Unreliable point facility location problems on networks

Justo Puerto[1], Federica Ricca[2], Andrea Scozzari[3]
[1]IMUS, Instituto de Matemáticas de la Universidad de Sevilla (Spain)
puerto@us.es
[2]Sapienza Università di Roma (Italy)
federica.ricca@uniroma1.it
[3]Università degli Studi "Niccolò Cusano", Roma, (Italy)
andrea.scozzari@unisucano.it

Abstract

In this paper we study facility location problems on graphs under the most common criteria, such as, median, center and centdian, but we incorporate in the objective function some reliability aspects. Assuming that facilities may become unavailable with a certain probability, the problem consists of locating facilities minimizing the overall or the maximum expected service cost in the long run, or a convex combination of the two. We show that the k-facility problem on general networks is NP-hard. Then, we provide efficient algorithms for these problems for the cases of $k = 1, 2$, both on general networks and on trees. We also explain how our methodology extends to handle a more general class of unreliable point facility location problems related to the ordered median objective function.

Keywords: Reliable facility location, point location, service disruption.

1 Introduction

The literature on facility location has grown a lot in the last years, one of the reasons being its close relationship with logistics [4, 15]. This connection has given rise to the incorporation into location models of very interesting new issues that had already appeared in the area of logistics analysis. Among the many extensions of interest that can be found in the literature, in this paper we would like to focus on reliability issues connected to the possibility of disruption of a facility. There are different sources of uncertainty that may cause the disruption of facilities, giving raise to situations where some facilities become temporary unavailable to provide the service to the customers. Some examples are system failures, natural disasters, terrorist attacks, labor strikes, etc.. As a result, logistic systems incur in extra transportation costs, since customers originally served by the closest facilities must be redirected to more distant ones (see, e.g., [10]). This has motivated an alternative approach to the "customer-to-closest facility" cost that consists of locating facilities that minimize the total expected service cost in the long run, assuming that failures are accidental, and their probabilities can be estimated in advance [1].

One of the assumptions made in [1] is that the probability of disruption is a function of the facility design and is not dependent on the facility location. Of course, this assumption is valid in some cases and in particular when probabilities of failure are equal as, for example, in [34]. Nevertheless, there are also cases where the facility disruption is dependent on the location, as

when disruptions are associated to natural disasters (such as, flooding or earthquakes). This is exactly the probability framework analyzed in this paper. We assume that independent failure probabilities are associated to the vertices of a network and that the failure probabilities of the points in the interior of an edge are induced by the failure probabilities of the end vertices of the edge. For the ease of presentation, and without loss of generality, we will assume that the probability of failure of a point in an edge (u, v) is the linear interpolation of the probabilities of failure of the vertices u and v. Therefore, the closer a point in an edge to a vertex, the greater the influence of such a vertex on the failure probability of the point. The reader is referred to Section 6 for an explanation of how this approach extends to be applied to general types of probability functions.

It is clear that these probabilities are location dependent implying that some of the results that were valid in [1] do not longer hold, and thus making our new model challenging. The reader may note that assuming that probabilities are location dependent has already been considered by some authors in the field of discrete location by simply considering different probabilities associated to each facility [2, 33, 35]. Nevertheless, the related location problem on networks has not been addressed yet in the literature.

The goal of this paper is to analyze reliability issues of location problems under this new failure probability model. To this purpose, we analyze 1- and 2-facility location problems on networks with the most commonly used objective functions: minimizing the overall sum (*median* criterion), the maximum (*center* criterion) expected service cost in the long run, or a convex combination of the sum and maximum (*centdian* criterion). We adopt tools of different types.

On the one hand, we rely on discretization. Since the seminal paper by Hakimi [7], much of the work related to location problems on networks has been devoted to identify a finite set of points such that some optimal solutions of the problem belong to it. This set, called *Finite Dominating Set* (FDS), is very useful in order to restrict the number of possible candidate points to be optimal solutions. This sort of discretization strategy has given very good results and it is at the basis of our approach to solve the problems considered in this paper. An overview of the literature involving characterizations of FDS shows a lot of papers that succeeded in finding such kind of sets. The reader is referred to [12] and to [22] and the references therein as literature sources on this subject (see, also [14, 17, 21, 28, 30]). More recent references dealing with other multifacility location problems on networks are [13, 16]. The former derives a FDS for the k-median problem with positive and negative weights; the latter reference solves the 2-facility case for different equity measures.

On the other hand, we shall make an extensive use of arrangements of planar and three dimensional curve patches and of some other tools borrowed from Computational Geometry that will allow us to state the complexity of our algorithms [23, 31, 32].

In this paper we develop efficient solution algorithms to tackle 1- and 2-facility problems for all the objective functions listed above. There are reasons for distinguishing between the 1- and 2-facility cases. In the 1-facility problem one is assuming that users are willing to try to get the service once before balking, whereas in the 2-facility case, customers will try twice (more than one) before balking which makes a difference in the analysis. Since it is reasonable to assume that in real-life situations a customer does not try more than a very limited number of times, we restrict ourselves to consider w.l.o.g. the case of two attempts to receive service before giving up. The main interest is for the study of 1- and 2-facility cases, although our analysis extends further to any fixed number k. It is important to point out that there is a substantial difference between our problems and standard location problems where multifacility issues have a completely different interpretation.

We consider the case of general networks, as well as, tree networks. On general networks we observe that the k-facility versions of the problems considered in this paper are NP-Hard since they contain the k-median, k-center and k-centdian problem, respectively (see, [18, 19]). Our final complexity results for $k = 1, 2$ on general networks and on trees are summarized in Table 1.

	1-Facility		2-Facility	
	Graphs	Trees	Graphs	Trees
Median	$O(mn \log n)$	$O(n)$	$O(m^2n^3)$	$O(n^3)$
Center	$O(m\lambda_5(n) \log n)$	$O(n\lambda_5(n) \log n)$	$O(m^2n^{4+\varepsilon})$	$O(n^{5+\varepsilon})$
Centdian	$O(m\lambda_5(n) \log n)$	$O(n\lambda_5(n) \log n)$	$O(m^2n^{4+\varepsilon})$	$O(n^{5+\varepsilon})$

Table 1: Summary of the results for unreliable facility location problems on networks. Function $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order s on n symbols.

The paper is organized as follows. Section 2 presents the notation and the definition of the problems analyzed in the paper. It also contains an interesting counterexample showing that vertex optimality is not ensured in this type of reliability problems, thus motivating the study of the continuous version of the problems. Section 3 focuses on median problems. In particular, Sections 3.1 and 3.2 consider the simplest version of the unreliable location problem, i.e., the 1-Median Unreliable Point problem, on graphs and on trees, respectively. Here, it is assumed that facilities may fail but that users will attempt to get the service only once and that they balk if the facility fails at the first trial. Sections 3.3 and 3.4 extend the above problem to the situation where users are willing to try twice to get the service before giving up. In Section 3.3, we provide an algorithm to solve the 2-Median Unreliable Points problem on graphs. Section 3.4 analyzes the same problem on trees developing a more efficient algorithm that takes advantage of the intrinsic properties of tree networks. Section 4 deals with similar problems under the hypothesis that one wishes to minimize the worst-case expected cost (center criterion) in the long run, with respect to the location of one or two facilities. In Section 5, we extend to a convex combination of the median and center objective functions, namely the 1- and 2-CentDian Unreliable Points problem. For these problems we provide solution algorithms based on the results of the previous sections. The paper ends in Section 6 with some concluding remarks on extensions of the problems addressed in this paper and future research on this topic.

2 Notation and problem definition

We are given an undirected connected simple graph $G = (V, E)$ without self loops, with $|V| = n$ and $|E| = m$. Assume that a non negative weight h_z is assigned to each vertex $z \in V$, and a positive length ℓ_{uv} is associated to each edge $e = (u, v)$. As customary, we suppose that the clients are located only at the vertices of G so that h_z corresponds to the level of demand at z. Let $A(G)$ denote the continuum set of points in the interior of the edges of G; given two points x and y in $A(G)$ (that may be vertices or points in the interior of some edges of G) $d(x, y)$ is the length of the shortest path from x to y. Here we assume that the edge (u, v) can be viewed as a continuous interval of values $[u, v]$ and in our notation $x \in [u, v]$ represents the point located at distance x from vertex u. In this paper we consider the probability that a facility is temporary non operating due to accidental events that may affect the network structure (*Unreliable Facility*

Location). In fact, differently from the probabilistic framework analyzed in [1, 27] here we assume that the probability of failure is associated to the place or the structure where the facility is located and not to the facility itself (i.e., to the facility "design"). Denote by $p_v \in [0,1]$ the probability of failure of vertex $v \in V$, and by $p_x^e \in [0,1]$ the probability of failure of a point x along an edge $e = (u,v)$. We assume that p_x^e is not independent from the probabilities of the end vertices u and v of edge e, so that for a facility located at a point x in (u,v) we define:

$$p_x^e = p_u\left(\tfrac{\ell_{uv}-x}{\ell_{uv}}\right) + p_v\left(\tfrac{x}{\ell_{uv}}\right) \quad \forall\, x \in [u,v]. \tag{1}$$

This assumption on the probability function is not restrictive at all, since we can handle or approximate up to any degree of accuracy any continuous probability function, as briefly discussed in the concluding remarks.

We also introduce a parameter $\beta_z \geq 0$ that represents a non negative penalty to be paid when z is not served or is served by a remote backup facility. Since β_z may be interpreted as the cost to serve a client in z from a facility outside the network when the located facilities fail, $\forall\, z \in V$, we impose $\beta_z \geq \max\limits_{u,v \in V} d(u,v)$, (see, also [1]).

In this paper we study the problem of locating one or two unreliable facilities minimizing objective functions based on the median and/or center criteria.

Consider the case of locating k facilities $\mathtt{x} = (x_1, \ldots, x_k)$. For a given $z \in V$, let $L_j^z(\mathtt{x})$, with $j = 1, 2, \ldots, k$, denote the point (or vertex) where the j-th closest facility to z is located. The general expression for the expected weighted cost for client z under the above probabilistic framework is given by:

$$\begin{aligned} f^z(\mathtt{x}) \;=\; & h_z \left[d(z, L_1^z(\mathtt{x}))(1 - p_{L_1^z(\mathtt{x})}) + d(z, L_2^z(\mathtt{x}))p_{L_1^z(\mathtt{x})}(1 - p_{L_2^z(\mathtt{x})}) + \ldots \right. \\ & \left. \ldots + d(z, L_k^z(\mathtt{x})) \prod_{j=1}^{k-1} p_{L_j^z(\mathtt{x})}(1 - p_{L_k^z(\mathtt{x})}) \right] + h_z \beta_z \prod_{j=1}^{k} p_{L_j^z(\mathtt{x})}. \end{aligned} \tag{2}$$

When referring to the average criterion we have the *k-Median Unreliable Points* (kMUP) location problem with the following objective function

$$f(\mathtt{x}) = \sum_{z \in V} f^z(\mathtt{x}) \tag{3}$$

while, when considering the worse case criterion, we have the *k-Center Unreliable Points* (kCUP) location problem with the following objective function

$$f(\mathtt{x}) = \max_{z \in V} f^z(\mathtt{x}). \tag{4}$$

In the rest of the paper, when we have to compute the objective function $f(\mathtt{x})$ restricted to a given set R, we will denote the corresponding restricted function by $f_R(\mathtt{x})$.

We remark that the above problems are different from the standard k facility location problems in which k points are located but each client is served by only one of them. In our problem, for each client all the k located facilities are different possibilities to be served: following the ordering of the distances, the client will be served first by its closest (operating) facility, then by

its second closest, and so on; if all the facilities fail, a penalty cost must be paid for redirecting the client to a backup facility.

In this paper we study the special cases of $k = 1, 2$ both for the median and center criterion. We call them *1-Median Unreliable Point* (1MUP), *2-Median Unreliable Points* (2MUP) problems and *1-Center Unreliable Point* (1CUP) and *2-Center Unreliable Points* (2CUP) problems, respectively. In addition, we also consider the centdian criterion and formulate and solve the *k-CentDian Unreliable Points* (kCDUP) problem ($k = 1, 2$).

The above unreliable facility location problems have a very different nature w.r.t. those introduced in [1]. For example, in our probability framework, the vertex optimality property for the median criterion does not hold any more, meaning that there might exist an optimal solution located in the interior of an edge. In particular, even in the simplest case of a tree network with equal vertex weights, the optimal solution may not satisfy the vertex optimality property. In fact, consider a tree T, and assume w.l.o.g that $\beta_z = 0$ and $h_z = 1$, for all $z \in V$. Given an edge $e = (u, v)$, denote by V_u and V_v the sets of vertices in T such that $d(z, u) < d(z, v)$ and $d(z, u) > d(z, v)$, respectively. For a point $x \in [u, v]$, let $a_e^z x + b_e^z$ be the linear function defining the distance of a vertex $z \in V$ from the point x depending on the position of x in the interior of (u, v). In particular, let $b_e^z = d(z, u)$, for all $z \in V$; for all the vertices $z \in V_u$ we have $a_e^z = 1$, while for the vertices $z \in V_v$ we have $a_e^z = -1$. With the above assumption, the objective function of 1MUP is:

$$\begin{aligned} f(x) &= \sum_{z\in V_u}(a_e^z x + b_e^z)(1 - p_x^e) + \sum_{z\in V_v}(a_e^z x + b_e^z)(1 - p_x^e) \\ &= (|V_u| - |V_v|)(1 - p_x^e)x + \sum_{z\in V} d(z, u)(1 - p_x^e) \\ &= (|V_u| - |V_v|)(1 - \big[p_u\big(\tfrac{\ell_{uv}-x}{\ell_{uv}}\big) + p_v\big(\tfrac{x}{\ell_{uv}}\big)\big])x + D(1 - \big[p_u\big(\tfrac{\ell_{uv}-x}{\ell_{uv}}\big) + p_v\big(\tfrac{x}{\ell_{uv}}\big)\big]), \end{aligned} \tag{5}$$

where $D = \sum_{z\in V} d(z, u)$. After some computations we obtain:

$$f(x) = (|V_u| - |V_v|)\big(\tfrac{p_u-p_v}{\ell_{uv}}\big)x^2 + \big[(|V_u| - |V_v|)(1 - p_u) + D\big(\tfrac{p_u-p_v}{\ell_{uv}}\big)\big]x + D(1 - p_u). \tag{6}$$

This is a quadratic function in one variable x which achieves its minimum either at the end vertices of (u, v) or at a stationary point $\bar{x}$ in the interior of $[u, v]$ (provided that it exists):

$$\bar{x} = -\frac{\big[(|V_u|-|V_v|)(1-p_u)+D\big(\frac{p_u-p_v}{\ell_{uv}}\big)\big]}{2(|V_u|-|V_v|)\big(\frac{p_u-p_v}{\ell_{uv}}\big)}. \tag{7}$$

Then, we have:

$$f(\bar{x}) = -\tfrac{1}{4}\big[\frac{\big((|V_u|-|V_v|)(1-p_u)+D\big(\frac{p_u-p_v}{\ell_{uv}}\big)\big)^2}{(|V_u|-|V_v|)\big(\frac{p_u-p_v}{\ell_{uv}}\big)}\big] + D(1 - p_u), \tag{8}$$

and, if $\bar{x}$ lies in the interior of $[u, v]$, we have to compare $f(\bar{x})$ with

$$\begin{aligned} f(u) &= D(1 - p_u) \\ f(v) &= D(1 - p_v) + (|V_u| - |V_v|)\ell_{uv}(1 - p_v). \end{aligned} \tag{9}$$

In particular, in the above case, if $|V_u| > |V_v|$ and $p_u > p_v$ we have $f(\bar{x}) < f(u)$ and $f(\bar{x}) < f(v)$ so that the minimum is achieved at a point x in the interior of (u, v). This proves that the vertex optimality property does not necessarily hold even in the simplest case of an unweighted tree network. Similar considerations apply to the 2MUP case.

We note that, the optimal point location may be in the interior of an edge also when considering the center criterion. In this case the proof follows straightforwardly from the fact that, setting in kCUP $p_v = 0$, $\forall\ v \in V$, implies that kCUP becomes the classic weighted 1-center problem for which the property does not necessarily hold [18]. Moreover, since the objective function is quadratic, an optimal point location for this problem may not even correspond to a point in the FDS of the center problem.

3 The Median Unreliable Point location problem

In this section we focus on the k-unreliable points location problem adopting the median criterion for $k = 1, 2$. We provide efficient algorithms both for general graphs and for tree networks.

3.1 The 1-Median Unreliable Point on graphs

Consider a graph $G = (V, E)$ and fix an edge $e = (u, v)$ and a vertex z (client). Let x_z be the point in (u, v) such that $d(z, u) + d(u, x_z) = d(z, v) + d(v, x_z)$ (possibly $x_z = u$ or $x_z = v$). The point x_z is the so called *breakpoint* in (u, v) w.r.t. vertex z. For each vertex of G, let $\{x_1, x_2, \ldots, x_n\}$ be the set of $O(n)$ breakpoints in (u, v). For the sake of simplicity, we assume that u and v are breakpoints in (u, v), so that $x_1 = u$ and $x_n = v$. Suppose that the facility is located at the point $x \in [x_i, x_{i+1}]$, $i = 1, \ldots, n-1$, the distance function of a vertex $z \in V$ w.r.t. x in the i-th interval $[x_i, x_{i+1}]$ is (see, Figure 1):

$$a_i^z x + b_i^z = \begin{cases} x + d(z, u), & \text{if } x_{i+1} \leq x_z \\ -x + [d(z, v) + \ell_{uv}], & \text{if } x_{i+1} > x_z. \end{cases} \tag{10}$$

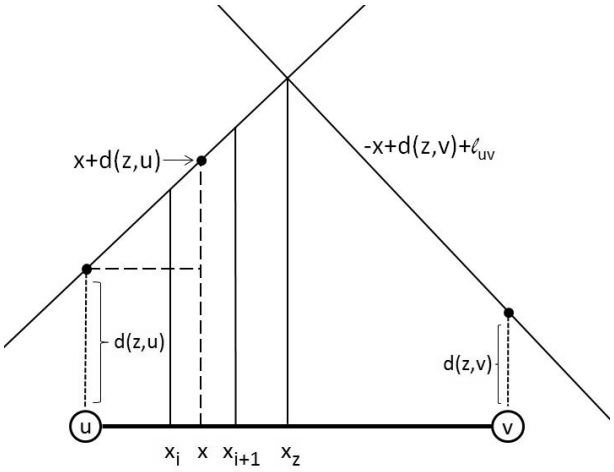

Figure 1: The distance function of a vertex $z \in V$ w.r.t. x.

Then, for each interval i, $i = 1, \ldots, n-1$, the objective function restricted to $x \in [x_i, x_{i+1}]$ is:

$$f_i(x) \quad = \quad \sum_{z \in V} h_z(a_i^z x + b_i^z)(1 - p_x^e) + \sum_{z \in V} h_z \beta_z p_x^e. \tag{11}$$

Since p_x^e is a linear function of x (see, (1)), function $f_i(x)$ is a quadratic function in the variable x, and it can be re-written in a more compact form:

$$f_i(x) \quad = \quad A_i x^2 + B_i x + C_i. \tag{12}$$

For each interval i, $i = 1, \ldots, n-1$, the minimum of $f_i(x)$ is obtained either at the extremes of the interval i or at a stationary point, provided that it belongs to $[x_i, x_{i+1}]$. In order to find the minimum of the objective function in the whole interval $[u, v]$, we consider the intervals delimited by consecutive breakpoints. Thus, when analyzing a point x in the interval $[x_{i+1}, x_{i+2}]$, we update the function $f_i(x)$ to obtain the new function $f_{i+1}(x)$. We observe that once all the breakpoints x_z, for all $z \in V$, have been computed, it is possible to sort them in a non decreasing order, so that when moving from one interval to the next, only one distance function $a_i^z x + b_i^z$ must be updated, precisely the one related to the client z that generated breakpoint $x_i = x_z$ (see, Figure 1). Relying on the definition of a_i^z and b_i^z, this can be done in constant time.
Hence, 1MUP can be solved on a general graph G with the following algorithm.

Algorithm 1MUP

1. For each edge $e = (u, v)$ of E
   - 1.1 compute and sort all the breakpoints $x_1, x_2, \ldots, x_n$
   - 1.2 for each interval $[x_i, x_{i+1}]$, $i = 1, \ldots, n-1$
     - 1.2.1 compute $f_i(x)$
     - 1.2.2 find the minimum $\bar{x}_i$ of $f_i(x)$ in $[x_i, x_{i+1}]$ and let $f_i(\bar{x}_i)$ be the corresponding objective function value
   - 1.3 let x_e be such that $f(x_e) = \min_{i:[x_i, x_{i+1}] \subset [u,v]} f_i(\bar{x}_i)$
2. let x^* be such that $f(x^*) = \min_{e \in E} f(x_e)$

Theorem 1 *Algorithm 1MUP solves the 1-Median Unreliable Point location problem on a general graph $G = (V, E)$ in $O(mn \log n)$ time.*

Proof:
The major effort of the algorithm consists of computing and sorting the set of $O(n)$ breakpoints of an edge. This requires $O(n \log n)$ time. As discussed above, computing and, in particular, updating $f_i(x)$ requires constant time, as well as, finding the minimum of $f_i(x)$ in $[x_i, x_{i+1}]$. Thus, for each edge $e = (u, v)$ the time complexity for finding x_e is $O(n \log n)$. Hence, the overall time complexity of the above algorithm is $O(mn \log n)$. □

3.2 The 1-Median Unreliable Point on trees

Let us now consider 1MUP on a weighted tree $T = (V, E)$ with $|V| = n$. Given a vertex r, we root T at r and denote by T_r the resulting rooted tree. For each edge $e = (u, v)$ we do not need to compute the breakpoints x_i, since, in a tree, it is always possible to partition the vertices of T w.r.t. edge (u, v) into the two sets V_u and V_v already introduced in Section 2. Suppose that

for (u, v), v is closer to the root r than u and set $v = p(u)$, meaning that v is the parent of u in T_r. Let $V(T_u)$ be set of vertices in the subtree T_u rooted at u (with $u \in V(T_u)$). Denote by $V(T_r \backslash T_u)$ the corresponding set of vertices in the remaining tree. Let $S(u)$ be the set of the children of u in T_r, so that a vertex u is a leaf if and only if $|S(u)| = 0$. It is well-known that visiting T_r bottom-up level by level from the leaves to r and top-down from r to the leaves, one can compute, for each vertex u, the following quantities (see, i.e., [20, 24, 26]):

1. $H_B(u)$, the sum of the weights of all the vertices in $V(T_u)$;
2. $H_A(u)$, the sum of the weights of all the vertices in $V(T_r \backslash T_u)$;
3. $D_B(u)$, the sum of the weighted distances of all the vertices in $V(T_u)$ to u;
4. $D_A(u)$, the sum of the weighted distances of all the vertices in $V(T_r \backslash T_u)$ to u.

All these quantities can be computed once in a preprocessing phase in $O(n)$ time. Given an edge (u, v) and a point $x \in [u, v]$, we already introduced the distance from a vertex z to x as a function of x, that, for T_r, can be re-written as (here $V(T_u)$ and $V(T_r \backslash T_u)$ correspond to V_u and V_v, respectively):

$$a_e^z x + b_e^z = \begin{cases} x + d(z, u), & \text{if } z \in V(T_u) \\ -x + d(z, u), & \text{if } z \in V(T_r \backslash T_u), \end{cases} \tag{13}$$

so that the objective function restricted to $x \in [u, v]$ is given by the following quadratic function:

$$f_e(x) \quad = \quad \sum_{z \in V} h_z (a_e^z x + b_e^z)(1 - p_x^e) + \sum_{z \in V} h_z \beta_z p_x^e. \tag{14}$$

When visiting the tree bottom-up, for each edge $e = (u, v)$ we can compute the minimum of the function $f_e(x)$ for $x \in [u, v]$. We observe that, once all the necessary quantities listed above have been computed, the quadratic function $f_e(x)$ can be evaluated and updated in constant time for each edge of T_r. Thus, the same algorithm already used in Section 3.1 for general graphs can be applied here. The fact that for a tree we do not have to compute and sort the set of breakpoints in each edge implies a straightforward relevant reduction of the computational complexity which is stated in the following theorem.

Theorem 2 *Algorithm 1MUP solves the 1-Median Unreliable Point problem on trees in $O(n)$ time.*

3.3 The 2-Median Unreliable Points on graphs

In this section we study 2MUP on a general graph $G = (V, E)$. We consider two possible locations x and y of two facilities in the interior of edge $e = (u, v)$ and $g = (r, s)$, respectively. For a given vertex $z \in V$, denote by $L_1^z(x, y)$ and $L_2^z(x, y)$ the location of the closest and the second closest facility to vertex z, that is:

$$L_1^z(x, y) \in \arg\min\{d(z, x), d(z, y)\} \qquad \text{and} \qquad L_2^z(x, y) \in \arg\max\{d(z, x), d(z, y)\}. \tag{15}$$

Ties are broken arbitrarily. For 2MUP, the expected cost for a client $z \in V$ w.r.t. x and y is defined as follows:

$$\begin{aligned} f^z(x,y) &= h_z\big[d(z,L_1^z(x,y))(1-p_{L_1^z(x,y)}) + d(z,L_2^z(x,y))p_{L_1^z(x,y)}(1-p_{L_2^z(x,y)})\big] \\ &\quad + h_z\beta_z p_{L_1^z(x,y)}p_{L_2^z(x,y)}. \end{aligned} \tag{16}$$

Hence, the total expected weighted cost is given by:

$$\sum_{z\in V} f^z(x,y). \tag{17}$$

In this section we generalize the approach provided in Section 3.1 for 1MUP to the two facilities case. For every pair of edges $e = (u,v)$ and $g = (r,s)$ (possibly $u = r$ and $v = s$), we have to determine the distance from each vertex $z \in V$ to points x in (u,v) and y in (r,s). In fact, considering the two edges (u,v) and (r,s) as two continuous intervals of points $[u,v]$ and $[r,s]$, we have to find a *subdivision* (or *arrangement*) of the subspace $[u,v]\times[r,s] \subset \mathbb{R}^2$ such that, in each subset of the subdivision, $L_1^z(x,y)$ and $L_2^z(x,y)$ are univocally identified for all $z \in V$, and do not change. Recalling that in our notation x and y represent also the distances from vertex u and r, respectively, the above arrangement is induced by the following set of six linear equations:

$$\begin{aligned} x + d(z,u) &= (\ell_{uv} - x) + d(z,v) \\ y + d(z,r) &= (\ell_{rs} - y) + d(z,s) \\ x + d(z,u) &= y + d(z,r) \\ x + d(z,u) &= (\ell_{rs} - y) + d(z,s) \\ y + d(z,r) &= (\ell_{uv} - x) + d(z,v) \\ (\ell_{uv} - x) + d(z,v) &= (\ell_{rs} - y) + d(z,s). \end{aligned} \tag{18}$$

Applying equations (18) for all $z \in V$, we obtain an arrangement of $[u,v]\times[r,s]$ induced by a collection of $O(n)$ lines [31]. For the pair of edges e and g, we denote the corresponding arrangement by $\mathcal{A}_{eg}$. An element of the arrangement $\mathcal{A}_{eg}$ corresponds to a cell c delimited by some of the equations (18). The arrangement of $O(n)$ lines can be obtained in $O(n^2)$ time with a procedure provided in [31] that, for each cell c, produces the list of equations that define c along with the coordinates of the vertices of c.
Also in the particular case when $u = r$ and $v = s$ we can find the arrangement of lines of the subspace $[u,v]\times[u,v]$ by equations (18) and the additional inequalities $0 \le x \le y \le \ell_{uv}$. The analysis for $y \le x$ can be skipped since it leads to the same solution with interchanged names for the two facilities.

Let us now consider the objective function restricted to a cell $c \in \mathcal{A}_{eg}$, we have:

$$
\begin{aligned}
f_c(x,y) \;&= \sum_{z:L_1^z(x,y)=x} h_z\big[(a_c^z x + b_c^z)(1-p_x^e) + (a_c^z y + b_c^z)p_x^e(1-p_y^g)\big] \\
&+ \sum_{z:L_1^z(x,y)=y} h_z\big[(a_c^z y + b_c^z)(1-p_y^g) + (a_c^z x + b_c^z)p_y^g(1-p_x^e)\big] \\
&+ \sum_{z\in V} h_z \beta_z p_x^e p_y^g,
\end{aligned} \tag{19}
$$

where $a_c^z x + b_c^z$ and $a_c^z y + b_c^z$ are linear functions with $a_c^z \in \{-1, 0, 1\}$ and b_c^z is computed on the basis of the distances $d(z,u)$, $d(z,v)$, $d(z,r)$ and $d(z,s)$ and depends on which equations in (18) define cell c. Thus, $f_c(x,y)$ is a function of the two unknown x and y corresponding to the following polynomial (in general form):

$$
f_c(x,y) \;= \alpha_{21}x^2y + \alpha_{12}xy^2 + \alpha_{20}x^2 + \alpha_{02}y^2 + \alpha_{11}xy + \alpha_{10}x + \alpha_{01}y + \alpha_{00}, \tag{20}
$$

where the subscripts of the coefficients refer to the power of x and y in the corresponding monomial, respectively. We observe that, for a given cell $c \in \mathcal{A}_{eg}$ the coefficients α_{ij} can be computed in $O(n)$ time. The minimum of $f_c(x,y)$ is attained either at a stationary point, provided that it belongs to cell c, or at the boundary of c, including its vertices. In order to find the stationary points, we have to solve the following system of two equations in x and y:

$$
\begin{cases}
\frac{\partial}{\partial x} f_c(x,y) = 2\alpha_{21}xy + \alpha_{12}y^2 + 2\alpha_{20}x + \alpha_{11}y + \alpha_{10} = 0 \\
\\
\frac{\partial}{\partial y} f_c(x,y) = 2\alpha_{12}xy + \alpha_{21}x^2 + 2\alpha_{02}y + \alpha_{11}x + \alpha_{01} = 0
\end{cases} \tag{21}
$$

Solving the first equation w.r.t. x we obtain:

$$
\bar{x} \;= -\tfrac{\alpha_{12}y^2+\alpha_{11}y+\alpha_{10}}{2(\alpha_{21}y+\alpha_{20})}. \tag{22}
$$

When $y \neq -\frac{\alpha_{20}}{\alpha_{21}}$, we can substitute $\bar{x}$ in the second equation and obtain:

$$
\alpha_{21}\big(-\tfrac{\alpha_{12}y^2+\alpha_{11}y+\alpha_{10}}{2(\alpha_{21}y+\alpha_{20})}\big)^2 + 2\alpha_{12}\big(-\tfrac{\alpha_{12}y^2+\alpha_{11}y+\alpha_{10}}{2(\alpha_{21}y+\alpha_{20})}\big)y + 2\alpha_{02}y + \alpha_{11}\big(-\tfrac{\alpha_{12}y^2+\alpha_{11}y+\alpha_{10}}{2(\alpha_{21}y+\alpha_{20})}\big) + \alpha_{01} = 0 \tag{23}
$$

which is a polynomial of degree 4 in y whose roots can be computed in constant time. Call them y_h^c, $h = 1,2,3,4$. Then, we check which of the four stationary points (x_h^c, y_h^c), $h = 1,2,3,4$, belong to c and compute (if it exists) the minimum of $f_c(x,y)$ in the interior of the cell $c \in \mathcal{A}_{eg}$.

Notice that if $y = -\frac{\alpha_{20}}{\alpha_{21}}$ one has $2(\alpha_{21}y + \alpha_{20}) = 0$ implying that the term in x in $\frac{\partial}{\partial x} f_c(x,y)$ vanishes and one has only to check whether the roots of $\alpha_{12}y^2 + \alpha_{11}y + \alpha_{10} = 0$ are solutions of system (21) or not. This additional check can be done in constant time, too.

Moreover, we have to find the minimum of $f_c(x,y)$ on the boundary of c. To do this, we consider a parametrization of x and y w.r.t. the direction t along any given segment of the boundary of c. Then, substituting $x(t)$ and $y(t)$ in $f_c(x,y)$ in the bounded interval defined by the segment, we obtain a polynomial in t of degree 3:

$$
f_c(x(t), y(t)) \;= \phi_3 t^3 + \phi_2 t^2 + \phi_1 t + \phi_0, \tag{24}
$$

where the coefficients ϕ_i are conveniently defined, and for which the minimum can be easily computed. Finally, we have to evaluate the function $f_c(x, y)$ at each vertex of c.

Summarizing all the above steps, we obtain the following algorithm for solving 2MUP.

Algorithm 2MUP

1. For each pair of edges $e = (u, v)$ and $g = (r, s)$ (possibly $u = r$, and $v = s$) of E
   - 1.1 determine the arrangement $\mathcal{A}_{eg}$ of the subspace $[u, v] \times [r, s] \subset \mathbb{R}^2$
   - 1.2 for each cell $c \in \mathcal{A}_{eg}$
     - 1.2.1 compute $f_c(x, y)$
     - 1.2.2 find the minimum of $f_c(x, y)$ (if it exists) in the interior of c
     - 1.2.3 compute the minimum of $f_c(x, y)$ on the boundary of c
     - 1.2.4 evaluate $f_c(x, y)$ at each vertex of c
     - 1.2.5 let (x_c, y_c) be the pair that provides the minimum objective function value w.r.t. cell c
   - 1.3 let (x_{eg}, y_{eg}) be such that $f(x_{eg}, y_{eg}) = \min_{c \in \mathcal{A}_{eg}} f_c(x_c, y_c)$
2. let (x^*, y^*) be such that $f(x^*, y^*) = \min_{e,g \in E} f(x_{eg}, y_{eg})$

Theorem 3 *Algorithm 2MUP solves the 2-Median Unreliable Points problem on graphs in $O(m^2n^3)$ time.*

Proof:
Determining the arrangement $\mathcal{A}_{eg}$ of $[u, v] \times [r, s]$ by using equations (18) requires $O(n^2)$ time. This arrangement produces $O(n^2)$ cells c [31]. For a function $f_c(x, y)$, steps 1.2.2 - 1.2.5 can be implemented in constant time. Different from Algorithm 1MUP, for each cell we have now to compute the function $f_c(x, y)$ from scratch each time we move from c to an adjacent cell. This takes $O(n)$ time for each function $f_c(x, y)$, and implies that for each pair of edges $e = (u, v)$ and $g = (r, s)$, the point (x_{eg}, y_{eg}) can be found in $O(n^3)$ time. Repeating the procedure for all the pairs of edges in E leads to an overall time complexity of $O(m^2n^3)$. □

To conclude this section, we observe that when all the edges of G have the same length (for instance, all lengths equal to 1), for a given pair of edges $e = (u, v)$ and $g = (r, s)$ the set of equations (18) generates in $[u, v] \times [r, s]$ always a constant number of lines, whichever the value of n, thus producing an arrangement $\mathcal{A}_{eg}$ of $[u, v] \times [r, s]$ formed by only a fixed number of (namely, 8) cells. Thus, since for a given cell $c \in \mathcal{A}_{eg}$, computing function $f_c(x, y)$ requires $O(n)$ time, the point (x_{eg}, y_{eg}) can be also found in $O(n)$ time. This means that, in this special case, the overall time complexity of 2MUP on a graph reduces to $O(nm^2)$.

3.4 The 2-Median Unreliable Points on trees

In this section we solve the 2MUP problem on a weighted tree $T = (V, E)$. We refer to the same notation already introduced in Section 3.2 for a tree T_r rooted at vertex r. Consider two edges $e = (u, p(u))$ and $g = (v, p(v))$, and two points $x \in [u, p(u)]$ and $y \in [v, p(v)]$. It is clear that all the vertices $z \in V(T_u)$ have $L_1^z(x, y) = x$ and $L_2^z(x, y) = y$, while, for

all the vertices $z \in V(T_v)$, $L_1^z(x,y) = y$ and $L_2^z(x,y) = x$ holds. For all the other vertices $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$, $L_1^z(x,y)$ and $L_2^z(x,y)$ depend on the distance of z from x and y which are decreasing linear functions of x and y, respectively. These functions are characterized by different intercepts $b_e^z = d(z,u) = d(z,p(u)) + \ell_{up(u)}$ and $b_g^z = d(z,v) = d(z,p(v)) + \ell_{vp(v)}$ with slope $a_e^z = a_g^z = -1$. In a tree, the advantage is that b_e^z, $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$, are increasing monotone, as well as, b_g^z, $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$, are decreasing monotone when z moves from $p(u)$ to $p(v)$. On this basis, in $[u,p(u)] \times [v,p(v)]$ we are able to find an arrangement $\mathcal{A}_{eg}$ in $O(n)$ cells such that in each $c \in \mathcal{A}_{eg}$ it is univocally determined for which, among the vertices $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$, x is the closest facility and y the second closest, and for which the opposite holds. More precisely, the arrangement $\mathcal{A}_{eg}$ is now induced only by the last equation in (18) for each $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$. For a given z, such an equation can be re-written as:

$$y = x + (b_g^z - b_e^z). \tag{25}$$

In particular, for the vertices z along the path $P(p(u),p(v))$, the intercept $(b_g^z - b_e^z)$ has a monotonic behavior and it decreases when z becomes closer to $p(v)$ while it increases when z becomes closer to $p(u)$. The same holds for all the vertices in $V(T_z)$ for which the first and the second closest facilities are the same as for z. This means that, in order to detect all the possible situations in which $L_1^z(x,y)$ and $L_2^z(x,y)$ remain univocally identified for all $z \in V(T_r)\backslash[V(T_u) \cup V(T_v)]$, it suffices to consider only the vertices z along the path $P(p(u),p(v))$. Therefore, we can consider only the set of $O(n)$ parallel lines corresponding to the equations (25) related to such vertices. Since $(b_g^z - b_e^z)$ are decreasing for z from $p(u)$ to $p(v)$, in the resulting arrangement $\mathcal{A}_{eg}$ we have $O(n)$ adjacent cells each one induced by two equations of type (25) corresponding to two consecutive vertices z_c^1 and z_c^2 in the path $P(p(u),p(v))$ with $d(z_c^1,p(v)) > d(z_c^2,p(v))$ and $L_1^{z_c^1}(x,y) = x$ and $L_1^{z_c^2}(x,y) = y$ (see, Figure 2). This guarantees that we can scan, sequentially and without repetitions, the adjacent cells of the arrangement $\mathcal{A}_{eg}$ by following the order given by the sequence of edges in the path $P(p(u),p(v))$ from $p(u)$ to $p(v)$.

The above properties can be exploited to provide an efficient algorithm for 2CUP on a tree. As before, for two fixed edges $e = (u,p(u))$ and $g = (v,p(v))$, we restrict to minimize the objective function in each cell $c \in \mathcal{A}_{eg}$ independently. The restricted objective function $f_c(x,y)$ can be computed in constant time relying to the quantities already introduced in Section 3.2. For the sake of presentation, we assume that in T_r edge $(v,p(v))$ is not in the path from $p(u)$ to r (see, Appendix 1 for the other case). Recalling that a cell c refers to a specific edge in $P(p(u),p(v))$, in the following formula we also assume that cell c corresponds to the edge $(z_c,p(z_c))$ lying in the path from $p(u)$ to r, and for which $L_1^{z_c}(x,y) = x$ and $L_1^{p(z_c)}(x,y) = y$ (the case in which z_c is in the path from $p(v)$ to r is similar). The objective function restricted to a cell c can be then computed as follows:

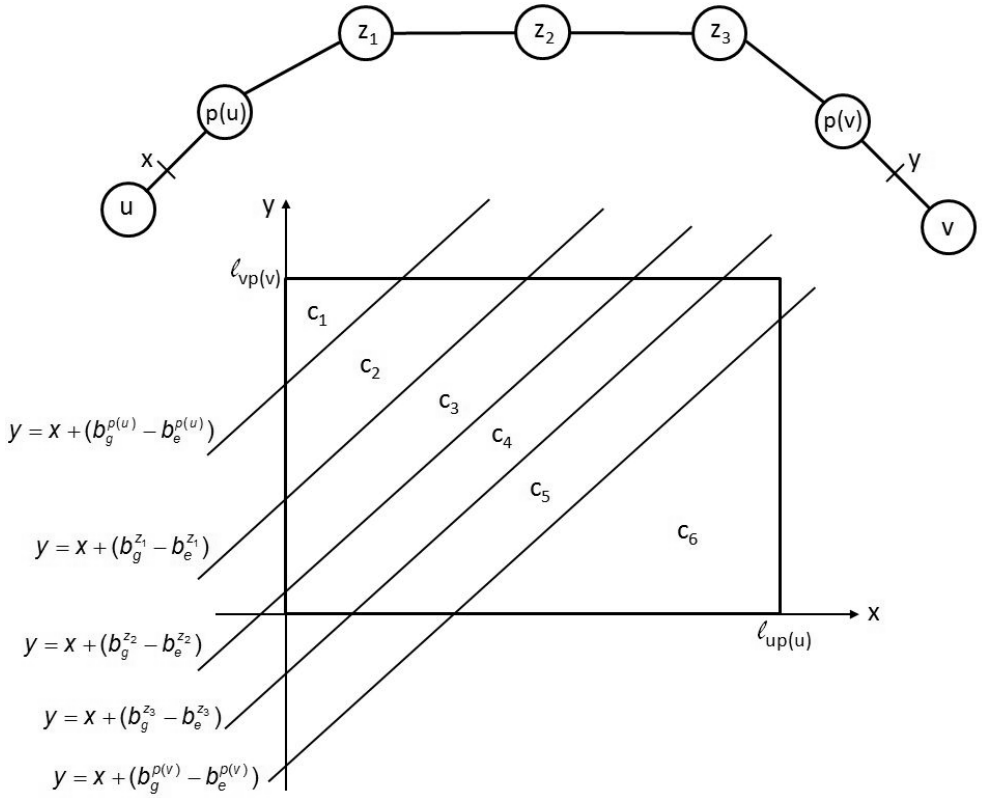

Figure 2: The arrangement $\mathcal{A}_{eg}$ for $e = (u, p(u))$ and $g = (v, p(v))$.

$$
\begin{aligned}
f_c(x,y) = \; & \big\{[D_B(u) + H_B(u)x] + \big[D_A(p(u)) - [D_A(z_c) + H_A(z_c)d(z_c,p(u))]\big] \\
& + [H_A(p(u)) - H_A(z_c)](\ell_{up(u)} - x) + \big[D_B(p(u)) - [D_B(u) + H_B(u)\ell_{up(u)}]\big] \\
& + [H_B(p(u)) - H_B(u)](\ell_{up(u)} - x)\big\}(1 - p_x^e) \\
& + \big\{D_B(z_c) + H_B(z_c)[d(z_c,p(v)) + (\ell_{vp(v)} - y)]\big\}p_x^e(1 - p_y^g) \\
& + \big\{[D_B(v) + H_B(v)y] + \big[D_A(p(v)) - [D_B(z_c) + H_B(z_c)d(z_c,p(v))]\big\} \\
& + [H_A(p(v)) - H_B(z_c)](\ell_{vp(v)} - y) + \big[D_B(p(v)) - [D_B(v) + H_B(v)\ell_{vp(v)}]\big] \\
& + [H_B(p(v)) - H_B(v)](\ell_{vp(v)} - y)\big\}(1 - p_y^g) \\
& + \big\{D_A(z_c) + H_A(z_c)[d(z_c,p(u)) + (\ell_{up(u)} - x)]\big\}p_y^g(1 - p_x^e) \\
& + \sum_{z \in V} h_z \beta_z p_x^e p_y^g.
\end{aligned}
\tag{26}
$$

The above function (26) is a polynomial which, as before, can be written in a general form by using suitable coefficients α_{ij}. As in Section 3.3, $f_c(x, y)$ can be minimized in constant time in

order to find the optimal pair of points (x_c, y_c) located in $c \in \mathcal{A}_{eg}$. We point out that when x and y belong to the same edge $e = (u, p(u))$, we can still rely on the same available quantities to compute in constant time the objective function restricted to edge e, say $f_e(x, y)$. In this case, taking into account the additional constraints $0 \leq x \leq y \leq \ell_{up(u)}$, $L_1^z(x, y)$ and $L_2^z(x, y)$ are well defined for all $z \in V$, and they do not change for x and y varying in e, so that $f_e(x, y)$ can be computed as follows:

$$\begin{aligned} f_e(x,y) = \; & \big\{[D_B(u) + H_B(u)x](1-p_x^e) + [D_B(u) + H_B(u)y]p_x^e(1-p_y^e)\big\} \\ & + \big\{[D_A(p(u)) + H_A(p(u))(\ell_{up(u)} - y)] + \big[D_B(p(u)) - [D_B(u) + H_B(u)\ell_{up(u)}]\big] \\ & + [H_B(p(u)) - H_B(u)](\ell_{up(u)} - y)\big\}(1-p_y^e) \\ & + \big\{[D_A(p(u)) + H_A(p(u))(\ell_{up(u)} - x)] + \big[D_B(p(u)) - [D_B(u) + H_B(u)\ell_{up(u)}]\big] \\ & + [H_B(p(u)) - H_B(u)](\ell_{up(u)} - x)\big\}p_y^e(1-p_x^e)\big\} \\ & + \sum_{z \in V} h_z \beta_z p_x^e p_y^e. \end{aligned} \tag{27}$$

The minimum of $f_e(x, y)$ subject to $0 \leq x \leq y \leq \ell_{up(u)}$ can be also found in constant time.

Theorem 4 *Algorithm 2MUP solves the 2-Median Unreliable Points problem on trees in $O(n^3)$ time.*

Proof:

For a given pair of edges $e = (u, p(u))$ and $g = (v, p(v))$ in T_r, function $f_c(x, y)$ can be computed in constant time for each cell $c \in \mathcal{A}_{eg}$. Computing the minimum of this function can be done in constant time, too. Since in $\mathcal{A}_{eg}$ we have $O(n)$ cells, finding the optimal pair of points in $[u, p(u)] \times [v, p(v)]$ requires $O(n)$ time. Considering all the possible pairs of edges e and g with $e \neq g$ we get an overall time complexity for solving 2MUP on trees equal to $O(n^3)$ time. Notice that the evaluation of the case when x and y belong to the same edge requires $O(n)$ time in total, and thus it does not affect the previous time complexity. □

As in Section 3.3, we analyze here the reduction in the time complexity for solving 2MUP on a tree when all its edges have the same length. In this case, for a given pair of edges $e = (u, p(u))$ and $g = (v, p(v))$ the arrangement $\mathcal{A}_{eg}$ of $[u, p(u)] \times [v, p(v)]$ is induced by the set of equations (25) and it always produces only two cells, whichever the value of n (namely, the two halves of $[u, p(u)] \times [v, p(v)]$ generated by the bisector $y = x$). Since for trees, function $f_c(x, y)$ can be computed in constant time, the optimal location of the two points x in e and y in g, with $e \neq g$, can be also found in constant time, implying an overall time complexity of $O(n^2)$.

4 The Minmax Unreliable Point location problems

In this section we focus on the k-unreliable points location problem based on the worse case approach and for $k = 1, 2$. We provide efficient algorithms both for general graphs and for tree networks.

4.1 The 1-Center Unreliable Point on graphs

The median criterion is adopted when one wants to minimize the average (expected) cost of the clients living in a given region to be served by a facility. However, it is well-known that, when locating, for instance, an emergency service point, much attention must be paid to those people living far from the facility in order to minimize the cost/time of these clients for reaching the point in case of emergency. This second criterion is usually modeled by considering the minimization of the maximum cost to serve a client (or a set of clients) from the located service point. In this paper we consider minimax location problems in the unreliable framework, that is, considering the possibility that a facility may fail. Referring to our probability model, given a weighted graph $G = (V, E)$, the *1-Center Unreliable Point* (1CUP) problem can be stated as follows referring to a point x located in a generic edge:

$$\min_{x \in A(G)} \quad \max_{z \in V} \{h_z d(z, x)(1 - p_x) + h_z \beta_z p_x\}. \tag{28}$$

Let us consider an edge $e = (u, v)$ and a possible location of a facility in a point $x \in [u, v]$ with the usual assumption that x also corresponds to the distance from vertex u. Given a vertex z, the expected cost of z w.r.t. x is:

$$f^z(x) = h_z d(z, x)(1 - p_x^e) + h_z \beta_z p_x^e \quad = h_z(a_e^z x + b_e^z)(1 - p_x^e) + h_z \beta_z p_x^e. \tag{29}$$

and therefore, 1CUP restricted to edge $e = (u, v)$ can be stated as follows

$$\min_{x \in [u,v]} \quad f_e(x) = \max_{z \in V} f^z(x). \tag{30}$$

Let us observe that for the linear function $a_e^z x + b_e^z$, similarly to the median case, we have:

$$a_e^z x + b_e^z = \begin{cases} x + d(z, u) & \text{if } x \le x_z \\ -x + [d(z, v) + \ell_{uv}] & x > x_z. \end{cases} \tag{31}$$

with x_z the breakpoint generated by vertex z in (u, v).

We note that for every edge (u, v), the distances $d(z, u)$ and $d(z, v)$, for all $z \in V$, are known once all the distances between every pair of vertices of G have been computed. In a general graph, this can be done in advance in $O(n^3)$ time by using the Floyd-Warshall algorithm. However, as we will see, this does not affect the overall time complexity of the algorithm for solving 1CUP. For each vertex z its expected cost function is represented by two arcs of quadratic functions in the two intervals $[u, x_z]$ and $[x_z, v]$.

$$f^z(x) = h_z a_e^z \big(\tfrac{p_u - p_v}{\ell_{uv}}\big) x^2 + h_z \big[a_e^z(1 - p_u) + \big(\tfrac{p_u - p_v}{\ell_{uv}}\big)(b_e^z + \beta_z)\big] x + h_z \big[b_e^z(1 - p_u) + h_z \beta_z p_u\big]. \tag{32}$$

Different from the median case, for solving 1CUP restricted to edge (u, v) we have now to minimize in $[u, v]$ the upper envelope of a set of quadratic functions each referring to a vertex $z \in V$. Finding the upper envelope in $[u, v]$ provides a set of breakpoints $\{x_1, \ldots, x_q\}$ that partition $[u, v]$ into $q - 1$ intervals $[x_i, x_{i+1}]$. Within each interval $[x_i, x_{i+1}]$ the maximum among the $f^z(x)$, $z \in V$, is univocally determined and denoted by $f_i(x)$, so that in $[x_i, x_{i+1}]$ we have to find the minimum of $f_i(x)$ w.r.t. x.

To compute the upper envelope of arcs of quadratic functions, we refer to a result given in [32], that we report below for the sake of completeness:

Theorem 5 *([32], Theorem 6.5). Given a set of n x_1-monotone Jordan arcs with at most s intersections between any pair of arcs, its upper envelope has an $O(\lambda_{s+2}(n))$ complexity, and it can be computed in $O(\lambda_{s+1}(n)\log n)$ time.*

Function $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order s on n symbols (see, Chapter 3 in [32] for the exact definitions and properties of the functions $\lambda_s(n)$). The exact formulas for the function $\lambda_s(n)$ are reported below:

- for even values of $s \geq 4$ we have: $\lambda_s(n) = n \cdot 2^{\frac{1}{t!}\alpha(n)^t(1+o(1))}$, with $t = \frac{s-2}{2}$;
- for odd values of $s \geq 5$ we have: $\lambda_s(n) = n \cdot 2^{\frac{1}{t!}\alpha(n)^t \log(\alpha(n))(1+o(1))}$, with $t = \frac{s-3}{2}$;

where $\alpha(n)$ is the inverse of the Ackermann function. Since we consider intersections of quadratic functions in one variable, in our case s is at most 4. We also note that the number q of breakpoints generated in an edge is $O(\lambda_6(n))$. Basing on the discussion provided in this section, in the following we describe the algorithm for solving 1CUP on general graphs.

Algorithm 1CUP

1. For each edge $e = (u, v)$ of E
   - 1.1 compute the quadratic function $f^z(x)$ for each vertex $z \in V$
   - 1.2 compute the upper envelope of the quadratic functions $f^z(x)$, $z \in V$
   - 1.3 identify the intervals $[x_i, x_{i+1}]$, $i = 1, \ldots, q-1$
   - 1.4 for each interval $[x_i, x_{i+1}]$, $i = 1, \ldots, q-1$
     - 1.4.1 let $f_i(x)$ be the arc of the quadratic function corresponding to the upper envelope in $[x_i, x_{i+1}]$
     - 1.4.2 let $\bar{x}_i$ be the point that provides the minimum of $f_i(x)$ in $[x_i, x_{i+1}]$.
   - 1.5 let x_e be such that $f(x_e) = \min_{i:[x_i,x_{i+1}]\subset[u,v]} f_i(\bar{x}_i)$
2. let x^* be such that $f(x^*) = \min_{e\in E} f(x_e)$

Theorem 6 *Algorithm 1CUP solves the 1-Center Unreliable Point problem on general graphs in $O(m\lambda_5(n)\log n)$ time.*

Proof:

Given an edge (u, v), computing the quadratic functions $f^z(x)$, $\forall\ z \in V$, requires $O(n)$ time, provided that we have already computed the distances between all pairs of vertices in G. Computing the upper envelope of the quadratic functions $f^z(x)$, $z \in V$, can be done in $O(\lambda_5(n)\log n)$ time. Notice that the procedure for finding the upper envelope provides us not only the interval $[x_i, x_{i+1}]$, $i = 1, \ldots, q-1$, but also the arc of quadratic function $f_i(x)$ corresponding to the upper envelope of the $f^z(x)$ in $[x_i, x_{i+1}]$. Minimizing $f_i(x)$ on $[x_i, x_{i+1}]$ requires constant time, so that the point x_e can be found in $O(\lambda_6(n))$ time. Hence, repeating the above steps for all the edges leads to an overall complexity of $O(m\lambda_5(n)\log n)$ time. □

In case of tree networks, the same approach as the one presented above can be adopted, but without too much improvement in the time complexity w.r.t the previous case. In fact, following this approach, we can efficiently exploit the tree structure only in the computation of the distances between every pair of vertices in the tree which requires $O(n^2)$ time instead of $O(n^3)$. Hence, for tree networks, the overall time complexity of the above algorithm amounts to $O(n\lambda_5(n)\log n)$.

4.2 The 2-Center Unreliable Points on graphs

In this section we consider the problem of locating 2 points minimizing the center criterion with probability of disruption. Given a weighted graph $G = (V, E)$, and considering the expected costs $f^z(x, y)$, $z \in V$, introduced in Section 3.3, the *2-Center Unreliable Points* (2CUP) problem can be formulated as follows:

$$\min_{x,y \in A(G)} \quad f(x, y) := \max_{z \in V} f^z(x, y). \tag{33}$$

We solve problem (33) by repeatedly minimizing $f(x, y)$ restricted to every pair of edges $e = (u, v)$ and $g = (r, s)$. The procedure for finding the minimum of $f(x, y)$ in the subspace $[u, v] \times [r, s]$ starts by determining the arrangement $\mathcal{A}_{eg}$ of $[u, v] \times [r, s]$ induced by linear equations (18). The difficult issue in 2CUP is that we have to find the upper envelope of functions $f^z(x, y)$, $z \in V$, in each cell $c \in \mathcal{A}_{eg}$. Actually, function $f^z(x, y)$ restricted to $c \in \mathcal{A}_{eg}$ is defined as follows:

$$f^z_c(x, y) = \begin{cases} h_z\big[d(z, x)(1 - p^e_x) + d(z, y)p^e_x(1 - p^g_y) + \beta_z p^e_x p^g_y\big] & \text{if} \quad d(z, x) \le d(z, y) \\ \\ h_z\big[d(z, y)(1 - p^g_y) + d(z, x)p^g_y(1 - p^e_x) + \beta_z p^e_x p^g_y\big] & \text{otherwise.} \end{cases} \tag{34}$$

We recall that in each cell $c \in \mathcal{A}_{eg}$ we are able to determine which of the two points x and y is the closest to vertex z, so that function $f^z_c(x, y)$ is well identified. The above function has a compact expression of the following type:

$$f^z_c(x, y) = \begin{cases} \alpha^z_{20}x^2 + \alpha^z_{12}xy^2 + \alpha^z_{11}xy + \alpha^z_{10}x + \alpha^z_{01}y + \alpha^z_{00} & \text{if} \quad d(z, x) \le d(z, y) \\ \\ \alpha^z_{02}y^2 + \alpha^z_{21}x^2y + \alpha^z_{11}xy + \alpha^z_{10}x + \alpha^z_{01}y + \alpha^z_{00} & \text{otherwise.} \end{cases} \tag{35}$$

Thus, in each cell c we have $O(n)$ (x, y)-*monotone third degree polynomial surface patches*. A surface patch is (x_1, x_2)-*monotone* in $\mathbb{R}^3$ if every line parallel to the x_3-axis intersects the surface patch in at most one point (see, [31] for more details). In a given cell $c \in \mathcal{A}_{eg}$ we have to compute the upper envelope of these $O(n)$ functions. The so called *maximization diagram* of c, $\mathcal{D}_c$, is then obtained by projecting the upper envelope of the surface patches onto the cell c. This provides a decomposition of c into connected relatively open semialgebraic sets, such that each set is a maximal connected portion of c. According to [31], the maximization diagram can be computed under some specific assumptions on the surface patches which hold also for our set of functions $f^z_c(x, y)$, $z \in V$.
In [31], the *combinatorial complexity* of the upper (lower) envelope of a set of $O(n)$ surface patches is defined to be the number of (open semialgebraic) sets of all dimensions of $\mathcal{D}_c$. In our case it amounts to $O(n^{2+\epsilon})$, for any $\epsilon > 0$ (see, [9, 23, 31]). Moreover, computing the upper envelope of $O(n)$ surface patches in three dimensions requires $O(n^{2+\epsilon})$ time [31].

Consider the maximization diagram $\mathcal{D}_c$, and let γ be a maximal connected portion of $\mathcal{D}_c$ of dimension two (i.e., a surface area of $\mathcal{D}_c$). We denote by $f^z_{c,\gamma}(x,y)$ the function corresponding to the upper envelope restricted to γ. We observe that these functions are univocally determined by the procedure for obtaining the maximization diagram itself.
Then, we can solve the problem restricted to $\gamma \in \mathcal{D}_c$ for each γ separately. For a given $\gamma \in \mathcal{D}_c$, this (restricted) problem can be solved in constant time by first finding the stationary points of $f^z_{c,\gamma}(x,y)$ and checking whether they belong to γ. The next problem consists of minimizing $f^z_{c,\gamma}(x,y)$ w.r.t. a set γ of dimension one. The upper envelope procedure presented in [23, 31] provides also the surfaces that produced such sets γ. Thus, the problem consists of minimizing one of such surfaces, say $f^z_{c,\gamma}(x,y)$, subject to the polynomial functions that define γ. In fact, we have to minimize $f^z_{c,\gamma}(x,y)$ subject to a polynomial function of fixed (and constant) degree δ. This can be done by resorting to solve the system generated by the Karush-Kuhn-Tucker (KKT) conditions. Note that in our technique we refer to only algebraic (arithmetic and comparison) operations over the field extension $\mathcal{F}$ of the rationals generated by the coefficients of the nonlinear equations derived from the KKT conditions. Since, even in the simplest cases, it is possible that the numerical solution to that system does not lie in $\mathcal{F}$, we describe the solution indirectly. By a polynomial predicate we refer here to a predicate whose atomic formulae are polynomial equalities or inequalities of constant size. That is, they depend on the number of equations (three in our case) and the degree of the equations which are bounded above by δ. Thus, the output of the algorithm is a polynomial predicate, each of its solutions is a candidate solution for our problem. Such a solution can be computed to any required approximation, in time polynomial in the maximum degree of the involved polynomials and the bit-size of their coefficients, therefore in constant time since in our framework those parameters are constant [29]. Hence, the minimization of $f^z_{c,\gamma}(x,y)$ can be done in constant time for each $\gamma \in \mathcal{D}_c$.

We can now describe the algorithm for solving 2CUP.

Algorithm 2CUP

1. For each pair of edges $e=(u,v)$ and $g=(r,s)$ of E
   - 1.1 compute the arrangement $\mathcal{A}_{eg}$ of $[u,v]\times[r,s]$
   - 1.2 for each cell $c \in \mathcal{A}_{eg}$
     - 1.2.1 compute the maximization diagram $\mathcal{D}_c$ and the functions $f^z_{c,\gamma}(x,y)$, for all $\gamma \in \mathcal{D}_c$ and $z \in V$
     - 1.2.2 for each set $\gamma \in \mathcal{D}_c$
       - 1.2.2.1 compute the minimum (x_γ, y_γ) of the function $f^z_{c,\gamma}(x,y)$
     - 1.2.3 let (x_c,y_c) be such that $f^z_c(x_c,y_c) = \min_{\gamma\in\mathcal{D}_c} f^z_{c,\gamma}(x_\gamma,y_\gamma)$
   - 1.3 let (x_{eg},y_{eg}) be such that $f(x_{eg},y_{eg}) = \min_{c\in\mathcal{A}_{eg}} f^z_c(x_c,y_c)$
2. let (x^*,y^*) be such that $f(x^*,y^*) = \min_{e,g\in E}(x_{eg},y_{eg})$

Theorem 7 *Algorithm 2CUP solves the 2-Center Unreliable Points problem on general graphs in $O(m^2n^{4+\epsilon})$ time.*

Proof:

For a given pair of edges $e=(u,v)$ and $g=(r,s)$ computing the arrangement $\mathcal{A}_{eg}$ of $[u,v]\times[r,s]$ takes $O(n^2)$ time. In each cell $c \in \mathcal{A}_{eg}$ the functions $f^z_c(x,y)$ can be computed in

$O(n)$ time for all $z \in V$. The maximization diagram $\mathcal{D}_c$ requires $O(n^{2+\epsilon})$ time and, for each $\gamma \in \mathcal{D}_c$, it produces the function $f^z_{c,\gamma}(x,y)$. As discussed before, the minimization of the function $f^z_{c,\gamma}(x,y)$ takes constant time in each subset γ, so that the point (x_c, y_c) can be found in $O(n^{2+\epsilon})$ time which is also the combinatorial complexity of the maximization diagram. Since the number of cells $c \in \mathcal{A}_{eg}$ is $O(n^2)$, the pair (x_{eg}, y_{eg}) can be found in $O(n^{4+\epsilon})$ time. Hence, repeating all these steps for all pairs of edges in E provides an overall time complexity of $O(m^2 n^{4+\epsilon})$. $\square$

We observe that we can use Algorithm 2CUP also for solving 2CUP on trees. In this case, for a pair of edges e and g in a rooted tree T_r, the arrangement $\mathcal{A}_{eg}$ produces only $O(n)$ cells c (see, Section 3.4). Hence, for 2CUP on trees we have an overall time complexity of $O(n^{5+\epsilon})$.

Additionally, by similar arguments as those at the end of Sections 3.3 and 3.4, we can improve the time complexity of our approach to $O(m^2 n^{2+\epsilon})$ time when the edges of a general graph G are of equal length, and to $O(n^{4+\epsilon})$ time for trees with equal edge lengths.

We conclude this section by considering the probability model introduced in [1] and related to the possibility of facility design failure. In that case each facility is characterized by its own probability of failure which is independent of its location. Let $\mathtt{x} = (x_1, \ldots, x_k)$ be the location of the facilities, and $z \in V$ be a generic vertex for which we denote by $z(1), \ldots, z(k)$ the facilities sorted by their distance w.r.t. z. Then, under the model in [1], the probability of failure when z visits its i-th closest facility is given by $p_{z(i)}$, $i = 1, \ldots, k$. Our aim is to show that our technique can be used also in this probability framework for solving the k-Center Unreliable Points location problem, for a fixed $k \geq 2$.

Consider a set of k distinct edges $e_1, \ldots, e_k$ of the graph G, and compute the arrangement $\mathcal{A}_k$ of the subspace of dimension k generated by this set of edges. This arrangement can be found by solving a system of $O(nk^2)$ linear equations analogous to (18) and provides a subdivision of the subspace induced by the edges $(e_1, \ldots, e_k)$ in $O(n^k k^{2k})$ cells (see, [5]). Let $\mathtt{x} = (x_1, \ldots, x_k)$ be the location of the facilities along the edges $e_1, \ldots, e_k$. After having computed the arrangement $\mathcal{A}_k$, in any cell $c \in \mathcal{A}_k$ the distance function of each vertex $z \in V$ can be easily computed. Actually, each cell c provides a fixed distance ordering of each vertex z w.r.t. the set $\mathtt{x}$. Let $x_{z(1)}, \ldots, x_{z(k)}$ denote the location of the facilities ordered w.r.t. z in c. The cost function of z restricted to c is given by:

$$f^z_c(\mathtt{x}) = h_z[d(z, x_{z(1)})(1 - p_{z(1)}) + \ldots + d(z, x_{z(k)}) \prod_{i=1}^{k-1} p_{z(i)}(1 - p_{z(k)})] + h_z \beta_z \prod_{i=1}^{k} p_{z(i)}.$$

Since in the probability framework of [1], the probabilities are not functions of $\mathtt{x}$, once the relative order of distances to z does not change, the above function can be re-written as follows:

$$f^z_c(\mathtt{x}) = a^z_c \mathtt{x} + b^z_c.$$

Thus, in a given cell $c \in \mathcal{A}_k$ we have to compute the maximization diagram $\mathcal{D}_c$ of a set of n hyperplanes. This can be done in $\Theta(n^{\lfloor \frac{k}{2} \rfloor})$ time (see, [23]), thus obtaining a decomposition of c in $\Theta(n^{\lfloor \frac{k}{2} \rfloor})$ sets γ. We have now to find the minimum of a linear function $f^z_{c,\gamma}(\mathtt{x})$ on a polytope γ in $\mathbb{R}^{k-1}$ (with k fixed). The ellipsoid method [6] guarantees that one can find a solution in time polynomial in n, say $poly(n)$. Hence, by applying the same algorithm that for 2CUP, the k-Center Unreliable Points location problem with probabilities depending on the facility design can be solved in $O(m^k n^k k^{2k} n^{\lfloor \frac{k}{2} \rfloor} poly(n))$ time.

5 The CentDian Unreliable Points on graphs

In this section, we consider unreliable points location on graphs minimizing a convex combination of the center and the median objective functions. We analyze only the (more difficult) 2-*CentDian Unreliable Points* problem on a general graph (2CDUP) since it is clear that the 1CDUP problem can be solved both on general graphs and on tree networks by adopting the same techniques used for the 1MUP and 1CUP problems (see, Table 1 for the overall time complexity of 1CDUP on graphs and trees). For a given $\alpha \in [0,1]$, we introduce the centdian objective function:

$$\min_{x,y\in A(G)} \alpha \sum_{z\in V} h_z\big[d(z,L_1^z(x,y))(1-p_{L_1^z(x,y)}) + d(z,L_2^z(x,y))p_{L_1^z(x,y)}(1-p_{L_2^z(x,y)}) + \beta_z p_{L_1^z(x,y)}p_{L_2^z(x,y)}\big]$$

$$+(1-\alpha)\max_{z\in V}\{h_z\big[d(z,L_1^z(x,y))(1-p_{L_1^z(x,y)}) + d(z,L_2^z(x,y))p_{L_1^z(x,y)}(1-p_{L_2^z(x,y)}) + \beta_z p_{L_1^z(x,y)}p_{L_2^z(x,y)}\big]\}. \tag{36}$$

We can apply the same methodology presented in the previous sections and based on the solution of the problem restricted to a pair of edges $e=(u,v)$ and $g=(r,s)$. It is easy to prove that the overall time complexity is determined by the minimization of the function referring to the center criterion, which is the most expensive operation. In fact, let $[u,v]\times[r,s]$ be the subspace induced by edges e and g, and let $\mathcal{A}_{eg}$ be the corresponding arrangement into $O(n^2)$ cells. Denote by $f_c(x,y)$ the median function restricted to a cell $c\in\mathcal{A}_{eg}$ (see, equation (19)). Consider now the center function in the objective (36), in order to minimize this function in a cell $c\in\mathcal{A}_{eg}$, we have to further decompose c to obtain the maximization diagram $\mathcal{D}_c$ as in Section 4.2. Adopting the same notation, for 2CDUP the minimization problem restricted to $\gamma\in\mathcal{D}_c$ is now the following:

$$\min\big\{\alpha f_c(x,y) + (1-\alpha)f^z_{c,\gamma}(x,y)\big\}. \tag{37}$$

We notice that in each cell $c\in\mathcal{A}_{eg}$ the function $f_c(x,y)$ can be computed once, and remains the same for each $\gamma\in\mathcal{D}_c$. This implies that, when analyzing a cell c, the greatest effort relates to the computation of the maximization diagram $\mathcal{D}_c$, since, the minimization of the function $f_c(x,y)+(1-\alpha)f^z_{c,\gamma}(x,y)$ over a set $\gamma\in\mathcal{D}_c$ can be done in constant time. As before, this computation requires $O(n^{2+\epsilon})$ time for each cell $c\in\mathcal{A}_{eg}$. The algorithm for solving 2CDUP follows.

Algorithm 2CDUP

1. For each pair of edges $e=(u,v)$ and $g=(r,s)$ of E
   - 1.1 compute the arrangement $\mathcal{A}_{eg}$
   - 1.2 for each cell $c\in\mathcal{A}_{eg}$
     - 1.2.2 compute $f_c(x,y)$, the maximization diagram $\mathcal{D}_c$, and the functions $f^z_{c,\gamma}(x,y)$, for all $\gamma\in\mathcal{D}_c$, and for all $z\in V$
     - 1.2.3 for each subset $\gamma\in\mathcal{D}_c$
       - 1.2.2.1 compute the minimum (x_γ,y_γ) of the function $f_c(x,y)+(1-\alpha)f^z_{c,\gamma}(x,y)$
     - 1.2.3 let (x_c,y_c) be such that $f_c(x_c,y_c)+(1-\alpha)f^z_{c,\gamma}(x_c,y_c) := \min_{\gamma\in\mathcal{D}_c} f_c(x_\gamma,y_\gamma)+(1-\alpha)f^z_{c,\gamma}(x_\gamma,y_\gamma)$

1.3 let (x_{eg}, y_{eg}) be such that $f(x_{eg}, y_{eg}) := \min_{c \in \mathcal{A}_{eg}} f_c(x_c, y_c) + (1-\alpha) f^z_{c,\gamma}(x_c, y_c)$

2. let (x^*, y^*) be such that $f(x^*, y^*) := \min_{e,g \in E} f(x_{eg}, y_{eg})$

Theorem 8 *Algorithm 2CDUP solves the 2-Center Unreliable Points problem on general graphs in $O(m^2 n^{4+\epsilon})$ time.*

The proof follows from the complexity results discussed in Theorems 3 and 7.

In the special case of 2CDUP on trees the overall time complexity is $O(n^{5+\epsilon})$. Furthermore, these complexities reduce to $O(m^2 n^{2+\epsilon})$ and $O(n^{4+\epsilon})$ for graphs and trees with equal edge lengths, respectively.

6 Concluding Remarks

We would like to remark that the assumption made on the shape of the function modeling the probability of failure on the edges of the network is not restrictive at all because this probability framework allows representing - or, at least, approximating up to any degree of accuracy - any probability function on the edges of the graph. Indeed, let us assume that we restrict ourselves to an edge $e \in E$. If the function is defined in e by a piecewise linear function, then one has only to consider the breakpoints of this function as new pseudonodes on V, giving rise to a new augmented graph $G' = (V', E')$ to which the framework of this paper applies directly. On the other hand, if the probability function is not piecewise linear, we can approximate it by a new piecewise linear function with as many breakpoints as needed to ensure the required accuracy. After such an approximation, we are in the previous case and the same approach is valid.

Another natural extension of the results of this paper is to apply the reliability approach, under our probabilistic paradigm, to the more general class of *ordered median functions* [14, 17, 21, 22]. The reader may realize that all the results in this paper extend to that class of objective functions, provided that sufficiently refined arrangements are considered. Actually, one should guarantee that within each cell of the arrangement the relative order of the distances of any point in the cell w.r.t. the vertices of the network does not change. This is always possible, for instance for the single facility case, refining the arrangement $\mathcal{A}_e$ by adding the set of linear functions $d(z, x) = d(z', x)$ for any pair z and z', with $z \neq z' \in V$. The 2-facility case is handled similarly.

To conclude, we also point out that most of our algorithms can be extended, with a polynomial worse case complexity to the case of $k > 2$ facilities, provided that k is fixed. In this case, we would need to construct arrangements on the k-dimensional space but still our tools are applicable and the results will extend under the Real Arithmetic Mode (RAM) of computation (see, [29]) by using techniques similar to the ones applied in Section 4.2.

Appendix

Referring to Section 3.4, for a rooted tree T_r, we present the expression of the function $f_c(x, y)$ when, given two edges $e = (u, p(u))$ and $g = (v, p(v))$, the vertex v is in the path from $p(u)$ to the root r, with $r \neq p(v)$. The objective function restricted to cell $c \in \mathcal{A}_{eg}$ is given by:

$$
\begin{aligned}
f_c(x,y) \;=\; & \big\{[D_B(u)+H_B(u)x]+\big[D_A(p(u))-[D_A(z_c)+H_A(z_c)d(z_c,p(u))]\big] \\
& +[H_A(p(u))-H_A(z_c)]\big(\ell_{up(u)}-x\big)+\big[D_B(p(u))-[D_B(u)+H_B(u)\ell_{up(u)}]\big] \\
& +[H_B(p(u))-H_B(u)](\ell_{up(u)}-x)\big\}(1-p_x^e) \\
& +\big\{D_B(z_c)+H_B(z_c)\big[d(z_c,v)+y\big]\big\}p_x^e(1-p_y^g) \\
& +\big\{[D_A(p(v))+H_A(p(v))(\ell_{vp(v)}-y)]+\big[D_B(p(v))-[D_B(v)+H_B(v)\ell_{vp(v)}]\big] \\
& +[H_B(p(v))-H_B(v)](\ell_{vp(v)}-y)+\big[D_B(v)-[D_B(z_c)+H_B(z_c)d(z_c,v)]\big] \\
& +[H_B(v)-H_B(z_c)]y\big\}(1-p_y^g) \\
& +\big\{D_A(z_c)+H_A(z_c)\big[d(z_c,p(u))+\ell_{up(u)}-x\big]\big\}p_y^g(1-p_x^e) \\
& +\sum_{z\in V} h_z\beta_z p_x^e p_y^g.
\end{aligned}
\tag{38}
$$

Acknowledgements

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 trough grant number MTM2007-67433-C02-01; the first author is also grateful to the Junta de Andalucía for partially supporting this research trough grant number FQM-5849.

References

[1] Berman, O., Krass, D., and Menezes, M. (2007). Facility reliability issues in network p-median problems: Strategic centralization and co-location effects. *Operations Research*, 55(2):332–350.

[2] Cui, T., Ouyang, Y., and Shen, Z. (2010). Reliable facility location design under the risk of disruptions. *Operations Research*, 58:998–1011.

[3] Drezner, Z. (1987). Heuristic solution methods for two location problems with unreliable facilities. *Journal of the Operations Research Society*, 38:509–514.

[4] Drezner, Z., and Hamacher, H., *Facility Location: Applications and theory*. Springer, 2004.

[5] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*. Springer, 1987.

[6] Grötschel, M., Lovász, L., and Schrijver, A., *Geometric Algorithms and Combinatorial Optimization*. Springer, 1993.

[7] Hakimi, S.L. (1964). Optimum location of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459.

[8] Hakimi, S.L., Schmeichel, E.F., and Labbè, M. (1993). On locating path-or tree shaped facilities on networks. *Networks*, 23:543–555.

[9] Halperin, D., and Sharir, M. (1994). New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete and Computational Geometry*, 12:313–326.

[10] Herivin, H., and Mahjoub, A. (2005). Design of survivable networks: A survey. *Networks*, 46(1):1–21.

[11] Hollander, Y., and Prashker, J. (2006). The applicability of non-cooperative game theory in transport analysis. *Transportation*, 33(5):481–496.

[12] Hooker, J.N., Garfinkel, R.S., and Chen, C.K. (1991). Finite dominating sets for network location problems. *Operations Research*, 39(1):100–118.

[13] Kalcsics, J. (2011). The multi-facility median problem with pos/neg weights on general graphs. *Computer & Operations Research*, 38:674–682.

[14] Kalcsics, J., Nickel, S., and Puerto, J. (2003). Multi-facility ordered median problems: A further analysis. *Networks*, 41(1):1–12.

[15] Kalcsics, J., Nickel, S., Puerto, J., and Rodríguez-Chía, A.M. (2010). Distribution Systems Design With Role Dependent Objectives. *European Journal of Operational Reserch*, 202:491-501.

[16] Kalcsics, J., Nickel, S., Puerto, J., and Rodríguez-Chía, A.M. (2012). Several 2-facility location problems on networks with equity objectives. *Working Paper*.

[17] Kalcsics J., Nickel S., Puerto J., and Tamir, A. (2002). Algorithmic results for ordered median problems defined on networks and the plane. *Operations Research Letters*, 30:149–158.

[18] Kariv, O., and Hakimi, S.L. (1979). An algorithmic approach to network location problems. I: The p-centers. *SIAM Journal on Applied Mathematics*, 37:513–538.

[19] Kariv, O., and Hakimi, S.L. (1979). An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, 37:539–560.

[20] Kim, T.U., Lowe, T.J., Tamir, A., and Ward, J.E. (1996). On the location of a tree-shaped facility. *Networks*, 28:167–175.

[21] Nickel, S., and Puerto, J. (1999). A unified approach to network location problems. *Networks*, 34:283–290.

[22] Nickel, S., and Puerto, J., *Facility Location - A Unified Approach*. Springer Verlag, 2005.

[23] Meyerovitch, M. (2006). Robust, Generic, and Efficient Construction of Envelopes of Surfaces in Three-Dimensional Space. *Proc. of the 14th Annual European Symposium on Algorithms (ESA)*, 792–803, Zurich, Switzerland.

[24] Puerto, J., Ricca, F., and Scozzari, A. (2009). The continuous and discrete path-variance problems on trees. *Networks*, 53:221–228.

[25] Puerto, J., Ricca, F., and Scozzari, A. (2009). Extensive facility location problems on networks with equity measures. *Discrete Applied Mathematics*, 157:1069–1085.

[26] Puerto, J., Ricca, F., and Scozzari, A. (2012). Range minimization problems in path-facility location on trees. *Discrete Applied Mathematics*, 160:2294–2305.

[27] Puerto, J., Ricca, F., and Scozzari, A. (2012). Reliability problems related to path-shaped facility location on trees. *Technical Report 24/2012*, Department of Statistical Sciences, Sapienza University of Rome.

[28] Puerto, J., and Rodríguez-Chía, A.M. (2005). On the exponential cardinality of finite dominating sets for the multifacility ordered median problem. *Operations Research Letters*, 33:641–651.

[29] Renegar, J. (1988). A faster PSPACE algorithm for deciding the existential theory of the reals. *Proc. 29th IEEE Symposium on Foundations of Computer Science*, pp 291–295.

[30] Rodríguez-Chía, A.M., Puerto, J., Pérez-Brito, D., and Moreno, J.A. (2005). The p-facility ordered median problem on networks. *TOP*, 13:105–126.

[31] Sharir, M. (1994). Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete and Computational Geometry*, 12:327–345.

[32] Sharir, M., and Agarwal, P.K., *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press, 1995.

[33] Shen, Z., Shan, R., and Shang, J. (2011). The reliable facility location problem: formulations, heuristics and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482.

[34] Snyder, L., and Daskin, M.S. (2005). Reliability models for facility location: The expected failure cost case. *Transportation Science*, 39(3):400–416.

[35] Zhang, Y., Berman, O., and Verter, V. (2009). Incorporating congestion in preventive healthcare facility network design. *European Journal of Operational Research*, 198(3):922–935.