

Robust 1-median location problem on a tree

Rim KALAI*, Mohamed Ali ALOULOU*, Philippe VALLIN* and Daniel VANDERPOOTEN*

*Paris-Dauphine University/LAMSADE

Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16

Email: kalai,aloulou,vallin,vdp@lamsade.dauphine.fr

Abstract—In combinatorial optimization, and particularly in location problems, the most used robustness criteria rely either on maximal cost or on maximal regret. However, it is well known that these criteria are too conservative. In this paper, we present a new robustness approach, called *lexicographic α -robustness*, which compensates for the drawbacks of the criteria based on the worst case. We apply this notion to the 1-median location problem under uncertainty and we give a polynomial algorithm to determine robust solutions in the case of a tree graph.

Index Terms—Robustness, 1-median location problem, minmax cost/regret, scenario-based uncertainty.

I. INTRODUCTION

ROBUSTNESS analysis looks for solutions in a context where the imprecise, uncertain and generally badly known parameters of a problem make inappropriate the search of optimal solutions [19] [23]. Unlike deterministic or stochastic approaches which are aimed at determining the best solution for a certain instance of values (or scenario), robust approaches try to find a solution or a set of solutions that is acceptable for any considered scenario. In combinatorial optimization and particularly in location problems, the most used robustness criteria rely either on the maximal cost or on the maximal regret [14]: a robust solution is one that minimizes the maximal cost or regret among all scenarios. Nevertheless, grasping the notion of robustness through only one measure (the maximal cost or regret) is questionable, since this often leads to favor only the worst case scenario. Furthermore, no tolerance is considered in this measure.

These two drawbacks of the criteria founded on the worst case suggest considering alternative robustness criteria. In the case of deterministic public location problems, Ogryczak departs from considering only the worst case by introducing the notion of *lexicographic minimax* [16]. In this paper, we use and extend this idea in order to define a new robustness approach when the set of scenarios is finite and propose a polynomial time

algorithm to compute the corresponding robust solutions for the 1-median problem on a tree.

Our paper is organized as follows. In section 2, we define the *1-median problem* and review the main works on robustness for this problem. In section 3, we introduce a relation called *α -leximax*, and use it to define a set of robust solutions. We also present a general algorithm which computes this robust set when the set of solutions is finite and apply it to the *vertex 1-median problem*. In section 4, we apply our robustness approach to the *1-median problem* on a tree for which we present a specific algorithm that finds the robust points of the tree. In a final section, we summarize the important points of this work and suggest some perspectives.

II. LITERATURE REVIEW

Network location problems are aimed at locating new facilities in order to meet the demand of a certain number of customers [8]. Demand and travel between demand sites and facilities are assumed to occur only on a graph $G = (V, E)$ composed of a set $V = \{v_i, i = 1, \dots, n\}$ of n nodes (or vertices) and a set E of m edges. The length of each edge (v_i, v_j) , i.e. the distance between site v_i and site v_j , is denoted c_{ij} . We assume that demands occur only at the nodes of the network and that they can be characterized by a weight vector $W = (w_1, w_2, \dots, w_n)$ where w_i is the weight associated with node v_i for $i = 1, \dots, n$.

The *absolute 1-median problem* is to locate the *absolute median* of a graph G , that is the point of G which minimizes the total weighted distance to all nodes of the graph. A *point* of the graph corresponds either to a node or to any point on an edge. Let us denote $d(a, b)$ the minimum distance between two points a and b of G . The 1-median problem is formulated as follows :

$$\min_{x \in G} C(x) = \sum_{i=1}^n w_i d(x, v_i) \quad (1)$$

The Hakimi property stipulates that an absolute median of a graph is always at a vertex of the graph [12].

The absolute 1-median problem is then equivalent to the *vertex 1-median problem* which can be written as :

$$\min_{v \in V} C(v) = \sum_{i=1}^n w_i d(v, v_i) \quad (2)$$

Consequently, given all distances $d(v_i, v_j)$, the problem can be solved in linear time by enumerating and evaluating the n possible solutions.

Deterministic approaches assume that the problem parameters (node weights and edge lengths) are fixed and well known. In practice, however, it often appears difficult to determine in a reliable and irrevocable way all the data of a given problem. The decision-maker is often confronted with uncertainty that makes the deterministic reasoning inappropriate. Uncertainty situations are divided into two classes: if it exists a perfectly known probability distribution on the set of the nature states, we are in a *risk situation*. Otherwise, it is impossible to allocate probabilities to the possible outcomes of a decision and we say that we are in an *uncertainty* (or *true uncertainty*) *situation*. The latter case arises, for example, when the outcome of a decision may depend on a simultaneous or subsequent decision of a competitor whose objectives conflict with one's own, or on future external events of non-repeatable variety, for which the estimation of probabilities is a dubious exercise [18]. Robustness analysis concerns uncertainty situations.

Let us assume that the node weights and the edge lengths can take many different values and that there is a (finite or infinite) set S of possible scenarios (possible values of the parameters). For a given scenario s and a point x of G , the cost function under scenario s is defined as follows:

$$C^s(x) = \sum_{i=1}^n w_i^s d^s(x, v_i) \quad (3)$$

where w_i^s and $d^s(a, b)$ denote respectively the weight of node v_i and the minimum distance between points a and b under scenario s . The regret of solution x (also called *opportunity loss* or *absolute deviation* [14]) is the difference between the cost of x under scenario s and the cost of the best solution under the same scenario:

$$R^s(x) = C^s(x) - C^s(x^{*s}) \quad (4)$$

where x^{*s} is the optimal solution of the 1-median problem under scenario s .

To determine the robust solutions for the 1-median problem, authors often attempted to optimize the worst case performance of the system by minimizing the maximal cost or the maximal regret (see [2], [4], [5],

[7], [14] and [22]). The *minmax 1-median problem* is defined as follows:

$$\min_{x \in G} \max_{s \in S} C^s(x) \quad (5)$$

and the *minmax regret 1-median problem* has the following expression :

$$\min_{x \in G} \max_{s \in S} R^s(x) = \min_{x \in G} \max_{s \in S} (C^s(x) - C^s(x^{*s})) \quad (6)$$

In the literature on minmax (regret) 1-median problem, the authors distinguish many models according to the graph structure (tree, network), the location sites (on nodes or on edges) as well as the nature of the scenario set. Indeed, uncertainty on a parameter may be modelled either as a discrete set of scenarios, or as an interval data. Figure 1 summarizes the main results with regard to the minmax regret 1-median problem on a tree, the uncertainty being on weights (assumed to be positive). When weights can be negative and are represented by uncertainty intervals, Burkard and Dollani give an algorithm in $O(n^2)$ for the problem on a tree [6]. As for the minmax regret 1-median problem on a general network (with uncertainty on weights), Averbakh and Berman present in [4] two approaches in $O(nm^4)$ time and $O(mn^2 \log n)$ time for the absolute problem (location anywhere on the graph), the vertex problem having an order of complexity of $O(n^3)$. When edge lengths are uncertain, Chen and Lin [7] show that, in the case of a tree graph and interval data, the problem can be reduced to the deterministic problem under the scenario with maximal lengths. On the other hand, on a general network, the problem with uncertain lengths becomes NP-hard [2].

It is generally admitted that minmax cost and minmax regret criteria are too conservative since they are based only on the worst case. Besides, the worst case performance is often reached for a scenario with a small likelihood of occurrence, especially when uncertainty is represented by intervals. To remedy the conservatism of the minmax regret model for the *p-median* problem (p is the number of facilities to locate), Daskin et al [9] introduce a new variant of this problem called *α -reliable p-minimax regret* problem. In this model, the decision-maker associates a probability with each scenario. The model then selects a subset of scenarios whose collective probability of occurrence is at least some user-specified value α ($0 \leq \alpha \leq 1$) which is called *reliability level*. The model identifies the solution that minimizes the maximum regret with respect to the chosen subset of scenarios. An appropriate choice of α guarantees that the solution is not based on a scenario with a very small likelihood of occurrence.

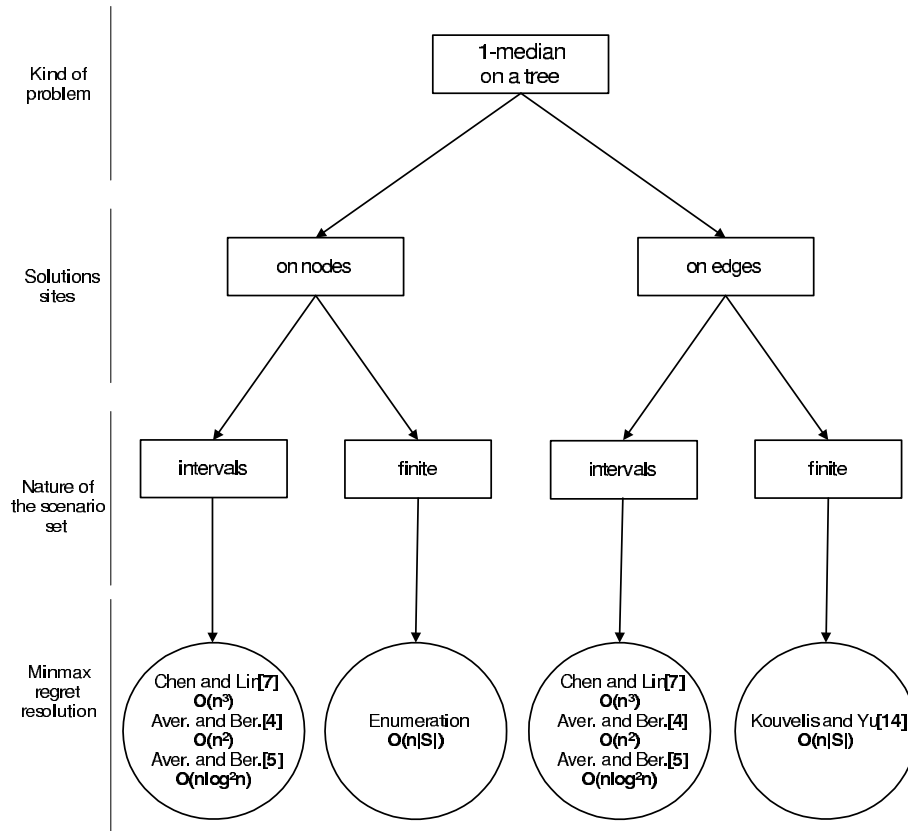


Fig. 1. Minmax regret 1-median problem on a tree with uncertainty on node weights

In a recent work, Snyder and Daskin [21] present the *stochastic p -robust P -median problem (p -SPMP)* (P is the number of facilities to locate). They use a measure called *p -robustness* which was first introduced by Kouvelis et al in [13]. This measure imposes a constraint dictating that the cost under each scenario must be within $(100 + p)\%$ of the optimal cost for that scenario, where $p \geq 0$ is an external parameter (completely independent of P the number of facilities). Moreover, the authors assign a probability to each scenario. Thus, they build a new robustness measure consisting in determining the p -robust solutions which minimize the expected-cost. Snyder and Daskin prove that p -SPMP is NP-hard and discuss a mechanism for detecting infeasibility since p -robust solutions may not exist (especially for small values of p).

The main drawback of these last two approaches is that they require to express a probability for each scenario.

According to this review, we can distinguish two families of approaches to find robust solutions for a given problem. The first family looks for solutions which optimize a certain objective function (e.g. minmax approaches) whereas the second one imposes conditions that solutions must satisfy in order to be considered as

robust (e.g. p -robustness). In the following, we define a new robustness approach which belongs to the second family of approaches.

III. DEFINITION OF A NEW ROBUSTNESS APPROACH

Let us suppose that, for a given problem, one (or several) of the parameters cannot be determined in a certain and definite way and that there is a finite set S of scenarios. Let X denote the set of feasible solutions and q the number of scenarios. Since the reasoning and the results are valid for costs as for regrets, we use in what follows the term “cost” and the notation C indifferently for cost and regret. A robust solution according to the maximal cost criterion is a solution that verifies:

$$\min_{x \in X} \max_{s \in S} C^s(x) \tag{7}$$

In the next subsections, we introduce a new preference relation that we call α -leximax and use it to define a set of robust solutions.

A. The α -leximax relation

Let x be a solution of X . We associate to x a cost vector denoted by $C(x) = (C^{s^1}(x), \dots, C^{s^q}(x))$ where $C^{s^j}(x)$ is the cost of solution x under scenario s^j ,

$1 \leq j \leq q$. By ordering the coordinates of $C(x)$ in a non-increasing order, we get a vector $\hat{C}(x)$ called *disutility vector* [15]. We have $\hat{C}^1(x) \geq \hat{C}^2(x) \geq \dots \geq \hat{C}^q(x)$. Thus, $\hat{C}^j(x)$ is the j^{th} largest cost of x .

Definition 1: Let x and y be two solutions of X , $\hat{C}(x)$ and $\hat{C}(y)$ the associated disutility vectors. The *leximax* relation, denoted by \succ_{lex} , is defined as follows [11]:

$$\begin{aligned} x \succ_{lex} y &\Leftrightarrow \exists k \in \{1, \dots, q\} : \hat{C}^k(x) < \hat{C}^k(y) \\ &\text{and } \forall j \leq k - 1, \hat{C}^j(x) = \hat{C}^j(y) \end{aligned}$$

x is said to be (strictly) preferred to y in the sense of the *leximax* relation.

$$x \sim_{lex} y \Leftrightarrow \forall k \in \{1, \dots, q\}, \hat{C}^k(x) = \hat{C}^k(y)$$

x and y are said to be equivalent in the sense of the *leximax* relation.

In other words, comparing two cost vectors in the sense of the *leximax* relation is equivalent to comparing the first distinct coordinates of the disutility vectors. Remark that reordering cost vector implies that we implicitly assume that the vector obtained by the permutation of the cost vector coordinates is equivalent to the original cost vector (the *leximax* relation is said to be *anonymous* [16]). This is justified by the fact that, in a situation of true uncertainty, none of the scenarios can be distinguished. The *leximax* relation is complete, reflexive and transitive. Therefore, it is a *weak order*.

The previous definition of the *leximax* relation requires a perfect equality between the disutility vector coordinates of two solutions in order to consider them equivalent. Nevertheless, in practice, it may exist a tolerance threshold under which the decision-maker either cannot perceive the difference between two elements, or refuse to give his opinion on the preference for one of them [20]. Taking an indifference threshold α into account leads to the following definition:

Definition 2: Let x and y be two solutions of X , $\hat{C}(x)$ and $\hat{C}(y)$ the associated disutility vectors, and α a positive real value. The α -*leximax* relation, denoted by \succ_{lex}^α , is defined as follows :

$$\begin{aligned} x \succ_{lex}^\alpha y &\Leftrightarrow \exists k \in \{1, \dots, q\} : \hat{C}^k(x) < \hat{C}^k(y) - \alpha \\ &\text{and } \forall j \leq k - 1, |\hat{C}^j(y) - \hat{C}^j(x)| \leq \alpha \end{aligned}$$

x is said to be (strictly) preferred to y in the sense of the α -*leximax* relation.

$$x \sim_{lex}^\alpha y \Leftrightarrow \forall k \in \{1, \dots, q\}, |\hat{C}^k(y) - \hat{C}^k(x)| \leq \alpha$$

x and y are said to be indifferent in the sense of the α -*leximax* relation.

The α -*leximax* relation is a lexicographic aggregation of semiorders. It is known that, for $\alpha \neq 0$, such an aggregation is not a semiorder [17] (for $\alpha = 0$, the relation resulted from the aggregation is none other than the weak order *leximax*). Actually, neither its asymmetric part \succ_{lex}^α nor its symmetric part \sim_{lex}^α are transitive, which may lead to preference cycles.

Example 1: $\hat{C}(x) = (3, 3, 2)$, $\hat{C}(y) = (5, 0, 0)$, $\hat{C}(z) = (4, 3, 0)$ and $\alpha = 1$.

We have $x \succ_{lex}^\alpha y$ and $y \succ_{lex}^\alpha z$ but $z \succ_{lex}^\alpha x$.

Despite its failure to comply with some properties, the α -*leximax* preference relation remains suitable for the determination of robust solutions since it takes into account several measures (costs under different scenarios), offers some tolerance (indifference threshold) and takes into account, at least initially, the solutions given by the minmax criteria (cost/regret).

B. Lexicographic α -robust solutions

We want to determine the set of robust solutions by relying on the α -*leximax* relation. As noticed at the end of section II, there are two families of robustness approaches: the first family looks for solutions given by optimizing a chosen criterion and the second one imposes some robustness properties that solutions must satisfy. Let x^* be an ideal solution (most of the time fictitious) such that:

$$\hat{C}(x^*) = (\hat{C}^1(x^*), \hat{C}^2(x^*), \dots, \hat{C}^q(x^*)) \quad (8)$$

where $\hat{C} = (\hat{C}^1, \hat{C}^2, \dots, \hat{C}^q)$ is the disutility vector and $x_k^* = \arg \min_{x \in X} \hat{C}^k(x)$ for all $k \in \{1, \dots, q\}$. Let us consider the following set:

$$\begin{aligned} A(\alpha) &= \{x \in X : \text{not}(x^* \succ_{lex}^\alpha x)\} \\ &= \{x \in X : x \sim_{lex}^\alpha x^*\} \end{aligned} \quad (9)$$

where the second equality results from the fact that \succ_{lex}^α is complete and that we cannot have $x \succ_{lex}^\alpha x^*$ by definition of x^* .

Using the definition of α -*leximax* relation, the set $A(\alpha)$ can also be written as follows:

$$A(\alpha) = \{x \in X : \forall k \leq q, \hat{C}^k(x) - \hat{C}^k(x_k^*) \leq \alpha\} \quad (10)$$

Any solution of $A(\alpha)$ performs well with regard to the disutility vector since $A(\alpha)$ is the set of solutions whose the k^{th} largest cost is close to the minimum for all $k \leq q$. If we consider this last condition as a robustness

property, then we can consider $A(\alpha)$ as a set of robust solutions that we will call set of *lexicographic α -robust solutions*.

It is obvious that for small values of α , this set can be empty. The minimum value of α that guarantees the existence of lexicographic α -robust solutions is:

$$\alpha_{min} = \min_{x \in X} \max_{1 \leq k \leq q} (\hat{C}^k(x) - \hat{C}^k(x_k^*)) \quad (11)$$

Moreover, the set of lexicographic α -robust solutions is stable with regard to parameter α , that is if $\alpha' \leq \alpha$ then $A(\alpha') \subseteq A(\alpha)$.

We present, in appendix I, a simple algorithm named α -*LEXROB*, for the determination of lexicographic α -robust solutions in the case of a finite set of solutions. Algorithm α -*LEXROB* is based on an iterative procedure which determines, in each iteration $k \in \{1, \dots, q\}$, the subset:

$$A^k(\alpha) = \{x \in A^{k-1}(\alpha) : \hat{C}^k(x) - \hat{C}^k(x_k^*) \leq \alpha\} \quad (12)$$

The algorithm requires $O(|X|q)$ elementary operations where $|X|$ is the number of elements of X and q the number of scenarios.

C. Example

Let us consider the vertex 1-median problem. We have $X = V$ and $|X| = n$ where V is the set of all nodes of the graph and n the number of nodes. For this problem, algorithm α -*LEXROB* is polynomial ($O(nq)$). We consider the complete graph of figure 2.

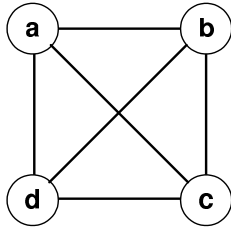


Fig. 2. A complete graph with four vertices

All edges are 1 unit long whereas node weights can take two possible values depending on the scenario (see table I). The cost of each vertex is computed according to equation (3). We look for lexicographic α -robust solutions among vertices a, b, c and d .

According to the maximal cost criterion, the robust solution is b since it has the minimum cost in the worst case. It is clear that this solution is not so robust since it does not perform well under all scenarios. Indeed, its cost under scenarios S_1 and S_2 is rather high.

We give hereafter the lexicographic α -robust solutions for different values of α .

TABLE I

Weights and costs of nodes under scenarios S_1 and S_2

	weights		costs		dis. vect.	
vertex	S_1	S_2	S_1	S_2	\hat{C}^1	\hat{C}^2
a	14	3	14	30	30	14
b	3	8	25	25	25	25
c	1	17	27	16	27	16
d	10	5	18	28	28	18

- $\alpha = 1 \Rightarrow A(1) = \emptyset$.
- $\alpha = 2 \Rightarrow A(2) = \{c\}$.
- $\alpha = 3 \Rightarrow A(3) = \{c\}$.

IV. LEXICOGRAPHIC α -ROBUST 1-MEDIAN PROBLEM ON A TREE

We consider the 1-median problem on a tree in the case of uncertainty on node weights. Kouvelis and Yu [14] consider the minmax cost and the minmax regret versions of this problem using scenario-based weights. They present an $O(nq)$ algorithm to determine the minmax (regret) 1-median of the tree, where n is the number of nodes and q the number of scenarios.

Instead of finding a unique robust 1-median on the tree, we want to determine the lexicographic α -robust set $A(\alpha)$, if it is not empty, in order to define *robust segments* of the tree. Since the feasible solution set is infinite, algorithm α -*LEXROB* cannot be applied. After reminding the principle of Kouvelis and Yu's algorithm, we present a specific polynomial algorithm for the *lexicographic α -robust 1-median* problem on a tree. We recall that the notation C and the word "cost" refer indifferently to cost or to regret.

A. Principle of Kouvelis and Yu's algorithm

Let T be a tree. The removal of any edge (v_i, v_j) of T partitions the tree into two connected components made up of node subsets V_i and V_j . For each point x on edge (v_i, v_j) of length c_{ij} , we denote by y the distance between node v_i and x ($0 \leq y \leq c_{ij}$). The minimum maximal cost is defined as :

$$\min_{(v_i, v_j) \in T} \max_{0 \leq y \leq c_{ij}} \max_{s \in S} C_{ij}^s(y) \quad (13)$$

where $C_{ij}^s(y)$ is the cost under scenario s of the point of (v_i, v_j) at a distance y from node v_i . For the 1-median problem on a tree, the cost $C_{ij}^s(y)$ can be written as:

$$C_{ij}^s(y) = \lambda_{ij}^s + \mu_{ij}^s y \quad (14)$$

with :

$$\begin{aligned} \mu_{ij}^s &= \sum_{v_k \in V_i} w_k^s - \sum_{v_k \in V_j} w_k^s, \\ \lambda_{ij}^s &= \sum_{v_k \in V_i} w_k^s d(v_i, v_k) + \sum_{v_k \in V_j} w_k^s (d(v_j, v_k) + c_{ij}) \end{aligned}$$

if C represents the cost and

$$\lambda_{ij}^s = \sum_{v_k \in V_i} w_k^s d(v_i, v_k) + \sum_{v_k \in V_j} w_k^s (d(v_j, v_k) + c_{ij}) - C^s(x^{*s})$$

if C represents the regret, $C^s(x^{*s})$ being the minimum cost under scenario s .

In their approach, Kouvelis and Yu determine, for a given edge (v_i, v_j) , the solution y_{ij}^* which minimizes the maximal cost on the edge. They describe a procedure that computes y_{ij}^* by solving:

$$C_{ij}(y_{ij}^*) = \min_{0 \leq y \leq c_{ij}} \max_{s \in S} C_{ij}^s(y) \quad (15)$$

After applying this procedure to all edges of the tree, they use a linear time algorithm to find the minmax (regret) 1-median by determining, among all points y_{ij}^* found, one with a minimum maximal cost (regret).

B. Determination of the robust segments of the tree

1) Principle and notations:

We want to find the robust segments of a tree T , that is the set of lexicographic α -robust solutions when $X = T$. We present here an algorithm which determines acceptable intervals that are reduced at each iteration and finally gives robust intervals.

For a given edge (v_i, v_j) of length c_{ij} and a point $x \in (v_i, v_j)$, $\hat{C}_{ij}^k(x)$ represents the k^{th} largest cost of x on interval $[0, c_{ij}]$, $1 \leq k \leq q$. It is obvious that, unlike $C_{ij}^s(\cdot)$, costs $\hat{C}_{ij}^k(\cdot)$ are not linear functions on $[0, c_{ij}]$.

We define the following subsets for $k \in \{1, \dots, q\}$ and $(v_i, v_j) \in E$:

$$I_{ij}^k(\alpha) = \{y \in [0, c_{ij}] : \hat{C}_{ij}^k(y) - \hat{C}_{ij}^k(x_k^*) \leq \alpha\} \quad (16)$$

where $x_k^* = \arg \min_{x \in X} \hat{C}_{ij}^k(x)$. Subsets $I_{ij}^k(\alpha)$ are called *acceptable intervals of order k* .

Let $A_{ij}^k(\alpha)$ be the acceptable subsets defined as follows:

$$\begin{aligned} A_{ij}^1(\alpha) &= I_{ij}^1(\alpha) \text{ and} \\ A_{ij}^k(\alpha) &= A_{ij}^{k-1}(\alpha) \cap I_{ij}^k(\alpha) \text{ for } k \geq 2 \end{aligned} \quad (17)$$

Then, the acceptable subset $A^k(\alpha)$, $k \geq 1$, defined in equation (12) can be written as :

$$A^k(\alpha) = \bigcup_{(v_i, v_j) \in E} A_{ij}^k(\alpha) \quad (18)$$

Therefore, in order to determine the set of lexicographic α -robust solutions, we use the following algorithm.

Algorithm α -LEXROB(IMT)

Begin

$A^0(\alpha) \leftarrow X$;
 $A_{ij}^0(\alpha) \leftarrow [0, c_{ij}]$ for all $(v_i, v_j) \in E$;
 $k \leftarrow 1$;
while ($k \leq q$ and $A^{k-1}(\alpha) \neq \emptyset$) **do**
 Compute x_k^* ;
 for all $(v_i, v_j) \in E$ **do**
 Determine $I_{ij}^k(\alpha)$;
 Determine $A_{ij}^k(\alpha) \leftarrow A_{ij}^{k-1}(\alpha) \cap I_{ij}^k(\alpha)$;
 Determine $A^k(\alpha) \leftarrow \bigcup_{(v_i, v_j) \in E} A_{ij}^k(\alpha)$;
 $k \leftarrow k + 1$;

End.

If for a given $k \leq q$, $A^k(\alpha) = \emptyset$, then it is obvious that $A(\alpha) = \emptyset$.

In the following, we detail the procedures required by the algorithm.

2) Determination of x_k^* :

We begin by determining, for each edge (v_i, v_j) , the point $y_{ij}^{*(k)}$ which minimizes the cost \hat{C}_{ij}^k on (v_i, v_j) . The point x_k^* corresponds to the point $y_{ij}^{*(k)}$ with the minimum cost \hat{C}^k . For $k > 1$, functions $\hat{C}_{ij}^k(\cdot)$ are piecewise linear but not convex unlike functions $\hat{C}_{ij}^1(\cdot)$. As a result, it is not possible to use the Kouvelis and Yu's procedure since it is based on the convexity of $\hat{C}_{ij}^1(\cdot)$. We propose another approach consisting in determining, for each interval $[0, c_{ij}]$, all points corresponding to a breakpoint of function $\hat{C}_{ij}^k(\cdot)$. Indeed, if it is different from 0 and c_{ij} , $y_{ij}^{*(k)}$ is bound to be one of the points where the function slope changes (see figure 3). We call these points $z_{ij}^{h(k)}$, $1 \leq h \leq h_{ij}^k$, where h_{ij}^k is the number of breakpoints of function \hat{C}_{ij}^k on $[0, c_{ij}]$. Points $z_{ij}^{1(k)}$ and $z_{ij}^{h_{ij}^k(k)}$ correspond respectively to 0 and c_{ij} .

We present, in appendix II, a procedure ($Find(y_{ij}^{*(k)})$) giving for an edge (v_i, v_j) all points $z_{ij}^{h(k)}$, $1 \leq h \leq h_{ij}^k$, as well as the point $y_{ij}^{*(k)}$ whose cost \hat{C}_{ij}^k is minimum.

The main idea of procedure $Find(y_{ij}^{*(k)})$ is to determine the scenarios $s_{ij}^{h(k)}$, $h \in \{1, \dots, h_{ij}^k\}$, which give the k^{th} largest cost function \hat{C}_{ij}^k on the interval $[0, c_{ij}]$. At each iteration, the procedure determines, at the current breakpoint $z_{ij}^{h(k)}$, the line that must be chosen from those (two or more) available (see figure 3) as well as the adjacent breakpoint.

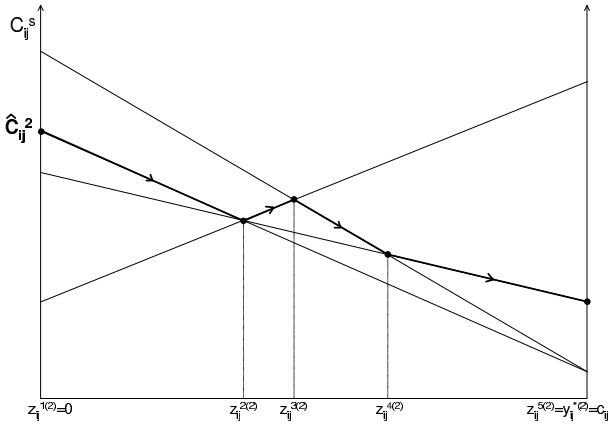


Fig. 3. Illustration of procedure $Find(y_{ij}^{*(k)})$ when $k = 2$ and $q = 4$

Lemma 1: The complexity of procedure $Find(y_{ij}^{*(k)})$ is $O(q^{7/3} \log q)$

Proof: see appendix III.

Lemma 2: Finding x_k^* requires $O(nq^{7/3} \log q)$ elementary operations.

Proof: Lemma 2 follows directly from lemma 1 and from the following relation:

$$x_k^* = \arg \min_{x \in X} \hat{C}^k(x) = \arg \min_{(v_i, v_j) \in E} \hat{C}_{ij}^k(y_{ij}^{*(k)}) \quad (19)$$

We remind that a tree has $n - 1$ edges. □

3) Determination of acceptable intervals $I_{ij}^k(\alpha)$:

Unlike acceptable intervals of order 1, subsets $I_{ij}^k(\alpha)$, $2 \leq k \leq q$, are not necessarily connected because of the non convexity of functions $\hat{C}_{ij}^k(\cdot)$. Nevertheless, for the sake of convenience, we will continue to call them *acceptable intervals*. Let us notice that the nature of these intervals depends on parameter α . A subset $I_{ij}^k(\alpha)$ can be represented by a unique interval for a given value of α , and change into the union of several intervals for a different value of this parameter.

If it is not empty, the acceptable interval $I_{ij}^k(\alpha)$ can be represented by the union of p_{ij}^k elementary intervals as follows:

$$I_{ij}^k(\alpha) = [y_{ij}^{1(k)}, y_{ij}^{2(k)}] \cup \dots \cup [y_{ij}^{2p_{ij}^k-1(k)}, y_{ij}^{2p_{ij}^k(k)}] \quad (20)$$

with $0 \leq y_{ij}^{1(k)} \leq \dots \leq y_{ij}^{2p_{ij}^k(k)} \leq c_{ij}$ and $p_{ij}^k \in \mathbb{N}^*$.

Let us denote by E_{ij}^k the set of points corresponding to the bounds of $I_{ij}^k(\alpha)$:

$$E_{ij}^k = \{y_{ij}^{1(k)}, y_{ij}^{2(k)}, \dots, y_{ij}^{2p_{ij}^k-1(k)}, y_{ij}^{2p_{ij}^k(k)}\} \quad (21)$$

Remark that for $k = 1$, $E_{ij}^1 = \{y_{ij}^1, y_{ij}^2\}$ and $p_{ij}^1 = 1$.

Remind that $z_{ij}^{h(k)}$, $1 \leq h \leq h_{ij}^k$, are the breakpoints of \hat{C}_{ij}^k on $[0, c_{ij}]$ determined by procedure $Find(y_{ij}^{*(k)})$. It is obvious that for a given $h \in \{1, \dots, h_{ij}^k - 1\}$, if we have $z_{ij}^{h(k)} \notin I_{ij}^k(\alpha)$ and $z_{ij}^{h+1(k)} \notin I_{ij}^k(\alpha)$ (that is $\hat{C}_{ij}^k(z_{ij}^{h(k)}) > \hat{C}_{ij}^k(x_k^*) + \alpha$ and $\hat{C}_{ij}^k(z_{ij}^{h+1(k)}) > \hat{C}_{ij}^k(x_k^*) + \alpha$), then $[z_{ij}^{h(k)}, z_{ij}^{h+1(k)}] \cap I_{ij}^k(\alpha) = \emptyset$. Similarly, if $z_{ij}^{h(k)} \in I_{ij}^k(\alpha)$ and $z_{ij}^{h+1(k)} \in I_{ij}^k(\alpha)$, then $[z_{ij}^{h(k)}, z_{ij}^{h+1(k)}] \subset I_{ij}^k(\alpha)$ (see figure 4). On the other hand, if one of them belongs to $I_{ij}^k(\alpha)$ and not the other, then only a part of the interval $[z_{ij}^{h(k)}, z_{ij}^{h+1(k)}]$ is included in $I_{ij}^k(\alpha)$. Therefore, we just have to enumerate the points $z_{ij}^{h(k)}$ in a non-decreasing order for h varying from 1 to h_{ij}^k in order to determine the parts of intervals $[z_{ij}^{h(k)}, z_{ij}^{h+1(k)}]$ which belong to $I_{ij}^k(\alpha)$ and afterwards deduce the set E_{ij}^k .

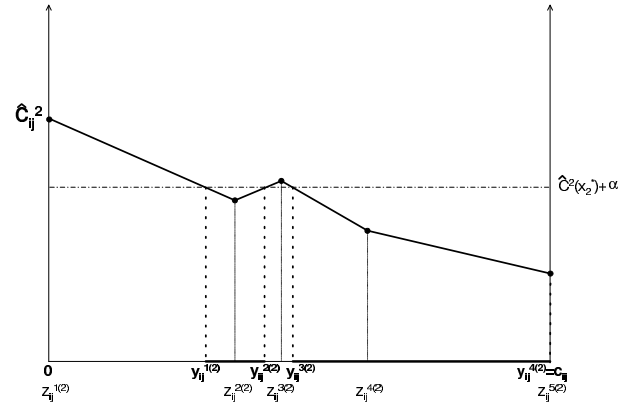


Fig. 4. Illustration of procedure $Find(I_{ij}^k)$ when $k = 2$

A detailed procedure, $Find(I_{ij}^k)$, is presented in appendix II. Since we go over all points $z_{ij}^{h(k)}$, $1 \leq h \leq h_{ij}^k$, the complexity of procedure $Find(I_{ij}^k)$ is $O(h_{ij}^k)$, so $O(q^{4/3})$ (see proof of lemma 1 in appendix III).

Lemma 3: Given x_k^* , finding the set $I^k(\alpha)$ which is the union of $I_{ij}^k(\alpha)$ over E requires $O(nq^{4/3})$ elementary operations.

4) Determination of the acceptable subsets $A_{ij}^k(\alpha)$:

We have $A_{ij}^1(\alpha) = I_{ij}^1(\alpha)$. For $k \geq 2$,

$$A_{ij}^k(\alpha) = A_{ij}^{k-1}(\alpha) \cap I_{ij}^k(\alpha) \quad (22)$$

$A_{ij}^k(\alpha)$ is then the union of r_{ij}^k subintervals of $[0, c_{ij}]$:

$$A_{ij}^k(\alpha) = [a_{ij}^{1(k)}, a_{ij}^{2(k)}] \cup \dots \cup [a_{ij}^{2r_{ij}^k-1(k)}, a_{ij}^{2r_{ij}^k(k)}] \quad (23)$$

with $0 \leq a_{ij}^{1(k)} \leq \dots \leq a_{ij}^{2r_{ij}^k(k)} \leq c_{ij}$ and $r_{ij}^k \in \mathbb{N}^*$.

Let us denote by B_{ij}^k the set of points corresponding to the bounds of $A_{ij}^k(\alpha)$:

$$B_{ij}^k = \{a_{ij}^{1(k)}, a_{ij}^{2(k)}, \dots, a_{ij}^{2r_{ij}^k-1(k)}, a_{ij}^{2r_{ij}^k(k)}\} \quad (24)$$

For $k = 1$, $B_{ij}^1 = \{a_{ij}^1, a_{ij}^2\} = \{y_{ij}^1, y_{ij}^2\}$ and $r_{ij}^1 = 1$.

Given B_{ij}^{k-1} (bounds of $A_{ij}^{k-1}(\alpha)$) and E_{ij}^k (bounds of $I_{ij}^k(\alpha)$), then $B_{ij}^k \subset (B_{ij}^{k-1} \cup E_{ij}^k)$. Consequently, to find the bounds of $A_{ij}^k(\alpha)$, we have to look for them among those of $A_{ij}^{k-1}(\alpha)$ and $I_{ij}^k(\alpha)$. The figure 5 shows how to find the set B_{ij}^k (see appendix II for the detailed procedure).

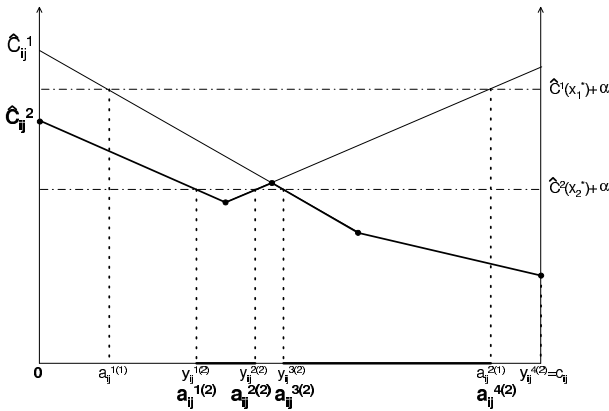


Fig. 5. Illustration of procedure $Find(A_{ij}^k)$ when $k = 2$

Lemma 4: The complexity of procedure $Find(A_{ij}^k)$ is $O(q^3)$.

Proof: see appendix III.

5) Complexity of lexicographic α -robust segments of the tree:

Theorem 1: Lexicographic α -robust 1-median on a tree can be solved in $O(nq^4)$ time.

Proof: Theorem 1 follows immediately from lemmas 2, 3 and 4. Indeed, the largest number of elementary operations needed is due to the determination of sets $A^k(\alpha)$ for all $k \leq q$. \square

C. Example

Consider the tree T of figure 6 where values on edges represent lengths. Uncertainty on node weights is modelled by four scenarios as shown in table II (v_s^* is the median under scenario s , $s \in \{S_1, S_2, S_3, S_4\}$).

The minmax median x^* of the tree is the point of (v_1, v_4) at a distance 2.5 from node v_1 and the minmax

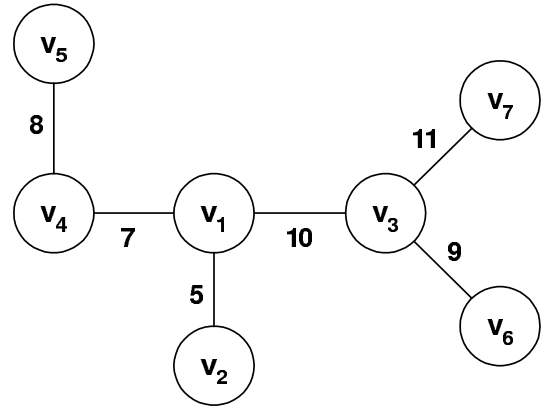


Fig. 6. Example

TABLE II
Weight scenarios

weights	S_1	S_2	S_3	S_4
v_1	1	1	1	1
v_2	1	10	1	1
v_3	1	1	10	1
v_4	1	1	1	1
v_5	1	1	1	10
v_6	1	1	1	1
v_7	1	1	1	1
v_s^*	v_1	v_2	v_3	v_5

cost is 197. We present in figure 7 the cost functions on edge (v_1, v_4) (C_{14}^j denotes the cost function on interval $[0, 7]$ under scenario j). The points which minimize costs \hat{C}^k , $k = 1, \dots, 4$, are $x_1^* = x^*$, x_2^* the point of edge (v_1, v_3) at a distance 2.5 from node v_1 , $x_3^* = v_3$ and $x_4^* = v_1$. If we choose a threshold $\alpha = 45$, the first iteration of algorithm α -LEXROB(IMT) gives the set:

$$A^1(45) = \{x \in T : \hat{C}^1(x) - 197 \leq 45\} \quad (25)$$

represented by bold segments in figure 8.

After four iterations, we get the set $A(45)$ of lexicographic α -robust solutions of the tree. $A(45)$ is represented by the union of three segments (v_1, v'_2) , (v_1, v'_3) and (v_1, v'_4) where v'_2 is the point of edge (v_1, v_2) at a distance 1.78 from node v_1 , v'_3 the point of edge (v_1, v_3) at a distance 1 from node v_1 and v'_4 the point of edge (v_1, v_4) at a distance 0.83 from node v_1 (see figure 9). Remark that x^* , the minmax robust solution, is outside the lexicographic α -robust set. Indeed, it performs well for the maximal cost function, but not well enough for \hat{C}^2, \hat{C}^3 and \hat{C}^4 , compared with node v_1 for example.

The minimum value of parameter α which guarantees the existence of lexicographic α -robust solutions

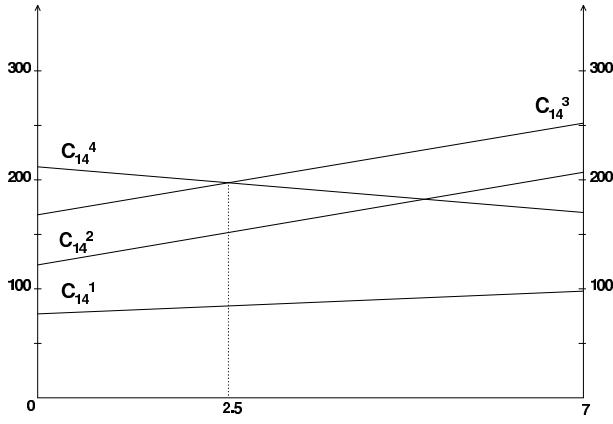


Fig. 7. Cost functions on edge (v_1, v_4)

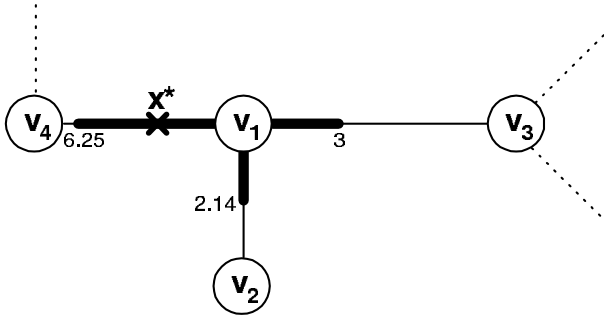


Fig. 8. Set $A^1(45)$

is $\alpha_{min} = 35$, that is $\alpha < 35 \Rightarrow A(\alpha) = \emptyset$ and $\alpha \geq 35 \Rightarrow A(\alpha) \neq \emptyset$.

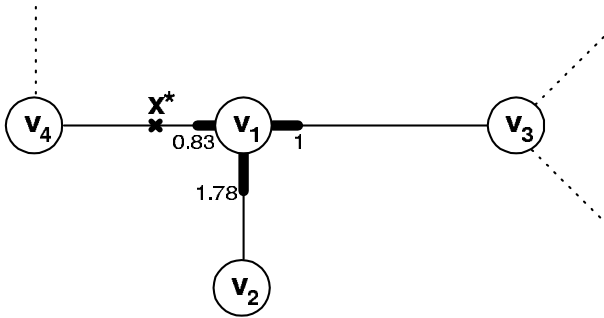


Fig. 9. Minmax robust solution (x^*) and lexicographic α -robust solutions of the tree for $\alpha = 45$

V. CONCLUSION AND PERSPECTIVES

In this paper, we introduced a new robustness approach, called *lexicographic α -robustness*, suitable for the scenario-based uncertainty. Compared with minmax criteria, this approach has three main advantages. First, it takes into account several measures, that is to say costs. Second, it offers some tolerance since it includes an indifference threshold α . Finally, it keeps the importance of the largest cost. We presented a simple algorithm

for the determination of lexicographic α -robust solutions when the set of solutions is finite. We also studied a special problem in the case of an infinite set of solutions, the 1-median location problem on a tree graph and we presented a polynomial algorithm for this problem.

It is obvious that lexicographic α -robustness adds some more complexity to minmax versions of a problem, that is why it is reasonable to apply this approach only to problems which remain polynomially solvable under minmax criteria. In facility location context, the *minmax 1-center* problem on a tree ([3], [6]) and the *minmax 1-median* problem on a general network ([4]) are shown to remain polynomially solvable in the case of interval uncertainty on weights. We recall that the *1-center* problem is to locate a facility on a graph such that it minimizes the maximal distance between the facility and different nodes. We consider the application of lexicographic α -robustness to these problems to be an avenue for future research.

ACKNOWLEDGEMENTS

This research was partially funded by the cooperation agreement CNRS/CGRI-FNRS n° 18 227.

APPENDIX I

Algorithm α -LEXROB

Input: $\hat{C}(x)$ for all $x \in X$ and α .

Output: $A(\alpha)$.

Begin

$A^0 \leftarrow X$;

$k \leftarrow 1$;

while ($k \leq q$ and $A^{k-1} \neq \emptyset$) **do**

$x_k^* \leftarrow \arg \min_{x \in X} \hat{C}^k(x)$;

$A^k \leftarrow \emptyset$;

for all $x \in A^{k-1}$ **do**

if $\hat{C}^k(x) - \hat{C}^k(x_k^*) \leq \alpha$ **then** $A^k \leftarrow A^k \cup \{x\}$;

$k \leftarrow k + 1$;

$A(\alpha) \leftarrow A^{k-1}$;

End.

The condition $A^{k-1} \neq \emptyset$ is used for stopping the algorithm if $A^{k-1}(\alpha)$ is empty, since if $A^{k-1}(\alpha) = \emptyset$ then for all $l \geq k$, $A^l(\alpha) = \emptyset$.

APPENDIX II

Procedure Find($y_{ij}^{*(k)}$)**Input:** λ_{ij}^s and μ_{ij}^s for all $s \in S$,**Output:** $y_{ij}^{*(k)}$, $\hat{C}_{ij}^k(y_{ij}^{*(k)})$, h_{ij}^k , $z_{ij}^{h(k)}$ and $s_{ij}^{h(k)}$ for $h \in \{1, \dots, h_{ij}^k\}$.**Begin**

$$z_{ij}^{1(k)} \leftarrow 0;$$

$$h \leftarrow 1;$$

repeat1) Compute costs $\hat{C}_{ij}^1(z_{ij}^{h(k)})$ to $\hat{C}_{ij}^k(z_{ij}^{h(k)})$;2) Find set of scenarios $S^k(z_{ij}^{h(k)})$ such that:

$$S^k(z_{ij}^{h(k)}) = \{s \in S; \lambda_{ij}^s + \mu_{ij}^s z_{ij}^{h(k)} = \hat{C}_{ij}^k(z_{ij}^{h(k)})\};$$

3) Let d be the first index such that:

$$\hat{C}_{ij}^{k-d}(z_{ij}^{h(k)}) \neq \hat{C}_{ij}^k(z_{ij}^{h(k)});$$

4) Let $s_{ij}^{h(k)}$ be the scenario of $S^k(z_{ij}^{h(k)})$ with the d^{th} largest slope (slope = μ_{ij}^s);5) Compute the intersection points of segment $\{\lambda_{ij}^{s_{ij}^{h(k)}} + \mu_{ij}^{s_{ij}^{h(k)}} z, z \in]z_{ij}^{h(k)}, c_{ij}]\}$ with all lines corresponding to other scenarios;6) The point $z_{ij}^{h+1(k)}$ is the nearest intersection point from $z_{ij}^{h(k)}$:

$$z_{ij}^{h+1(k)} \leftarrow \min\{c_{ij};$$

$$\min_{s \in S \setminus \{s_{ij}^{h(k)}\}} \{z^s = \frac{\lambda_{ij}^{s_{ij}^{h(k)}} - \lambda_{ij}^s}{\mu_{ij}^s - \mu_{ij}^{s_{ij}^{h(k)}}}; z^s \in]z_{ij}^{h(k)}, c_{ij}]\}\};$$

7) $h \leftarrow h + 1$;**until** $z_{ij}^{h(k)} = c_{ij}$;

$$h_{ij}^k \leftarrow h;$$

$$y_{ij}^{*(k)} \leftarrow z_{ij}^{t(k)} \text{ where } t \leftarrow \arg \min_{1 \leq h \leq h_{ij}^k} \{\hat{C}_{ij}^k(z_{ij}^{h(k)})\};$$

End.

Step 1 computes the k largest costs of the current breakpoint $z_{ij}^{h(k)}$. Steps 2, 3 and 4 are used for determining the line that must be chosen from those (two or more) available in order to determine the scenario which gives \hat{C}^k (see figure 3). Steps 5 and 6 are used to determine the adjacent breakpoint $z_{ij}^{h+1(k)}$.

Procedure Find(I_{ij}^k)**Input:** $\hat{C}^k(x_k^*)$, α , $\hat{C}_{ij}^k(y_{ij}^{*(k)})$, h_{ij}^k , $z_{ij}^{h(k)}$, $s_{ij}^{h(k)}$ and $\hat{C}_{ij}^k(z_{ij}^{h(k)})$ for $h \in \{1, \dots, h_{ij}^k\}$.**Output:** E_{ij}^k , p_{ij}^k .**Begin****if** $\hat{C}_{ij}^k(y_{ij}^{*(k)}) > \hat{C}^k(x_k^*) + \alpha$ **then** $E_{ij}^k \leftarrow \emptyset$;**else**

$$E_{ij}^k \leftarrow \emptyset;$$

$$p \leftarrow 0;$$

if $\hat{C}_{ij}^k(z_{ij}^{1(k)}) \leq \hat{C}^k(x_k^*) + \alpha$ **then**

$$E_{ij}^k \leftarrow E_{ij}^k \cup \{z_{ij}^{1(k)}\};$$

$$p \leftarrow p + 1;$$

if $\hat{C}_{ij}^k(z_{ij}^{h_{ij}^k(k)}) \leq \hat{C}^k(x_k^*) + \alpha$ **then**

$$E_{ij}^k \leftarrow E_{ij}^k \cup \{z_{ij}^{h_{ij}^k(k)}\};$$

$$p \leftarrow p + 1;$$

for $h = 1$ to $h_{ij}^k - 1$ **do****if** $\hat{C}_{ij}^k(z_{ij}^{h(k)}) > \hat{C}^k(x_k^*) + \alpha$ **then****if** $\hat{C}_{ij}^k(z_{ij}^{h+1(k)}) \leq \hat{C}^k(x_k^*) + \alpha$ **then**

$$y \leftarrow \frac{\hat{C}^k(x_k^*) + \alpha - B_{ij}^{s_{ij}^{h(k)}}}{A_{ij}^{s_{ij}^{h(k)}}};$$

$$E_{ij}^k \leftarrow E_{ij}^k \cup \{y\};$$

$$p \leftarrow p + 1;$$

else**if** $\hat{C}_{ij}^k(z_{ij}^{h+1(k)}) > \hat{C}^k(x_k^*) + \alpha$ **then**

$$y \leftarrow \frac{\hat{C}^k(x_k^*) + \alpha - B_{ij}^{s_{ij}^{h(k)}}}{A_{ij}^{s_{ij}^{h(k)}}};$$

$$E_{ij}^k \leftarrow E_{ij}^k \cup \{y\};$$

$$p \leftarrow p + 1;$$

$$p_{ij}^k \leftarrow p/2;$$

End.

Procedure $Find(A_{ij}^k)$

Input: r_{ij}^{k-1} , p_{ij}^k , $a_{ij}^{t(k-1)}$ for $t \in \{1, \dots, 2r_{ij}^{k-1}\}$ and $y_{ij}^{l(k)}$ for $l \in \{1, \dots, 2p_{ij}^k\}$.

Output: B_{ij}^k , r_{ij}^k .

Begin

$B_{ij}^k \leftarrow \emptyset$;

$r \leftarrow 0$;

for $t = 1$ to r_{ij}^{k-1} **do**

for $l = 1$ to p_{ij}^k **do**

if $y_{ij}^{2l-1(k)} \geq a_{ij}^{2t-1(k-1)}$ **then**

if $y_{ij}^{2l-1(k)} \leq a_{ij}^{2t(k-1)}$ **then**

$B_{ij}^k \leftarrow B_{ij}^k \cup \{y_{ij}^{2l-1(k)}\}$;

$r \leftarrow r + 1$;

if $y_{ij}^{2l(k)} \leq a_{ij}^{2t(k-1)}$ **then**

$B_{ij}^k \leftarrow B_{ij}^k \cup \{y_{ij}^{2l(k)}\}$;

$r \leftarrow r + 1$;

else

$B_{ij}^k \leftarrow B_{ij}^k \cup \{a_{ij}^{2t(k-1)}\}$;

$r \leftarrow r + 1$;

else

if $y_{ij}^{2l(k)} \geq a_{ij}^{2t-1(k-1)}$ **then**

$B_{ij}^k \leftarrow B_{ij}^k \cup \{a_{ij}^{2t-1(k)}\}$;

$r \leftarrow r + 1$;

if $y_{ij}^{2l(k)} \leq a_{ij}^{2t(k-1)}$ **then**

$B_{ij}^k \leftarrow B_{ij}^k \cup \{y_{ij}^{2l(k)}\}$;

$r \leftarrow r + 1$;

else

$B_{ij}^k \leftarrow B_{ij}^k \cup \{a_{ij}^{2t(k-1)}\}$;

$r \leftarrow r + 1$;

$r_{ij}^k \leftarrow r/2$;

End.

APPENDIX III

Proof of lemma 1: For a given edge (v_i, v_j) , a given order k ($k \in \{1, \dots, q\}$) and a given iteration h ($h \in \{1, \dots, h_{ij}^k\}$), step 1 requires a sorting in a set of q elements $(C_{ij}^s(z_{ij}^{h(k)}), s \in S)$, therefore it can be solved in $O(q \log q)$ time. Steps 2, 5 and 6 can be solved in $O(q)$ time, whereas steps 3 and 4 have a complexity of $O(k)$. Consequently, for a given edge (v_i, v_j) and a given order k , procedure $Find(y_{ij}^{*(k)})$ can be computed in $O((h_{ij}^k - 1)q \log q)$ since $k \leq q$ and $h \leq h_{ij}^k$.

Moreover, based on Dey's theorem ([1], [10]), the number of segments of \hat{C}_{ij}^k is $O(q(q-k)^{1/3})$. So we can write $O(h_{ij}^k) = O(q(q-k)^{1/3}) = O(q^{4/3})$ for all $k \leq q$.

The point $y_{ij}^{*(k)}$ is the one of edge (v_i, v_j) with the lowest cost \hat{C}_{ij}^k among all $z_{ij}^{h(k)}$ found by the procedure $Find(y_{ij}^{*(k)})$. Thus, it can be found in $O(q^{7/3} \log q)$ time. \square

Proof of lemma 4: Given an edge (v_i, v_j) and an order k , solving procedure $Find(A_{ij}^k)$ requires $O(r_{ij}^{k-1} \cdot p_{ij}^k)$ elementary operations. p_{ij}^k is the number of subintervals of $[0, c_{ij}]$ given by the intersection of a straight line $(D = \hat{C}^k(x_k^*) + \alpha)$ with at most q lines and possibly the lines $y_{ij} = 0$ and $y_{ij} = c_{ij}$. Therefore, $p_{ij}^k \leq \frac{q+2}{2}$. On the other hand, $B_{ij}^k \subset (E_{ij}^k \cup B_{ij}^{k-1})$ implies that:

$$r_{ij}^k \leq p_{ij}^k + r_{ij}^{k-1} \leq \frac{q+2}{2} + r_{ij}^{k-1} \leq \dots \leq (k-1)\left(\frac{q+2}{2}\right) + r_{ij}^1$$

As $r_{ij}^1 = 1$ and $k \leq q$, we get $r_{ij}^k \leq q\left(\frac{q+2}{2}\right) + 1$, for all $k \in \{1, \dots, q\}$.

Consequently, $r_{ij}^{k-1} \cdot p_{ij}^k \leq (q\left(\frac{q+2}{2}\right) + 1) \cdot \frac{q+2}{2}$, so $O(r_{ij}^{k-1} \cdot p_{ij}^k) = O(q^3)$. \square

REFERENCES

- [1] AGARWAL P.K. and SHARIR M., "Arrangements and their applications," in *Handbook of Computational Geometry*, SACK J.-R. and URRUTIA J., Eds. Elsevier Sciences, 2000, pp. 49–120.
- [2] AVERBAKH I., "Complexity of robust single facility location problems on networks with uncertain edge length," *Discrete Applied Mathematics*, vol. 127, pp. 505–522, 2003.
- [3] AVERBAKH I. and BERMAN O., "Algorithms for the robust 1-center problem on a tree," *European Journal of Operational Research*, vol. 123, pp. 292–302, 2000.
- [4] —, "Minmax regret median location on a network under uncertainty," *INFORMS Journal on Computing*, vol. 12, no. 2, pp. 104–110, 2000.
- [5] —, "An improved algorithm for the minmax regret median problem on a tree," *Networks*, vol. 41, no. 2, pp. 97–103, 2003.
- [6] BURKARD R.E. and DOLLANI H., "Robust location problems with pos/neg weights on a tree," *Networks*, vol. 38, pp. 102–113, 2001.
- [7] CHEN B. and LIN C.S., "Min-max regret robust 1-median location on a tree," *Networks*, vol. 31, pp. 93–103, 1998.
- [8] DASKIN M.S., *Network and Discrete Location: Models, Algorithms and Applications*. Wiley, 1995.
- [9] DASKIN M.S., HESSE S.M., and REVELLE C.S., " α -reliable p-minimax regret: a new model for strategic facility location modeling," *Location Science*, vol. 5, no. 4, pp. 227–246, 1997.
- [10] DEY T.K., "Improved bounds on planar k-sets and related problems," *Discrete Computational Geometry*, vol. 19, no. 3, pp. 373–382, 1998.
- [11] GRABISCH M. and PERNY P., "Agrégration multicritère," in *Logique Floue, principes, aide à la décision*, BOUCHON-MEUNIER B. and MARSALA C., Eds. Hermes-Lavoisier, 2003, pp. 81–120.
- [12] HAKIMI S.L., "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Research*, vol. 12, pp. 450–459, 1964.

- [13] KOUVELIS P., KURAWARWALA A.A., and GUTIRREZ G.J., "Algorithms for robust single and multiple period layout planning for manufacturing systems," *European Journal of Operational Research*, vol. 63, no. 2, pp. 287–303, 1992.
- [14] KOUVELIS P. and YU G., *Robust Discrete Optimization and its Applications*, ser. Non Convex Optimization and Its Applications. Kluwer Academic Publishers, 1997.
- [15] MOULIN H., "Social welfare orderings," in *Axioms of cooperative decision making*. Cambridge University Press, 1988, pp. 30–60.
- [16] OGRYCZAK W., "On the lexicographic minimax approach to location problems," *European Journal of Operational Research*, vol. 100, pp. 566–585, 1997.
- [17] PIRLOT M. and VINCKE P., *Semiorders: Properties, Representations, Applications*. Kluwer Academic Publishers, 1997.
- [18] ROSENHEAD J., ELTON M., and GUPTA S.K., "Robustness and optimality criteria for strategic decisions," *Operational Research Quaterly*, vol. 23, no. 4, pp. 413–423, 1972.
- [19] ROY B., "A missing link in OR-DA : robustness analysis," *Foundations of computing and decision sciences*, vol. 23, no. 3, pp. 141–160, 1998.
- [20] ROY B. and VINCKE Ph., "Relational systems of preference with one or more pseudo-criteria: some new concepts and results," *Management Science*, vol. 30, no. 11, pp. 1323–1335, 1984.
- [21] SNYDER L.V. and DASKIN M.S., "Stochastic p-robust location problems," Lehigh University, Dept. of ISE, Technical report 04T-014, July 2004.
- [22] VAIRAKTARAKIS G.L. and KOUVELIS P., "Incorporation dynamic aspects and uncertainty in 1-median location problems," *Naval Research Logistics*, vol. 46, pp. 147–168, 1999.
- [23] VINCKE P., "Robust solutions and methods in decision-aid," *Journal of Multi-criteria Decision Analysis*, vol. 8, pp. 181–187, 1999.