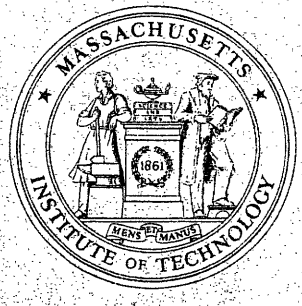
OPERATIONS RESEARCH CENTER

working paper



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

The Probabilistic Minimum Spanning Tree, Part I: Complexity and Combinatorial Properties

by

Dimitris Bertsimas

OR 183-88 August 1988

The probabilistic minimum spanning tree, Part I: complexity and combinatorial properties

by

Dimitris Bertsimas

Sloan School of Management,

Massachusetts Institute of Technology, Rm E53-359

Cambridge, Mass. 02139, USA

Abstract

In this paper we consider a natural probabilistic variation of the classical minimum spanning tree (MST) problem, which we call the probabilistic minimum spanning tree problem (PMST). In particular, we consider the case where not all the points are deterministically present, but are present with certain probability. We discuss the applications of the PMST and find a closed form expression for the expected length of a given spanning tree. Based on these expressions we prove that the problem is $NP - complete$. We further examine some combinatorial properties of the problem, establish the relation of the PMST problem with the MST problem and the network design problem and examine some cases where the problem is solvable in polynomial time.

Key words: Probabilistic combinatorial optimization problems, minimum spanning tree, network design, $NP - completeness$.

1 Introduction

The classical minimum spanning tree (MST) problem plays an important role in combinatorial optimization. It possesses the matroidal property which allows the greedy algorithm to solve the problem optimally, and thus it is the prototype for problems solvable in polynomial time. For a summary of its properties and algorithms for its solution see Papadimitriou and Steiglitz [7]. From a practical point of view, it has important applications in transportation, communications, distribution systems, etc.

In this paper we consider a natural probabilistic variation of this classical problem. In particular, we consider the case where not all the points are deterministically present, but are present with certain probability. Formally, given a weighted graph $G = (V, E)$ and a probability of presence $p(S)$ for each subset S of V, we want to construct an **a priori** spanning tree of minimum expected length in the following sense: on any given instance of the problem delete the vertices and their adjacent edges among the set of absent vertices provided that the tree remains connected. The problem of finding an a priori spanning tree of minimum expected length is the probabilistic minimum spanning tree (PMST) problem. In order to clarify the definition of the PMST problem, consider the example in Figure 1. If the a priori tree is T and nodes $2, 7, 9$ are the only ones not present, the tree becomes T_1. One can easily observe that if every node is present with probability $p_i = 1$ for all $i \in V$ then the problem reduces to the classical MST problem.

This paper is part of a more general investigation of the properties of combinatorial optimization problems when instances are modified probabilis-

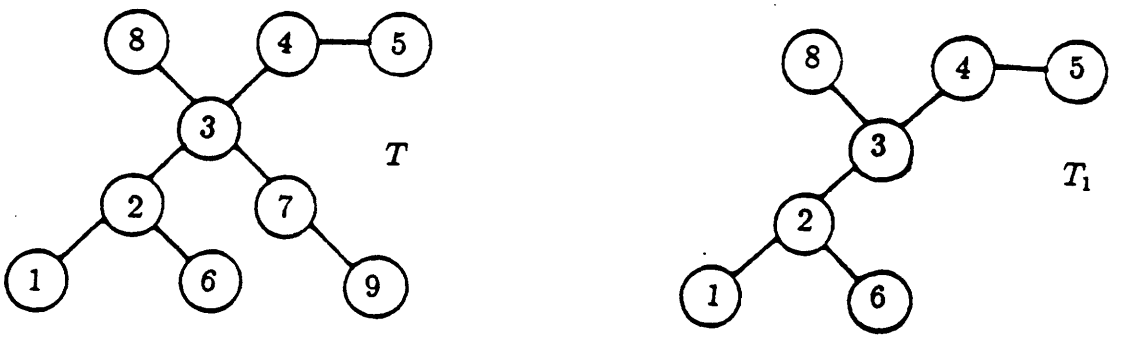


Figure 1: The PMST methodology

tically. Interest in this class of problems started with the Ph.D thesis of Jaillet [4] on the probabilistic traveling salesman problem (PTSP), where he posed the problem, examined some of its combinatorial properties and proved asymptotic theorems in the plane. In Bertsimas [1] further properties of the PTSP are derived and the results of Jaillet [4] are sharpened. Bertsimas [1] includes also results on the probabilistic vehicle routing problem, defined in Jaillet and Odoni [5] and probabilistic facility location problems. To our knowledge, the PMST problem has never been defined before in the literature despite its intrinsic interest as well as its applicability. In Bertsimas [2], which is a sequel of the present paper, we perform probabilistic analysis of the PMST and prove that surprisingly the PMST problem is asymptotically equivalent to the strategy of re-optimization, in which we re-optimize every instance of the problem.

In the next section we discuss some applications of the PMST problem, while in section 3 we address the question of finding an explicit expression for the expected length of an a priori tree T. In section 4 we investigate

the complexity of the problem and we prove that even a restricted version of the problem with all weights equal is $NP - complete$, which in view of the simplicity of the MST problem is a quite surprising result. In section 5 we examine some combinatorial properties of the problem (bounds, relation of PMST and MST, relation with re-optimization strategies and the network design problem). In section 6 we exploit these combinatorial properties of the problem and find some special cases which are solvable in polynomial time. The final section includes some concluding remarks.

2 Discussion and Applications of the PMST Problem

It is natural to ask why the PMST problem is interesting. We first observe that the problem defines an efficient strategy to update minimum spanning tree solutions when problem's instances are modified probabilistically because of the absence of certain nodes from the graph. We denote this strategy with Σ_{T_p}, where T_p is the optimal a priori tree. Then in the instance S, i.e. when only nodes in the set S are present, the strategy produces a tree $T_p(S)$ with length $L_{T_p}(S)$, which is the length of the tree that connects nodes from the set S of present nodes using parts of T_p. In the context of this discussion the letter Σ denotes the strategy used.

Two possible other strategies are:

1. A re-optimization strategy Σ_{MST}, in which we find the minimum spanning tree (MST) of the set of present nodes in every instance. We

denote with $L_{MST}(S)$ the length of the MST of the nodes in the set S.

2. A re-optimization strategy $\Sigma_{STEINER}$, in which we find the minimum Steiner tree of the set of present nodes in every instance. We denote with $L_{STEINER}(S)$ the length of the Steiner tree of the nodes in the set S, using possibly nodes from the set $V - S$.

Remark: The above definition of the re-optimization strategy $\Sigma_{STEINER}$ applies only for the case of a fixed network, as opposed to the case where the points are located in the Euclidean plane. In this case $L_{STEINER}(S)$ is the length of the Steiner tree in the plane of the points from the set S.

Why don't we use these re-optimization strategies Σ_{MST}, $\Sigma_{STEINER}$, rather than the strategy Σ_{T_p} we are proposing?

Concerning the $\Sigma_{STEINER}$ strategy, it is clear that $L_{STEINER}(S) \leq L_{T_p}(S)$, because the tree connecting the set S using only parts of the tree T_p is also a solution to the Steiner problem. The disadvantage of the Steiner strategy is that we have to solve an $NP - hard$ problem in every instance, something which is feasible only for small problem instances.

With the strategy Σ_{MST} it is clear that we can compute $L_{MST}(S)$ in $O(|S|^2)$, using the greedy algorithm, but it is not clear that $L_{MST}(S) \leq L_{T_p}(S)$. In fact in section 5 we construct examples where the probabilistic strategy Σ_{T_p} we are proposing is better than the Σ_{MST}. Furthermore, in [2] we prove that asymptotically under reasonable probabilistic assumptions the probabilistic strategy Σ_{T_p} is at least as good as the Σ_{MST}.

What is more important is the fact that in many applications we need a real-time strategy to modify the solution when the instances are modified. Clearly, the PMST strategy satisfies this criterion, since the tree $T(S)$ can be found in $O(n)$ time as follows:

1. Start with the a priori tree T.

2. Until there are no unmarked leaves in T:
   find an unmarked leaf in T;
   if $i \in S$ mark it; else delete i from T.

3. The resulting tree is the tree $T(S)$.

Since we are only looking at every node at most once, this is an $O(n)$ algorithm. Note that the two re-optimization strategies are superlinear. In addition, we may not have the computer resources to re-optimize. An even more important motivation in favor of the Σ_{T_p} strategy is that this strategy does not change the underlying network structure, while both the re-optimization strategies can result in a completely different network structure for different problem instances, by adding new edges and deleting old ones. In a communication network for example, it may be very expensive or even impractical to create new communication links for each problem instance.

After this discussion of the various strategies available when problem instances are modified, let us consider some potential areas of application of the PMST problem. Consider for example a VLSI environment. Suppose on a circuit, there are n processors subject to failure and processor i becomes

inactive with probability p_i. Then we would like to connect the active processors using a spanning tree structure, which minimizes the manufacturing cost. Communication of two active processors through some inactive processors means that the inactive processors allow communication. Since in this example changing the underlying network structure is impractical, the PMST strategy is a good solution to the problem.

In a communication network, nodes may represent communication centers. arcs represent communication links and link costs are the communication costs among centers. The probability of failure p_i is the probability of blocked communication in center i. If the centers are blocked, they can only be used to establish communication between unblocked centers. Then the problem of finding an a priori network structure of minimum expected cost is the PMST problem.

A more unusual application of the PMST problem is in the area of organizational structures. For instance, a rather intriguing paradigm for the PMST problem in the area of organizational structure design might be the following: Suppose the n points that we wish to interconnect represent our agents or spies in a foreign country. They will undertake in the future a series of missions, each mission involving a different subset of agents. A mission, in our context, is an instance of the problem. We are looking for an a priori organizational structure in which, for obvious reasons, each agent will know only the people immediately above or below him/her in the structure; this implies a spanning-tree-like structure. The probability p_i associated with point i is the a priori probability that agent i will have to participate in any random mission undertaken by the network. For any given mission, only

that part of the organization which is necessary to interconnect all the agents participating in that particular mission is activated. (For example, if the tree T of Figure 1 represents the network and agents 1, 3, 5, 6 and 8 must be involved in a particular mission, the tree T_1 of Figure 1 represents the network and subset of agents that will be activated.) The distance between points i and j is interpreted as the cost or risk of exposure incurred when agents i and j must communicate or work with each other. Given p_i for $i = 1, 2, \ldots, n$ and the distance matrix for all possible pairs (i, j), the PMST gives the organizational structure which, in the expected value sense, minimizes the risk of exposure of the network on a random mission.

In a routing context, a company may want to connect all demand locations, using a tree-like structure. The cost between demand locations can represent transportation cost and the probability p_i represents the probability of having a demand at location i. The company would like to find an a priori spanning tree for the demand locations, such as to minimize the expected transportation cost.

Other areas of potential application can be in the areas of transportation and of strategic planning.

One might object that all the examples we have discussed represent some idealization of reality. Nevertheless, the PMST is a generic problem, which in many applications where a particular type of randomness is present can be a more appropriate model than the classical MST. It also addresses the question of finding a spanning tree which is optimal on the average, rather than a solution which is optimal on a particular instance. The essential characteristic therefore of the PMST problem is that it is a more global

problem than the MST problem; as well the optimal solution to the PMST is a robust solution.

Unfortunately, as we prove in section 4, one pays for these nice properties (robustness, globality) by changing the complexity of the problem radically. While the MST problem is easily solvable, the PMST problem is $NP-hard$.

3 The Expected Length of a Given Spanning Tree

As we noted in the previous section the PMST problem defines an efficient strategy for updating spanning tree solutions when problem instances are modified probabilistically in response to the absence of certain nodes from the graph. Given an a priori tree T we define $L_T(S)$ to be the length of the tree which connects nodes from the set S of present nodes using only parts of T. For example in Figure 1, $S = \{1, 3, 5, 6, 8\}$ and $L_T(S)$ is the length of the tree T_1.

Then if the set S of points present has probability $p(S)$, the problem can be defined formally as follows:

Problem definition:

Given a graph $G = (V, E)$, not necessarily complete, a cost function $c : E \rightarrow R$, a probability function $p : 2^V \rightarrow [0, 1]$ we want to find a tree T that minimizes the expected length $E[L_T]$:

$$E[L_T] = \sum_{S \subseteq V} p(S) L_T(S), \tag{1}$$

where the summation is taken over all subsets of V, the instances of the problem.

Note that at this level of generality we can model dependencies among the probabilities of presence of sets of nodes. An additional observation is that with this formulation one would need $O(n2^n), (|V| = n)$ effort to compute the expected length of a given tree T. We would like to be able to compute $E[L_T]$ efficiently. The question we address in this section is for which probability functions $p(S)$ we can compute efficiently $E[L_T]$ for a given tree T.

If we define

$h(S) \triangleq Pr\{$ none of the nodes in S is present $\} = \sum_{R \subseteq V-S} p(R)$, then

Theorem 1

Given an a priori tree T its expected length is given by the expression

$$E[L_T] = \sum_{e \in T} c(e)\{1 - h(K_e) - h(V - K_e) + h(V)\}, \qquad (2)$$

where $K_e, V - K_e$ are the subsets of nodes contained in the two subtrees obtained from T by removing the edge e (see Figure 2).

Proof:

Given a tree T let us consider how much each edge $e \in T$ contributes to $E[L_T]$. By the definition of the problem only the edges in T contribute in this expectation. If we define the events:

$A(K_e) \triangleq$ at least one node in K_e is present,

then the contribution of every edge e is

$$c(e)Pr\{A(K_e) \cap A(V - K_e)\},$$

because the edge e is used if and only if there exists at least one node present

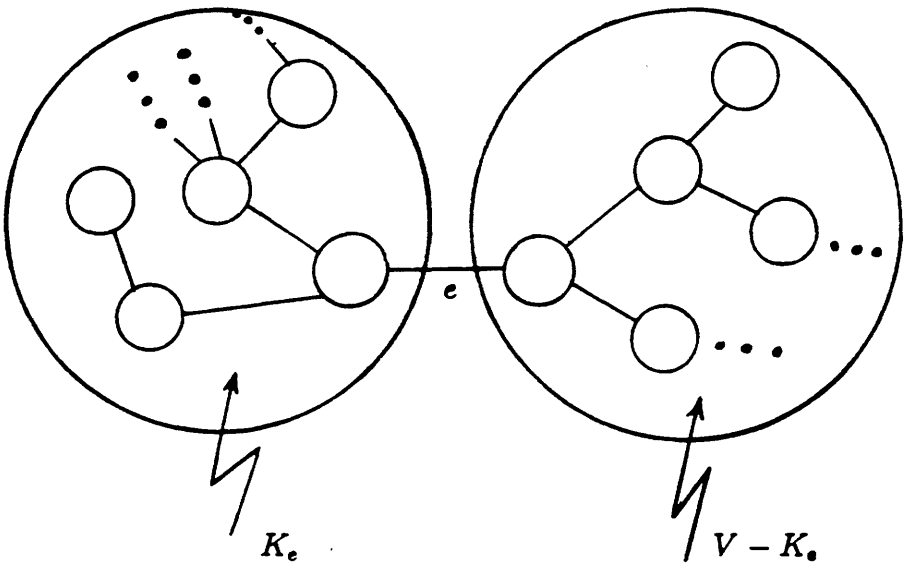


Figure 2: The sets $K_e, V - K_e$

in K_e and at least one node present in $V - K_e$. As a result,

$$E[L_T] = \sum_{e \in T} c(e) Pr\{A(K_e) \cap A(V - K_e)\}.$$

But

$$Pr\{A(K_e) \cap A(V - K_e)\} = Pr\{[A^c(K_e) \cup A^c(V - K_e)]^c\} = 1 - Pr\{A^c(K_e) \cup A^c(V - K_e)\}$$

$$= 1 - Pr\{A^c(K_e)\} - Pr\{A^c(V - K_e)\} + Pr\{A^c(K_e) \cap A^c(V - K_e)\}.$$

But since $Pr\{A^c(K_e)\} = Pr\{\text{none of the nodes in } K_e \text{ is present}\} = h(K_e)$, we easily obtain (2). •

Thus if instead of the probability function $p(S)$ we are given the function $h(S)$ we can compute $E[L_T]$ for any given tree T in $O(n)$, assuming we can compute $h(S)$ in $O(1)$, since we can find all the sets K_e for all e in $O(n)$ by

starting the computation at the leaves. An interesting case, and important in practice, is when the nodes are present independently. Then we can find an explicit expression for $E[L_T]$.

Theorem 2

If node i is present with probability p_i, then the expectation $E[L_T]$ of a given tree T is given by the expression

$$E[L_T] = \sum_{e \in T} c(e)\{1 - \prod_{i \in K_e}(1-p_i)\}\{1 - \prod_{i \in V - K_e}(1-p_i)\}. \qquad (3)$$

Proof:

In this case, because nodes are present independently

$$h(S) = \prod_{i \in S}(1-p_i).$$

Substituting the above expression in (2), we easily obtain (3). •

From (3) we can compute $E[L_T]$ in $O(n^2)$, since we can compute $h(S)$ in $O(|S|)$. By organizing the computation carefully, we can compute $E[L_T]$ in $O(n)$ as follows:

1. Let $a = \prod_{i \in V}(1-p_i)$; let $a_i = 1$; Let $MARKED$=set of leaves.

2. Until node set is empty:
   if i is a leaf, let $a_i = (1-p_i)\prod_{j \in MARKED,\ (i,j) \in T} a_j$;
   add i to the set $MARKED$; delete i from T.

3. $E[L_T] = \sum_{e=(i,j) \in T} c(e)(1-a_i)(1-a/a_i)$.

An important special case is when $p_i = p$ for all i. Then $E[L_T]$ becomes

$$E[L_T] = \sum_{e \in T} c(e)\{1 - (1-p)^{|K_e|}\}\{1 - (1-p)^{n-|K_e|}\}. \tag{4}$$

If we define

$$\phi(k) \triangleq \{1 - (1-p)^k\}\{1 - (1-p)^{n-k}\}, \tag{5}$$

then

$$E[L_T] = \sum_{e \in T} c(e)\phi(|K_e|). \tag{6}$$

Based on these closed form expressions we will prove in the next section that the decision version of the PMST problem, even with $p_i = p$ for all i and $c(e) = 1$ is $NP - complete$. An additional importance of the expressions (6) is that they will assist us in deriving some key combinatorial properties of the optimal solution to the PMST problem.

4 The Complexity of the PMST Problem

In this section we prove that the simplest possible case of the PMST problem with equal weights $c(e) = 1$ and $p_i = p$ is $NP - complete$. We first define formally the decision version of this restricted problem.

The Restricted PMST Problem (RPMST)

Instance: Given a graph $G = (V, E)$, costs $c(e) = 1$ for all $e \in E$, a rational number $p, 0 < p < 1$ and a bound B.

Question: Is there a spanning tree T for G with

$$E[L_T] = \sum_{e \in T} c(e)\{1 - (1-p)^{|K_e|}\}\{1 - (1-p)^{n-|K_e|}\} \leq B?$$

In order to prove that the RPMST problem is $NP-complete$ we will need some properties of the function $\phi(k) = (1-x^k)(1-x^{n-k})$, $x = 1-p$ defined in (5).

Proposition 3

The function $\phi(k)$ has the following properties:

1. If $k < m < \frac{n}{2}$, then $\phi(k) < \phi(m)$.
2. $\phi(k+m) < \phi(k) + \phi(m)$.
3. $3\phi(3) - 2\phi(4) > 0$.

Proof:

These properties follow easily from elementary algebraic manipulations as follows:

1. $\phi(k) - \phi(m) = (x^m - x^k)(1 - x^{n-m-k}) < 0$ if $k < m$ and $m + k < \frac{n}{2} + \frac{n}{2} = n$.
2. $\phi(k) + \phi(m) - \phi(k+m) = (1-x^k)(1-x^m)(1+x^{n-k-m}) > 0$.
3. $3\phi(3) - 2\phi(4) = 3(1-x^3)(1-x^{n-3}) - 2(1-x^4)(1-x^{n-4}) \geq (1-x^{n-4})(1+2x^4-3x^3) = (1-x^{n-4})(1-x)[1+x(1-x^2)+x^2(1-x)] > 0$.

•

We now have all the required tools to prove that the RPMST problem is $NP-complete$.

Theorem 4

The RPMST problem is $NP-complete$.

Proof:

Clearly, RPMST belongs to the class NP, since given a tree T we can compute $E[L_T]$ in polynomial time ($O(n)$) and compare it with the given bound B. In order to prove the completeness of the problem we will reduce the $NP-complete$ problem EXACT COVER BY 3-SETS (Garey and Johnson [3]) to it.

EXACT COVER BY 3-SETS (E-3C)

Instance: A family $S = \{\sigma_1, \ldots, \sigma_s\}$ of 3-element subsets of a set $C = \{c_1, \ldots, c_{3c}\}$.

Question: Is there a subfamily $S_1 \subset S$ of pairwise disjoint sets such that $\cup_{\sigma \in S_1} \sigma = C$?

Given an instance of the E-3C problem, we define the following instance of the RPMST problem:

$G = (V, E)$,

$V = R \cup S \cup C$,

$R = \{a_0, \ldots, a_r\}$,

$r = s + 3c$,

$E = \{(a_i, a_0), i = 1, \ldots, r\} \cup \{(a_0, \sigma), \sigma \in S\} \cup \{(\sigma, c), c \in \sigma\}$,

p arbitrary rational with $0 < p < 1$,

$B = (r + 3c)\phi(1) + c\phi(4)$,

$\phi(k) = (1 - x^k)(1 - x^{n-k})$, $x = 1 - p, n = r + 1 + s + 3c$.

As an example if $S = \{\{c_1, c_2, c_3\}, \{c_2, c_3, c_5\}, \{c_2, c_4, c_5\}, \{c_4, c_5, c_6\}\}$, $c = 2, s = 4$, the corresponding graph is presented in Figure 3.

Let T be a feasible ($E[L_T] \leq B$) spanning tree of G. Clearly $(a_i, a_0) \in T$. We now show that if $E[L_T] \leq B$, then $(a_0, \sigma) \in T$ for all $\sigma \in S$. Suppose

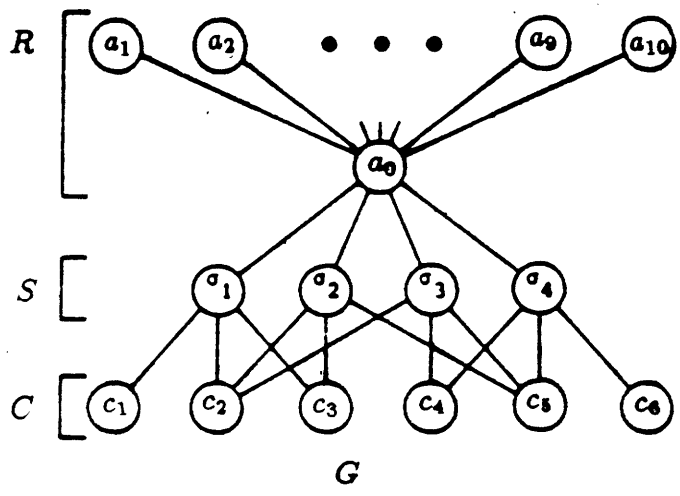


Figure 3: Equivalent instances of E-3C and RPMST

first there exists only one $(a_0, \sigma) \notin T$ for some $\sigma \in S$. We will show that $E[L_T] > B$. Since $(a_0, \sigma) \notin T$, there exists $i \in S$ and $j \in C$ such that $(a_0, i), (i, j), (j, \sigma) \in T$ (see Figure 4a). We define

$g_l \triangleq$ the number of nodes in $C - \{j\}$ that are adjacent to l in T. In the example in Figure 4a, $g_i = 1, g_\sigma = 2$.

We also define

$s_l \triangleq$ the number of nodes in $S - \{i, \sigma\}$ in T that are adjacent to exactly l vertices from C in T $(l = 0, 1, 2, 3)$.

From these definitions we get

$$s_1 + 2s_2 + 3s_3 = 3c - g_i - g_\sigma - 1 \Rightarrow s_3 = \frac{1}{3}(3c - 2s_2 - s_1 - g_i - g_\sigma - 1). \quad (7)$$

We now write an expression for $E[L_T]$.

$$E[L_T] = r\phi(1) + (3c - g_i - g_\sigma - 1)\phi(1) + s_1\phi(2) + s_2\phi(3) + s_3\phi(4) + \phi(g_i + g_\sigma + 3) +$$

$$+(g_i + g_\sigma)\phi(1) + \phi(2 + g_\sigma) + \phi(1 + g_\sigma),$$

where the first term ($r\phi(1)$) is from the contributions of the r edges (a_i, a_0), the second term is from the contributions of the edges connecting the nodes in C except the ones that are connected with i, σ, and the terms $\phi(g_i + g_\sigma + 3), \phi(2+g_\sigma), \phi(1+g_\sigma)$ are from the contributions of the edges $(a_0, i), (i, j), (j, \sigma)$ respectively. Then

$$E[L_T] > B = (r + 3c)\phi(1) + c\phi(4) \Leftrightarrow$$

$$s_1\phi(2)+s_2\phi(3)+s_3\phi(4)+\phi(g_i+g_\sigma+3)+\phi(2+g_\sigma)+\phi(1+g_\sigma)-\phi(1)-c\phi(4) > 0.$$

Substituting (7) we get

$$E[L_T] > B \Leftrightarrow \frac{1}{3}s_1[3\phi(2) - \phi(4)] + \frac{1}{3}s_2[3\phi(3) - 2\phi(4)]+$$

$$\frac{1}{3}[3\phi(g_i+g_\sigma+3)-(g_i+g_\sigma)\phi(4)+\phi(2+g_\sigma)]+\frac{1}{3}[2\phi(2+g_\sigma)-\phi(4)]+[\phi(1+g_\sigma)-\phi(1)] > 0.$$

Using proposition 3 we can easily check that all the terms in [ ] are strictly positive and thus $E[L_T] > B$.

Suppose now that there are $(a_0, \sigma_1), \ldots, (a_0, \sigma_k) \notin T$ (see Figure 4b). Since T is a tree, there exist $i \in S$ and $j \in C$ such that $(a_0, i), (i, j), (j, \sigma_k) \in T$. Then if we add the edge (a_0, σ_k) and delete the edge (j, σ_k) we get a new tree T_{k-1}, in which there are only $k-1$ nodes $\sigma_1, \ldots, \sigma_{k-1}$ not connected with a_0. If we denote the tree T with T_k in order to represent the fact that there are k nodes in T not connected to a_0, we claim that

$$E[L_{T_k}] > E[L_{T_{k-1}}].$$

Let u_i, u_j, u_{σ_k} be the number of nodes in the subtrees from nodes i, j, σ_k respectively (see also Figure 4b). The contribution of edges in T_k, T_{k-1} that

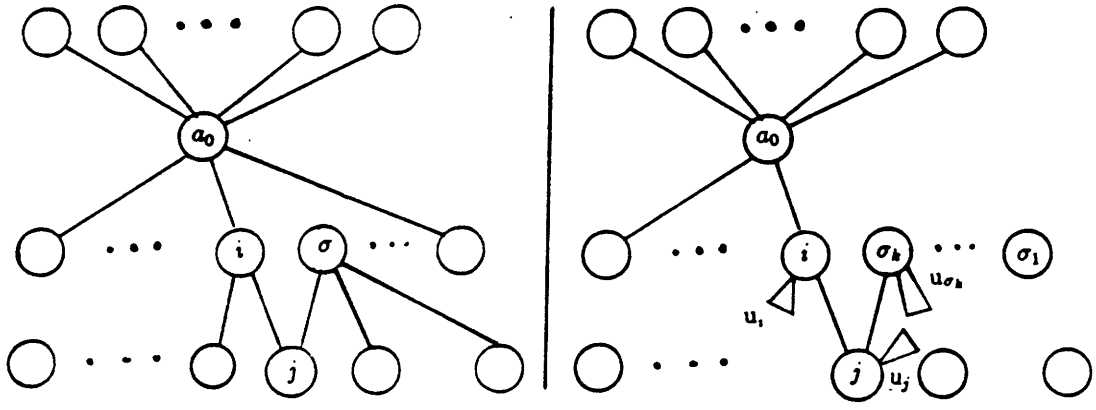


Figure 4: The cases $(a_0, \sigma) \notin T$ and $(a_0, \sigma_1), \ldots, (a_0, \sigma_k) \notin T$

are not involved in the cycle created by adding the edge (a_0, σ_k) is the same. Then

$$E[L_{T_k}] - E[L_{T_{k-1}}] = \phi(u_i+1+u_j+1+u_{\sigma_k}+1)+\phi(u_j+1+u_{\sigma_k}+1)+\phi(u_{\sigma_k}+1)-$$

$$\phi(u_i + 1 + u_j + 1) - \phi(u_j + 1) - \phi(u_{\sigma_k} + 1),$$

where $\phi(u_i + 1 + u_j + 1 + u_{\sigma_k} + 1)$, $\phi(u_j + 1 + u_{\sigma_k} + 1)$, $\phi(u_{\sigma_k} + 1)$ are the contributions in T_k of (a_0, i), (i, j) and (j, σ_k) respectively. Similarly in T_{k-1} the terms $\phi(u_i+1+u_j+1)$, $\phi(u_j+1)$ and $\phi(u_{\sigma_k}+1)$ are from (a_0, i), (i, j) and (a_0, σ_k) respectively. By proposition 3 we have that $\phi(u_i+1+u_j+1+u_{\sigma_k}+1) > \phi(u_i + 1 + u_j + 1)$ and $\phi(u_j + 1 + u_{\sigma_k} + 1) > \phi(u_j + 1)$. As a result, $E[L_{T_k}] > E[L_{T_{k-1}}]$. Note that we have used the fact $r = s + 3c$, since in order for proposition 3 to hold we need $u_i+1+u_j+1+u_{\sigma_k}+1 < s+3c < \frac{n}{2} = \frac{r+1+s+3c}{2}$.

As a result, the expected cost of T decreases by adding one missing arc (a_0, σ_k). Making this transformation inductively we find:

$$E[L_T] = E[L_{T_k}] > E[L_{T_{k-1}}] > \ldots > E[L_{T_1}].$$

But since the tree T_1 has only one missing arc (a_0, σ_1), we have already proved that in this case $E[L_{T_1}] > B$.

Therefore, it follows that for the tree T to be feasible all edges $(a_0, \sigma_i) \in T$. We will now show that

$$E[L_T] \leq B \Leftrightarrow \text{E-3C has a solution.}$$

By using the quantities s_l $(l = 0, 1, 2, 3)$ defined above we have

$$s_1 + 2s_2 + 3s_3 = 3c, \quad s_0 + s_1 + s_2 + s_3 = s.$$

The expected cost of T is then given by

$$E[L_T] = (r + 3c)\phi(1) + s_1\phi(2) + s_2\phi(3) + s_3\phi(4).$$

Thus

$$E[L_T] \leq B \Leftrightarrow s_1\phi(2) + s_2\phi(3) + (s_3 - c)\phi(4) \leq 0 \Leftrightarrow$$
$$\frac{1}{3}s_1[3\phi(2) - \phi(4)] + \frac{1}{3}s_2[3\phi(3) - 2\phi(4)] \leq 0. \qquad (8)$$

From proposition 3, $3\phi(2) - \phi(4) > 0$ and $3\phi(3) - 2\phi(4) > 0$. As a result, inequality (8) holds if and only if $s_1 = s_2 = 0$ and hence $s_3 = c$, which is equivalent to E-3C having a solution. Thus $E[L_T] \leq B \Leftrightarrow$ E-3C has a solution, and hence the RPMST problem is $NP - complete$. •

We can add some insight to why the problem is hard by noticing the following remarkable fact. As $p \rightarrow 1$ the PMST approaches the MST which is easily solvable. What is the limit as $p \rightarrow 0$? In this case

$$\phi(k) = (1-(1-p)^k)(1-(1-p)^{n-k}) \rightarrow p^2 k(n-k).$$

As a result,

$$E[L_T] \rightarrow p^2 \sum_{e \in T} c(e)|K_e|(n-|K_e|).$$

The expression $\sum_{e \in T} c(e)|K_e|(n-|K_e|)$ is the objective function of another famous problem, the NETWORK DESIGN PROBLEM on a tree, which is defined as follows:

NETWORK DESIGN PROBLEM

Instance: A graph $G = (V, E)$, a weight $c(e)$ for each $e \in E$ and a bound B.

Question: Is there a spanning tree T for G such that, if $W(\{u, v)\})$ denotes the sum of the weights of the edges on the path joining u and v in T, then

$$f(T) = \sum_{u,v \in V} W(\{u, v\}) \leq B?$$

It is easily seen by considering the contribution of every edge e that $f(T) = \sum_{e \in T} c(e)|K_e|(n-|K_e|)$. The network design problem on a tree was proved $NP-complete$ in Johnson, Lenstra and Rinnooy Kan [6]. Thus the PMST problem approaches as $p \rightarrow 0$ an $NP-complete$ problem which gives some intuition as to why the problem is hard. In fact, it is this observation that originally led us to suspect that the PMST problem is hard.

We have proved that the restricted version of the PMST with equal costs on a non-complete graph is $NP-complete$. We now prove that even if the

graph is complete, but the costs are either small or large, the problem is still hard.

The PMST problem on a complete graph

Instance: A complete graph K_n, a cost $c(e) \in \{1, M\}$, a bound B and a probability $p, 0 < p < 1$.

Question: Is there a spanning tree T with $E[L_T] \leq B$?

Theorem 5

The PMST problem on a complete graph is $NP - complete$.

Proof:

Clearly the problem is in NP because of the closed form expressions we have found. To prove that the problem is complete we use the same reduction as in the proof of theorem 4. In order to make the graph complete we add the remaining edges but with very high cost, i.e $c(e) = \frac{(r+3c)\phi(1)+c\phi(4)+1}{\phi(1)}$. Then if we include any edge of this type, its contribution would be $c(e)\phi(|K_e|) \geq c(e)\phi(1) = B + 1$, i.e. it can not be included in the tree. Therefore, the proof remains unchanged since edges with large costs never appear in a tree with $E[L_T] \leq B$. •

In section 6 by exploiting some combinatorial properties of the problem, we examine some special cases of the PMST in which the problem can be solved in polynomial time. For example we prove that in a complete graph with all costs equal the problem is solvable in $O(n)$. The previous theorems indicate that the problem is hard if either the graph is complete and the costs are 1 or M or the graph is non-complete but the costs are equal. If we combine these two requirements (complete graph, equal costs) the problem becomes easy.

5 Combinatorial and Functional Properties of the PMST

In this section we examine the case with equal probabilities $p_i = p$. In this case we are trying to find a spanning tree that minimizes the expression

$$f(p) \triangleq \min_T f_T(p) = \min_T \{ \sum_{e \in T} c(e)\phi(|K_e|) \}. \tag{9}$$

5.1 Functional Properties of the PMST

Expression (9) is clearly a function of the coverage probability p. For different values of p the corresponding optimal probabilistic trees which minimize (9) are different. We first address the question of specifying the properties of the function $f(p)$. From the results of section 4 we have seen that it would be difficult to find $f(p)$ for a particular value of p, but can we find some global properties of this function which will give some insight into the problem? We call these properties functional because they are related to the function $f(p)$. Some initial observations are stated in the following proposition.

Proposition 6

The function $f(p)$ is continuous, increasing, piecewise differentiable. For $np > 2$ it is also concave if the costs are positive.

Proof:

We examine the properties of the function

$$\phi_k(p) \triangleq (1 - (1-p)^k)(1 - (1-p)^{n-k}).$$

We can easily check that

$$\frac{d}{dp}\phi_k(p) = (1-p)^{k-1}[k(1-(1-p)^{n-2k}) + n(1-p)^{n-2k}(1-(1-p)^k)] > 0,$$

$$\frac{d^2}{dp^2}\phi_k(p) = \begin{cases} [n(n-1)(1-p)^{n-k} - k(k-1) - \\ -(n-k)(n-k-1)(1-p)^{n-2k}](1-p)^{k-2}, & k \geq 2; \\ (n-1)(1-p)^{n-3}(2-np), & k = 1. \end{cases}$$

It can be easily checked that $\frac{d^2}{dp^2}\phi_k(p) < 0$ for all $k \geq 2$ and $\frac{d^2}{dp^2}\phi_1(p) < 0$ if $np > 2$. Thus the function $f_T(p)$ is continuous and differentiable since it is a polynomial and furthermore it is increasing and concave for $np > 2$, since it is a weighted sum with positive weights ($c(e) \geq 0$). Therefore, the function $f(p)$ is concave for $np > 2$ and continuous, since it is the minimum of a finite number of concave and continuous functions. Furthermore, $f(p)$ is increasing because for $p_1 < p_2$ if $f(p_i) = f_{T_i}(p_i), i = 1, 2$, then $f(p_1) = f_{T_1}(p_1) \leq f_{T_2}(p_1) < f_{T_2}(p_2) = f(p_2)$. Finally there is a finite number of trees, which can possibly minimize $f(p)$. Thus the function $f(p)$ has a finite number of breakpoints. Between successive breakpoints p_i, p_{i+1}, $f(p) = f_{T_i}(p), p_i \leq p \leq p_{i+1}$ for some T_i. Hence $f(p)$ is piecewise differentiable. •

We can now combine the above proposition 6 and our previous observations that as $p \to 1$ the PMST tends to the MST, i.e. the optimal tree for p close to 1 is the MST, and as $p \to 0$ the optimal PMST is the solution to the network design problem, to sketch a possible graph of the function $f(p)$ in Figure 5.

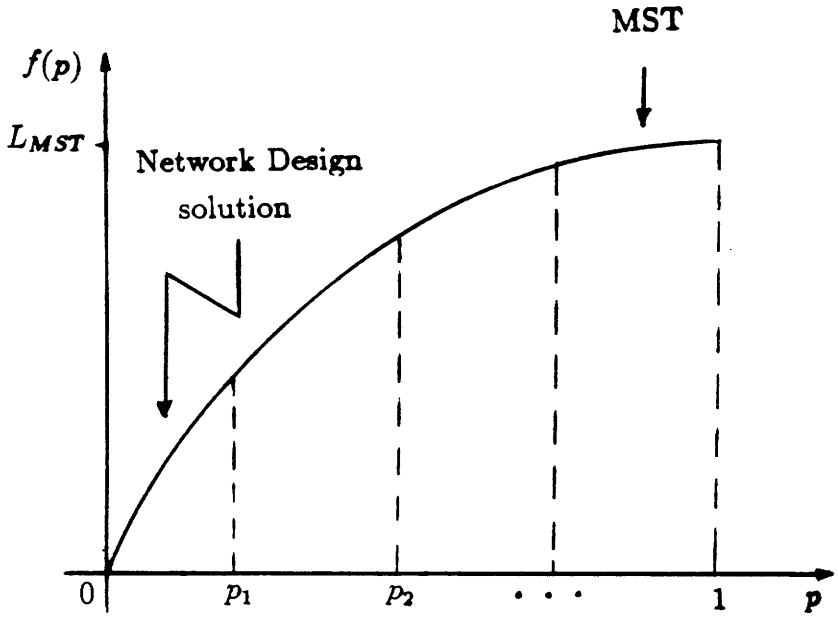


Figure 5: The PMST problem as a function of the coverage probability p

5.2 Bounds for the PMST

Based on the above functional properties of $f(p)$ and some properties of $\phi(k)$ from proposition 3 we can prove the following proposition.

Proposition 7

If T_p is the optimal PMST and L_T is the length of the tree T, then

$$\max[pL_{MST}, p(1-(1-p)^{n-1})L_{T_p}] \leq E[L_{T_p}] \leq (1-(1-p)^{\lceil \frac{n}{2} \rceil})^2 L_{MST}. \quad (10)$$

Proof:

From the concavity of the function $f(p)$ we get that

$$f(p) \geq pf(1) + (1-p)f(0) = pL_{MST},$$

where clearly $L_{MST} \triangleq$the length of the minimum spanning tree, which is the solution of PMST for $p = 1$.

From proposition 3 we get

$$\phi(1) \leq \phi(|K_e|) \leq \phi(\lfloor \frac{n}{2} \rfloor).$$

From the closed form expression (6) for $E[L_T]$ we find

$$\phi(1)L_T = \phi(1)\sum_{e \in T} c(e) \leq E[L_T] = \sum_{e \in T} c(e)\phi(|K_e|) \leq \phi(\lfloor \frac{n}{2} \rfloor)L_T.$$

Since $E[L_{T_p}] \leq E[L_{MST}]$, we easily derive (10). •

Exploiting these bounds, we address the question of how good is the MST as a solution to the PMST problem. The following is an obvious corollary of the bounds (10).

Proposition 8

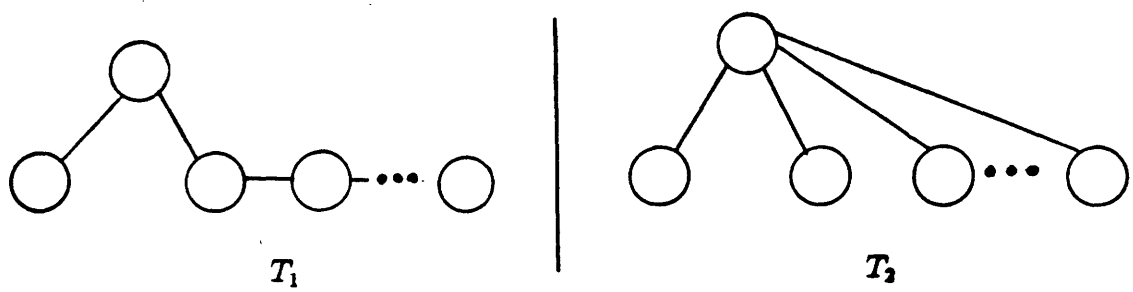


Figure 6: The trees T_1, T_2

$$\frac{E[L_{MST}] - E[L_{T_p}]}{E[L_{T_p}]} \leq \frac{(1-p)(1-(1-p)^{\lfloor \frac{n}{2} \rfloor - 1})}{p}. \tag{11}$$

Proof:

Since $E[L_{T_p}] \leq E[L_{MST}] \leq \phi(\lfloor \frac{n}{2} \rfloor) L_{MST} \leq (1-(1-p)^{\lfloor \frac{n}{2} \rfloor}) L_{MST}$, and $E[L_{T_p}] \geq pL_{MST}$ we can easily derive (11). Note that as $p \to 0$ the bound becomes $O(n)$. •

These bounds indicate that for p large enough (say $p > 1/2$) the MST solution is a good approximation for the solution of the PMST problem, which is consistent with our intuition. However, as $p \to 0$ and $n \to \infty$ this bound is not informative. In fact, the following example confirms our intuition that the MST can be a very poor solution to the PMST problem.

Consider a complete graph K_{n+1} with cost function: $c(i, i+1) = 1$, $i = 1, \ldots, n$ and $c(e) = 2$ for all $e \neq (i, i+1)$. Note that the cost function in this example satisfies the triangle inequality. If the tree T_1 is the path $1, 2, \ldots, n+1$ and T_2 is the star tree rooted at node $n+1$ (see Figure 6), then clearly T_1 is the MST. Then $E[L_{T_2}] = (2n-1)\phi(1)$ and $E[L_{T_1}] =$

$2\sum_{i=1}^{\frac{n}{2}} \phi(i) = n(1+(1-p)^n) - 2\frac{(1-p)+p(1-p)^{\frac{n}{2}}-(1-p)^n}{p}$ (assuming n is an even number). Then if T_p is the minimal PMST, we obtain

$$\frac{E[L_{MST}]}{E[L_{T_p}]} \geq \frac{E[L_{T_1}]}{E[L_{T_2}]} = \frac{n(1+(1-p)^n) - 2\frac{(1-p)+p(1-p)^{\frac{n}{2}}-(1-p)^n}{p}}{(2n-1)p(1-(1-p)^n)}.$$

If $p = \frac{a}{n}$ for some constant $a > 2$ we easily obtain as $n \to \infty$ that

$$\frac{E[L_{MST}]}{E[L_{T_p}]} \geq \Omega(n).$$

Thus from (11) we always have

$$\frac{E[L_{MST}]}{E[L_{T_p}]} = \mathrm{O}(n),$$

and we have found an example for which

$$\frac{E[L_{MST}]}{E[L_{T_p}]} = \Theta(n).$$

As a result, we conclude that the bound (11) is the best possible.

Furthermore, we can address the opposite question. How good is the PMST solution to the MST problem?

Similarly we can show

Proposition 9

$$\frac{L_{T_p} - L_{MST}}{L_{MST}} \leq \frac{1-p}{p(1-(1-p)^{n-1})}. \tag{12}$$

Proof:

Inequality (12) follows from the inequality (10) as follows:

$$p(1-(1-p)^{n-1})L_{T_p} \leq E[L_{T_p}] \leq L_{MST}.\bullet$$

5.3 Relation of the PMST problem and Re-optimization Strategies

We have suggested in section 2 the idea that the PMST problem defines an efficient strategy to update the solution to minimum spanning tree problems, when problem instances are modified probabilistically because of the absence of certain nodes from the graph. In section 2 we introduced two other alternative strategies Σ_{MST} and $\Sigma_{STEINER}$, in which we find the minimum spanning tree (MST) or the minimum Steiner tree of the set of present nodes in every instance. If $L_{MST}(S)$ and $L_{STEINER}(S)$ denote the length of the MST or the Steiner tree respectively of the nodes in the set S, we then define the expectation of these re-optimization strategies as follows:

$$E[\Sigma_{MST}] \triangleq \sum_{S \subseteq V} p(S) L_{MST}(S), \tag{13}$$

$$E[\Sigma_{STEINER}] \triangleq \sum_{S \subseteq V} p(S) L_{STEINER}(S), \tag{14}$$

where $p(S)$ was defined earlier to be the probability that only nodes in S are present. In this section, we address the question of comparing the expectation of the re-optimization strategies with the expectation of the PMST strategy.

In general it is difficult to find a closed form expression for $E[\Sigma_{MST}]$, since we have to compute a sum of $O(2^n)$ terms. Instead, we will find a bound on the $E[\Sigma_{MST}]$.

Proposition 10

If every node is independently present with probability p then

$$E[\Sigma_{MST}] \geq \frac{np + (1-p)^n - 1}{n-1} L_{MST}. \tag{15}$$

Proof:

$$E[\Sigma_{MST}] = \sum_{k=2}^{n} p^k (1-p)^{n-k} \sum_{S \subseteq V, |S|=k} L_{MST}(S).$$

We define $D_k \triangleq \sum_{S \subseteq V, |S|=k} L_{MST}(S)$ and thus

$$E[\Sigma_{MST}] = \sum_{k=2}^{n} p^k (1-p)^{n-k} D_k. \tag{16}$$

We claim that

$$D_k \geq \frac{k-1}{n-1} \binom{n}{k} L_{MST}. \tag{17}$$

We will prove the above claim by backward induction. Consider the n sets $S_i = V - \{i\}$. Then

$$L_{MST}(S_i) + c(i,j) \geq L_{MST}(V) = L_{MST} \quad \forall (i,j) \in MST, \tag{18}$$

because the tree created by adding the edge (i,j) to the MST on S_i is a feasible solution to the instance V. We apply (18) for $i = 1 \ldots n$ and since it holds for any edge in the MST, we choose for every i the corresponding edge (i,j) from the MST, such that the $n-1$ edges (i,j) are distinct, and one edge is the one with the minimum cost among all edges in the MST. Summing over all i we get

$$\sum_{i=1}^{n} L_{MST}(S_i) + \sum_{i=1}^{n} c(i,j) \geq n L_{MST}.$$

In order to choose $n-1$ edges (i,j) to be distinct and the one remaining the least in cost we perform the following algorithm:

1. Find the edge e^* with smallest cost $c(e^*)$.

2. Until the node set is non-empty,

   if i is a leaf in the MST then let (i, j_i) be the unique edge in MST.

   If $(i, j_i) \neq e^*$ delete i.

3. For the two remaining nodes let e^* be their corresponding edge.

Since there are $n-1$ edges (i,j) that are distinct, then

$$\sum_{i=1}^{n} c(i,j) = L_{MST} + c(e^*) \leq (1 + \frac{1}{n-1}) L_{MST}.$$

As a result,

$$D_{n-1} = \sum_{i=1}^{n} L_{MST}(S_i) \geq (n - 1 - \frac{1}{n-1}) L_{MST} = \frac{n(n-2)}{n-1} L_{MST}.$$

Consider now the $t = \binom{n}{k}$ subsets of V of cardinality k, $A_1, A_2, \ldots, A_t$. For all A_i let $A_{i,j} \triangleq A_i - \{j\}$. Arguing as before

$$\sum_{j} L_{MST}(A_{i,j}) \geq \frac{k(k-2)}{k-1} L_{MST}(A_i).$$

Adding with respect to i we get

$$\sum_{i,j} L_{MST}(A_{i,j}) \geq \frac{k(k-2)}{k-1} D_k. \tag{19}$$

But,

$$\sum_{i,j} L_{MST}(A_{i,j}) = (n-k+1) D_{k-1}, \tag{20}$$

since in the summation in (20) we count each distinct subset of the $\binom{n}{k-1}$ subsets of V of cardinality $k-1$, $n-k+1$ times. Combining (19), (20) we find

$$D_{k-1} \geq \frac{k(k-2)}{(k-1)(n-k+1)} D_k. \tag{21}$$

Applying (21) inductively we easily obtain (17) Then from (16), (17) we find

$$E[\Sigma_{MST}] \geq \sum_{k=1}^{n} p^k (1-p)^{n-k} \frac{k-1}{n-1} \binom{n}{k} L_{MST}.$$

Therefore,

$$E[\Sigma_{MST}] \geq \frac{np + (1-p)^n - 1}{n-1} L_{MST}.$$

Note that as $n \to \infty$ the bound becomes

$$E[\Sigma_{MST}] \geq pL_{MST}.\bullet$$

It is not clear that $E[\Sigma_{MST}] \leq E[L_{T_p}]$. In fact, we give an example where $E[\Sigma_{MST}] > E[L_{T_p}]$. Let $G = (V, E)$ be a complete graph K_n with $c(i,j) = c_i + c_j$, $c_1 \leq c_2 \leq \ldots \leq c_n$. Then, the MST is the star tree rooted at node 1. As will be shown in proposition 12 the optimal PMST is the same star tree. As a result,

$$E[L_{T_p}] = p(1-(1-p)^{n-1})[(n-1)c_1 + \sum_{k=2}^{n} c_k].$$

In this example we will be able to find a closed form expression for $E[\Sigma_{MST}]$ by exploiting the special structure of the cost function. If the ith node is present and the 1st,..., $i-1$th nodes are not present then the optimal tree is the star tree rooted at node i. From this observation we can write a closed form expression for $E[\Sigma_{MST}]$:

$$E[\Sigma_{MST}] = \sum_{i=1}^{n-1} p(1-p)^{i-1} E[L_{T_i} | \text{node} \quad i \quad \text{is} \quad \text{present}],$$

where $E[L_{T_i}]$ means the expected length in the PMST sense of the star tree rooted at node i with leaves $i+1, \ldots, n$. Since $E[L_{T_i} | i \text{ is present}] = pL_{T_i} =$

$p[(n-1-i)c_i + \sum_{k=i+1}^{n} c_k]$, after some algebraic manipulations, we easily find that

$$E[\Sigma_{MST}] = p\sum_{i=1}^{n} c_i[p(n-i)(1-p)^{i-1} + 1 - (1-p)^{i-1}].$$

Choosing $c_i = i$ we find

$$E[L_{T_p}] = \frac{n(n+1)-1}{2}p + n - \frac{3}{p} + O((1-p)^n),$$

$$E[\Sigma_{MST}] = \frac{n^2+3n-4}{2}p + O((1-p)^n).$$

Letting $np = c$ and $n \to \infty$ we see

$$E[L_{T_p}] \to (c/2 + 1 - 3/c)n, \quad E[\Sigma_{MST}] \to cn/2.$$

Thus as $n \to \infty$ $E[L_{T_p}] > E[\Sigma_{MST}]$ for $c > 3$, but $E[L_{T_p}] < E[\Sigma_{MST}]$ for $c < 3$.

6 Some Special Cases

In this section we exploit some of the combinatorial properties, which were proved in section 5, to find some special cases in which we can solve the PMST problem in polynomial time. In section 4 we have seen that the more restricted versions of the PMST problem with $c(e) = 1$ in a non-complete graph and $c(e) \in \{1, M\}$ in a complete graph are $NP - complete$.

6.1 The Role of the Star Tree

The first natural question concerns the complexity of the problem when we combine the above restrictions, i.e. when we have a complete graph with all

costs $c(e) = 1$. We prove a more general theorem, which includes this case and characterizes the optimal solution.

Theorem 11

In the case where $p_i = p$ for all $i \in V$, whenever the optimum solution of the MST problem is a star tree T_*, then T_* is also the solution to the PMST problem.

Proof:

For all trees T

$$E[L_T] = \sum_{e \in T} c(e)\phi(|K_e|) \geq \phi(1)L_T$$

from proposition 3. But

$$E[L_{T_*}] = \sum_{e \in T} c(e)\phi(1) = \phi(1)L_{T_*}.$$

Since T_* is by assumption the MST $L_T \geq L_{T_*}$ for all trees. Combining the above inequalities

$$E[L_{T_*}] \leq E[L_T].$$

Therefore, the star tree T_* solves the PMST problem. •

Theorem 11 characterizes the optimal solution whenever the MST is a star tree T_*. But are there interesting examples in which T_* is the MST?

Proposition 12

In the following examples the MST is a star tree T_* and thus, by theorem 11, T_* is the PMST.

1. A complete graph, with $c(i,j) = c_i + c_j$.

2. A complete graph, with $c(i,j) = c_i c_j, c_i \geq 0$.

3. A complete graph, with $c(i,j) = c_i + c_j + d_i d_j$,with $c_1 = \min c_i$ and $d_1 = \min d_i$.

4. A complete graph, with $c(i,j) = \min(c_i, c_j)$.

Proof:

Consider an arbitrary tree T. Without loss of generality we assume that $c_1 = \min_i c_i$. Since T is connected its cost is $L_T = c(2,i_2)+\ldots+c(n,i_n)$, where at least one i_j is 1. Then $L_T = c_2+c_{i_2}+\ldots+c_n+c_{i_n} \geq (n-1)c_1+c_2+\ldots+c_n = L_{T_*}$, i.e. T_* is the MST, with T_* rooted in node 1.

With exactly the same argument we can prove that T_* is the MST in the other cases. •

A corollary of theorem 11 is that in a complete graph with $c(e) = 1$ the MST is a star tree T_* and thus T_* is also the PMST. Hence, in this case the optimal solution can be found in $O(n)$ time.

6.2 The Case $p_i \neq p_j$

We have shown that the optimum PMST, in the case $p_i = p$, is a star tree T_*, whenever T_* is the MST. Does this result continue to hold even in the case $p_i \neq p_j$? The following theorem answers this question.

Theorem 13

If the probability of presence of node i is p_i, $p_1 = \min_i p_i$ and the MST is a star tree T_* rooted at node 1, then T_* is the PMST.

Proof:

From the closed form expression (3) for any tree T

$$E[L_T] = \sum_{e \in T} c(e)\{1 - \prod_{i \in K_e}(1-p_i)\}\{1 - \prod_{i \in V-K_e}(1-p_i)\}.$$

Let $x_i \triangleq 1 - p_i$ and without loss of generality we assume that $1 \in K_e$. But

$$(1-\prod_{i \in K_e} x_i)(1-\prod_{i \in V-K_e} x_i)-(1-x_1)(1-\prod_{i=2}^{n} x_i) = (x_1-\prod_{i \in V-K_e} x_i)(1-\prod_{i \in K_e-\{1\}} x_i) \geq 0,$$

because $\prod_{i \in K_e-\{1\}} x_i \leq 1$ and $x_1 \geq x_i \geq \prod_{i \in V-K_e} x_i$.

As a result,

$$E[L_T] \geq p_1(1 - \prod_{i=2}^{n}(1-p_i)) \sum_{e \in T} c(e).$$

By the assumption that $L_{T_*} \leq L_T$ we find that

$$E[L_T] \geq E[L_{T_*}].$$

The star tree T_* is the PMST. •

As a corollary, in a complete graph with $c(e) = 1$ the PMST is the star tree rooted at node l, where $p_l = \min_i p_i$.

6.3 Sensitivity Analysis

We investigate next the conditions under which the star tree T_*, which was optimal for certain special cases, remains optimal in the case $p_i = p$ when the cost function is arbitrary.

We define a node in a tree to be an outer node if the degree of the node is one, an inner node if the degree of the node is two or more. If we erase all outer nodes from a tree of n nodes then the remaining graph is again a tree formed by inner nodes. This tree will be referred to as the inner tree T_I. In

the tree T_I, there again must be some nodes of degree one and these nodes are called extreme inner nodes.

Theorem 14

Let a, b, c be the costs of three sides of any triangle, i.e. any set of three nodes, in the $n - node$ network ($n \geq 4$), with $a \leq b \leq c$. If there exists a positive t with

$$t \leq (1-p)(1-(1-p)^{\lfloor \frac{n}{2} \rfloor - 1})^2 / p(\lfloor \frac{n}{2} \rfloor - 1)(1-(1-p)^{n-1}) = \Theta(\frac{1}{n}),$$

such that

$$a + tb \geq c$$

for all triangles in the network, then there exists a PMST which is a star tree.

Proof:

Note that the smaller the value of t, the more restrictive the inequality. If $t = 0$, then it restricts all sides of any triangle to be of equal length. If $t = 1$ it reduces to the regular triangle inequality. Since the value of t must be less than 1, this condition is stronger than the triangle inequality, i.e. more restrictive.

It is sufficient to show that we can reduce the number of inner nodes in any spanning tree, which is not a star tree, without increasing the expected cost. So, let T be any spanning tree which contains at least two inner nodes. Let N_q be an extreme inner node in T with a neighbor N_p which is an inner node. Since N_q is an extreme inner node, all its neighbors except N_p must be of degree one in T. Call these nodes $N_1, \ldots N_{k-1}$. This is shown in Figure 7, where the distance between N_p and N_i is denoted by c_i.

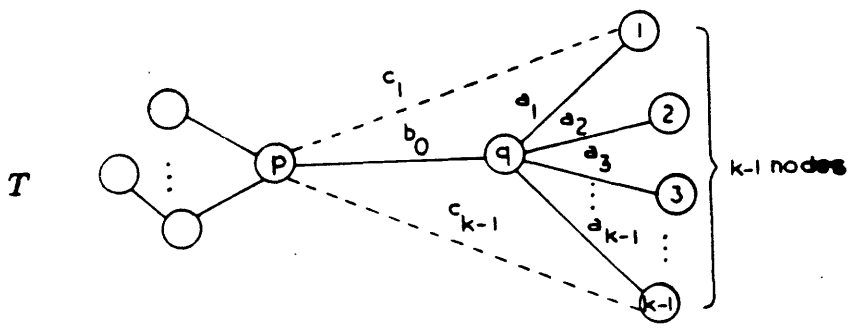


Figure 7: The tree T

Let us construct a new spanning tree T_1 which is the same as T except that the nodes N_i $(i = 1, \ldots, k-1)$ are connected to N_p directly. In the new tree T_1, N_q is no longer an inner node. Thus the number of inner nodes has decreased by one. We will show that

$$E[L_{T_1}] \leq E[L_T].$$

If we apply this idea recursively to the trees $T_1, T_2, \ldots$, we will finally get a tree T_* which is a star tree. Since the part of the tree to the left of N_p is exactly the same for both T and T_1, we need only calculate the expected cost for the part to the right of N_p. If $x = 1 - p$, then

$$E[L_T] - E[L_{T_1}] = b_0(1 - x^k)(1 - x^{n-k}) + \sum_{i=1}^{k-1} a_i(1 - x)(1 - x^{n-1}) -$$

$$b_0(1 - x)(1 - x^{n-1}) - \sum_{i=1}^{k-1} c_i(1 - x)(1 - x^{n-1}).$$

The net decrease of expected cost in changing from T to T_1 is

$$E[L_T] - E[L_{T_1}] = (1 - x)(1 - x^{n-1}) \sum_{i=1}^{k-1} [a_i - c_i + b_0 \frac{x(1 - x^{k-1})(1 - x^{n-k-1})}{(k-1)(1 - x)(1 - x^{n-1})}].$$

The decrease will be positive if each term is positive, namely

$$a_i + b_0 \frac{x(1-x^{k-1})(1-x^{n-k-1})}{(k-1)(1-x)(1-x^{n-1})} \geq c_i.$$

For a fixed n, $x(1-x^{k-1})(1-x^{n-k-1})/(k-1)(1-x)(1-x^{n-1})$ is smallest when k is largest, that is when $k = \lfloor n/2 \rfloor$. Thus it is sufficient to have

$$a_i + b_0 \frac{(1-p)(1-(1-p)^{\lfloor \frac{n}{2} \rfloor - 1})^2}{p(\lfloor \frac{n}{2} \rfloor - 1)(1-(1-p)^{n-1})} \geq c_i.$$

Assuming $a_i \leq b_0 \leq c_i$ gives the strongest inequality, and we have the statement of the theorem. •

7 Some Concluding Remarks

We have seen that a natural probabilistic variation of a classical combinatorial problem has the potential to model various practical situations, offers an alternative way to update solutions to problem instances which are modified probabilistically and leads to very different properties in comparison with its deterministic counterpart. The simplest possible version of the PMST problem was proved to be $NP - complete$, in sharp contrast with the fact that the MST problem is solved by a greedy, most straightforward algorithm.

Surprisingly, our analysis of the combinatorial properties of the problems established some interesting connections with the network design problem and naturally with the MST. In particular, as the probability of presence p tends to 0, the PMST approaches the solution to the network design problem. This limiting behavior suggests the idea of solving the network design problem as a sequence of PMST problems, which is a topic of future research.

At a final step, we examined the special role of the star tree, which can be the solution of the PMST problem under some conditions.

As a general conclusion, probabilistic variations of classical combinatorial optimization problems raise interesting and entirely new questions compared with their deterministic counterparts and in addition, understanding of the properties of the probabilistic problem can add insight to deterministic problems, as it was the case with the network design problem.

Acknowledgments

I would like to thank my thesis advisor Professor Amedeo Odoni for several useful comments and interesting discussions. I want also to thank Professor Patrick Jaillet for his constructive remarks. This work was partially supported from the National Science Foundation under grant ECS-8717970.

References

[1] Bertsimas D., (1988), "Probabilistic Combinatorial Optimization Problems", Ph.D thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[2] Bertsimas D., (1988), "The Probabilistic Minimum Spanning Tree Problem, Part II: Probabilistic Analysis and Asymptotic Results", submitted for publication.

[3] Garey M. and Johnson D., (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco.

[4] Jaillet P., (1985), "Probabilistic Traveling Salesman Problems", (Ph.D. Thesis) Technical Report No. 185, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[5] Jaillet P. and Odoni A., (1988), The Probabilistic Vehicle Routing Problem, in Vehicle Routing; Methods and Studies, edited by B.L. Golden and A.A. Assad, North Holland, Amsterdam.

[6] Johnson D., Lenstra J.K., A. Rinnooy Kan, (1978), "The Complexity of the Network Design Problem", Networks, 8, 279-285.

[7] Papadimitriou C.H. and Steiglitz K., (1982), Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, New Jersey.