

# Shortest Path Problems with Node Failures

Patrick Jaillet

MSIS Department

The University of Texas at Austin

and Laboratoire de Mathématiques et Modélisation

ENPC, France\*

Consider the problem of finding the shortest paths from a node source  $s$  to a node sink  $t$  in a complete network. On any given instance of the problem, only a subset of the intermediate nodes can be used to go from  $s$  to  $t$ , the subset being chosen according to a given probability law. We wish to find an a priori path from  $s$  to  $t$  such that, on any given instance of the problem, the sequence of nodes defining the path is preserved but only the permissible nodes are traversed, the others being skipped. The problem of finding an a priori path of minimum expected length is defined as the Probabilistic Shortest Path Problem (PSPP). Note that if the network is not originally complete, the PSPP methodology can still be used if we first add each missing edge, together with a *deterministic* length (being defined by an alternative path using nodes that have no probability of failure). In this paper, after discussing potential applications of the PSPP, we study the complexity of this class of problems. We first show that the problem is, in general, NP-hard and then we develop polynomial time procedures for special cases of it. We also consider the complexity of a related problem: the Probabilistic Minimum Spanning Tree Problem (PMSTP). Finally, we provide a discussion of the implications of the results.

© 1992 John Wiley & Sons, Inc.

## 1. INTRODUCTION

Combinatorial optimization problems on graphs are important. They cover a wide variety of models with practical applications and the theory behind them is, in general, very rich.

Most (but not all) of the research devoted to this area has concentrated on deterministic situations. By that, we mean situations in which the number of nodes, the length of the arcs, etc., are known with certainty before a particular optimization problem is solved. One can identify, however, a practically endless variety of problems in which one or more of these parameters are random variables, i.e., subject to uncertainty in accordance with some probability law. Most of the existing literature addressing uncertainty has been confined to the analysis of problems under the assumptions of random distances between

\*Supported in part by a joint CNRS/NSF Grant.

nodes (see [13] for an introduction to the analysis of network problems under uncertainty; see also [1, 5, 6]).

Here, we are concerned with a somewhat different approach toward introducing and analyzing uncertainty in combinatorial problems. This approach was initiated in [8] with the traveling salesman problem and since then has been extended to the vehicle routing problem (see [9, 12]), the spanning tree problem, and the traveling salesman facility location problem (see [2]). Our main concern is to define and analyze probabilistic versions of well-known combinatorial optimization problems while keeping their original (combinatorial) flavor. There are several motivations behind this work. Among them, two are particularly important—The first one is the desire to formulate and analyze models that are more appropriate for real-world problems where randomness is present. The second motivation is an attempt to analyze the robustness (with respect to optimality) of optimal solutions for deterministic problems when the network for which the problem has been solved is modified (in our case the perturbation of the problem instances is simulated by the presence or not of the given nodes).

In this paper, we consider special versions of the Probabilistic Shortest Path Problem (PSPP) and of the Probabilistic Minimum Spanning Tree Problem (PMSTP). We first analyze the PSPP and, after showing that the problem is, in general, NP-hard, we give polynomial time procedures for special cases of it. We then show that the recognition versions of the analogous special cases for the PMSTP are NP-complete. Finally, we provide a discussion of the implications of these results.

## 2. THE PROBABILISTIC SHORTEST PATH PROBLEM

### 2.1. Introduction

The Deterministic Shortest Path Problem (DSPP) is one of the most studied problems in network optimization and can be easily solved in polynomial time through a number of well-known algorithms (for instance, see [14] for a survey). Stochastic versions of this problem have received little attention so far and have been limited mainly to the case of random arc lengths (see [1, 6]).

The PSPP can be described as follows: Consider the problem of finding a shortest path between a node source  $s$  and a node sink  $t$  in a complete network having a length associated with each arc. On any given instance of the problem, only a subset among intermediate nodes can be used to go from  $s$  to  $t$ , the subset being chosen according to a given probability law. We wish to construct a priori path such that, on any given instance of the problem, the sequence of nodes defining the path is preserved but only the permissible nodes are traversed, the others being skipped. The problem of finding an a priori path of minimum expected length is defined as a PSPP. This problem was first introduced in [10] in which a branch-and-bound scheme was proposed. Note that this definition is based on a very specific strategy, called here a priori optimization, which differs radically from the strategy of reoptimization. More specifi-

cally, rather than reoptimizing every potential instance, we wish to find an a priori solution to the original problem and then update in a simple way this a priori solution to answer each particular instance. Also, it is worth mentioning that if the physical underlying network is not complete the model can still be used by first extending the network to a complete one. In that case, a crucial condition for the network completion to be valid with respect to the PSPP is the following: Given the selected a priori path  $(s, n_1, n_2, \dots, n_k, t)$ , if nodes  $n_{i+1}$  through  $n_{i+r}$  fail, then they are skipped by either taking the arc  $(n_i, n_{i+r+1})$  if it exists or by taking an alternative path from  $n_i$  to  $n_{i+r+1}$  *using only nodes that have no probability of failure*. This latter case is to ensure that we do not define a “random” length for this missing edge, i.e., an edge whose length would depend on the presence or not of probabilistic nodes.

The generic PSPP, as stated, can be of interest in many modeling applications. First, consider a network in which arcs represent streets of a city and nodes are intersections, and suppose we want to go from an origin  $s$  to a destination  $t$  along this network. The length of an arc  $(i, j)$  is defined to be the time needed to go from  $i$  to  $j$ , a value that may vary greatly. Usually, one either uses the means and solves a deterministic shortest path problem or one considers the travel times as random variables and tries to solve one of many possible problems such as finding the path with maximum probability of being the shortest, or finding the path with minimum variance among the shortest, etc. (see [1] for a discussion on these formulations). However, if the traffic is close to saturation, a little perturbation in the flow can create a blocking situation. In that case, we have a critical situation in which the travel time does not vary much anymore (being constantly high because of congestion) except when a blocking situation is faced (a street unexpectedly blocked by unloading trucks or garbage collecting trucks; an intersection with conflicting flows of vehicles; etc.). Depending on the frequencies of these blockings, one might then choose longer but less risky paths to go from one place to another. Let us see how the PSPP offers an analytical way to choose between paths with these uncertainties. First of all, one can model a risky street  $(i, j)$  by adding an artificial “probabilistic” node  $k$  that can or cannot be traversed: In the first case, the length of the street would be its normal travel time; in the second case, it would be much higher. In the case of a risky intersection, the model is even simpler: We simply model the intersection as a “probabilistic” node. The only apparent difficulty in applying the PSPP model for this kind of problem is that the physical network of streets is usually not complete. But this can be resolved by adding weighted arcs as follows (see Fig. 1 for the case of a simple symmetric network):

- First, delete all “probabilistic” nodes and their adjacent arcs, and call the remaining network the *backbone*  $B$ . Make  $B$  complete by using shortest paths. In our example, edges  $(s, t)$ ,  $(1, 3)$ , and  $(1, t)$  are thus added. Note that if  $B$  is disconnected, every arc added between disconnected components will then be given a large chosen weight, say  $M$  (this case is not covered in our example).

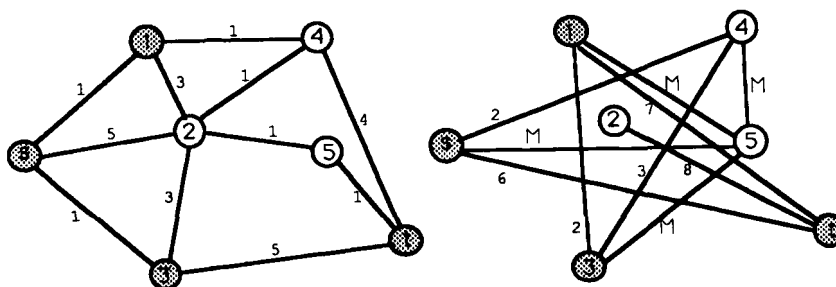


FIG. 1. An original network and the addition of missing weighted arcs.

- Then, consider a “probabilistic” node, say  $i$ , and add it (together with adjacent arcs) to  $B$ . From  $i$ , add all necessary arcs with weight equal to shortest paths that do not use  $t$  as an intermediate node. Then, remove  $i$  from  $B$  and repeat the procedure for the other “probabilistic” nodes. In our example, the edges  $(2,t)$ , then  $(s,4)$ ,  $(3,4)$ , and, finally,  $(s,5)$ ,  $(1,5)$ ,  $(3,5)$  are thus added. Again, if the “probabilistic” node is not connected to  $B$  by a node other than  $t$ , each added arc will then be given the large value  $M$  (in our example, it is the case for node 5).
- Finally, consider two nonadjacent “probabilistic” nodes, say  $i$  and  $j$ , and add them (with adjacent arcs) to  $B$ . Add arc  $(i,j)$  with weight equal to the shortest path between  $i$  and  $j$  that does not use  $t$  as an intermediate node. Then, remove the pair  $i,j$ , and repeat the procedure for the other pairs of nonadjacent “probabilistic” nodes [in our example, edge  $(4,5)$  is thus added].

Note that this entire procedure require (see above) the choice of a single large value  $M$ . This quantity can be interpreted as the penalty (in units of time) of being, at one point along the path, in the impossibility of moving forward on a given instance of the problem.

Let us now turn our attention to some other applications. In the context of flying operations, nodes  $s$  and  $t$  can be airports and the other nodes can represent geographical areas (mountains, countries, etc.) that can or cannot be flown over by aircraft going from  $s$  to  $t$  (e.g., because of weather conditions, unexpected military restrictions, etc.). The modifications of the route the plane has to take, because of such unexpected restrictions, can be very costly if not planned ahead. The idea would be to include explicitly these uncertainties in the model in order to find routes of minimum expected costs. The PSPP model is quite appropriate for such a situation.

More important, consider the following class of problems: Suppose one has to go from a “city”  $s$  to another city  $t$ , possibly going through other cities in which one can receive some “revenue” with a given probability. With the objective of minimizing the expected total net cost, this problem can be modeled as a PSPP: The underlying network is a complete graph built on all cities of interest, and the length of an arc  $(i,j)$  is the net cost between city  $i$  and city  $j$

[minimum cost of traveling from  $i$  to  $j$  minus half the total revenue in the cities,  $c_{ij} - (r_i + r_j)/2$ ].

Let us finally mention another useful characteristic of the PSPP: In the context of network reliability, the PSPP can be interpreted as a local (hence, easily implementable) strategy of handling node failures (“we skip them”) that is optimal in a global sense (minimization of the expected cost). This can be useful either for describing operating strategies of unreliable networks or for having a polynomial time computable estimate of “connecting cost versus node failures” for such networks.

**2.2. Notation and Assumptions**

$G = (N, A, d)$  denotes a complete, loopless, directed, weighted graph where  $N$  is the node set,  $A$  is the set of arcs joining the nodes of  $N$ , and  $d$  is a function  $A \mapsto R$ . We consider a node source  $s$ , a node sink  $t$ , and paths from  $s$  to  $t$ . A path  $c$  will be given by the sequence of nodes defining it, i.e.,  $c = (s, n_1, n_2, \dots, n_k, t)$ . The set of nodes  $N$  is partitioned into two subsets  $N_1$  and  $N_2$ :

$N_1$  is the set of nodes without failure or always working (“black” nodes), of cardinality  $|N_1| = m (m \geq 2$  since  $s$  and  $t$  belong to  $N_1)$ .

$N_2$  is the set of nodes with possible failure or not always working (“white” nodes), of cardinality  $|N_2| = n$ .

We assume given a probability  $\mathbb{P}$  on  $\Omega$ , the power set of  $N_2$  (an outcome  $\omega$  defines the subset of white nodes with no failure on this particular instance). We restrict  $\mathbb{P}$  to be such that all outcomes of equal cardinality have the same probability of occurring, i.e.:

$$\forall \omega_1 \in \Omega, \quad \forall \omega_2 \in \Omega, \quad |\omega_1| = |\omega_2| \Rightarrow \mathbb{P}(\omega_1) = \mathbb{P}(\omega_2). \tag{1}$$

If  $W$  is the random variable that represents the number of white nodes with no failure, we have

$$\Pr(W = |\omega|) = \binom{n}{|\omega|} \mathbb{P}(\omega). \tag{2}$$

Hence, our probabilistic models can be specified equivalently by giving the probability  $\mathbb{P}$  or the probability distribution of  $W$ . Note also that the restriction imposed on  $\mathbb{P}$  implies that, given  $W = k$ , the  $k$  nodes are selected uniformly at random among the set of  $n$  nodes; any probability  $\mathbb{P}$  satisfying (1) will then be said to be *node invariant*.

One important specific example (hereafter, named  $\mathbb{P}_1$ ) is

$$\mathbb{P}_1(\omega) = p^{|\omega|}(1 - p)^{n-|\omega|}, \tag{3}$$

which corresponds to the case for which each white node has a probability  $p$  of being present, independently of the others; we then speak informally of a Bernouilli process with parameter  $p$ .

For a given a priori path  $c$  between  $s$  and  $t$ , the length  $L_c$  covered in traversing the nodes without failure on each instance of the problem is a random variable. The general PSPP can then be stated as follows:

**Problem PSPP:** Given a network  $G = (N, A, d)$  with node source  $s$  and node sink  $t$  and a probability  $\mathbb{P}$ , find an a priori path  $c$  of minimum expected length,  $E[L_c]$ .

### 2.3. Properties and Characteristics of Optimal Solutions to PSPPs

In this subsection, we investigate the complexity of the PSPP and its links to its deterministic counterpart.

#### 2.3.1. The Expected Length of a Given Path

Let us consider a PSPP problem. For a given path  $c = (s, n_1, \dots, n_k, t)$ ,  $L_c$  is a random variable that can have up to  $2^k$  different values. By considering all cases, its expected value would then be obtained in  $O(k2^k)$  additions in the worst case, which is prohibitive. In fact, one can develop efficient methods to compute  $E[L_c]$  for general node invariant probabilities. The most general result, based on an extension of results obtained for the PTSP (see [11]), is the following:

**Theorem 1.** *Given a node invariant PSPP and a path  $c = (0, 1, \dots, k, k+1)$  from  $s$  to  $t$  (by convention  $0 \equiv s$  and  $k + 1 \equiv t$ ), we have*

$$E[L_c] = \sum_{r=0}^{k-2} \alpha_r A_c^{(r)} + \sum_{r=0}^{k-1} \beta_r B_c^{(r)} + \gamma_k C_c^{(k)}, \tag{4}$$

where

$$A_c^{(r)} = \sum_{i=1}^{k-1-r} d_c(i, i + r + 1)$$

$$B_c^{(r)} = d_c(0, r + 1) + d_c(k - r, k + 1)$$

$$C_c^{(k)} = d_c(0, k + 1)$$

$$\alpha_r = \sum_{j=r}^{n-2} \left( \binom{n-2-r}{j-r} / \binom{n}{j} \right) \Pr(W = n - j) \text{ for all } r \in [0 \dots k - 2]$$

$$\beta_r = \sum_{j=r}^{n-1} \left( \binom{n-1-r}{j-r} / \binom{n}{j} \right) \Pr(W = n - j) \text{ for all } r \in [0 \dots k - 1]$$

$$\gamma_k = \sum_{j=k}^n \left( \binom{n-k}{j-k} / \binom{n}{j} \right) \Pr(W = n - j),$$

with the following convention:

$d_c(i, i + r + 1) = \sum_{e=0}^s d(b_e, b_{e+1})$ , where  $b_0 \equiv i$ ,  $b_{s+1} \equiv i + r + 1$ , and  $(b_1, \dots, b_s)$  is the sequence of black nodes drawn from  $(i + 1, \dots, i + r)$ .

*Proof.* The proof is similar to the one given in [11] for the PTSP and is not reproduced here in all the details. Suppose first that  $m = 2$ . On any given instance of the problem, the arc  $(i, i + r + 1)$  is in the resulting path if and only if the nodes  $i$  and  $i + r + 1$  are working and the nodes  $i + 1, \dots, i + r$  are not working. The probability of presence of this arc will then depend on three cases:

1.  $i \neq 0$  and  $i + r + 1 \neq k + 1$ , the probability of presence is  $\alpha_r$ .
2.  $i = 0$  or  $i + r + 1 = k + 1$ , the probability is  $\beta_r$ .
3.  $i = 0$  or  $i + r + 1 = k + 1$ , the probability is  $\gamma_k$ .

The definitions of  $A_c$ 's,  $B_c$ 's, and  $C_c^k$  are then based on a regrouping of arcs with equal probabilities (as an illustration of these quantities, see Fig. 2 for a simple example). Finally, when  $m > 2$ , we introduce a "transformed" distance  $d_c$  that reflects the possibility of having black nodes (hence, always working) between  $i$  and  $i + r + 1$  along the path  $c$ . It is then straightforward, but cumbersome, to verify that (4) indeed leads to the right probability of presence for all edges of the given path. For example, consider the case in which the path contains one black node, say node 3. Then, from our theorem, an edge such as  $(0,3)$  will not only appear in  $B_c^{(2)}$ , but also in all  $B_c^{(r)}$  for  $r > 2$  and in  $C_c^k$ . The total weight for this edge will then be

$$\sum_{r=2}^{k-1} \beta_r + \gamma_k = \sum_{r=2}^{k-1} \sum_{j=r}^{n-1} \left( \binom{n-1-r}{j-r} / \binom{n}{j} \right) \Pr(W = n - j) + \sum_{j=k}^n \left( \binom{n-k}{j-k} / \binom{n}{j} \right) \Pr(W = n - j).$$

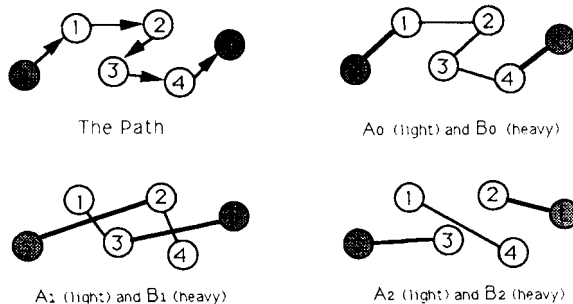


FIG. 2. A path and the arcs representing some  $A_c$ 's and  $B_c$ 's.

This simplifies to

$$\sum_{j=2}^n \left( \binom{n-2}{j-2} / \binom{n}{j} \right) \Pr(W = n - j),$$

which is, indeed, the true probability of presence of edge (0,3). ■

**Remarks:**

1. When  $k = 1$  or when  $k = 0$ , Eq. (4) gives, respectively,  $E[L_c] = \beta_0 B_c^{(0)} + \gamma_1 C_c^{(1)}$  and  $E[L_c] = \gamma_0 C_c^{(0)}$ .
2. The closed form formula (4) gives the expected length of a path from  $s$  to  $t$  through  $k$  intermediate nodes in  $O(k^2)$  elementary operations (for a general node invariant  $\mathbb{P}$  and assuming that the  $\alpha_r$ 's have been previously computed). As a consequence, it shows that the recognition version of this problem belongs to the class *NP*.
3. For the case of a Bernoulli process  $\mathbb{P}_1$ ,  $\alpha_r = p^2(1-p)^r$ ,  $\beta_r = p(1-p)^r$ , and  $\gamma_k = (1-p)^k$ .

**2.3.2. The Complexity of the PSPP and Its Relationship to the DSPP**

The DSPP is a special case of the PSPP in which all nodes are black; it is then natural to investigate the possible links between the two problems. We show in this section that the two problems are fundamentally different so that the PSPP requires entirely new solution procedures. The following results are easily proved and confirm previous knowledge about the drastic changes that randomness causes to well-known combinatorial problems (see [1, 2, 11]).

**Result 1.** *The PSPP is NP-hard. Indeed, consider the special case in which  $\mathbb{P} = \mathbb{P}_1$  and in which we have only two black nodes ( $s$  and  $t$ ). From Theorem 1, the expected length of a path containing  $k$  ( $k \leq n$ ) white nodes depends on  $d(s,t)$  via a weight equal to  $(1-p)^k$ . Since  $(1-p)^k \geq (1-p)^n$ , if we take  $d(s,t)$  arbitrarily large (everything else being equal), we can force the potential candidates for the corresponding PSPP to go through all the  $n$  white nodes. But this last problem is the Probabilistic Hamiltonian Path Problem that is NP-hard (being equivalent to the PTSP, see [8]).*

**Result 2.** *Suppose that the distances do not satisfy the triangular inequality, since, otherwise, the optimal PSPP path would simply be the arc  $(s,t)$ . Under that condition, it is easy to construct examples in which an optimal DSPP path is arbitrarily bad for the corresponding PSPP. See, e.g., Figure 3 in which all arcs not shown are of length 4 and in which  $M > 4$  and  $\mathbb{P} = \mathbb{P}_1$ . The optimal DSPP path  $(s,1,2,t)$  of length 3 has an expected length depending on  $M$ , the length of arc  $(1,t)$ , which is traversed when node 1 works and node 2 fails; it can then be made arbitrarily large as compared to the expected length of path  $(s,t)$  (of value 4).*



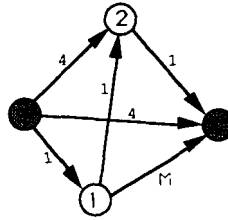


FIG. 3. The optimal deterministic shortest path can be bad.

**Result 3.** *The principle of optimality (which helps solving the DSPP) does not work here. The main reason is that the expected length of a path is not an additive functional (as compared to the length of a path), in the sense that in general  $E[L_{c_1 \oplus c_2}] \neq E[L_{c_1}] + E[L_{c_2}]$ , where  $c_1 \oplus c_2 = (i_1, \dots, i_k)$  stands for the concatenation of the two paths  $c_1 = (i_1, \dots, i_j)$  and  $c_2 = (i_j, \dots, i_k)$ .*

From Result 1, a polynomial time algorithm for the PSPP seems out of reach; from Result 2, the optimal DSPP path cannot be considered as a good candidate for approximating the corresponding PSPP; and from Result 3, one has to be careful about transferring classical DSPP algorithms to the PSPP.

A practical consequence of these results is the necessity to develop entirely new solution procedures. As mentioned in the Introduction, a branch-and-bound scheme was proposed in [10]. We will concentrate in the next section on showing that certain subclasses of the PSPP are solvable in polynomial time.

**2.4. Polynomial Procedures for Special Cases of the Problem**

In this section, we consider special cases of the PSPP for which we are able to give polynomial time procedures. Let us first consider the simplest of these cases and show how this can be done. The other cases will be straightforward extensions of the main idea developed here.

**2.4.1. A Simple Special Case**

Let us assume that  $m = 2$ , i.e., the only black nodes are  $s$  and  $t$ . Also, let us consider a probability  $\mathbb{P}$  such that

$$\Pr(W \leq n - 2) = 0 \quad \text{and} \quad \Pr(W = n - 1) > 0,$$

i.e., either all white nodes are working or only one of them has a failure. A path  $c = (s, 1, \dots, k, t)$  will then have an expected length given by

$$E[L_c] = \begin{cases} \alpha_0 A_c^{(0)} + \beta_0 B_c^{(0)} + \alpha_1 A_c^{(1)} + \beta_1 B_c^{(1)} & \text{if } k \geq 2 \\ \beta_0 B_c^{(0)} + \gamma_1 d(s, t) & \text{if } k = 1 \\ d(s, t) & \text{if } k = 0. \end{cases} \quad (5)$$

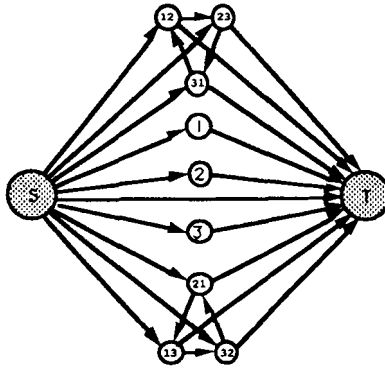


FIG. 4. The auxiliary network when  $N_2 = \{1,2,3\}$ .

Indeed, (5) is obtained from Eq. (4) by noticing that  $\alpha_r = 0$  and  $\beta_r = 0$  for all  $r \geq 2$ ,  $\gamma_k = 0$  if  $k \geq 2$ , and  $\gamma_0 = 1$ .

Let us construct an auxiliary network  $(V, E, \varphi)$  as follows (see Fig. 4 for an illustration):

- The set of nodes is  $V = (N_2 \otimes N_2) \cup N_1 \cup N_2$  of cardinality  $|V| = n^2 + 2$  (where the notation  $A \otimes A$  stands for  $(A \times A) \setminus \text{diag}(A)$ ).
- The set of arcs  $E$  and the arc-length function  $\varphi$  are defined by

- Arc  $(s, t)$  of length  $\varphi(s, t) = d(s, t)$ .
- Arcs  $(s, v)$  for all  $v = i$  in  $N_2$  of length  $\varphi(s, i) = \beta_0 d(s, i) + (\gamma_1/2)d(s, t)$ .
- Arcs  $(s, v)$  for all  $v = (i, j)$  in  $N_2 \otimes N_2$  of length  $\varphi(s, (i, j)) = \beta_0 d(s, i) + (\alpha_0/2)d(i, j) + \beta_1 d(s, j)$ .
- Arcs  $(v, t)$  for all  $v = i$  in  $N_2$  of length  $\varphi(i, t) = \beta_0 d(i, t) + (\gamma_1/2)d(s, t)$ .
- Arcs  $(v, t)$  for all  $v = (i, j)$  in  $N_2 \otimes N_2$  of length  $\varphi((i, j), t) = \beta_0 d(j, t) + (\alpha_0/2)d(i, j) + \beta_1 d(i, t)$ .
- Arcs  $(v, w)$  for all  $v = (i, j)$ ,  $w = (j, k)$ ,  $i \neq j \neq k$  of length  $\varphi((i, j), (j, k)) = (\alpha_0/2)(d(i, j) + d(j, k)) + \alpha_1 d(i, k)$ .

The cardinality of  $E$  is then

$$|E| = n(n - 1)(n - 2) + 1 + 2(n(n - 1) + n) = n^3 - n^2 + 2n + 1.$$

One can now give the fundamental relationship between the two networks:

**Lemma 1.** *There is a one-to-one correspondence between the set of paths from  $s$  to  $t$  in  $(N, A, d)$  and the set of paths from  $s$  to  $t$  in  $(V, E, \varphi)$ . Moreover, the expected length of each path from  $s$  to  $t$  in  $(N, A, d)$  is equal to the length of a corresponding path in  $(V, E, \varphi)$ .*

*Proof.* For each path  $c$  from  $s$  to  $t$  in  $(N, A, d)$ , let us exhibit a path  $C$  in  $(V, E, \varphi)$  such that  $E[L_c] = L_C$ .

Let  $c = (s, 1, \dots, k, t)$  be a path in  $(N, A, d)$ . Three cases have to be considered:

- $k = 0$ , i.e.,  $c = (s, t)$ , then  $C = (s, t)$  is the corresponding path. From Eq. (5) and the construction of  $\varphi$ , we have

$$E[L_c] = d(s, t) = \varphi(s, t) = L_C.$$

- $k = 1$ , i.e.,  $c = (s, i, t)$ , then  $C = (s, i, t)$  is the corresponding path. From Eq. (5), we have

$$E[L_c] = \beta_0(d(s, i) + d(i, t)) + \gamma_1 d(s, t).$$

Also, from the construction of  $\varphi$ , we have

$$\begin{aligned} L_C &= \varphi(s, i) + \varphi(i, t) \\ &= \beta_0 d(s, i) + (\gamma_1/2)d(s, t) + \beta_0 d(i, t) \\ &\quad + (\gamma_1/2)d(s, t). \end{aligned}$$

So  $E[L_c] = L_C$ .

- $k \geq 2$ , i.e.,  $c = (s, 1, \dots, k, t)$ , then  $C = (s, (1, 2), (2, 3), \dots, (k - 1, k), t)$  is the corresponding path. From Eq. (5), we have

$$\begin{aligned} E[L_c] &= \alpha_0 \sum_{i=1}^{k-1} d(i, i + 1) + \beta_0(d(s, 1) + d(k, t)) \\ &\quad + \alpha_1 \sum_{i=1}^{k-2} d(i, i + 2) + \beta_1(d(s, 2) + d(k - 1, t)). \end{aligned}$$

Also, from the construction of  $\varphi$ , we have

$$\begin{aligned} L_C &= \varphi(s, (1, 2)) + \sum_{i=1}^{k-2} \varphi((i, i + 1), (i + 1, i + 2)) + \varphi((k - 1, k), t) \\ &= \beta_0 d(s, 1) + (\alpha_0/2)d(1, 2) + \beta_1 d(s, 2) \\ &\quad + \sum_{i=1}^{k-2} ((\alpha_0/2)(d(i, i + 1) + d(i + 1, i + 2)) + \alpha_1 d(i, i + 2)) \\ &\quad + \beta_0 d(k, t) + (\alpha_0/2)d(k - 1, k) + \beta_1 d(k - 1, t). \end{aligned}$$

So  $E[L_c] = L_C$ .

Finally, given a path in  $(V, E, \varphi)$ , its corresponding path in  $(N, A, d)$  is uniquely obtained by reversing the previous argument. ■

We are now in a position to give our main result:

**Theorem 2.** *Given a node invariant PSPP with  $m = 2$  and with a probability  $\mathbb{P}$  such that  $\Pr(W \leq n - 2) = 0$  and  $\Pr(W = n - 1) > 0$ , one can find an optimal PSPP path between  $s$  and  $t$  in time  $O(n^3)$ .*

*Proof.* From Lemma 1, one can solve such a PSPP by finding the shortest path in the auxiliary network  $(V, E, \varphi)$ . This can be done by a careful implementation of the Dijkstra algorithm in time  $O(|E| + |V| \log |V|)$  (see [3] or [4]). ■

#### 2.4.2. Generalizations

We can extend the idea of the previous subsection in several directions:

We can first consider cases in which several white nodes can fail at the same time. For example, let us look at the case in which  $\Pr(W \leq n - 3) = 0$ . The set of nodes  $V$  will be augmented to include all triplets  $(i, j, k)$  with  $i \neq j \neq k$ , i.e., of  $n(n - 1)(n - 2)$  nodes. The set of arcs  $E$  and the function  $\varphi$  will remain the same except that we add arcs of the types  $(s, (i, j, k))$ ,  $((i, j, k), t)$ , and  $((i, j, k), (j, k, l))$  (with  $l \neq i$ ) and that we delete arcs of the type  $((i, j), (j, k))$ . The length of the added arcs are defined with the same idea as before, e.g.:

$$\begin{aligned} \varphi(s, (i, j, k)) &= (2\alpha_0/3)d(i, j) + (\alpha_0/3)d(j, k) + (\alpha_1/2)d(i, k) \\ &\quad + \beta_0d(s, i) + \beta_1d(s, j) + \beta_2d(s, k) \end{aligned}$$

$$\begin{aligned} \varphi((i, j, k), (j, k, l)) &= (\alpha_0/3)(d(i, j) + d(j, k) + d(k, l)) \\ &\quad + (\alpha_1/2)(d(i, k) + d(j, l)) + \alpha_2d(i, l). \end{aligned}$$

Now a path  $c = (s, 1, \dots, k, t)$  in  $(N, A, d)$  will correspond to the path  $C = (s, (1, 2, 3), (2, 3, 4), \dots, (k - 2, k - 1, k), t)$  in  $(V, E, \varphi)$ . We have

$$|V| = 2 + n + n(n - 1) + n(n - 1)(n - 2) = O(n^3)$$

$$\begin{aligned} |E| &= 1 + 2n + 2n(n - 1) + 2n(n - 1)(n - 2) + n(n - 1)(n - 2)(n - 3) \\ &= O(n^4). \end{aligned}$$

Hence, the PSPP can be solved in  $O(n^4)$ .

Another example of generalization is to consider the case of Subsection 2.4.1 but with more than two black nodes: The set  $V$  would be augmented to include  $m - 2$  additional nodes, and we would have to add arcs of the types  $(i, v)$ ,  $(v, i)$ ,  $((i, j), v)$ , and  $(v, (i, j))$  for all  $v$  in  $N_1 \setminus \{s, t\}$  and  $i, j, i \neq j$  in  $N_2$ , and arcs of the types  $(s, v)$ ,  $(v, t)$ ,  $(v, w)$ , and  $(w, v)$  for all  $v \neq w$  in  $N_1 \setminus \{s, t\}$ . The weight of these addi-

tional edges would be defined as before [e.g.,  $\varphi(v,w) = d(v,w)$ ,  $\varphi(v,(i,j)) = \beta_0 d(v,i) + (\alpha_0/2)d(i,j) + \beta_1 d(v,j)$ ]. The most general result is obtained by considering a combination of the two previous extensions:

**Theorem 3.** *Given a node invariant PSPP with  $m$  black nodes ( $m \geq 2$ ) and a probability  $\mathbb{P}$  such that  $\Pr(W \leq n - k - 1) = 0$  and  $\Pr(W = n - k) > 0$ , one can find an optimal PSPP path between  $s$  and  $t$  in time  $O(mn^{k+1} + n^{k+2} + m^2)$ .*

*Proof.* The theorem results from the fact that

$$\begin{aligned} |V| &= m + n + n(n - 1) + \dots + n(n - 1) \dots (n - k) \\ &= O(m + n^{k+1}) \end{aligned}$$

$$\begin{aligned} |E| &= (2m - 2)(|V| - m) + n(n - 1) \dots (n - k - 1) + (m - 1)(m - 2) + 1 \\ &= O(mn^{k+1} + n^{k+2} + m^2). \quad \blacksquare \end{aligned}$$

### 3. PROBABILISTIC MINIMUM SPANNING TREE PROBLEM

#### 3.1. Introduction

The Probabilistic Minimum Spanning Tree Problem (PMSTP) was first introduced in [2] and its rationale is based on the same motivations as for the PTSP, PSPP, etc. (see Section 1). The problem is a probabilistic generalization of the classical minimum spanning tree problem. Given a set of nodes in a network (not necessarily complete), only a random subset of them is present on any particular instance of the problem. We wish to find a priori a spanning tree through all the nodes, so that, for any subsequent random subset of nodes, the tree is retraced deleting only the nodes that are not present (with their adjacent edges) provided the deletion does not disconnect the tree. The problem of finding an a priori spanning tree of minimum expected length is the PMST problem.

For a description of many applications of the PMSTP in the areas of communication, infrastructure planning, circuit design, routing, and organizational structures, see [2].

After specifying the notation and the probabilistic assumptions to be used throughout Section 3, we first give in Section 3.3 efficient closed-form expressions for the expected length of a given spanning tree. In Section 3.4, we show that the simplest version of the PMSTP is NP-hard.

#### 3.2. Notation and Assumptions

Throughout this section,  $G = (V,E,c)$  is a loopless, undirected, weighted graph where  $V$  is the node set,  $E$  is the set of edges joining the nodes of  $V$ , and  $c$  is a function  $E \mapsto R$ . Otherwise, the notation and assumptions are the same as

in Section 2.2 except that, for simplicity, we do not consider black nodes. With our assumptions, the PMSTP can be formulated as follows:

**Problem PMSTP.** Given a network  $G = (V, E, c)$  and a probability  $\mathbb{P}$ , find an a priori tree  $T$  of minimum expected length,  $E[L_T]$ .

### 3.3. The Expected Length of a Given Tree

One can develop an efficient closed-form expression to compute  $E[L_T]$  for general node invariant probabilities. The result is the following:

**Theorem 4.** *Given a PMSTP and a spanning tree, we have*

$$E[L_T] = \sum_{e \in T} \sum_{k=1}^n c(e) \alpha(e, k) \Pr(W = k), \tag{6}$$

where

$$\alpha(e, k) = 1 - \frac{\binom{|K_e|}{k} + \binom{n - |K_e|}{k}}{\binom{n}{k}}$$

with  $K_e$  being the subset of nodes contained in the smallest of the two subtrees obtained from  $T$  by removing the edge  $e$ , and with the convention that  $\binom{y}{x} = 0$  if  $y < x$ .

*Proof.* By conditioning on the number of present nodes, we have  $E[L_T] = \sum_{k=1}^n E[L_T | W = k] \Pr(W = k)$ . Now  $E[L_T | W = k] = \sum_{e \in T} c(e) \alpha(e, k)$ , where  $\alpha(e, k)$  is the conditional probability that the edge  $e$  remains in the tree. To evaluate this last quantity, note that the edge  $e$  is absent from the tree if and only if the set of present nodes is fully contained either in the set  $K_e$  or in its complement. The probability of such an event is exactly  $1 - \alpha(e, k)$ . ■

#### Remarks.

1. This theorem gives the expected length of a spanning tree in  $O(n^2)$  elementary operations (for a general node invariant  $\mathbb{P}$ ).
2. For the case of a Bernoulli process  $\mathbb{P}_1$ , we have  $\sum_{k=1}^n \alpha(e, k) \Pr(W = k) = (1 - (1 - p)^{|K_e|})(1 - (1 - p)^{n - |K_e|})$ , a result obtained in [2]. The expected length can then be computed in  $O(n)$ .
3. Similar results are obtained in [2] for the case in which each node  $i$  has a probability  $p_i$  of being present, independently of the others (this case cannot be modeled by a node invariant probability  $\mathbb{P}$ ).

### 3.4. The Intrinsic Complexity of the Problem

In [2] it has been shown that the simple PMSTP with  $\mathbb{P} = \mathbb{P}_1$  and with equal weights  $c(e) = 1$  is NP-hard. We show in this section that the simpler case with  $\mathbb{P}$  such that  $\Pr(W < n - 1) = 0$  and with equal weights  $c(e) = 1$  is also NP-hard. Let us define the recognition version of this simple problem.

*SPECIAL PMSTP (SPMSTP)*

Instance: Given a weighted graph  $G = (V, E, c)$ , a probability  $\mathbb{P}$  such that  $\Pr(W = n - 1) > 0$  and  $\Pr(W < n - 1) = 0$ , weights  $c(e) = 1$  for all  $e \in E$ , and a bound  $B$ .

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

**Theorem 5.** *SPMSTP is a NP-complete problem.*

*Proof.* The problem is in NP, since for a given tree we can compute  $E[L_T]$  in polynomial time and compare it with  $B$ . The proof of the completeness is obtained by reducing the NP-complete problem MAXIMUM LEAF SPANNING TREE (see [7]) to it.

*MAXIMUM LEAF SPANNING TREE PROBLEM*

Instance: Graph  $G = (V, E)$ , positive integer  $K \leq |V|$ .

Question: Is there a spanning tree for  $G$  in which  $K$  or more nodes have degree 1?

Given an instance of this problem, we define an instance of the SPMSTP using the same graph  $G$ . Now from Theorem 4,  $E[L_T]$  is, in fact, equal to  $\sum_{e \in T} (\Pr(W = n - 1)\alpha(e, n - 1) + \Pr(W = n)\alpha(e, n))$ . Since  $\alpha(e, n) = 1$  and  $\alpha(e, n - 1) = (n - 1)/n$  or 1 depending if  $|K_e| = 1$  or not, we finally have

$$E[L_T] = \sum_{e \in T: |K_e|=1} (1 - \Pr(W = n - 1)/n) + \sum_{e \in T: |K_e| \neq 1} 1.$$

It is now obvious that we can reduce the problem MAXIMUM LEAF SPANNING TREE to our problem [take  $B = n - 1 - \Pr(W = n - 1)K/n$ ]. ■

## 4. CONCLUSION AND DISCUSSION

In Section 2, we have analyzed the PSPP, a conceptual model for many potential application areas. The analysis, presented here, implies that the problem presents sufficiently different features from its deterministic counterpart to necessitate devising entirely new solution procedures. In Section 3, we have looked at PMSTPs under the assumptions of general node invariant probabilities. We showed that the expected length of a given spanning tree can still be

obtained efficiently via closed form expressions (see Theorem 4). We finally demonstrated that the simplest version of the PMSTP in which only one node can at any time be absent remains NP-hard.

The main contribution of this paper was to delineate the computational complexity of the PSPP and PMSTP and, most importantly, to give polynomial time procedures for special versions of the first problem.

With respect to the first point, we are facing the following peculiar phenomenon. Although the spanning tree problem is “easier” to solve than the shortest path problem in a deterministic network, it becomes “harder” as soon as one node (chosen at random) may or may not be working (the first problem becomes NP-hard and the other one remains easy to solve). Moreover, we showed that, if the number of malfunctioning nodes is, at any time, bounded by a constant, the PSPP can still be solved in polynomial time (in general, however, it is also NP-hard). A possible explanation of this changing status is that the presence or not of a given node has a *local* effect in the case of the PSPP (we go through or skip this node), but a *global* effect in the case of the PMSTP (the tree has to remain connected).

With respect to the second point, one should note that this is the first time, since the beginning of the analysis of probabilistic combinatorial optimization problems, that we have been able to find a large subclass of problems solvable by polynomial time procedures. The techniques developed in Section 2.4 can be applied to special versions of other problems such as the PTSP (transforming a PTSP into a TSP). Finally, it is worthwhile to mention that these results can be used to obtain good heuristics to the PSPP or PTSP when the probability distribution of  $W$  is such that  $\Pr(W \leq n - k) = \varepsilon$  with  $\varepsilon$  very small (e.g., when  $\mathbb{P} = \mathbb{P}_1$  and  $p$  very close to 1). One can also analyze precisely the quality of these heuristics using the general framework given in [8, pp. 163–169].

I would like to thank two anonymous referees for useful comments that helped in improving the presentation. Also, special thanks to one of the referees who brought to my attention the Fibonacci heaps of Fredman and Tarjan (or the relaxed heaps of Driscoll, Gabow, Shrairman, and Tarjan) that allow implementation of Dijkstra’s algorithm in  $O(|E| + |V| \log |V|)$  time, rather than the traditional  $O(|E| \log |V|)$  time. This led to an improvement of the computational bound in Theorem 3.

## REFERENCES

- [1] G. Andreatta, F. Ricaldone, and L. Romeo, Exploring stochastic shortest path problems. *ATTI, Giornate di Lavoro 1985*. Tecnoprint, Bologna, Italy (1985).
- [2] D. Bertsimas, The probabilistic minimum spanning tree problem. *Networks* **20** (1990) 245–275.
- [3] J. Driscoll, H. Gabow, R. Shrairman, and R. Tarjan, Relaxed heaps: An alternative to Fibonacci heaps with applications to parallel computation. *Commun. ACM* **31**(11) (1988) 1343–1354.
- [4] M. Fredman and R. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM* **34**(3) (1987) 596–615.
- [5] A. Frieze, On the value of a random minimum spanning tree problem. *Discrete Appl. Math.* **10** (1985) 47–57.



- [6] A. Frieze and G. Grimmet, The shortest path problem for graphs with random arc lengths. *Discrete Appl. Math.* **10** (1985) 57–77.
- [7] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco (1979).
- [8] P. Jaillet, *The Probabilistic Traveling Salesman Problems*. Technical Report 185, Operations Research Center, MIT, Cambridge, MA (1985).
- [9] P. Jaillet, Stochastic routing problems. *Stochastics in Combinatorial Optimization* (G. Andreatta, F. Mason, and P. Serafini, Eds.). World Scientific, Singapore (1987).
- [10] P. Jaillet, On some probabilistic combinatorial optimization problems defined on graphs. *Flow Control of Congested Network* (A. Odoni, L. Bianco, and G. Szegö, Eds). NATO ASI Series F. 38, Springer-Verlag, Berlin (1987).
- [11] P. Jaillet, *A priori solution of a traveling salesman problem in which a random subset of the customers are visited*. *Operations Res.* **36**(6) (1988) 929–936.
- [12] P. Jaillet and A. Odoni, Probabilistic vehicle routing problems. *Vehicle Routing: Methods and Studies* (B. Golden and A. Assad, Eds.). Studies in Management Science and Systems, Vol. 16. North-Holland, Amsterdam (1988).
- [13] R. Larson and A. Odoni, *Urban Operations Research*. Prentice-Hall, Englewood Cliffs, NJ (1981).
- [14] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ (1982).

Received April 1989

Accepted April 1992