Discrete Optimization 12 (2014) 61–72

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Reliability problems in multiple path-shaped facility location on networks

Justo Puerto[a], Federica Ricca[b], Andrea Scozzari[c,*]

[a] IMUS, Instituto de Matemáticas de la Universidad de Sevilla, Spain

[b] Sapienza, Università di Roma, Italy

[c] Università degli Studi "Niccolò Cusano", Roma, Italy

ARTICLE INFO

Article history:
Received 28 March 2013
Received in revised form 22 January 2014
Accepted 25 January 2014
Available online 16 February 2014

Keywords:
Reliable location
Path-shaped facilities
Multifacility path location
Network location

ABSTRACT

In this paper we study a location problem on networks that combines three important issues: (1) it considers that facilities are extensive, (2) it handles simultaneously the location of more than one facility, and (3) it incorporates reliability aspects related to the fact that facilities may fail. The problem consists of locating two path-shaped facilities minimizing the expected service cost in the long run, assuming that paths may become unavailable and their failure probabilities are known in advance. We discuss several aspects of the computational complexity of problems of locating two or more reliable paths on graphs, showing that multifacility path location – with and without reliability issues – is a difficult problem even for 2 facilities and on very special classes of graphs. In view of this, we focus on trees and provide a polynomial time algorithm that solves the 2 unreliable path location problem on tree networks in $O(n^2)$ time, where n is the number of vertices.

1. Introduction

One of the most important strategic decisions in the design of infrastructures is the location of facilities. This has motivated a lot of research on different facility location problems during many years (see, e.g., [1,2]) and, in particular, on several extensive path- or tree-shaped facility location problems (see [3–7] and the references therein).

There are several sources of uncertainty that must be considered when facing a location problem. Costs, customer demands or production capacities may be unknown at the moment of making a decision. A different uncertainty aspect maybe related to facilities themselves that may disrupt due to unexpected events. This gives rise to situations where some facilities become temporarily unavailable to provide service to customers, due to system failures, natural disasters, terrorist attacks, labor strikes, etc. Modeling issues should handle as best as they can these unknown or unpredictable situations whenever they occur.

Realistically, no decision-maker would accept a solution with very high operating costs just to hedge against very rare facility disruptions, unless high penalties must be paid to costumers in case of uncovered service. Failures typically result in extra transportation costs, as customers originally served by the closest facilities must be redirected to more distant ones (see, e.g., [8]). In order to balance the normal and failure operation costs, the facility location should depend on how likely facilities may get disrupted, as well as, on their closeness to the potential customers. This has motivated an alternative approach to the "customer-to-closest facility cost" criterion that consists of locating facilities that minimize the

* Corresponding author. Tel.: +39 3398146279; fax: +39 0670307312.
E-mail addresses: puerto@us.es (J. Puerto), federica.ricca@uniroma1.it (F. Ricca), andrea.scozzari@unicusano.it (A. Scozzari).

http://dx.doi.org/10.1016/j.disopt.2014.01.003
1572-5286/

total expected service cost in the long run, assuming that failures are accidental, and their probabilities can be estimated in advance [9,10]. Needless to say, this approach is not unique and another way to model disruption is to consider it motivated by intentional attacks [11–13]. Although these models are very interesting, they are beyond the scope of this paper.

The literature in the area of reliable location models can be traced back, at least, to the paper [14], where both a median and a center objective functions are considered, and the situation where a fixed number of facilities might fail is tackled. However, one can observe that the current interest in reliability issues in location problems has been recently restarted with [15] and other works by the same authors. In addition, the following papers [9,16,17] have also contributed decisively to this increasing interest. Robustness analysis of transportation networks has been also widely studied in the literature from alternative points of view (see, e.g., [18–23]).

As can be seen from the literature review, there are models that consider reliability aspects of point location and some other models that apply integer programming tools to design routes, but one does not find reliability models for the simultaneous location of extensive facilities. The goal of this paper is to combine three crucial aspects of location models: (i) the existence of more than one service facility; (ii) the assumption of the extensive nature of service facilities (frequently more realistic than the assumption of point facilities); (iii) the minimization of the total expected service cost in the long run, assuming that failures are accidental and their probabilities are known. In particular, referring to (i) and (ii), we point out that there are few papers in the literature that consider the problem of locating two or more path-shaped facilities on networks. The combination of the three above aspects makes the location problem studied in this paper even more difficult under the computational complexity viewpoint, also for graphs with a very simple structure. We discuss several aspects related to the computational complexity of problems that are strictly related to our unreliable path location problem and, in fact, are special cases of it. We show that multiple facility path location problems are NP-Hard even on very simple classes of graphs, implying that the same holds for our reliability problem. This suggests that there is little hope to solve even the two unreliable path location problem on networks more general than a tree.

In view of the above considerations, we focus on trees and study the following problem: given two path-shaped facilities that may fail with given probabilities, find two paths in the tree where the two facilities can be located in order to minimize the total expected service cost. Here we assume that each customer is first assigned to its closest facility, then, if this fails, to the second closest, and, if both facilities fail, he/she is assigned to a backup facility modeled by a penalty cost. Assuming that the tree has n vertices, the problem can be solved by brute force by evaluating the objective function on each of the $O(n^4)$ different pairs of paths of the tree. Since each evaluation can be done in linear time, this approach would lead to an overall complexity of $O(n^5)$ time. In spite of that, in this paper we present an $O(n^2)$ time complexity algorithm for solving this problem. It is also worth noticing that our complexity result for locating two paths on a tree equals the one obtained by [9] for the corresponding two points location problem.

Although the multiple unreliable path location problem studied in this paper can be considered as a natural extension of the point location version already analyzed in [9], some additional issues arise when handling pairs of paths instead of pairs of points. As we will see, this requires new algorithmic solutions, in particular for an efficient evaluation of pairs of *intersecting* paths.

The paper is organized as follows. In Section 2 we provide some notation and definitions, while basic properties are introduced in Section 3. In Section 4 we discuss the computational complexity of the problem of locating $K \geq 1$ paths on networks also in the case of unreliable paths. Section 5 illustrates the algorithm for solving the two path-shaped facilities location problem with probabilities of failure. In Section 6 we discuss further extensions and draw some conclusions.

2. Notation and definitions

Let $T = (V, E)$ be a tree with $|V| = n$. Suppose that a positive real length $\ell(e) = \ell(u, v)$ is assigned to each edge $e = (u, v) \in E$. Let $A(T)$ denote the continuum set of the points in the edges of T. Each subgraph of T is also viewed as a subset of $A(T)$. The edge lengths induce a distance function on $A(T)$ that associates to each pair of points x and y in $A(T)$ (i.e., vertices or points in the interior of an edge) a distance $d(x, y)$ corresponding to the length of the (unique) path $P(x, y)$ from x to y. Therefore, $A(T)$ is a metric space with respect to such a distance function [24]. In the following, we avoid to specify one or both endpoints of a path when it is not necessary. When T is rooted at a vertex r it is denoted by T_r. We denote by $V(T_r)$ the set of vertices of T_r. For any vertex v, let T_v be the subtree of T_r rooted at vertex v, $S(v)$ the set of the children of v in T_r, and $p(v)$ the father of v in T_r. Clearly, a vertex v is a leaf if and only if $|S(v)| = 0$. A path P is *discrete* if both its endpoints are vertices of T, otherwise it is *continuous*. We denote by $V(P)$ the set of vertices belonging to P, and $d(u, P)$ the distance from a vertex u to a path P, that is, the length of the shortest path from u to a vertex or an endpoint of P.

Given a weighted tree $T = (V, E)$, the 2 *Unreliable Median Paths* (2UMP) problem consists of locating in T two path-shaped facilities, $\mathcal{P}^1$ and $\mathcal{P}^2$, each characterized by a given probability of disruption, say, p_1 and p_2, respectively. For a given location of the two facilities, that we denote by $L = \{L(\mathcal{P}^1), L(\mathcal{P}^2)\}$, a client v is first associated to its closest facility, but, in case of disruption, he/she is re-directed to the other one, and, if this also fails (meaning that no operating facilities are available to serve client v), a fixed positive penalty is applied in order to take into account the cost for the dissatisfaction of client v. For each vertex $v \in V$, let $0 \leq h_v \leq 1$ be a weight representing the fraction of the total population in v, and β_v be the fixed non negative penalty to be paid when v is not served. Since β_v is considered as the cost to serve a client in v from a facility outside the network when all the located facilities fail, we impose $\beta_v \geq \max_{u \in V} d(u, v)$. Problem 2UMP is a natural extension to path-shaped facilities of the problem studied in [9] for locating two point facilities.

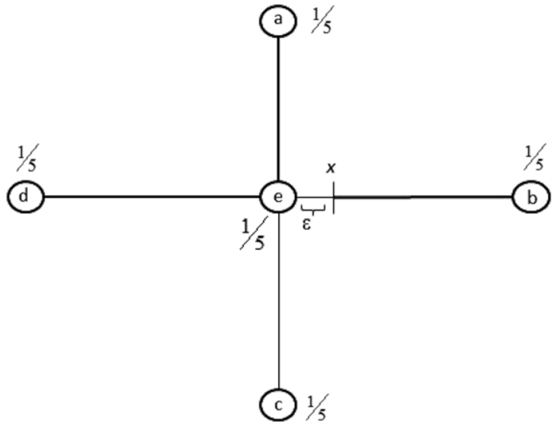

Fig. 1. An example where an optimal solution of 2UMP does not exist when intersection between paths is not allowed.

We denote by L_k^v, with $k = 1, 2$, the location of the k-th closest facility to v, so that, if, for example, for a vertex v, $\mathcal{P}^2$ is the closest facility and $\mathcal{P}^1$ the second closest, one has $L_1^v = L(\mathcal{P}^2)$ and $L_2^v = L(\mathcal{P}^1)$.

To evaluate the general objective function for a given path location L, for each vertex v in V, we compute the total expected weighted cost to serve v with L:

$$Z_v[L] = h_v \left[d(v, L_1^v)\ (1 - p_{L_1^v}) + d(v, L_2^v)\ p_{L_1^v}(1 - p_{L_2^v})\right] + h_v p_{L_1^v} p_{L_2^v}\ \beta_v. \tag{1}$$

Then, the objective function of 2UMP is

$$Z[L] = \sum_{v\in V} h_v \left[d(v, L_1^v)\ (1 - p_{L_1^v}) + d(v, L_2^v)\ p_{L_1^v}(1 - p_{L_2^v})\right] + \sum_{v\in V} h_v p_{L_1^v} p_{L_2^v}\ \beta_v \tag{2}$$

and the problem can be stated as follows: find in T a location L of two path-shaped facilities $\mathcal{P}^1$ and $\mathcal{P}^2$ such that (2) is minimized.

The key-aspect in the above 2UMP problem is that, for each client v, the distance function induces a complete order $\prec_v$ on the set of the two facilities stating which facility is assigned to v first, i.e., $\mathcal{P}^1 \prec_v \mathcal{P}^2$ if and only if $d(v, L(\mathcal{P}^1)) < d(v, L(\mathcal{P}^2))$. When the two facilities are equidistant from v the tie can be broken arbitrarily, but, since for the computation of the objective function we need to distinguish which facility is assigned to v for first, in the rest of the paper we will assume that when $d(v, L(\mathcal{P}^1)) = d(v, L(\mathcal{P}^2))$, the first facility for v is always the one with lower probability of disruption, and, if this still produces a tie, we establish that the first facility for v is $\mathcal{P}^1$.

3. Basic properties and problem structure

As widely discussed in [9], in location of unreliable facilities a main issue is *co-location* which for the point location problem corresponds to the possibility of locating different facilities at the *same point* of the network. Extending this notion to path-shaped facilities means that the two optimal (different) co-located facilities correspond to the *same path*. However, in path location an additional aspect must be discussed related to the possibility that the two optimal paths may partially intersect. Actually, the possibility of intersection between the two located paths leads to some important issues characterizing the structure of the problem. We provide a first remark related to the existence of solutions for 2UMP.

Remark 1. When intersection between paths is not allowed, the optimal solution of 2UMP may not exist.

Consider the tree T shown in Fig. 1 with five vertices, a, b, c, d and e. Suppose that all edge lengths are equal to 1, and $h_a = h_b = h_c = h_d = h_e = \frac{1}{5}$. Consider the two disjoint paths P_{ad} and $P_{x(\varepsilon)b}$ where $x(\varepsilon)$ is a point along edge (e, b) located at distance $\varepsilon > 0$ from vertex e. Consider the location of facility $\mathcal{P}^1$ in P_{ad} and facility $\mathcal{P}^2$ in $P_{x(\varepsilon)b}$, i.e., $P_{ad} = L(\mathcal{P}^1)$ and $P_{x(\varepsilon)b} = L(\mathcal{P}^2)$, with the corresponding objective function value given by:

$$Z^\varepsilon[L] = \frac{1}{5}\ p_1(1 - p_2)\ [3 + 4\varepsilon] + \frac{1}{5}\ (1 - p_1)(1 + p_2) + \frac{1}{5}\ p_1 p_2[\beta_a + \beta_b + \beta_c + \beta_d + \beta_e]. \tag{3}$$

When $\varepsilon \to 0$, $Z^\varepsilon[L]$ decreases but it never reaches its infimum if the intersection between P_{ad} and $P_{x(\varepsilon)b}$ is forbidden. On the other hand, if the intersection is allowed, an optimal solution of the above problem is given by $L^*(\mathcal{P}^1) = P_{ad}$ and $L^*(\mathcal{P}^2) = P_{cb}$ with objective function value

$$Z[L^*] = \frac{2}{5}\ p_1(1 - p_2) + \frac{2}{5}\ p_2(1 - p_1) + \frac{1}{5} p_1 p_2[\beta_a + \beta_b + \beta_c + \beta_d + \beta_e]. \tag{4}$$

This example shows that there are some instances of 2UMP that, when the intersection between paths is not allowed, are not well defined, implying that the optimal solution of 2UMP might not exist. Therefore, we allow intersections and consider the case of pairs of optimal paths that intersect as a third possibility to be added to the cases of disjoint and co-located optimal paths.

The example also shows that the intersection between the two optimal paths does not necessarily imply that they are also co-located. In fact, if in the example we restrict ourselves to co-located paths, the best solution corresponds to any pair of paths $\bar{L}$ connecting two tips of T, and its objective function value is worse than the one of L^*. Indeed,

$$Z[\bar{L}] = \frac{2}{5}\ p_1(1-p_2) + \frac{2}{5}\ (1-p_1) + \frac{1}{5}p_1p_2[\beta_a + \beta_b + \beta_c + \beta_d + \beta_e] > Z[L^*]. \tag{5}$$

The following proposition states other desirable properties of optimal paths for 2UMP.

Proposition 1. *In an optimal solution of* 2UMP *the two optimal paths always connect two leaves of* T.

Proof. We start by proving that in an optimal solution of 2UMP the two optimal paths are always discrete. Let us consider the following situation. Let $L(\mathcal{P}^1) = P(u, v)$ and $L(\mathcal{P}^2) = P(x, d)$ be the paths where the two facilities $\mathcal{P}^1$ and $\mathcal{P}^2$ are located, and assume that u, v and d are vertices of T, while the other endpoint x belongs to the interior of an edge (a, b). W.l.o.g., we assume that vertex $b \in V(P(x, d))$. For $\mathcal{P}^2$, we evaluate the possibility of replacing $P(x, d)$ by the discrete path $P(a, d)$. For a vertex z in T, two cases may arise: (i) the order induced by the distance function on the set of the two facilities for client z does not change when $P(a, d)$ replaces $P(x, d)$; (ii) $P(x, d)$ was the second closest facility for z, but, after the replacement, $P(a, d)$ becomes the first one. The only relevant case is (ii). In fact, in case (i), $P(a, d)$ provides a better solution than $P(x, d)$, since for a client z either no distance changes or the expected cost of z strictly decreases. In particular, we observe that the second situation holds at least for one of the clients. For example, the expected cost of a strictly decreases, since the distance from a to $P(a, d)$ is zero.

In case (ii), one has $d(z, a) < \min_{w\in V(P(x,d))} d(z, w)$ and $d(z, P(x, d)) = d(z, a) + d(a, x)$. For client z the following holds:

$$d(z, P(u, v)) \le d(z, P(x, d)) \quad \text{and} \quad d(z, P(u, v)) > d(z, a) = d(z, P(a, d)).$$

The cost to serve z when $L_1^z = P(u, v)$ and $L_2^z = P(x, d)$ is:

$$h_z d(z, P(u, v))(1-p_1) + h_z d(z, P(x, d))p_1(1-p_2) + h_z p_1 p_2 \beta_z \tag{6}$$

while, when $L_1^z = P(a, d)$ and $L_2^z = P(u, v)$, it is:

$$h_z d(z, P(a, d))(1-p_2) + h_z d(z, P(u, v))p_2(1-p_1) + h_z p_1 p_2 \beta_z. \tag{7}$$

The difference between (6) and (7) is

$$h_z[d(z, P(u, v)) - d(z, a)](1-p_1)(1-p_2) + h_z d(a, x)p_1(1-p_2) > 0,$$

showing that the objective function (2) strictly decreases when facility $\mathcal{P}^2$ is located in the discrete path $P(a, d)$ instead of in $P(x, d)$. We observe that, when more than one endpoint of the two paths $P(u, v)$ and $P(x, d)$ are not vertices of T, we can apply the same above arguments independently to each of them to show that one can always enlarge the two paths and find two discrete paths for the location of $\mathcal{P}^1$ and $\mathcal{P}^2$ with a lower total expected cost. The above discussion implies that, whenever in a solution of 2UMP we have a path $P(x, d)$ for which at least one endpoint x is in the interior of an edge (a, b), with $b \in V(P(x, d))$, we can always improve on the expected cost of the solution by enlarging $P(x, d)$ up to vertex a. It follows that, optimal paths for 2UMP always connect two leaves of T, and this completes the proof. □

4. Some complexity results on locating multiple unreliable facilities

In this section we discuss the computational complexity of the problem of locating $K \ge 1$ path-shaped facilities characterized by a probability of failure, on general networks and on special classes of graphs. We also analyze the closely related problem of locating one or more paths with the additional constraint that they must be vertex disjoint. As we will see, this will provide insights on the computational complexity of the problem of locating $K \ge 2$ vertex disjoint paths minimizing the sum of the distances (with and without reliability issues).

We start by observing that on general graphs 2UMP is NP-Hard since it contains as a special case the (classical) median path location problem (i.e., the problem of locating one path on a graph minimizing the sum of the distances [25]). In fact, when $p_1 = 0$ and $p_2 = 1$ or vice versa, 2UMP corresponds to the location of the unique operating facility (a single path). This negative result can be further specialized. In fact, the proof of the NP-Completeness of finding a median path on a network is based on a reduction from the Hamiltonian path problem. Since the latter problem is NP-Hard even on the very simple class of planar cubic 3-connected graphs [26,27], it can be shown that the median path problem is NP-Hard on the same class, and therefore this holds for 2UMP, too. At present, this is the strongest complexity result related to our reliable path location problem.

Let us now consider the problem of locating $K \ge 2$ *vertex disjoint* paths minimizing the sum of the distances. We observe that, when $K = 2$, this problem is again a special case of 2UMP. Actually, assume $p_1 = 0$ and $p_2 = 0$; then, for a given

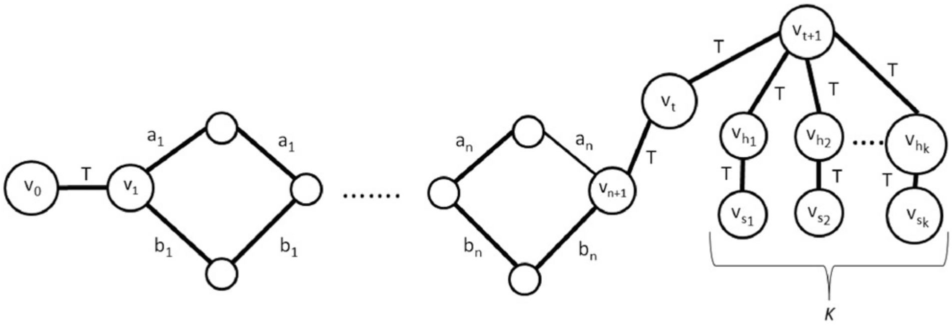

Fig. 2. The cactus graphs used in the NP-Completeness proof of Problem 1.

path location L, the total expected weighted cost function (1) simply returns the cost of assigning a vertex $v \in V$ to its closest facility between $\mathcal{P}^1$ and $\mathcal{P}^2$, and thus, the total objective function (2) computes exactly the sum of the distances of all the vertices $v \in V$ to their closest path. In view of this close relation between 2UMP and the problem of locating two or more vertex disjoint median path-shaped facilities on networks, it is worth studying the complexity of the latter problem on general networks and trying to specialize and update the current computational complexity results. In [28] it is shown that the problem of finding $K \geq 2$ vertex disjoints median paths, for fixed K, is NP-Hard in arbitrary graphs by a reduction from the Covering by K Paths problem, according to which, given a graph G, one must decide whether there exists a set of K vertex disjoint paths covering all the vertices of G (i.e., each vertex belongs to exactly one path in the set). This result can be strengthened by taking into account that in [28] Covering by K Paths is proved to be NP-Hard on general graphs by a transformation from the Hamiltonian path problem. Since the latter problem is NP-Hard on planar, cubic, 3-connected graphs, referring to the reduction proposed in [28, see, p. 550, Fig. 5], we note that Covering by K paths remains NP-Hard on planar graphs with vertex degree at most 4 for $K = 2$, and at most $K + 1$ for $K \geq 3$. This implies that also the problem of locating two vertex disjoint median paths remains NP-Hard on planar graphs with vertex degree at most 4. In the literature, the $K = 2$ vertex disjoint median path location problem was only solved on tree networks via a $O(n)$ time dynamic programming algorithm [29,30]. In [30] the authors also provide an algorithm for locating $K > 2$ paths in $O(n^{K-1})$ time.

All the results reported above do not consider the case when there is also a bound on the total length (or cost) of the located paths. When $K \geq 2$, the total length is given by the sum of all the lengths of the edges of the K located paths. In the following we provide some new results on the complexity of solving the problem of locating $K \geq 1$ median paths with bounded length on networks also under the unreliability condition.

First, let us consider the case $K = 1$. It is known that the problem of locating a median path of length at most ℓ is NP-Complete on the simple classes of cactus and grid graphs [31–33]. We observe that this problem remains NP-Complete on cactus even if we remove the constraint on the length, but require the paths to have sum of the distances exactly equal to a given value (see, [34]).

When $K \geq 2$ the problem becomes much more difficult to solve. Actually, if the number K of (vertex disjoint) paths to be located is an input variable, the problem is NP-Hard even on tree networks [28]. If K is fixed, the problem is polynomially solvable on trees although, to the best of our knowledge, no specialized algorithm for this case was provided in the literature yet. In the following, we present a new complexity result that could help both in the complexity analysis of the problem of locating a fixed number $K \geq 2$ of disjoint median paths on graphs, and of the 2 unreliable median paths problem under study. Let us consider the following problem:

Problem 1. Let $G = (V, E)$ be a cactus graph with non negative weights for the vertices and positive lengths (costs) for the edges. Consider $\ell > 0$, a fixed positive value D, and a fixed integer $K \geq 2$. Is it possible to find in G a set of K vertex disjoint paths having sum of the distances at most D and such that the sum of their edge lengths is less than or equal to ℓ?

Theorem 1. *Problem 1 is NP-Complete.*

Proof. The proof is by reduction from the following Partition with disjoint pairs problem that was shown to be NP-complete in [33]. Given two vectors of integers, $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$, such that

$$\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i = T.$$

Partition with disjoint pairs asks for finding a subset of indices $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} a_i + \sum_{i \notin S} b_i = \frac{T}{2}$.

Consider the graph in Fig. 2 where the length of an edge is equal to a_i, or b_i, $i = 1, \ldots, n$ or T.

Assume that the weights on the vertices are all equal to 1, and set $D = \frac{T}{2}$ and $\ell = T(K + 5)$. From a solution of Partition with disjoint pairs problem we can construct a solution of Problem 1 by finding one path starting from vertex v_0 passing through vertex v_{t+1} and ending at one of the vertices v_{s_r}, $r = 1, 2, \dots, K$. Let v_{s_j} be the ending vertex of this path, with $j \in \{1, 2, \dots, K\}$. The path must also pass through the edges with weights b_i if $i \in S$ and a_i if $i \notin S$. It is easy to verify that this path has sum of the distances exactly equal to D and length equal to $6T$. The other $K-1$ vertex disjoint paths correspond to the paths (v_{h_r}, v_{s_r}), with $r \in \{1, 2, \dots, K\}$, $r \neq j$, with total length equal to $T(K-1)$. Hence, the total cost ℓ of the resulting K vertex disjoint paths is exactly equal to $T(K+5)$ with sum of the distances exactly equal to D.

Conversely, consider a solution of Problem 1 that must be formed by K vertex disjoint paths. At least $K-1$ paths are needed to cover all vertices $v_{s_j}, j = 1, \dots, K$. If $K-1$ paths are used, then they cannot cover vertex v_t and, therefore, the K-th path must include both vertices v_0 and v_t. If K paths are used to cover vertices $v_{s_j}, j = 1, \dots, K$, then one of them must include v_0, too, and therefore also v_t. In both cases, we refer to the K-th path including v_0 as P_1, and we observe that P_1 necessarily includes also vertex v_{n+1}. Let S be defined as the subset of indices i in $\{1, \dots, n\}$ for which both edges with length b_i are in P_1. Then P_1 must pass through the edges with weights b_i if $i \in S$ and a_i if $i \notin S$, since otherwise the sum of the distances from it alone would be greater than D. Let ℓ_0 denote the sum of the lengths of the edges of the K paths, and D_0 denote the sum of the distances to these paths. The following equation derives from the computation of the sum of the lengths of all the edges in the graph:

$$\ell_0 + 2D_0 + T(K-1) = T(2K+5).$$

In fact, the left-hand side adds the total length of the edges in the K paths to the total length of those edges outside such paths; on the other hand, the right-hand side of the previous equation sums all the edge lengths of the graph. The above equation can be reduced to:

$$\ell_0 + 2D_0 = T(K+6)$$

which can hold only if $D_0 = D = \frac{T}{2}$, and $\ell_0 = \ell = T(K+5)$. Hence, the set S provides a solution for Partition with disjoint pairs problem. Problem 1 is clearly in NP so that it is NP-Complete and this completes the proof. □

Remark 2. In the case of $K = 2$, the problem remains NP-Complete if we do not consider any constraint on the total length of the 2 paths, but require that the sum of the distances must be exactly equal to D.

The NP-Completeness result provided in Theorem 1 directly applies to the 2UMP problem with length constraint (when $K = 2$ and considering $p_1 = p_2 = 0$), implying that 2UMP is NP-Complete on the same class of cactus graphs, as well.

All the complexity results provided in this section lead to the conclusion that locating more than one path is a difficult problem even on graphs with a very simple structure. The additional unreliability aspect does not help for this task. In view of this, we focus on trees and, in the following section, we provide an $O(n^2)$ algorithm for solving 2UMP on a tree without length constraint. The case when there is a bound on the length of the path is still an open problem that will be a subject of our future work.

5. Solution algorithm for the discrete 2UMP

In [9] the problem of locating two facilities in a tree under reliability issues is already addressed for facilities corresponding to points. The authors show that a "nodal optimality" property holds if co-location is allowed, that, in their case, corresponds to locating the two facilities at the same point. The solution algorithm proposed is then based on searching separately for the best co-located solution and the best disjoint one. Here we follow a similar approach for 2UMP, taking into account that, in our case, the optimal pair of paths may be co-located, or partially intersecting, or vertex-disjoint. We describe an *ad hoc* procedure for finding the best pair of intersecting paths in Section 5.1, while, in Section 5.2 we show that an approach similar to the one in [9] can be adapted to deal with the location of two vertex-disjoint paths.

5.1. Search for intersecting paths for 2UMP

W.l.o.g., we assume that $p_1 \le p_2$, so that the two facilities $\mathcal{P}^1$ and $\mathcal{P}^2$ are univocally identified by their probabilities of disruption p_1 and p_2, respectively, and, as stated before, if a tie occurs for a client v, he/she is assigned to $\mathcal{P}^1$ first.

When the two optimal paths intersect, we distinguish two cases: (i) they intersect in at least one edge (i, j); (ii) they intersect in exactly one vertex r. We analyze these two cases separately, since they rely on two different solution strategies.

Case (i). Consider an edge (r_1, r_2) and the two subtrees T_{r_1} and T_{r_2} rooted at vertices r_1 and r_2, respectively (see Fig. 3(a)). For each edge (r_1, r_2) we have to find the pair of paths intersecting at least in (r_1, r_2) that minimizes the total expected cost (2). We call these paths *best paths* w.r.t. (r_1, r_2), and denote by $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ the best paths for the location of the facilities with probability of disruption p_1 and p_2, respectively. The idea is to construct such paths starting from r_1 (resp. r_2) and ending in some leaves of T_{r_1} (resp. T_{r_2}) to find the two branches of $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ in T_{r_1} (resp. T_{r_2}). This can be done by independently visiting top-down the two subtrees T_{r_1} and T_{r_2}. We focus our analysis only on T_{r_1}, the one for T_{r_2} being the same. To facilitate the computations, w.l.o.g., we assume that T_{r_1} is binary, since, otherwise, we can transform it into a binary tree by applying the linear time procedure provided in [24,35]. Notice that, going down in T_{r_1}, the branches of the two paths may share additional vertices and edges up to some vertex v which may be even a leaf.

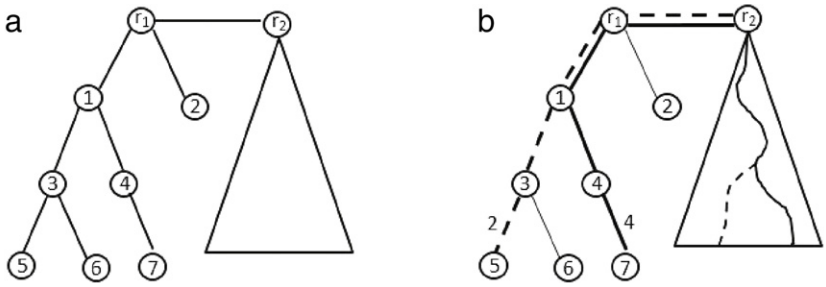

Fig. 3. (a) An example of the two subtrees T_{r_1} and T_{r_2} of a given tree T. (b) The two branches of $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ in T_{r_1} assuming $h_v = 1, \forall v \in V$. Edges (3, 5) and (4, 7) have length equal to 2 and 4, respectively. All the other edge lengths are equal to 1. The bold line refers to the path with probability of disruption p_1, the dashed line refers to the path with probability of disruption p_2 (see Appendix A for the computation of the recursive formulas).

For a given node $v \in V(T_{r_1})$ we denote by v_1 and v_2 its left and right children, respectively. W.l.o.g., when a node v has only one child, we always consider it as the left child of v.

For a given root r_1, we denote by $P^1_{r_1}$ and $P^2_{r_1}$ the two branches of $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ in T_{r_1}. In order to find $P^1_{r_1}$ and $P^2_{r_1}$, along with their contribution to the objective function value that we denote by $Z[L^{r_1}]$, with $L^{r_1} = \{P^1_{r_1}, P^2_{r_1}\}$, the procedure relies on the computation of *saving functions* which, at each vertex v, provide the gain in the objective function (2) that can be obtained by extending the paths in T_v.

Consider the binary rooted tree T_{r_1}. In the top-down visit of T_{r_1} three different situations may arise at a given vertex v:

1. The two paths followed the same track up to vertex v, but they separate after v. Two cases are possible depending on which path passes through the left and the right child of v, v_1 and v_2, respectively.
2. The two paths followed the same track up to vertex v and, after v, they proceed together towards either v_1, or v_2. Also here we have two cases, depending on towards which child of v the two paths proceed.
3. The two paths followed the same track up to some ancestor of v, but just one of them passes through v into T_v.

To cope with the three above cases, we associate to each vertex v of T_{r_1} six quantities labeled as follows:

1. $S^{p_1p_2}_{\wedge}(v)$ is the maximum saving in the objective function when the path with probability of disruption p_1 passes through the left child of v, i.e., vertex v_1, ending in a leaf of T_{v_1}, while the path with probability of disruption p_2 passes through the right child of v, i.e., vertex v_2, and ends in a leaf of T_{v_2}. A similar quantity $S^{p_2p_1}_{\wedge}(v)$ is defined for the opposite case.
2. $S^{v_1}_{\shortparallel}(v)$ is the maximum saving in the objective function when the two paths proceed together towards v_1. A similar quantity $S^{v_2}_{\shortparallel}(v)$ is defined when the two paths proceed together towards v_2.
3. $BS^{p_1}(v)$ is the maximum saving in the objective function when only the path with probability p_1 extends from v up to a leaf of T_v. The same quantity $BS^{p_2}(v)$ is defined for the path with probability p_2.

The above quantities can be computed recursively during a bottom-up visit of T_{r_1}. Let $H_v = \sum_{u\in V(T_v)} h_u$ be the sum of the weights of vertices in T_v for which the bottom-up computation is well-known and straightforward [36,37].

The quantities $BS^{p_i}(v)$, $i = 1, 2$ can be computed applying the following recursive formulas:

$$BS^{p_i}(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ \max\{BS^{p_i}(v_1) + H_{v_1}d(v_1, v)(1 - p_i); BS^{p_i}(v_2) + H_{v_2}d(v_2, v)(1 - p_i)\} & \text{otherwise.} \end{cases} \tag{8}$$

The quantities $S^{p_ip_j}_{\wedge}(v)$, $i, j = 1, 2$, $i \neq j$ are computed as follows:

$$S^{p_ip_j}_{\wedge}(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ H_{v_1}d(v_1, v)(1 - p_i) + BS^{p_i}(v_1) + H_{v_2}d(v_2, v)(1 - p_j) + BS^{p_j}(v_2) & \text{otherwise.} \end{cases} \tag{9}$$

We note that when a vertex v has only one child the above quantities cannot be computed and we set $S^{p_ip_j}_{\wedge}(v) = 0$.

The quantities $S^{v_i}_{\shortparallel}(v)$, $i = 1, 2$, are determined as follows:

$$S^{v_i}_{\shortparallel}(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ H_{v_i}d(v_i, v)[(1 - p_1) + p_1(1 - p_2)] + MS(v_i) & \text{otherwise} \end{cases} \tag{10}$$

where

$$MS(v_i) = \max\{S^{p_1p_2}_{\wedge}(v_i),\ S^{p_2p_1}_{\wedge}(v_i),\ S^{v^i_1}_{\shortparallel}(v_i),\ S^{v^i_2}_{\shortparallel}(v_i)\}, \tag{11}$$

v^i_1 and v^i_2 being the two children of v_i. When a vertex v is a leaf we set $S^{v_1}_{\shortparallel}(v) = S^{v_2}_{\shortparallel}(v) = 0$. Furthermore, for $v = r_1$ we do not compute $MS(v)$ since this quantity is never used.

The objective function value associated to the pair of best paths $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ is then given by:

$$Z[L^{(r_1,r_2)}] = Z[L^{r_1}] + Z[L^{r_2}], \tag{12}$$

where, for $i = 1, 2$, one has:

$$Z[L^{r_i}] = \sum_{u \in V(T(r_i))} h_u d(u, r_i)[(1 - p_1) + p_1(1 - p_2)] - \max\{S_{\wedge}^{p_1 p_2}(r_i), S_{\wedge}^{p_2 p_1}(r_i), S_{\shortparallel}^{v_1^{r_i}}(r_i), S_{\shortparallel}^{v_2^{r_i}}(r_i)\}, \tag{13}$$

where $v_1^{r_i}$ and $v_2^{r_i}$ are the children of r_i.

Among all possible pairs of best paths $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$ that can be obtained by considering all the different edges (r_1, r_2) of T, the best location is given by the pair with the minimum expected service cost (12).

If one wants to provide the structure of the two paths, it suffices to trace back the sequence of choices made in the bottom-up computation of the above recursive formulas (see Appendix A for an example).

Case (ii). First of all we observe that two paths may intersect in exactly one vertex r if and only if the number of edges incident to it is greater than or equal to 4, i.e., the degree of r is at least 4. For all such vertices r, consider the tree rooted at r, say T_r. Let $T_{r_1}, \ldots, T_{r_m}$, with $m \geq 4$, be the subtrees rooted at each child of r. Notice that in each subtree T_{r_k}, $k = 1, \ldots, m$, we can locate only one branch of either facilities. Thus, in each subtree $T_{r_k} \cup (r_k, r)$, $k = 1, \ldots, m$, we can compute independently for the two facilities the branch that minimizes the expected cost for the vertices in T_{r_k} w.r.t. that facility.

For each T_{r_k}, $k = 1, \ldots, m$, we assume, w.l.o.g., that it is binary and find the above branches by independently computing bottom-up from the leaves of T_{r_k} to the root r_k the recursive quantities $BS^{p_i}(v)$, $i = 1, 2$. After the evaluation of all $T_{r_k} \cup (r_k, r)$, $k = 1, \ldots, m$, for the root r we have m different pairs of quantities that we denote by $BS^{p_i}_{r_k}(r)$, $i = 1, 2$, $k = 1, \ldots, m$.

Then, in order to avoid branches of the two facilities intersecting in an edge (r_k, r) for some k, we need to evaluate the 4 largest values for $BS^{p_1}_{r_k}(r)$ and the 4 largest for $BS^{p_2}_{r_k}(r)$, $k = 1, \ldots, m$, and select the two paths intersecting only in vertex r by choosing, among the above 8 branches, the 4 that pass through 4 different children of r and provide the maximum total saving $Sav(r)$. The objective function value associated to this solution is given by

$$Z[L^r] = \sum_{u \in V(T_r)} h_u d(u, r)[(1 - p_1) + p_1(1 - p_2)] - Sav(r).$$

5.2. *Search of disjoint paths for 2UMP*

In this section we discuss the case of a solution given by two vertex-disjoint paths (for short, *disjoint 2UMP*). As in [9] the approach developed in this section relies on the following basic observation. Suppose that the optimal solution of 2UMP on T corresponds to two vertex-disjoint paths $\bar{L} = \{\bar{L}(\mathcal{P}^1), \bar{L}(\mathcal{P}^2)\}$. Then it is always possible to find in T an edge (i, j) such that when it is removed from T the following holds for the two generated subtrees T_{ij} and T_{ji} containing vertex i and vertex j, respectively:

- $\bar{L}(\mathcal{P}^1)$ is in T_{ij} and $\bar{L}(\mathcal{P}^2)$ is in T_{ji} (or vice versa);
- all the vertices in T_{ij} are closer to $\bar{L}(\mathcal{P}^1)$ than to $\bar{L}(\mathcal{P}^2)$ and all the vertices in T_{ji} are closer to $\bar{L}(\mathcal{P}^2)$ than to $\bar{L}(\mathcal{P}^1)$ (or vice versa).

The idea of the procedure is to find the optimal pair $\bar{L} = \{\bar{L}(\mathcal{P}^1), \bar{L}(\mathcal{P}^2)\}$ by searching for the two paths separately in the two subtrees T_{ij} and T_{ji}. In order to do this, we follow here an approach similar to the one presented in [9] for the point location problem and extend the basic results to our 2UMP.

Discarding the constraint that in 2UMP a client must be served first by its closest facility and secondly by the other, one can state a variant of the 2UMP consisting of searching for an edge (i, j) in T and two vertex-disjoint paths, one located in T_{ij} and the other in T_{ji}, such that the following objective function is minimized:

$$\begin{aligned} F[\{L(\mathcal{P}^1), L(\mathcal{P}^2)\}|(i,j)] = &\sum_{v \in V(T_{ij})} h_v\ d(v, L(\mathcal{P}^1))\ (1 - p_{\mathcal{P}^1}) \\ &+ \sum_{v \in V(T_{ij})} h_v\ d(v, L(\mathcal{P}^2))\ (1 - p_{\mathcal{P}^2}) p_{\mathcal{P}^1} + \sum_{v \in V(T_{ji})} h_v\ d(v, L(\mathcal{P}^2))\ (1 - p_{\mathcal{P}^2}) \\ &+ \sum_{v \in V(T_{ji})} h_v\ d(v, L(\mathcal{P}^1))\ (1 - p_{\mathcal{P}^1}) p_{\mathcal{P}^2} + \sum_{v \in V} h_v p_{\mathcal{P}^1} p_{\mathcal{P}^2}\ \beta_v. \end{aligned} \tag{14}$$

The above expression corresponds to force a client in vertex v to be served first by the facility located in the same subtree where v lies when edge (i, j) is removed. Thus, the optimization problem is: $\min_{(i,j) \in E} F[(i, j)]$, where

$$F[(i,j)] := \min_{\substack{L(\mathcal{P}^1) \in T_{ij} \\ L(\mathcal{P}^2) \in T_{ji}}} F[\{L(\mathcal{P}^1), L(\mathcal{P}^2)\}|(i,j)]. \tag{15}$$

Problem (15) is different from 2UMP because here the customer's first facility is not necessarily the closest to him/her. Actually, for a given edge (i, j), it corresponds to 2UMP with the additional restriction imposing that for each customer $v \in T_{ij}$ the first facility is the one located in T_{ij}, and for a customer vertex u in T_{ji} the first facility is the one located in T_{ji}. We refer to this problem as *restricted* 2UMP w.r.t. (i, j). However, one can show that in the optimal solution of problem (15) with objective function $F[(i, j)]$ the located paths coincide with the optimal paths for 2UMP, provided that the optimal solution of 2UMP is given by two vertex-disjoint paths belonging to the two subtrees induced by the removal of edge (i, j). This result follows by a direct generalization of Theorem 5 in [9] presented below (we do not report the proof since it is basically the same as the one of Theorem 5 in [9]).

Theorem 2. *Let $\bar{L} = \{\bar{L}(\mathcal{P}^1), \bar{L}(\mathcal{P}^2)\}$ be a pair of vertex-disjoint paths corresponding to an optimal solution to* 2UMP. *Let (i^*, j^*) be the edge that minimizes $F[(i, j)]$ and $L^* = \{L^*(\mathcal{P}^1), L^*(\mathcal{P}^2)\}$ be the two paths minimizing $F[\{L(\mathcal{P}^1), L(\mathcal{P}^2)\}|(i^*, j^*)]$.*

Then one has $Z[\bar{L}] = F[\{L^*(\mathcal{P}^1), L^*(\mathcal{P}^2)\}|(i^*, j^*)] = Z[L^*]$.

The idea underlying the proof of the above theorem is that any optimal solution $\bar{L} = \{\bar{L}(\mathcal{P}^1), \bar{L}(\mathcal{P}^2)\}$ for the disjoint 2UMP in a tree T can be univocally associated to an edge $(\bar{i}, \bar{j})$ of T, and, therefore, to a specific ordering of the two facilities w.r.t. each customer vertex. Relying on this, it can be shown that solving disjoint 2UMP is equivalent to problem (15).

Notice that, for any fixed edge (i, j), the above ordering is forced in the optimal solution of the corresponding *restricted* 2UMP w.r.t. (i, j). Then, for a customer vertex v in T_{ij} his/her first facility will be necessarily the one located on path $\mathcal{P}^1$ in T_{ij}, and the distance from v to $\mathcal{P}^2$ in T_{ji} will depend only on $d(v, j)$. Similarly, for a customer vertex u in T_{ji} w.r.t. $\mathcal{P}^2$ and $\mathcal{P}^1$, respectively. This implies that for a fixed (i, j), $\mathcal{P}^1$ and $\mathcal{P}^2$ can be located independently as median paths in T_{ij} and T_{ji}, respectively.

After Theorem 2, we can find the optimal disjoint 2UMP by repeatedly removing an edge of the tree T and locating the two optimal paths w.r.t. (15). As in [9], this can be done provided that the vertex weights of each subtree are suitably adjusted (see Appendix B).

To conclude this section, we summarize the whole procedure to solve 2UMP which we call ALGORITHM 2UMP.

ALGORITHM 2UMP

1. For each edge (r_1, r_2) of T
   1.1 transform the two subtrees T_{r_i}, $i = 1, 2$, into binary trees rooted at r_i
   1.2 compute $Z[L^{(r_1,r_2)}]$ and find the corresponding pair of paths $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$.
2. For each vertex r with degree at least 4
   2.1 root T at r
   2.2 transform into binary trees the subtrees T_{r_k}, $k = 1, \dots, m$, rooted at the m children of r
   2.3 compute $Z[L^r]$ and find the corresponding pair of paths intersecting only in r.
3. Among all the pair of paths obtained above, choose the best pair w.r.t. the total expected cost (2), say (P'_1, P'_2).
4. For each edge (i, j) of T
   4.1 compute the adjusted weights of the vertices in T_{ij} and in T_{ji}
   4.2 find the median paths in T_{ij} and in T_{ji}, and denote them by (P_{ij}, P_{ji}).
5. Among the pairs (P_{ij}, P_{ji}), for all edges (i, j), choose the best pair w.r.t. (2), and denote it by $(\bar{P}_{ij}, \bar{P}_{ji})$.
6. Choose the best solution between (P'_1, P'_2) and $(\bar{P}_{ij}, \bar{P}_{ji})$.

Theorem 3. *For a given weighted tree T with n vertices, Problem* 2UMP *can be solved in $O(n^2)$ time.*

Proof. Transforming a tree T_{r_i} into a binary tree requires $O(|V(T_{r_i})|)$ time by applying the procedure in [24,35]. Thus, both Steps 1.1 and 2.2 can be performed in $O(n)$ time. The quantities (8)–(10), can be computed in $O(n)$ time by visiting a rooted tree bottom-up, so that finding the best pair (P'_1, P'_2) along with its objective function value takes $O(n^2)$ time. For solving disjoint 2UMP, after the removal of an edge (i, j), the adjusted vertex weights and the median paths in T_{ij} and T_{ji} can be found in linear time. Thus, also finding the optimal pair of vertex disjoint paths $(\bar{P}_{ij}, \bar{P}_{ji})$ requires $O(n^2)$ time.

6. Conclusions

We studied the problem of locating two path-shaped facilities that minimize the total expected service cost in a model where it is assumed that the paths may become unavailable with disruption probabilities that are known in advance. We showed that the problem is NP-Hard on very simple classes of planar graphs, while we provided a $O(n^2)$ time complexity algorithm for solving it on tree networks. Generalizing to a greater number of facilities, we point out that, under the additional constraint that the paths to be located must be vertex-disjoint, an analysis similar to the one followed in this paper to solve 2UMP can be applied also to solve 3UMP in $O(n^3)$ time. More generally, the K unreliable median vertex-disjoint paths problem with $K > 2$ can be still solved with the same approach, provided that K is considered as a fixed parameter.

Table 1
Computation of quantities (8) and (9) for the two branches of the optimal paths in T_{r_1} for the example shown in Fig. 3.

v	$BS^{p_i}(v), i = 1, 2$	$S_{\wedge}^{p_i p_j}(v), i, j = 1, 2\ i \neq j$
r_1	$\max\{12(1-p_i); (1-p_i)\}$	$12(1-p_i) + (1-p_j)$
1	$\max\{5(1-p_i); 6(1-p_i)\}$	$5(1-p_i) + 6(1-p_j)$
2	0	0
3	$\max\{2(1-p_i); (1-p_i)\}$	$2(1-p_i) + (1-p_j)$
4	$4(1-p_i)$	0
5	0	0
6	0	0
7	0	0

Table 2
Computation of quantities (10) for the two branches of the optimal paths in T_{r_1} for the example shown in Fig. 3.

v	$S_{\shortparallel}^{v_1}(v)$	$S_{\shortparallel}^{v_2}(v)$
r_1	$6[(1-p_1) + p_1(1-p_2)] + MS(1)$	$[(1-p_1) + p_1(1-p_2)]$
1	$3[(1-p_1) + p_1(1-p_2)] + MS(3)$	$2[(1-p_1) + p_1(1-p_2)]$
2	0	0
3	$2[(1-p_1) + p_1(1-p_2)]$	$[(1-p_1) + p_1(1-p_2)]$
4	$4[(1-p_1) + p_1(1-p_2)]$	$4[(1-p_1) + p_1(1-p_2)]$
5	0	0
6	0	0
7	0	0

Table 3
Computation of quantities (11) for the example shown in Fig. 3.

v	$MS(v)$
r_1	–
1	$\max\{5(1-p_1)+6(1-p_2); 5(1-p_2)+6(1-p_1); S_{\shortparallel}^{v_1}(1); S_{\shortparallel}^{v_2}(1)\}$
2	0
3	$\max\{(1-p_1)+2(1-p_2); (1-p_2)+2(1-p_1); 0; 0\}$
4	0
5	0
6	0
7	0

Acknowledgments

The present paper was developed during a research visit of the first author to the department of Statistical Sciences of Sapienza, University of Rome. The first and second authors wish to thank the Spanish Ministry of Science and Technology through grant number MTM2007-67433-C02-01; the first author is also grateful to the Junta de Andalucía for partially supporting this research through grant number FQM-5849.

Appendix A

We refer to the tree in Fig. 3 to show the computation of the branches in T_{r_1} of the two paths $P^1_{(r_1,r_2)}$ and $P^2_{(r_1,r_2)}$. Tables 1–3 reports the computation of the recursive quantities (8)–(10) w.r.t. generic values for p_1 and p_2. To complete the example, we assume that $p_1 = 0.5$ and $p_2 = 0.6$, so that the two branches in T_{r_1} correspond to those shown in Fig. 3(b). In particular, we have $Z[L^{r_1}] = 13.3 - \max\{6.4; 5.3; 9.2; 0.7\} = 4.1$.

Appendix B

Let us consider edge (i, j) and the two subtrees T_{ij} and T_{ji}. Assuming that the facility $\mathcal{P}^1$ is located in T_{ij} and $\mathcal{P}^2$ in T_{ji}, the objective function (14) is

$$\sum_{v\in V(T_{ij})} h_v\ d(v, L(\mathcal{P}^1))\ (1-p_{\mathcal{P}^1}) + \sum_{v\in V(T_{ij})} h_v\ d(v, L(\mathcal{P}^2))\ (1-p_{\mathcal{P}^2})p_{\mathcal{P}^1}$$
$$+ \sum_{v\in V(T_{ji})} h_v\ d(v, L(\mathcal{P}^2))\ (1-p_{\mathcal{P}^2}) + \sum_{v\in V(T_{ji})} h_v\ d(v, L(\mathcal{P}^1))\ (1-p_{\mathcal{P}^1})p_{\mathcal{P}^2} + \sum_{v\in V} h_v p_{\mathcal{P}^1} p_{\mathcal{P}^2}\ \beta_v. \qquad (16)$$

The distance $d(v, L(\mathcal{P}^2))$ of a vertex $v \in V(T_{ij})$ can be written as

$$d(v, L(\mathcal{P}^2)) = d(v, i) + d(i, j) + d(j, L(\mathcal{P}^2)).$$

Then, the second sum in formula (16) can be re-written as

$$\sum_{v \in V(T_{ij})} h_v \, [d(v, i) + d(i, j) + d(j, L(\mathcal{P}^2))] \, (1 - p_{\mathcal{P}^2}) p_{\mathcal{P}^1},$$

where it is clear that only the distance $d(j, L(\mathcal{P}^2))$ depends on the location of $\mathcal{P}^2$ in T_{ji}. The same can be done for the fourth sum in (16). Then, collecting all the terms that depend on the location of the two facilities, formula (16) becomes:

$$\sum_{v \in V(T_{ij})} h_v \, d(v, L(\mathcal{P}^1)) \, (1 - p_{\mathcal{P}^1}) + \left[d(j, L(\mathcal{P}^2)) \, (1 - p_{\mathcal{P}^2}) p_{\mathcal{P}^1} \sum_{v \in V(T_{ij})} h_v \right]$$
$$+ \sum_{v \in V(T_{ji})} h_v \, d(v, L(\mathcal{P}^2)) \, (1 - p_{\mathcal{P}^2}) + \left[d(i, L(\mathcal{P}^1)) \, (1 - p_{\mathcal{P}^1}) p_{\mathcal{P}^2} \sum_{v \in V(T_{ji})} h_v \right] + C[i, j] \tag{17}$$

where $C[i, j]$ is a constant including all terms independent of the facility locations (see also [9]). The adjusted weights for the vertices of T that must be computed at step 4.1 of ALGORITHM 2UMP are the following:

$$h'_v = \begin{cases} h_v \, (1 - p_{\mathcal{P}^1}), & \forall v \in T_{ij}, \ v \neq i \\ h_v \, (1 - p_{\mathcal{P}^2}), & \forall v \in T_{ji}, \ v \neq j \\ (1 - p_{\mathcal{P}^1}) \left[h_i + \sum_{v \in V(T_{ji})} h_v \, p_{\mathcal{P}^2} \right], & v = i \\ (1 - p_{\mathcal{P}^2}) \left[h_j + \sum_{v \in V(T_{ij})} h_v \, p_{\mathcal{P}^1} \right], & v = j. \end{cases} \tag{18}$$

Therefore, as in [9], we can re-write (17) as

$$\sum_{v \in V(T_{ij})} h'_v \, d(v, L(\mathcal{P}^1)) + \sum_{v \in V(T_{ji})} h'_v \, d(v, L(\mathcal{P}^2)) + C[i, j] \tag{19}$$

so that minimizing (19) actually corresponds to minimizing the sum of the weighted distances in the two subtrees T_{ij} and T_{ji}, separately.

References

[1] Z. Drezner, H. Hamacher, Facility Location: Applications and Theory, Springer, 2004.
[2] S. Nickel, J. Puerto, A.M. Rodriguez-Chía, An approach to location models involving sets as existing facilities, Math. Oper. Res. 28 (2003) 693–715.
[3] J.A. Mesa, J. Puerto, A. Tamir, Improved algorithm for several network location problems with equality measures, Discrete Appl. Math. 130 (2003) 437–448.
[4] J. Puerto, F. Ricca, A. Scozzari, Minimax regret path location on trees, Networks 58 (2011) 147–158.
[5] J. Puerto, F. Ricca, A. Scozzari, Range minimization problems in path-facility location on trees, Discrete Appl. Math. 160 (2012) 2294–2305.
[6] J. Puerto, A. Tamir, Locating tree-shaped facilities using the ordered median objective, Math. Program. Ser. A 102 (2005) 313–338.
[7] A. Tamir, J. Puerto, J.A. Mesa, A.M. Rodriguez-Chía, Conditional location of path and tree shaped facilites on trees, J. Algorithms 56 (2005) 50–75.
[8] H. Kerivin, A. Mahjoub, Design of survivable networks: a survey, Networks 46 (2005) 1–21.
[9] O. Berman, D. Krass, M. Menezes, Facility reliability issues in network p-median problems: strategic centralization and co-location effects, Oper. Res. 55 (2007) 332–350.
[10] J. Puerto, F. Ricca, A. Scozzari, Unreliable point facility location problems on networks, Discrete Appl. Math. (2013) http://dx.doi.org/10.1016/j.dam.2013.10.013.
[11] R.L. Church, M.P. Scaparra, R.S. Middleton, Identifying critical infrastructure: the median and covering facility interdiction problems, Ann. Assoc. Amer. Geogr. 94 (2004) 491–502.
[12] F. Liberatore, M.P. Scaparra, M.S. Daskin, Analysis of facility protection strategies against an uncertain number of attacks: the stochastic r-interdiction median problem with fortification, Comput. Oper. Res. 38 (2011) 357–366.
[13] R. Wood, Deterministic network interdiction, Math. Comput. Modelling 17 (1993) 1–18.
[14] Z. Drezner, Heuristic solution methods for two location problems with unreliable facilities, J. Oper. Res. Soc. 38 (1987) 509–514.
[15] L. Snyder, M.S. Daskin, Reliability models for facility location: the expected failure cost case, Transp. Sci. 39 (2005) 400–416.
[16] T. Cui, Y. Ouyang, Z. Shen, Reliable facility location design under the risk of disruptions, Oper. Res. 58 (2010) 998–1011.
[17] Z. Shen, R. Shan, J. Shang, The reliable facility location problem: formulations, heuristics and approximation algorithms, INFORMS J. Comput. 23 (2011) 470–482.
[18] A. Atamturk, M. Zhang, Two-stage robust network flow and design under demand uncertainty, Oper. Res. 55 (2007) 662–673.
[19] Y. Hollander, J. Prashker, The applicability of non-cooperative game theory in transport analysis, Transportation 33 (2006) 481–496.
[20] G. Laporte, A. Marín, J. Mesa, F. Perea, Designing robust rapid transit networks with alternative routes, J. Adv. Transp. 45 (2011) 54–65.
[21] G. Laporte, J. Mesa, F. Perea, A game theoretic framework for the robust railway transit network design problem, Transp. Res. B 44 (2010) 447–459.
[22] F. Perea, J. Puerto, Revisiting a game theoretic framework for the robust railway network design against intentional attacks, European J. Oper. Res. 226 (2013) 286–292.
[23] S. Ukkusuri, T. Mathew, S. Waller, Robust transportation network design under demand uncertainty, Comput.—Aided Civ. Infrastruct. Eng. 22 (2007) 6–18.
[24] T.U. Kim, T.J. Lowe, A. Tamir, J.E. Ward, On the location of a tree-shaped facility, Networks 28 (1996) 167–175.

[25] C.A. Morgan, J.P. Slater, A linear Algorithm for a core of a tree, J. Algorithms 1 (1980) 247–258.
[26] M.R. Garey, D.S. Johnson, R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, SIAM J. Comput. 5 (1976) 704–714.
[27] A. Itai, C.H. Papadimitriou, J.L. Szwarcfiter, Hamilton paths in grid graphs, SIAM J. Comput. 4 (1982) 676–686.
[28] S.L. Hakimi, E.F. Schmeichel, M. Labbé, On locating path- or tree-shaped facilities on networks, Networks 23 (1993) 543–555.
[29] B.-F. Wang, Finding a 2-core of a tree in linear time, SIAM J. Discrete Math. 15 (2002) 193–210.
[30] B.-F. Wang, J.-J. Lin, Finding a two-core of a tree in linear time, in: Lecture Notes in Computer Science, vol. 1969, 2000, pp. 467–478.
[31] R.I. Becker, I. Lari, A. Scozzari, G. Storchi, The location of median paths on grid graphs, Ann. Oper. Res. 150 (2001) 65–78.
[32] I. Lari, F. Ricca, A. Scozzari, Comparing different metaheuristic approaches for the median path problem with bounded length, European J. Oper. Res. 190 (2008) 587–597.
[33] M.B. Richey, Optimal location of a path or tree on a network with cycles, Networks 20 (1990) 391–407.
[34] I. Lari, F. Ricca, A. Scozzari, R.I. Becker, Locating median paths on connected outerplanar graphs, Networks 57 (2011) 294–307.
[35] A. Tamir, An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs, Oper. Res. Lett. 19 (1996) 59–64.
[36] J. Puerto, F. Ricca, A. Scozzari, The continuous and discrete path-variance problems on trees, Networks 53 (2009) 221–228.
[37] J. Puerto, F. Ricca, A. Scozzari, Extensive facility location problems on networks with equity measures, Discrete Appl. Math. 157 (2009) 1069–1085.