at most twice in $T'$ (either as real edges of $T'$, or for shortcut's bound). Furthermore, they will not further be encountered in any other shortcut. In Figure 11(b), the small lines crossing an edge of $\hat{T}$ indicate the number of times this edge encounters in $T'$. So, since $T'$ is also a spanning tree for $G$:

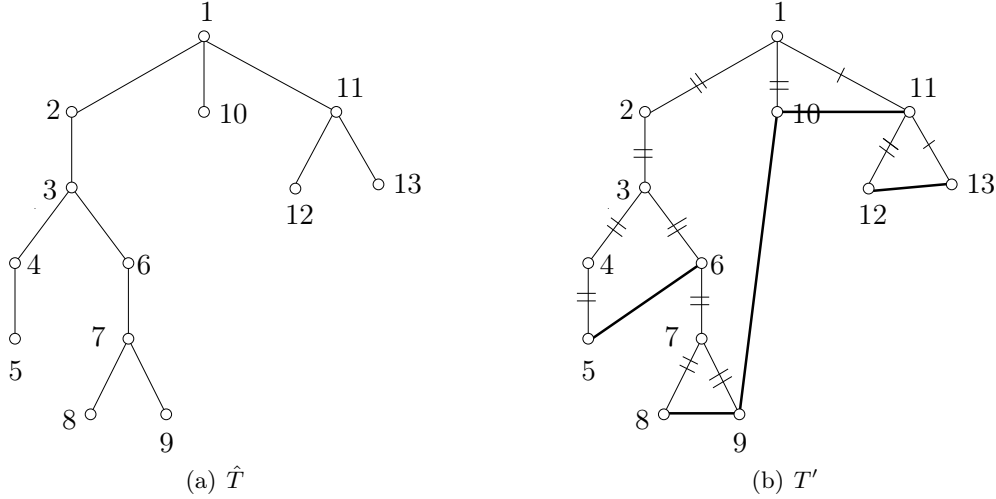$$m\left(G, T'\right) \leqslant 2\operatorname{opt}(G) \tag{29}$$



Figure 11: A tree $\hat{T}$ and the Hamiltonian path $T'$ produced from it.

Since $T'$ is a Hamiltonian path having the omnipresent vertex $v_1$ as one of its endpoints, any modification of it by the modification strategy `CLOSEST_ANCESTOR` due to the absence of any subset $V_i \subseteq V$, will produce a path $T'_i$ that always remains Hamiltonian for the graph $G'_i = G[V \setminus V_i]$. By the metric hypothesis, for any $V_i \subseteq V$:

$$m\left(G'_i, T'_i\right) \leqslant m\left(G, T'\right) \tag{30}$$

Then, the functional associated with $T'$ can be written as:

$$
\begin{aligned}
E\left(G, T'\right) &= \sum_{V_i \subseteq V} \Pr\left[V \setminus V_i\right] m\left(G'_i, T'_i\right) \overset{(30)}{\leqslant} m\left(G, T'\right) \times \sum_{V_i \subseteq V} \Pr\left[V \setminus V_i\right] \tag{31} \\
&= m\left(G, T'\right) \overset{(29)}{\leqslant} 2\operatorname{opt}(G) \overset{(15)}{\leqslant} \frac{2}{p} \times E\left(G, T^*\right) \tag{32}
\end{aligned}
$$

So, $T'$ is a $(2/p)$-approximation for PROBABILISTIC METRIC MIN SPANNING TREE and taking the best (with respect to the functional) among $\hat{T}$ and $T'$, the following result holds (according to (28) and (31) and concludes this subsection.

**Proposition 7.** PROBABILISTIC METRIC MIN SPANNING TREE *is approximable in polynomial time within ratio bounded above by* $\min\{1 + (n-2)(1-p), 2/p\}$.

### 4.3.2 PROBABILISTIC MIN SPANNING TREE 1,2

We now focus on a particular but natural and well-studied class of metric complete graphs where edge weights are either 1 or 2. It is easy to see that any such graph is metric.

We first prove the following easy approximation result for PROBABILISTIC MIN SPANNING TREE 1,2 that will be subsequently improved.

**Proposition 8.** *A minimum spanning tree of $G$ is a $(2-p)$-approximation for* PROBABILISTIC MIN SPANNING TREE 1,2.

**Proof.** Consider a minimum spanning tree $\hat{T}$ of $G$ computed, for instance, by Kruskal's algorithm. Then, the following fact holds.

**Fact 1.** If $\hat{T}$ has $k$ edges of weight 2, then in $E(G, T^*)$, (i.e., the optimal value for PROBABILISTIC MIN SPANNING TREE 1,2 under CLOSEST_ANCESTOR) there exist at least $k$ vertices with contribution $2p$ (recall that by (17), the functional can be defined as the sum of the contributions of the vertices, each such contribution expressed by (20)). ∎

Indeed, if $\hat{T}$ has $k$ edges of weight 2, then, by construction of $\hat{T}$, there exist $k+1$ distinct connected components $V_i$ in the graph $G'(V, E')$ (that contains only the edges of $G$ of weight 1); edges between any two vertices of any two of these components have weight 2 (suppose, w.l.o.g., that the omnipresent vertex $v_1$ belongs to $V_1$). So, the following holds:

$$\forall i \in \{1, \ldots, k+1\}, \ \forall v_j \in V_i, v_k \notin V_i, \ d_{jk} = 2 \tag{33}$$

Assume now a bfs on $\hat{T}$, starting from $v_1$ and denote by $u_i$ the first-visited vertex of $V_i$ (obviously, $u_1 = v_1$). For every $i > 1$, all the vertices visited before $u_i$ (a fortiori those on the path from $v_1$ to $u_i$), do not belong to $V_i$; hence, by (33), every edge linking $u_i$ to any of these vertices has weight 2. So, for $i = 2, \ldots, k+1$, the contribution $C(u_i)$ of $u_i$ to $E(G, \hat{T})$ is $2p$.

Discussion above holds for any spanning tree of $G$ and a fortiori for $T^*$; hence, truth of Fact 1 is immediately derived.

As a consequence of the discussion above, $T^*$ has at least $k$ vertices with contribution $2p$ to the functional; observe furthermore that the contribution of any of the other $n - k - 1$ vertices (recall that vertex $v_1$ has null contribution) is at least $p$. So:

$$E(G, T^*) \geqslant 2p \times k + (n - k - 1) \times p = p \times (n - 1 + k) \tag{34}$$

Revisit now $\hat{T}$ and consider that a vertex is "covered" by the edge of $\hat{T}$ entering it in the bfs visiting. Based upon the discussion above, there are $k$ vertices covered by edges of weight 2 and $n - k - 1$ vertices covered by edges of weight 1. Consider a vertex $w$ covered by an edge of weight 1. It is easy to see from (20) that its contribution $C(w)$ is at most $p(2-p)$ (considering that all the other edges incident to it in $G$ have weight at most 2). So:

$$E\left(G, \hat{T}\right) \leqslant p \times (2-p) \times (n - 1 - k) + 2pk = p(n-1) \times (2-p) + p^2 \times k \tag{35}$$

Combining (34) and (35) we get after some easy algebra: $E(G, \hat{T})/E(G, T^*) \leqslant 2 - p$ as claimed. ∎

In what follows we refine the result above. For this we consider an execution of Kruskal's algorithm that starts by introducing in the tree all the edges of weight 1 incident to the vertex $v_1$. Let us denote by $\tilde{T}$ the spanning tree so constructed; notice that $\tilde{T}$ is a minimum spanning tree for $G$.

**Proposition 9.** *The tree $\tilde{T}$ approximates $T^*$ within ratio:*

$$\frac{1 + (2-p)(n-2)}{n - 1 + \frac{(1-p)^2 - (1-p)^n}{p}}$$

**Proof.** Recall that, by Proposition 8, if the value of a minimum spanning tree $\tilde{T}$ is equal to $n+k$ then, for every spanning tree $T$ there exist at least $k$ (exactly $k$ for $\tilde{T}$) vertices each contributing with $2p$ to the functional associated with $T$. For simplicity, assume that this number is exactly $k$ for $T$ also (if not, i.e., if we have more than $k$ such vertices, the bounds derived below are even better). Among the remaining (less than) $n-1-k$ vertices of $T$ let us denote by $n_1$ the number of those that in $G$ are connected to $v_1$ by edges of weight 1 and by $U_2$ the set of those that in $G$ are connected to $v_1$ by edges of weight 2; set finally $n_2 = |U_2|$. Obviously, $n-1 = n_1 + n_2 + k$.

Consider a vertex $v_i \in U_2$, suppose that its height is $i$ and recall that, by definition, $d_{i1} = 2$. Then, the contribution $C_i$ of $v_i$ (see also (20)) in $E(G, T)$ is at least:

$$C_i \geqslant p\left(1 + (1-p)^{i-1}\right) \tag{36}$$

where in (36) it has been assumed that only edge $(v_1, v_i)$ has weight 2, while every other edge on the path from $v_1$ to $v_i$ in $T$ has weight 1.

Number the vertices of $T$ as they appear in a bfs ordering of $T$ and let $\ell_1, \ell_2, \ldots, \ell_{n_2}$ be the bfs numbers of the vertices of $U_2$. Fix a vertex $v_i \in U_2$ with bfs number, say, $\ell_i$ and notice that the $n_2 - i$ vertices of $U_2$ numbered $\ell_{i+1}, \ldots, \ell_{n_2}$ have all height at least equal to the height of $v_i$. The height of $v_i$ is at most $n - 1 - (n_2 - i) = n - 1 - n_2 + i$; hence, (36) becomes:

$$C_i \geqslant p\left(1 + (1-p)^{n-n_2+i-2}\right) \tag{37}$$

Using (37) and denoting by $C(U_2)$ the total contribution of the vertices of $U_2$ in $E(G, T)$, we get:

$$\begin{aligned} C(U_2) &\geqslant \sum_{i=n-n_2}^{n-1} p \times \left(1 + (1-p)^{i-1}\right) = pn_2 + p \times \sum_{i=n-n_2}^{n-1} (1-p)^i \\ &= p \times \left(n_2 + \frac{(1-p)^{n-n_2} - (1-p)^n}{p}\right) \end{aligned} \tag{38}$$

What it has been discussed until now holds for any spanning tree $T$, a fortiori for the optimal anticipatory tree $T^*$. So, using (38), the functional $E(G, T^*)$ becomes:

$$E(G, T^*) \geqslant p \times n_1 + p \times \left(n_2 + \frac{(1-p)^{n-n_2} - (1-p)^n}{p}\right) + 2p \times k \tag{39}$$

On the other hand, consider the tree $\tilde{T}$ and recall that, by its construction, $n_1$ of its vertices are directly linked to $v_1$. Then, their contribution to $E(G, \tilde{T})$ is equal to $pn_1$. According to the proof of Proposition 8, the contribution of any of the $n_2$ vertices linked in $G$ to $v_1$ by edges of weight 2 but covered in $\tilde{T}$ by edges of weight 1 is at most $p(2-p)$ and the contribution of any of the remaining $k$ vertices (covered in $\tilde{T}$ by edges of weight 2) is $2p$. Putting all this together, we derive:

$$E\left(G, \tilde{T}\right) \leqslant p \times n_1 + p(2-p) \times n_2 + 2p \times k \tag{40}$$

Combining (39) and (40) we get:

$$\frac{E\left(G, \tilde{T}\right)}{E(G, T^*)} \leqslant \frac{p \times n_1 + p(2-p) \times n_2 + 2p \times k}{p \times n_1 + p \times \left(n_2 + \frac{(1-p)^{n-n_2}-(1-p)^n}{p}\right) + 2p \times k} \tag{41}$$

$$\leqslant \frac{n_1 + (2-p) \times n_2}{n_1 + n_2 + \frac{(1-p)^{n-n_2}-(1-p)^n}{p}} \tag{42}$$

$$\leqslant \frac{n_1 + (2-p)(n-1-n_1)}{n-1 + \frac{(1-p)^{n_1-1}-(1-p)^n}{p}} \leqslant \frac{1 + (2-p)(n-2)}{n-1 + \frac{(1-p)^2-(1-p)^n}{p}} \tag{43}$$

where the rightmost expression of (42) is due to the fact that ratio's expression is decreasing with $k \geqslant 0$, the first expression of (43) is due to the fact that since $k$ is set to 0, $n - 1 = n_1 + n_2$ and the second expression of (43) is due to the fact that the first expression is decreasing with $n_1$ but $k$ and $n_1$ cannot be simultaneously equal to 0 (if $n_1 = 0$, then no vertex of $G$ is linked to $v_1$ with an edge of weight 1, so $k \geqslant 1$), hence, $n_1 \geqslant 1$. This proves the ratio claimed.
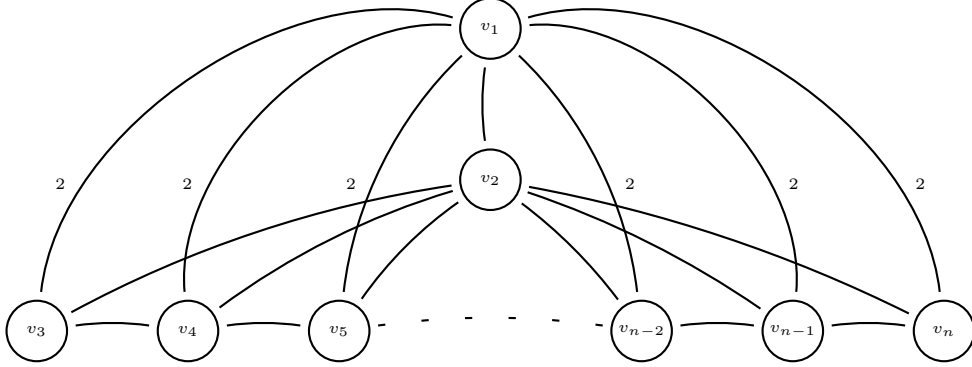


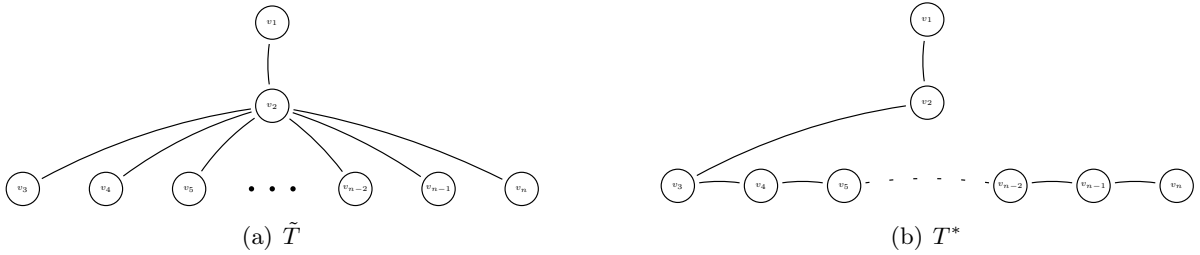Figure 12: A graph where the ratio claimed in Proposition 9 is tight.



Figure 13: The trees $\tilde{T}$ and $T^*$ from the graph of Figure 12.

For tightness, consider the graph of Figure 12. There, $\forall i \in \{3, \ldots, n\}$, $d_{1i} = 2$; every other edge has weight 1. So, any tree where $v_1$ is linked to $v_2$ is a tree $\tilde{T}$, potentially built by an execution of Kruskal's algorithm as that considered in the proposition. For instance, such a tree can be the tree of Figure 13(a). It is easy to see that the value of the functional of this tree is equal to $p(1 + (2 - p)(n - 2))$. On the other hand, the optimal anticipatory solution $T^*$ is shown in Figure 13(b); the value of the functional associated with $T^*$ is equal to $p(n - 1) + p \sum_{i=1}^{n-2}(1 - p)^i = p(n - 1) + ((1 - p) - (1 - p)^{n-1})$. ∎

One can see that when $p$ is fixed (i.e., independent on $n$), the approximation ratio achieved is strictly better than 2. On the other hand, when $p \sim 1/n$, then since $\lim_{n \to +\infty}(1 - p)^n/p = n/e$, the ratio claimed in Proposition 9 tends to 1.225, for large values of $n$. If $p \sim 1/n^k$, $k > 1$, then for large values of $n$, this ratio tends to 1. Finally, if $p \sim 1/n^k$, $k < 1$, then (always for large values of $n$) the ratio is asymptotically equal to 2.

## 5   PROBABILISTIC MIN SPANNING TREE under modification strategy ROOT

We will now analyze PROBABILISTIC MIN SPANNING TREE under another modification strategy. In this strategy, denoted by ROOT, present vertices whose fathers are absent are connected to the omnipresent root (see Figure 14 for an example). Clearly, its complexity is $O(n)$. ROOT is very

natural in terms of network design where, for instance, in case of failure of intermediate servers, surviving servers are connected to the central one in order to ensure total global connectivity of the remaining network.
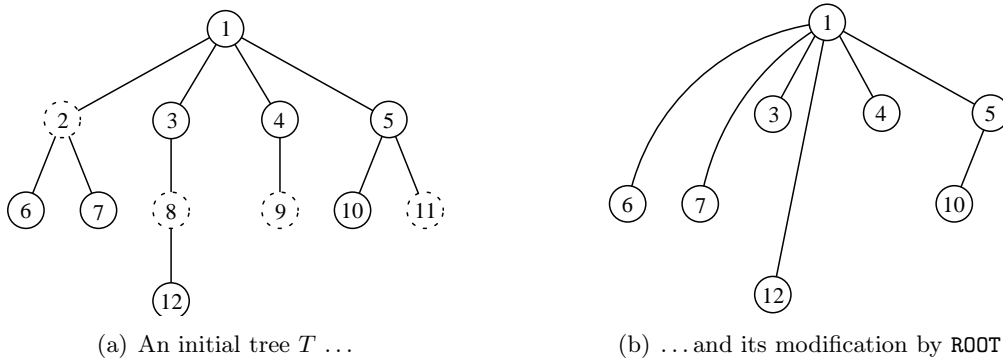


(a) An initial tree $T$ ...

(b) ...and its modification by `ROOT`

Figure 14: An anticipatory spanning tree $T$ and its adjustment.

## 5.1 The complexity of PROBABILISTIC MIN SPANNING TREE

With respect to `ROOT`, the contribution $C_i$ of a vertex $v_i$ can be written as follows:

$$C_i = C(v_i, T) = p_i \left( p_{f(i)} d_{i,f(i)} + \left( 1 - p_{f(i)} \right) d_{i,1} \right)$$

where $v_{f(i)}$ is the father of $v_i$ in $T$. As previously, summing the $C_i$'s, for each vertex of the graph, leads to the functional associated with the strategy `ROOT`:

$$E(G, T) = \sum_{v_i \in V} \left( p_i \left( p_{f(i)} d_{i,f(i)} + \left( 1 - p_{f(i)} \right) d_{i,1} \right) \right) \tag{44}$$

The value of $E(G, T)$ can be computed in polynomial time. We will now show that PROBABILISTIC MIN SPANNING TREE under `ROOT` belongs to **P**, by specifying a polynomial algorithm that solves this problem optimally.

**Proposition 10.** PROBABILISTIC MIN SPANNING TREE *under `ROOT` can be optimally solved in polynomial time.*

**Proof.** Transform the initial graph $G(V, E)$ into a directed one $G_d(V, A)$ having the same vertex-set as follows: for each edge $(v_i, v_j)$ of $E$ $(i, j \neq 1)$, add two arcs $(v_i, v_j)$ and $(v_j, v_i)$ in $G_d$ with weights:

$$\begin{aligned} d'_{i,j} &= p_i \left( p_j d_{i,j} + \left( 1 - p_j \right) d_{i,1} \right) \\ d'_{j,i} &= p_j \left( p_i d_{j,i} + \left( 1 - p_i \right) d_{j,1} \right) \end{aligned}$$

and for each edge $(v_1, v_i)$ of $E$, add one arc $(v_1, v_i)$ in $G_d$ with the same weight. Note that the number of arcs in $G_d$ is at most $2|E|$.

Run in $G_d$ Edmonds' polynomial algorithm for the ARBORESCENCE problem ([15, 16, 21]; this problem consists of finding a minimum-weight spanning tree in a directed graph), taking $v_1$ as root. It returns a rooted tree of minimum total edge-weight spanning all the vertices of $G_d$. This algorithm consists of determining the minimum weight arc entering each vertex (except for the root), and adding this edge to the solution. At this point, the solution has $n-1$ arcs, but may

contain loops. If the solution does not contain loops, then it is an optimal arborescence, otherwise loops are successively broken in an optimal way. Retain this tree as anticipatory solution for $G$ (simply delete orientations of arcs and unify the so obtained duplicate edges). Given the edge-weight function defined in $G_d$, this tree also minimizes the functional $E(G, T, \texttt{ROOT})$, i.e., it is optimal for PROBABILISTIC MIN SPANNING TREE. ∎

## 5.2 Stability of the solutions

### 5.2.1 Same probabilities for all vertices

We first focus on the case where all vertices have the same presence-probability, $p$ (except for $v_1$ which still has probability 1). We show that in such graphs, the deterministic and probabilistic solutions coincide, which is not true anymore when probabilities are distinct.

**Proposition 11.** *Let $G$ be an instance of* PROBABILISTIC MIN SPANNING TREE *problem under strategy* $\texttt{ROOT}$, *where all vertices, except for the root, have the same probability $p$. On such an instance, the optimal tree $\hat{T}$ and optimal a priori tree $T^*$ coincide.*

**Proof.** In an instance where all vertices are equiprobable, the general functional formula given in (44) can be reformulated as follows:

$$E(G, T) = \sum_{v_i \in V} \left( p \left( p d_{i,f(i)} + (1 - p) d_{i,1} \right) \right) \tag{45}$$

In this case, notice that the second member of the sum does not depend on the structure of the tree. Thus, one can rewrite the functional as:

$$E(G, T) = p \left( \sum_{(v_i, v_j) \in T} p d_{i,j} + \sum_{v_j \in V} (1 - p) d_{1,j} \right) \tag{46}$$

The first term in the parentheses of (46) is $p$ times the value of the spanning tree $T$, while the second term is the weight of the star rooted at $v_1$ (that is constant for a given graph), i.e.:

$$E(G, T) = p \left( p d(T) + (1 - p) d(S) \right) \tag{47}$$

So, in order to minimize $E(G, T)$ in (47), it suffices to minimize $d(T)$. ∎

### 5.2.2 Star as deterministic solution

Let us now consider the particular case where the anticipatory solution is a star rooted at $v_1$. Then, the following result holds.

**Proposition 12.** *Let $T$ be a star rooted at $v_1$. If $T$ is an optimal solution for* MIN SPANNING TREE *then it is also an optimal anticipatory solution for* PROBABILISTIC MIN SPANNING TREE *under* $\texttt{ROOT}$.

**Proof.** Consider $G_d$ as it has been built in the proof of Proposition 10. If the optimal solution for MIN SPANNING TREE in $G$ is a star, then:

$$\forall i, j \neq 1, d_{1,j} \leqslant d_{i,j} \tag{48}$$

According to the definition of the edge-weight function $d'$ in $G_d$, it can be easily seen that the property expressed by (48) extends also to $G_d$: $\forall i, j \neq 1, d'_{1,j} \leqslant d'_{i,j}$.

Since edges of $G$ incident to $v_1$ are not duplicated when they become arcs in $G_d$, in the case of stars rooted at $v_1$ the minimum-weight arc for each vertex $v_i$ will be the arc $(v_1, v_i)$ and no loop will appear. Thus, in this case, the probabilistic optimum is the same star. ∎
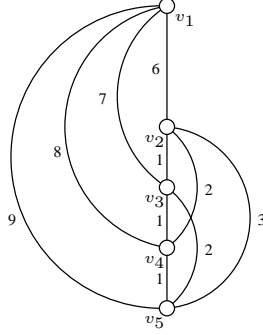
Figure 15: A graph $G$ with $n = 5$.

### 5.3 ROOT vs. CLOSEST_ANCESTOR

It has been shown that ROOT leads to a probabilistic model much easier to analyze and optimize than that induced by CLOSEST_ANCESTOR. But the optimum returned by the former might not be as good in terms of expected value as that returned by CLOSEST_ANCESTOR. Naturally, the choice between ROOT and CLOSEST_ANCESTOR will depend on the real world application so-modeled. But still, even if PROBABILISTIC MIN SPANNING TREE under ROOT is polynomial, comparing these two strategies in terms of quality of their respective returned solutions seems to be interesting and adds some further insight to the apprehension of the multiple difficulties occurring when handling probabilistic problems.

**Proposition 13.** *Let $E(G, T_{\mathtt{R}}^*)$ and $E(G, T_{\mathtt{CA}}^*)$ be the expected values of optimal anticipatory optima associated with* ROOT *and* CLOSEST_ANCESTOR, *respectively. Even in metric graphs, the approximation ratio between these two values is unbounded.*

**Proof.** Let us consider a graph $G$ of order $n$, some ordering on its vertices and assume that the optimal solution for MIN SPANNING TREE is a path. In this graph, vertices are equiprobable and weights are as follows:

$$
\begin{aligned}
d_{1,i} &= n + i - 1 & \forall i \neq 1 \\
d_{i,j} &= j - i & \forall i, j \;\; 1 < i < j \leqslant n
\end{aligned}
\tag{49}
$$

It can be easily seen that such graph is metric and that the path $P = (v_1, v_2, \ldots, v_n)$ is a minimum spanning tree (see Figure 15 for an example with $n = 5$).

From Proposition 11, $P$ is also the optimal anticipatory solution for PROBABILISTIC MIN SPANNING TREE under ROOT, since vertices are equiprobable. The weight of $P$ is $2n - 1$. So, revisiting (47), its expected value is:

$$
\begin{aligned}
E\left(G, T_{\mathtt{R}}^*\right) &= p\left( p(2n-1) + (1-p) \sum_{(v_j) \in V} (n + j - 1) \right) \\
&= p\left( p(2n-1) + (1-p) \left( n - 1 + \frac{n(n-1)}{2} \right) \right)
\end{aligned}
\tag{50}
$$

Consider now $P$ as the anticipatory solution for CLOSEST_ANCESTOR. Recall that this strategy reconnects each vertex to its closest present ancestor, so that, in $G$, any adjusted tree will be a path, with weight at most $d(P) = 2n - 1$. Figure 16 illustrates this assumption and depicts all the paths generated by CLOSEST_ANCESTOR, each of which has weight at most $2n - 1 = 7$.
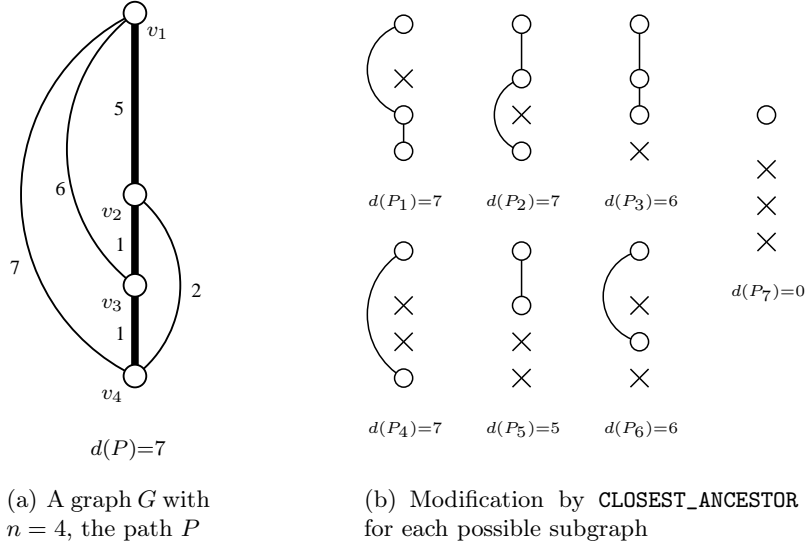
(a) A graph $G$ with $n = 4$, the path $P$

(b) Modification by `CLOSEST_ANCESTOR` for each possible subgraph

Figure 16: An anticipatory solution $P$ and all its possible adjustments.

Thus, a loose (but sufficient) lower bound for $E(G, T_{\mathtt{CA}}^*)$ is:

$$E\left(G, T_{\mathtt{CA}}^*\right) \leqslant 2n - 1 \tag{51}$$

Combining (50) and (51), we finally get for the ratio between the expected values associated with these two strategies:

$$\frac{E\left(G, T_{\mathtt{R}}^*\right)}{E\left(G, T_{\mathtt{CA}}^*\right)} \geqslant p^2 + \frac{p(1-p)(n-1)\left(1 + \frac{n}{2}\right)}{2n - 1} \tag{52}$$

that is $O(n)$ for large values of $n$ when $p$ is constant. ∎

In other words, there exist graphs where the expected value of the optimum with respect to `ROOT` is much larger than the expectation for `CLOSEST_ANCESTOR` when applied to this solution.

## 6   Conclusion

In this paper we have handled the PROBABILISTIC MIN SPANNING TREE problem in the framework of probabilistic combinatorial optimization. We have proposed two fast modification strategies (`CLOSEST_ANCESTOR` and `ROOT`) for reconstructing a second-stage tree and we have studied particular cases of anticipatory solutions. We have shown that the problem of optimizing the expectation of the second-stage cost by selecting an appropriate first-stage (anticipatory) solution is **NP**-hard under strategy `CLOSEST_ANCESTOR` while it is polynomial under `ROOT`. Under `CLOSEST_ANCESTOR` strategy we have given several approximation results for PROBABILISTIC METRIC MIN SPANNING TREE and PROBABILISTIC MIN SPANNING TREE 1,2.

There are several open questions subject for further research. To our opinion, the major among them are the complexities of PROBABILISTIC METRIC MIN SPANNING TREE and PROBABILISTIC MIN SPANNING TREE 1,2 (under `CLOSEST_ANCESTOR`; we conjecture that they are both **NP**-hard) and the improvement of their approximation ratios.

## References

[1] I. Averbakh, O. Berman, and D. Simchi-Levi. Probabilistic a priori routing-location problems. *Naval Res. Logistics*, 41:973–989, 1994.

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

[3] D. J. Bertsimas. On probabilistic traveling salesman facility location problems. *Transportation Sci.*, 3:184–191, 1989.

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

[5] D. J. Bertsimas, P. Jaillet, and A. Odoni. A priori optimization. *Oper. Res.*, 38(6):1019–1033, 1990.

[6] L. Bianchi, J. Knowles, and N. Bowler. Local search for the probabilistic traveling salesman problem: correlation to the 2-p-opt and 1-shift algorithms. *European J. Oper. Res.*, 161(1):206–219, 2005.

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

[8] N. Bourgeois, F. Della Croce, B. Escoffier, C. Murat, and V. Th. Paschos. Probabilistic coloring of bipartite and split graphs. *J. Comb. Optimization*, 17(3):274–311, 2009.

[9] F. Butelle. *Contribution à l'algorithmique distribuée de contrôle : arbres couvrants avec et sans contraintes.* PhD thesis, Université Paris VIII, 1994.

[10] B. Chazelle. A minimum spanning tree algorithm with inverse-ackerman type complexity. *J. Assoc. Comput. Mach.*, 47(6):1028–1047, 2000.

[11] D. Cheriton and R. E. Tarjan. Finding minimum spanning trees. *SIAM J. Comput.*, 5:724–742, 1976.

[12] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Grad. School of Industrial Administration, CMU, 1976.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* McGraw-Hill, 2nd edition, 2001.

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

[15] J. Edmonds. Optimum branchings. *J. Res. Natl. Bureau Standards*, 71B:233Ũ–240, 1967.

[16] L. Georgiadis. Arborescence optimization problems solvable by edmonds' algorithm. *Theoret. Comput. Sci.*, 301:427–437, 2003.

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

[18] P. Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Oper. Res.*, 36(6):929–936, 1988.

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

[20] P. Jaillet and A. Odoni. The probabilistic vehicle routing problem. In B. L. Golden and A. A. Assad, editors, *Vehicle routing: methods and studies.* North Holland, Amsterdam, 1988.

[21] B. Korte. *Combinatorial optimization: theory and algorithms*, volume 21 of *Algorithms and combinatorics*. Springer, 3rd edition, 2006.

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

[23] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 7(1):48–50, 1956.

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

[25] 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`.

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

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

[28] V. Th. Paschos, O. A. Telelis, and V. Zissimopoulos. Steiner forests on stochastic metric graphs. In A. Dress, Y. Xu, and B. Zhu, editors, *Proc. Conference on Combinatorial Optimization and Applications, COCOA'07*, volume 4616 of *Lecture Notes in Computer Science*, pages 112–123. Springer-Verlag, 2007.

[29] V. Th. Paschos, O. A. Telelis, and V. Zissimopoulos. Probabilistic models for the STEINER TREE problem. *Networks*, 56(1):39–49, 2010.

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

[31] R. E. Tarjan. Efficience of a good but not linear set-union algorithm. *J. Assoc. Comput. Mach.*, 22:215–225, 1975.

[32] A. Yao. An $O(|E|\log\log|V|)$ algorithm for finding minimum spanning trees. *Inform. Process. Lett.*, 4:21–23, 1975.