ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT
Discrete Optimization 2 (2005) 3–34
DISCRETE OPTIMIZATION
www.elsevier.com/locate/disopt

Facility location problems with uncertainty on the plane

Igor Averbakh[a,*], Sergei Bereg[b]

[a] *Division of Management, University of Toronto at Scarborough, 1265 Military Trail, Scarborough, Ont., Canada M1C 1A4*

[b] *Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA*

Received 24 June 2003; received in revised form 26 July 2004; accepted 17 December 2004

Abstract

We consider single facility location problems (1-median and weighted 1-center) on a plane with uncertain weights and coordinates of customers (demand points). Specifically, for each customer, only interval estimates for its weight and coordinates are known. It is required to find a "minmax regret" location, i.e. to minimize the worst-case loss in the objective function value that may occur because the decision is made without knowing the exact values of customers' weights and coordinates that will get realized. We present an $O(n^2 \log^2 n)$ algorithm for the interval data minmax regret rectilinear 1-median problem and an $O(n \log n)$ algorithm for the interval data minmax regret rectilinear weighted 1-center problem. For the case of Euclidean distances, we consider uncertainty only in customers' weights. We discuss possibilities of solving approximately the minmax regret Euclidean 1-median problem, and present an $O(n^2 2^{\alpha(n)} \log^2 n)$ algorithm for solving the minmax regret Euclidean weighted 1-center problem, where $\alpha(n)$ is the inverse Ackermann function.

Keywords: Facility location; Polynomial algorithm; Robust optimization

1. Introduction

Significant research efforts have been devoted to optimization problems with uncertainty in input data because of their importance for practice. Two ways of modeling uncertainty are usually used: the stochastic approach and worst-case analysis.

* Corresponding author. Tel.: +1 416 287 7329; fax: +1 416 287 7363.
E-mail addresses: averbakh@utsc.utoronto.ca (I. Averbakh), besp@utdallas.edu (S. Bereg).

1572-5286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2004.12.001

In the stochastic approach, uncertainty is modelled by means of assuming some probability distribution over the space of all possible scenarios (where a scenario is a specific realization of all parameters of the problem), and the objective is to find a solution with good probabilistic performance. Models of this type are handled using stochastic programming techniques [21,12].

In the worst-case approach, the set of possible scenarios is described deterministically, and one is looking for a solution that performs reasonably well for all scenarios, i.e. that has the best "worst-case" performance and hedges against the most hostile scenario. Specifically, the *minmax regret* version of the worst-case approach seeks to minimize the worst-case loss in the objective function value that may occur because the solution is chosen without knowing which scenario will take place. In other words, the minmax regret approach seeks to find a solution that is ε-optimal for any possible realization of parameters, with ε as small as possible. Minmax regret solutions are sometimes called *robust* solutions [24], although there are several different robustness concepts in the literature (see, e.g., [30,10,11]). Minmax regret optimization (MRO) has received increasing attention over the last decade, and by now it is a well-established area of research. A comprehensive treatment of the state of art in minmax regret discrete optimization up to 1997 and extensive references can be found in the book [24]. We also refer the reader to the book [24] for a comprehensive discussion of the motivation for the minmax regret approach in various types of application environments.

There are two natural ways to define the set of possible scenarios for MRO problems that have been used in the literature. First, when the set of scenarios is finite, all data instances from the set can be listed explicitly; such MRO problems are called *discrete-scenario* MRO problems. Second, the set of scenarios can be defined by specifying an interval of uncertainty for every numerical parameter, with the assumption that the parameter can take on any value within its interval of uncertainty regardless of the values taken by other numerical parameters; such MRO problems are called *interval data* MRO problems. Averbakh [4] gave examples of MRO problems that are polynomially solvable in the interval data version but are NP-hard in the discrete-scenario version.

The minmax regret approach was first applied to a location model by Kouvelis et al. [23], although location problems with deterministically modelled uncertainty in data had been considered earlier (e.g. [33]). Kouvelis et al. [23] presented polynomial algorithms for the minmax regret 1-median problem on a tree; for the interval data case, the complexity of their algorithm is $O(n^4)$, where n is the number of nodes. Chen and Lin [14] presented an alternative algorithm for the same problem on a tree with the order of complexity $O(n^3)$. For the same problem on a tree, Averbakh and Berman [7] further improved the order of complexity—they presented a simple algorithm with complexity $O(n^2)$. Averbakh and Berman have also developed a more complicated algorithm with complexity $O(n \log^2 n)$. For the same problem on a general network, Averbakh and Berman [7] presented a polynomial algorithm with the order of complexity $O(mn^2 \log n)$, where m is the number of edges, for the case where only node weights are uncertain. Averbakh [5] proved that if edge lengths are uncertain, then the problem is strongly NP-hard on general networks, even if there is no uncertainty in node weights. Vairaktarakis and Kouvelis [34] studied the minmax regret 1-median problem on a tree that combines dynamically evolving and uncertain parameters. An approach that combines the Pareto-optimality concept and the minmax regret criterion was considered in Fernandez et al. [19].

Minmax regret center location problems on networks have also been studied in the literature. Averbakh and Berman [8] showed that the interval data minmax regret weighted 1-center problem is polynomially solvable on trees when both node weights and edge lengths are uncertain. General results on MRO problems with a minimax type of objective function [3] imply that when only node weights are uncertain, the interval data minmax regret weighted p-center problem and the multicenter problem with mutual communication are polynomially solvable on any type of a network where their classical versions (i.e. without uncertainty) are polynomially solvable. Complexity orders of the corresponding algorithms can be found in [3]. However, Averbakh [5] proved that if edge lengths are uncertain, the minmax regret 1-center problem is strongly NP-hard on general networks.

In this paper, we study the interval data minmax regret single-facility location problems (the 1-median and the weighted 1-center problems) on a plane. Our main results correspond to the case of rectilinear distances, where we consider uncertainty both in the weights and the coordinates of customers. For the minmax regret rectilinear 1-median problem, which can be formulated as a linear programming problem with $O(n)$ variables and $O(n^3)$ constraints, we present an $O(n^2 \log^2 n)$ algorithm. For the minmax regret rectilinear weighted 1-center problem, we present an $O(n \log n)$ algorithm. We also consider the problems for the case of Euclidean distances with uncertainty only in node weights. We discuss possibilities of solving approximately the minmax regret Euclidean 1-median problem, and present an $O(n^2 2^{\alpha(n)} \log^2 n)$ algorithm for solving the minmax regret Euclidean weighted 1-center problem, where $\alpha(n)$ is the inverse Ackermann function.

2. Definitions and problem statement

Let C be a set of n customers located on the plane R^2. The location of each customer $c \in C$ is a point $v_c = (a_c, b_c) \in R^2$. Each customer $c \in C$ has an associated positive weight w_c. Weights w_c and coordinates a_c, b_c are not known; instead, for any customer $c \in C$ values $w_c^-, w_c^+, a_c^-, a_c^+, b_c^-, b_c^+$ such that $a_c^- \leqslant a_c^+,\ b_c^- \leqslant b_c^+,\ 0 < w_c^- \leqslant w_c^+$ are known, and it is known that $w_c^- \leqslant w_c \leqslant w_c^+,\ a_c^- \leqslant a_c \leqslant a_c^+,\ b_c^- \leqslant b_c \leqslant b_c^+$. We can assume that w_c, a_c, b_c are random variables with unknown distributions and can take on any values in the corresponding intervals of uncertainty $[w_c^-, w_c^+]$, $[a_c^-, a_c^+]$, $[b_c^-, b_c^+]$. The location v_c of a customer $c \in C$ belongs to the "*rectangle of uncertainty*" Γ_c with cornerpoints $(a_c^-, b_c^-),\ (a_c^-, b_c^+),\ (a_c^+, b_c^+),\ (a_c^+, b_c^-)$.

A natural motivation for modeling uncertain customers' locations by means of rectangles of uncertainty in the case of rectilinear distances is as follows. Rectilinear metric is usually used for modeling transportation in urban environment ("Manhattan metric"). At the time of choosing a location for the facility, the only available information about locations of customers may be that they belong to some specific districts (say, between certain streets and certain avenues). Or, the whole districts may be considered as (global) customers in a multi-level supply chain, with the assumption that there will be local distribution centers (e.g., retailers) in the district that will serve the (local) customers from the districts; the (upper-level) facility to be located (e.g., a warehouse) would be used to supply goods to the local distribution centers. If the locations of the local distribution centers have not been specified by the time when a location for the upper-level facility should be chosen, it is natural to model them by means of rectangles of uncertainty. In the case of the Euclidean metric, it is more difficult to find a natural way to model uncertain locations of customers; therefore, for the Euclidean case (considered in Sections 5 and 6), we consider only uncertainty in customers' weights.

Let us introduce the following notation: $v_c^{SW} = (a_c^-, b_c^-)$, $v_c^{NW} = (a_c^-, b_c^+)$, $v_c^{NE} = (a_c^+, b_c^+)$, $v_c^{SE} = (a_c^+, b_c^-)$ (*SW*, *NW*, *NE*, and *SE* stand for South–West, North–West, North–East, and South–East, respectively). Let S be the Cartesian product of intervals $[w_c^-, w_c^+]$, $[a_c^-, a_c^+]$, $[b_c^-, b_c^+]$, $c \in C$. Any $s \in S$ is called a *scenario* and represents a possible assignment of weights and location coordinates to customers, $s = \{w_c^{(s)}, a_c^{(s)}, b_c^{(s)} \mid c \in C\}$; then $v_c^{(s)} = (a_c^{(s)}, b_c^{(s)})$ is the location of customer c under scenario s. For an $\alpha \in \{NW, NE, SE, SW\}$, let S^α denote the set of scenarios $\{s \in S \mid v_c^{(s)} = v_c^\alpha \text{ for all } c \in C\}$. In the remainder of the paper, we use the notation $A = \{NW, NE, SE, SW\}$. Let V denote the set $\{v_c^\alpha \mid \alpha \in A,\ c \in C\}$ (i.e. V is the set of cornerpoints of the rectangles of uncertainty). For any $\alpha \in A$, let $V^\alpha = \{v_c^\alpha \mid c \in C\}$. For any points $x, y \in R^2$, let $d(x, y)$ denote the distance between x and y; in this paper, we consider Euclidean and rectilinear distances. For any point $g \in R^2$, $x_1(g)$ and $x_2(g)$ will denote the first and the second coordinates of g, respectively. A straight line in R^2 with slope 1 (with slope -1) will be called a *positive* (*negative*) *diagonal line*.

For any point $x \in R^2$, the vertical and horizontal straight lines that go through x divide the plane into four quadrants (points of the boundary of a quadrant also belong to the quadrant, so there are points that belong to more than one quadrant). We will call them North–West (NW), North–East (NE), South–East (SE), and South–West (SW) quadrants for x, with the natural correspondence between the names and the quadrants (e.g., the quadrant above and to the right of x is the NE-quadrant for x).

For a scenario $s \in S$, $s = \{w_c^{(s)}, a_c^{(s)}, b_c^{(s)} \mid c \in C\}$ and a point $x \in R^2$, let us define

$$F_1(s, x) = \sum_{c \in C} w_c^{(s)} d(v_c^{(s)}, x), \qquad F_2(s, x) = \max_{c \in C} w_c^{(s)} d(v_c^{(s)}, x).$$

That is, $F_1(s, x)$ ($F_2(s, x)$) is the sum (maximum) of weighted distances between x and the customers from C with weights $w_c^{(s)}$, $c \in C$ and locations $v_c^{(s)}$, $c \in C$ defined by the scenario s. Suppose that we want to choose a location x for a facility using the objective of minimizing a function $F(s, x)$. In this paper, we consider only the cases $F(s, x) = F_1(s, x)$ and $F(s, x) = F_2(s, x)$. If we know the real scenario s, then we have the problem

Problem OPT(s)**.** Minimize $\{F(s, x) | x \in R^2\}$.

In the case $F(s, x) = F_1(s, x)$, Problem OPT(s) is the classical 1-median problem and will be called Problem MED(s); in the case $F(s, x) = F_2(s, x)$, Problem OPT(s) is the classical weighted 1-center problem and will be called Problem CEN(s). An optimal solution to Problem MED(s) (Problem CEN(s)) is called a 1-median (1-center) for scenario s. Let $F^*(s)$ denote the optimal objective function value for Problem OPT(s).

If the real scenario is unknown but only the bounds $w_c^-, w_c^+, a_c^-, a_c^+, b_c^-, b_c^+$ for weights and coordinates of customers are available, different solution criteria for choosing a location for the facility are possible. The criterion used in this paper is to minimize the worst-case regret associated with the chosen location. Here, regret is the difference between the objective function value that corresponds to the chosen location and the optimal objective function value (under the realized scenario); the worst-case regret corresponds to the scenario that achieves the maximum regret for the chosen location.

Specifically, the worst-case regret associated with a location $x \in R^2$ is given by the following subproblem:

Subproblem MAXREGR(x)**.**

$$Z(x) = \max_{s \in S} \max_{y \in R^2} \{F(s, x) - F(s, y)\}. \quad (1)$$

For any $x, y \in R^2$, let us define value

$$REGR(x, y) = \max_{s \in S} (F(s, x) - F(s, y)). \quad (2)$$

Alternative ways to represent $Z(x)$ are

$$Z(x) = \max_{s \in S} (F(s, x) - F^*(s)), \quad (3)$$

$$Z(x) = \max_{y \in R^2} REGR(x, y). \quad (4)$$

An optimal solution to the right-hand side of (3) is called a *worst-case scenario* for x. An optimal solution to the right-hand side of (4) is called a *worst-case alternative* for x.

The following problem is considered in the paper:

Problem ROB. Find $x \in R^2$ that minimizes $Z(x)$.

Let X^* denote the set of optimal solutions for Problem ROB. If $F(s, x)$ is convex for all $s \in S$ (as is the case for all problems considered in the paper, see, e.g., [26]), then $Z(x)$ is a convex function and X^* is a convex set.

In the case $F(s, x) = F_1(s, x)$ ($F(s, x) = F_2(s, x)$), Problem ROB will be referred to as Problem ROBMED (Problem ROBCEN). An optimal solution to Problem ROBMED (Problem ROBCEN) is called a *robust* 1-*median* (*robust* 1-*center*).

Notice that Problem OPT(s) is a special case of Problem ROB (corresponding to the case where S consists of a single scenario, i.e. $a_c^- = a_c^+$, $b_c^- = b_c^+$, $w_c^- = w_c^+$ for all $c \in C$). To get a better intuition about the minmax regret problem, the following interpretation is useful. For an $\varepsilon > 0$ and a scenario s, an $x \in R^2$ is called an ε-optimal solution to Problem OPT(s) if $F(s, x) - F^*(s) \leqslant \varepsilon$. Let $X_\varepsilon(s)$ denote the set of all ε-optimal solutions to Problem OPT(s). One can look for a solution that is ε-optimal (for a given $\varepsilon > 0$) for all possible scenarios, that is, to look for an $x \in \bigcap_{s \in S} X_\varepsilon(s)$. For some values of ε such a solution exists, but for some (smaller) values of ε such a solution may not exist, because solutions good for one scenario may be bad for some other scenarios. Then, the solution x^* obtained by solving Problem ROB will be ε-optimal for all scenarios $s \in S$ for any $\varepsilon \geqslant Z(x^*)$; also, for any $\varepsilon < Z(x^*)$ we have $\bigcap_{s \in S} X_\varepsilon(s) = \emptyset$. So, value $Z(x^*)$ has the interpretation of the minimum possible ε such that there exists a solution which is ε-optimal for Problem OPT(s) for all scenarios $s \in S$; this value can be used as a measure of degree of uncertainty.

A set $S' \subset S$ of scenarios is called *locally sufficient for* $x \in R^2$ if value $Z(x)$ does not change if we replace in (3) maximization over S with maximization over S'. A set $S' \subset S$ of scenarios is called *globally sufficient* if it is locally sufficient for all $x \in R^2$.

Observation. If a set of scenarios S' contains a worst-case scenario for $x \in R^2$, then S' is locally sufficient for x.

3. Rectilinear median

Suppose that distances are rectilinear (i.e. for any $x = (x_1, x_2) \in R^2$ and $y = (y_1, y_2) \in R^2$, $d(x, y) = |x_1 - y_1| + |x_2 - y_2|$) and $F(s, x) = F_1(s, x)$. In Section 3.1, we study properties of Problem ROBMED that allow us to consider only a relatively small number of "efficient" scenarios; namely, we show that $O(n^3)$ scenarios define completely the objective function $Z(x)$ for all x, and $O(n^2)$ scenarios define the value of $Z(x)$ for a specific x. Based on these properties, in Section 3.2 we develop a linear programming formulation of the problem with $O(n)$ variables and $O(n^3)$ constraints. This formulation is not attractive computationally for large values of n, so in Section 3.3 we develop a specialized algorithm for solving Problem ROBMED with complexity $O(n^2 \log^2 n)$. The general idea of the algorithm is as follows. We identify two rectangular grids (one grid consists of lines parallel to the coordinate axes, the other grid consists of positive and negative diagonal lines) with the following property: Problem ROBMED restricted to the intersection of a cell of the first grid with a cell of the second grid can be solved in $O(n^2)$ time. Then, using convexity of the objective function, we perform binary search on the lines of the grids to identify optimal cells (i.e. cells that contain an optimal solution). The nontrivial part is to perform the binary search in almost quadratic time given that there are $O(n^3)$ lines in the grids. Using special properties of the problem and the "search using sorted matrices" technique from theoretical computer science, we perform the binary search in $O(n^2 \log^2 n)$ time.

3.1. General properties

Let H be the convex hull of the set V.

Lemma 1. *There is an optimal solution to Problem ROBMED that belongs to H.*

Proof. Consider an optimal solution x to Problem ROBMED. Suppose $x \notin H$. Let l_v (l_h) be the vertical (horizontal) straight line through x. The line l_v must have common points with H (otherwise, x could be shifted horizontally towards H so that $F_1(s, x)$ would decrease for any $s \in S$, which would contradict optimality of x for Problem ROBMED). Using the same argument, we can show that the line l_h must have common points with H. To be specific, suppose that l_h intersects H to the left of x, and l_v intersects H above x (other possible cases can be considered in an exactly similar way), see Fig. 1. Let l_d be the negative diagonal line through x, and let x_H be the point of $l_d \cap H$ closest to x. It follows from convexity of H that there are no points of H in the interior of the SE quadrant for x_H. Therefore, moving x to x_H will not increase value $F(s, x)$ for any $s \in S$, and, therefore, will not destroy optimality

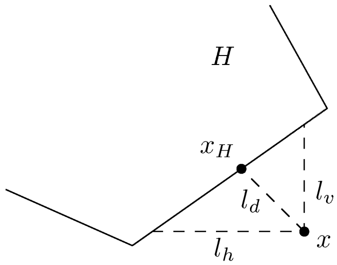

Fig. 1. Illustration for the proof of Lemma 1.

of x for Problem ROBMED. Thus, x_H is also an optimal solution for Problem ROBMED. This proves the lemma. □

The following observation will be used later to identify a relatively small globally sufficient set of scenarios.

Lemma 2. *For any $x \in R^2$, there exists a worst-case scenario s for x such that $s \in S^\alpha$ for some $\alpha \in A$.*

Proof. Let $x \in R^2$ be fixed, and suppose that $s' \in S$ is a worst-case scenario for x. (Existence of a worst-case scenario follows from compactness of S and H.) Let $y \in R^2$ be a 1-median for the scenario s'. Then y is a worst-case alternative for x, and $Z(x) = REGR(x, y) = F(s', x) - F(s', y)$.

Suppose that y belongs to the α-quadrant for x for some $\alpha \in A$. If we change the scenario s' by moving the location of each customer $c \in C$ to v_c^α without changing the weight of the customer, value $F(s', x) - F(s', y)$ will not decrease. Then, for the scenario $s'' \in S^\alpha$ that we obtain in this way, $F(s'', x) - F^*(s'') \geqslant F(s'', x) - F(s'', y) \geqslant F(s', x) - F(s', y) = Z(x)$. Since $F(s'', x) - F^*(s'') \leqslant Z(x)$, we have that $F(s'', x) - F^*(s'') = Z(x)$ and s'' is also a worst-case scenario for x. The lemma is proven. □

For any $\alpha \in A$ and any $x, y \in R^2$, let $s_\alpha(x, y)$ denote the scenario $s \in S^\alpha$ such that for any $c \in C$

$$w_c^{(s)} = \begin{cases} w_c^- & \text{if } d(x, v_c^\alpha) \leqslant d(y, v_c^\alpha), \\ w_c^+ & \text{if } d(x, v_c^\alpha) > d(y, v_c^\alpha). \end{cases} \tag{5}$$

Let $S_1^\alpha = \{s \in S^\alpha \mid \text{there exist } x, y \in R^2 \text{ such that } s = s_\alpha(x, y)\}$, $S_1 = \bigcup_{\alpha \in A} S_1^\alpha$.

Lemma 3. *For any $x \in R^2$, there exists a worst-case scenario s for x such that $s \in S_1$.*

Proof. Let $x \in R^2$ be fixed. Then, according to Lemma 2, for some $\alpha \in A$ there exists $s' \in S^\alpha$ such that s' is a worst-case scenario for x. Let $y \in R^2$ be a 1-median for the scenario s'. Then $Z(x) = REGR(x, y) = F(s', x) - F(s', y)$. Value $F(s', x) - F(s', y)$ will not decrease if s' is replaced with $s_\alpha(x, y)$. Therefore, $s_\alpha(x, y)$ is also a worst-case scenario for x. □

Corollary 1. *Set S_1 is globally sufficient.*

Proof. The statement follows immediately from the definition of a globally sufficient set and Lemma 3. □

Lemma 4. $|S_1| = O(n^3)$.

Proof. To prove the lemma, it is sufficient to prove that $|S_1^\alpha| = O(n^3)$ for any $\alpha \in A$. Suppose that an $\alpha \in A$ is fixed. For any $x, y \in R^2$, let us define the set $R^2(x, y) = \{g \in R^2 \mid d(x, g) > d(y, g)\}$. The boundary of the open set $R^2(x, y)$ consists of segments of at most three straight lines. Each of these lines is either parallel to a coordinate axis, or is a (positive or negative) diagonal line. These lines will be called *boundary lines* for $R^2(x, y)$. For each boundary line l for $R^2(x, y)$, its *normal vector* is defined as the unit vector orthogonal to l and pointing outside of $R^2(x, y)$. Notice that the boundary lines along with their normal vectors uniquely define the set $R^2(x, y)$. Notice also that the set $V^\alpha \cap R^2(x, y)$ will not change if $R^2(x, y)$ is extended by shifting each boundary line that does not contain points of V^α in the direction of its normal vector until it hits a point from V^α. Thus, each possible set $V^\alpha \cap R^2(x, y)$ is uniquely defined by a choice of at most three lines going through points of V^α and making angles of $(\pi/4)i$, $i = 0, 1, 2, 3$ with coordinate axis, and a choice of one of the two possible normal vectors for each of these lines. The statement of the lemma follows immediately. □

Let G_1 be the grid obtained by drawing all possible vertical and horizontal lines through points $v \in V$, and let $N(G_1)$ be the set of nodes of that grid. Clearly $|N(G_1)| = O(n^2)$. Let G_{1v} (G_{1h}) denote the set of vertical (horizontal) lines of grid G_1.

For any $\alpha \in A$, let S_2^α be the set of scenarios $s \in S$ such that $s = s_\alpha(x, y)$ for some $x \in R^2$ and $y \in N(G_1)$; let $S_2 = \bigcup_{\alpha \in A} S_2^\alpha$. Clearly $S_2^\alpha \subset S_1^\alpha$, $\alpha \in A$ and $S_2 \subset S_1$, therefore $|S_2| = O(n^3)$.

Lemma 5. *Set S_2 is globally sufficient.*

Proof. The proof is similar to the proof of Lemma 3, taking into account that for any $s \in S^\alpha$, $\alpha \in A$, there is a 1-median for the scenario s that belongs to $N(G_1)$ [26]. □

For any $x \in R^2$ and $\alpha \in A$, let us define $S_2^\alpha(x) = \{s_\alpha(x, y) \mid y \in N(G_1)\}$, $S_2(x) = \bigcup_{\alpha \in A} S_2^\alpha(x)$. For any $\alpha \in A$, let $S_\alpha^*(x) = \{s_\alpha(x, y) \mid y \in N(G_1), y$ is a 1-median for the scenario $s_\alpha(x, y)\}$. Let $S^*(x) = \bigcup_{\alpha \in A} S_\alpha^*(x)$. The following result is obvious:

Lemma 6. *For any $x \in R^2$, the set of scenarios $S^*(x)$ is locally sufficient for x.*

Observe that the set $S^*(x)$ has cardinality $O(n^2)$ (because $|S_2(x)| = O(n^2)$ and $S^*(x) \subset S_2(x)$).

3.2. Linear programming formulation

Let us show how to formulate Problem ROBMED as a linear programming problem if some finite globally sufficient set $S' \subset S_1$ of scenarios is known. The linear program will have $\mathrm{O}(n)$ variables and $\mathrm{O}(|S'|+n)$ constraints; also, to write down the linear programming problem, one would have to obtain values $F^*(s)$ for all $s \in S'$.

Suppose that $S' \subset S_1$ is a finite globally sufficient set of scenarios. Then for any specific scenario $s \in S'$, $F(s,x)$ is a convex piecewise linear function (as a two-variable function of the two coordinates of x), and according to (3) $Z(x)$ is a convex piecewise linear function as well. For any $x = (x_1, x_2) \in R^2$ and any $s \in S$, value $F(s,x)$ can be written as

$$F(s,x) = \min \left\{ \sum_{c \in C} y_c^s \mid y_c^s \geqslant w_c^{(s)} d(v_c^{(s)}, x),\ \ c \in C \right\},$$

where $y_c^s, c \in C$ are real variables. (Observe that in an optimal solution to the above problem, the inequalities inside the brackets of the minimization operator will hold as equalities.) Thus, taking into account (3), we have the following optimization problem to compute $Z(x)$:

$$Z(x) = \min \left\{ z \,|\, z \geqslant \sum_{c \in C} y_c^s - F^*(s),\ \ s \in S';\ \ y_c^s \geqslant w_c^{(s)} d(v_c^{(s)}, x),\ \ c \in C,\ \ s \in S' \right\}.$$

Since for any $s \in S_1$ and any $c \in C$, weight $w_c^{(s)}$ is equal to either w_c^+ or w_c^- and $v_c^{(s)} = v_c^\alpha$ for some $\alpha \in A$, it is sufficient to use only $8n$ y-variables $y_c^{\alpha+}, y_c^{\alpha-},\ \ \alpha \in A,\ \ c \in C$ and to write Problem ROBMED as

$$\text{minimize } z, \tag{6}$$

$$z \geqslant \sum_{c \in C} y_c^s - F^*(s), \quad s \in S', \tag{7}$$

$$y_c^{\alpha+} \geqslant w_c^+ d(v_c^\alpha, x),\ \ y_c^{\alpha-} \geqslant w_c^- d(v_c^\alpha, x), \quad c \in C,\ \ \alpha \in A, \tag{8}$$

$$x \in R^2;\ z,\ y_c^{\alpha+},\ \ y_c^{\alpha-} \text{—real variables}, \quad c \in C,\ \ \alpha \in A, \tag{9}$$

where $y_c^s = y_c^{\alpha+}$ ($y_c^s = y_c^{\alpha-}$) if $w_c^{(s)} = w_c^+$ ($w_c^{(s)} = w_c^-$) and $v_c^{(s)} = v_c^\alpha$. Since for any $x = (x_1, x_2) \in R^2$ and any $v = (a_v, b_v) \in R^2$, $d(v,x) = |x_1 - a_v| + |x_2 - b_v| = \max\{x_1 - a_v + x_2 - b_v, x_1 - a_v - x_2 + b_v, -x_1 + a_v + x_2 - b_v, -x_1 + a_v - x_2 + b_v\}$, each of constraints (8) can be written as four linear constraints; for example, instead of $y_c^{SW+} \geqslant w_c^+ d(v_c^{SW}, x)$ we will write $y_c^{SW+} \geqslant w_c^+(x_1 - a_c^- + x_2 - b_c^-)$, $y_c^{SW+} \geqslant w_c^+(x_1 - a_c^- - x_2 + b_c^-)$, $y_c^{SW+} \geqslant w_c^+(-x_1 + a_c^- + x_2 - b_c^-)$, $y_c^{SW+} \geqslant w_c^+(-x_1 + a_c^- - x_2 + b_c^-)$. Then, (6)–(9) will become a linear programming problem with $|S'| + 32n$ constraints and $8n + 3$ variables $y_c^{\alpha+},\ y_c^{\alpha-},\ c \in C,\ \alpha \in A,\ x_1, x_2, z$.

Solving Problem ROBMED via the linear programming formulation (6)–(9) does not seem to be computationally attractive for large values of n because of the size of the linear programming problem: if S_1 or S_2 is used as S', then (6)–(9) has $\mathrm{O}(n^3)$ constraints and $\mathrm{O}(n)$ variables. In the remainder of the section, we develop an algorithm that solves Problem ROBMED in $\mathrm{O}(n^2 \log^2 n)$ time.

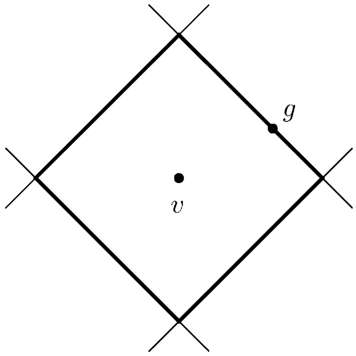

Fig. 2. An illustration for $Q(v, g)$.

3.3. An $O(n^2 \log^2 n)$ algorithm

For any $v \in V$ and $g \in N(G_1)$, consider the set $Q(v, g) = \{x \in R^2 \mid d(v, x) = d(v, g)\}$. Set $Q(v, g)$ is the boundary of the "diagonal" square with the center at v, point g on its boundary, and sides making angles $+\pi/4$ or $-\pi/4$ with the coordinate axes, see Fig. 2. Consider the set G_2 of all straight lines that contain sides of squares $Q(v, g)$, $v \in V$, $g \in N(G_1)$; G_2 forms a diagonal rectangular grid with $|G_2| = O(n^3)$ lines and $O(n^6)$ cells. Let G_{2+} (G_{2-}) denote the set of positive (negative) diagonal lines of the grid G_2.

Lemma 7. *If two points x' and x'' can be connected by a continuous curve in the plane that does not have common points with sides of squares $Q(v, g)$, $v \in V$, $g \in N(G_1)$, then $S^*(x') = S^*(x'')$.*

Proof. Observe that for any $y \in N(G_1)$ and any $\alpha \in A$, when x is moving along a continuous curve, scenario $s_\alpha(x, y)$ can change only when x is crossing a side of the square $Q(v_c^\alpha, y)$ for some $c \in C$. (This follows directly from the definitions.) The statement of the lemma follows immediately. □

Corollary 2. *For any cell of grid G_2, set $S^*(x)$ is the same for all interior points x of the cell.*

Notice also that for any cell of grid G_1, function $F(s, x)$ is linear inside the cell for any $s \in S_2$.

Lemma 8. *For any $x \in R^2$, values of all functions $F(s, x) - F^*(s)$, $s \in S^*(x)$, and their directional derivatives at x in a given direction can be computed in $O(n^2)$ total time.*

Proof. See the appendix. □

Lemma 9. *Problem* ROBMED *restricted to the intersection of a cell of grid G_1 with a cell of grid G_2 can be solved in $O(n^2)$ time.*

Proof. Let $\mathscr{A}$ be the intersection of a cell of grid G_1 with a cell of grid G_2. If $\mathscr{A}$ does not have interior points, then $\mathscr{A}$ is a point ($\mathscr{A}$ cannot be a line segment because no line of grid G_1 is parallel to any line of grid G_2), and the statement of the lemma is trivial. Suppose that $\mathscr{A}$ has interior points. Let x' be an interior point of $\mathscr{A}$. Then for any other interior point x of $\mathscr{A}$,

$$Z(x) = \max_{s \in S^*(x')} (F(s,x) - F^*(s)) \tag{10}$$

(according to Corollary 2 and Lemma 6). Eq. (10) holds also for any point x on the boundary of $\mathscr{A}$ (because of continuity of $Z(x)$). Therefore, Problem ROBMED restricted to $\mathscr{A}$ can be formulated as

$$\operatorname*{minimize}_{z \in R,\ x \in \mathscr{A}} \{z \mid z \geqslant F(s,x) - F^*(s),\ \ s \in S^*(x')\}. \tag{11}$$

Since $\mathscr{A}$ is a subset of a cell of grid G_1, functions $F(s,x) - F^*(s)$, $s \in S^*(x')$ are linear functions of x on $\mathscr{A}$, and according to Lemma 8 coefficients of these linear functions can be obtained in $O(n^2)$ time. Eq. (11) is a linear programming problem with 3 variables and $O(n^2)$ constraints; therefore, it can be solved in $O(n^2)$ time [28]. □

Finding the value $Z(x)$ at a point $x \in R^2$ and the directional derivative of $Z(x)$ at x in some specified direction (or in a fixed number of specified directions) will be called a *point test* at x.

Theorem 1 (*Complexity of the point test*). *For any $x \in R^2$, value $Z(x)$ and the directional derivatives of $Z(x)$ at x in a fixed number of directions can be computed in $O(n^2)$ time.*

Proof. Value $Z(x)$ can be obtained by taking the maximum of values $F(s,x) - F^*(s)$, $s \in S^*(x)$, that can be computed in $O(n^2)$ time according to Lemma 8.

Suppose a direction $r \in R^2$ is fixed. According to Lemma 8, directional derivatives of functions $F(s,x) - F^*(x)$, $s \in S^*(x)$ at x in the direction r can be computed in $O(n^2)$ time. If none of the points of $N(G_1)$ is a change point (the definition of a change point was given in the proof of Lemma 8), then the set $S^*(x)$ does not change for sufficiently small variations of x, and the directional derivative of $Z(x)$ at x in the direction r is the maximum of directional derivatives of functions $F(s,x) - F^*(s)$ at x in the direction r, where the maximum is taken over $s \in S^*(x)$ such that $F(s,x) - F^*(s) = Z(x)$, and thus can be computed in $O(n^2)$ time. If some points of $N(G_1)$ are change points, the situation is slightly more complicated because the set $S^*(x)$ may change for small variations of x. However, this difficulty can easily be avoided if we modify (5) as follows:

$$w_c^{(s)} = \begin{cases} w_c^+ & \text{if } d(x+\delta r, v_c^\alpha) > d(y, v_c^\alpha) \text{ for all sufficiently small positive values of } \delta, \\ w_c^- & \text{otherwise,} \end{cases}$$

keeping all other definitions as before. Clearly, all previous results will still hold after this change; but now set $S^*(x)$ does not change after sufficiently small shifts of x in the direction r, and therefore the directional derivative of $Z(x)$ in the direction r can be obtained from the directional derivatives of functions $F(s,x) - F^*(s)$, $s \in S^*(x)$, as described above. □

We will also use a *line test* which for a straight line L checks whether or not L contains an optimal solution to Problem ROBMED. In the former case, the line test finds an optimal solution to Problem ROBMED that belongs to L; in the latter case, the line test finds on which side from L the set X^* of optimal solutions to Problem ROBMED is. The details and the complexity of the line test will be discussed later.

The grids G_1 and G_2 are composed of four groups of parallel lines $G_{1h}, G_{1v}, G_{2-}, G_{2+}$. As follows from Lemma 1 and from convexity of the set X^*, for each of the four groups of parallel lines $G_{1h}, G_{1v}, G_{2-}, G_{2+}$ either there is a line of the group that contains a point from X^*, or all points of X^* lie in the strip between some two consecutive lines of the group. (Two lines of a group are called *consecutive* if there are no other lines of the group between them.)

Our approach to solving Problem ROBMED will be as follows. For each one of the four groups of parallel lines $G_{1h}, G_{1v}, G_{2-}, G_{2+}$, we perform a *group search* that either finds a line of the group that contains an optimal solution to Problem ROBMED, or finds two consecutive lines of the group such that the set of optimal solutions X^* is in the strip between them. The group search is performed using binary search on the lines of the group based on the line test. If at least one of the lines of grids G_1, G_2 has common points with X^*, an optimal solution to Problem ROBMED will be found at this stage; otherwise, the optimal cells (containing X^*) of grids G_1 and G_2 will be found at this stage. Given the optimal cells, an optimal solution to Problem ROBMED can be found in $O(n^2)$ time according to Lemma 9.

From this description, we see that we need to conduct the line test only for lines of grids G_1 and G_2. We will see that the line test for lines of grids G_1 and G_2 can be performed in $O(n^2 \log n)$ time, and using the line test, the binary search over the lines of each of the groups $G_{1h}, G_{1v}, G_{2-}, G_{2+}$ can be performed in $O(n^2 \log^2 n)$ time, which will result in the overall complexity of $O(n^2 \log^2 n)$ for Problem ROBMED.

Now let us discuss how to perform the line test for lines of grids G_1 and G_2. Consider a line L from one of the grids. Let Problem ROBMED(L) denote Problem ROBMED restricted to the line L (that is, in Problem ROBMED(L) it is required to minimize function $Z(x)$ over the line L). Let $X^*(L)$ be the set of optimal solutions to Problem ROBMED(L). Function $Z(x)$ is convex; if we solve Problem ROBMED(L) and find a point $x_L^* \in L$ that minimizes function $Z(x)$ over L, then the point test at x_L^* will either discover that x_L^* is an optimal solution to (unrestricted) Problem ROBMED, or will discover on which side from L the set X^* of optimal solutions to Problem ROBMED is. Thus, the line test has been reduced to solving Problem ROBMED(L).

Set $X^*(L)$ is convex because of convexity of $Z(x)$. An open interval of L that contains all points of $X^*(L)$ is called a *localization interval*. (Below, all intervals are assumed to be open unless stated otherwise.) Notice that there are $O(n^3)$ intersection points of the line L with other lines of grids G_1 and G_2 (further called *critical points*).

Lemma 10. *Given a localization interval that does not contain any critical points, Problem* ROBMED(L) *can be solved in* $O(n^2)$ *time.*

Proof. If a localization interval does not contain any critical points, then the localization interval belongs to the intersection of a cell of grid G_1 with a cell of grid G_2, and therefore according to the proof of Lemma 9 Problem ROBMED(L) can be formulated as a linear programming problem with $O(n^2)$ constraints and 3 variables, which can be solved in $O(n^2)$ time [28]. □

A straightforward way to solve Problem ROBMED(L) is as follows. Obtain all $O(n^3)$ critical points and sort them; this takes $O(n^3 \log n)$ time. Since $Z(x)$ is convex, the point test at a point x of the line L allows to determine whether x is optimal for Problem ROBMED(L), and if not, on which side from x the set $X^*(L)$ of optimal solutions to Problem ROBMED(L) is. Using binary search over the critical points and applying the point test at $O(\log n)$ critical points, we can find in $O(n^2 \log n)$ time a localization interval that does not contain critical points (or find an optimal solution to Problem ROBMED(L) if there is a critical point in $X^*(L)$). Then an optimal solution to Problem ROBMED(L) can be found in $O(n^2)$ time according to Lemma 10.

The straightforward approach described above conducts the line test in $O(n^3 \log n)$ total time; the bottleneck of complexity is finding explicitly and sorting the critical points. It is possible to improve the order of complexity of the line test to $O(n^2 \log n)$ (instead of $O(n^3 \log n)$) by exploiting special structure of the set of critical points, which allows to avoid obtaining explicitly and sorting this set. To reduce the order of complexity, we need to be able to do binary search over the critical points without explicitly generating all of them. Let us discuss the corresponding modification of the line test. We will need some auxiliary lemmas.

The following lemma is obvious (given the previous discussion) but we state it explicitly for reference purposes.

Lemma 11. *For any given set M of points of L with cardinality $O(n^2)$, it is possible to find in $O(n^2 \log n)$ time a localization interval that does not contain any points of M if M does not have common points with $X^*(L)$, or to find an optimal solution to Problem* ROBMED(L) *if $M \cap X^*(L)$ is not empty.*

The following lemma is the main tool for our algorithm.

Lemma 12. *Suppose that there is a real matrix $M = \|m_{ij}\|$ with k_1 rows and k_2 columns such that the entries in each row are nondecreasing from left to right and the entries in each column are nonincreasing from top to bottom.* (*Such a matrix will be called* sorted.) *The entries of M may be unknown, but we assume that given a row index i and a column index j, the entry m_{ij} can be found in $O(1)$ time. For any two real numbers $a, b, a \leqslant b$, let $r(a, b)$ denote the number of entries of M that are contained in the open interval (a, b). Suppose that two numbers $q_-, q_+, q_- < q_+$ are given. Then it is possible to find in $O(k_1 + k_2)$ time a number $\hat{q}$ such that $q_- \leqslant \hat{q} \leqslant q_+$ and $r(q_-, \hat{q}) \leqslant \frac{3}{4} r(q_-, q_+)$, $r(\hat{q}, q_+) \leqslant \frac{3}{4} r(q_-, q_+)$.*

Proof. The result of the lemma can be obtained using the standard "search using sorted matrices" technique [1,20]; details can be found in [6]. □

Lemma 13. *Suppose that a matrix M is as in the statement of the previous lemma with $k_1 = O(n^2)$ and $k_2 = O(n^2)$. If L is considered as a number line, and if the entries of M*

represent points on the line L (i.e., each entry m_{ij} of M represents the point of L that is m_{ij} units away from the origin in the positive direction), then it is possible to find in $O(n^2 \log n)$ *time a localization interval that does not contain any entries of M (or to find an optimal solution to Problem* ROBMED(*L*) *if the intersection of M and* $X^*(L)$ *is not empty).*

Proof. Using the result of the previous lemma and the point test, we can organize binary search so that at every iteration the number of entries of *M* that are contained in the current localization interval is reduced at least by factor $\frac{3}{4}$, and each iteration takes $O(n^2)$ time. The statement of the lemma follows immediately. □

To solve Problem ROBMED(*L*) in $O(n^2 \log n)$ time, we will use the following idea. We will identify several finite sets of points of the line *L* (called *covering sets*) that have the following properties:

(1) Together, they "cover" the set of critical points, that is, each critical point belongs to at least one of the covering sets.
(2) Each covering set either has cardinality $O(n^2)$, or it can be represented as a sorted matrix *M* with $O(n^2)$ rows and columns.

Then, according to Lemmas 11 and 13, for each covering set we can find in $O(n^2 \log n)$ time a localization interval that does not contain elements of the covering set (or to find an optimal solution to Problem ROBMED(*L*) if there is one in the covering set). The intersection of the localization intervals corresponding to the covering sets will give us a localization interval that does not contain any critical points; then, we can apply Lemma 10.

For any point $g \in N(G_1)$, let $l'(g)$ ($l''(g)$) denote the positive (negative) diagonal line that goes through *g*. Let K_L denote the set of critical points. In the following, a common point of two straight lines is called their *point of intersection* if the lines are *nonparallel*; that is, two lines that coincide are not considered to have points of intersection.

Consider the following sets of points of the line *L*. Set J_1 (J_2, J_3, J_4, respectively) is the set of points of intersection of *L* with the lines symmetrical to the lines $l'(g)$, $g \in N(G_1)$ about the vertical (horizontal, positive diagonal, negative diagonal, respectively) lines that go through points $v \in V$, see Fig. 3. Set J_5 (J_6, J_7, J_8, respectively) is the set of points of intersection of *L* with the lines symmetrical to the lines $l''(g)$, $g \in N(G_1)$, about the vertical (horizontal, positive diagonal, negative diagonal, respectively) lines that go through points $v \in V$. Observe that J_4 (J_7) is the set of points of intersection of the line *L* with the lines $l'(g)$, $g \in N(G_1)$ (with the lines $l''(g)$, $g \in N(G_1)$). Set J_9 is the set of points of intersection of *L* with the lines of grid G_1.

Observe that $K_L \subset \bigcup_{i=1}^{9} J_i$, because any line of grid G_2 is symmetrical to either $l'(g)$ or $l''(g)$ for some $g \in N(G_1)$ about either vertical, or horizontal, or positive diagonal, or negative diagonal line going through some $v \in V$. We consider separately the cases where *L* belongs to grid G_1 and where it belongs to grid G_2.

Case 1: Line *L* belongs to grid G_1. Suppose that *L* is horizontal (the case of a vertical line *L* is analogous). We will consider *L* as a number line with the positive direction defined by vector (1,0). Observe that $|J_9| = O(n)$, $|J_4| = O(n^2)$, $|J_7| = O(n^2)$, $|J_i| = O(n^3)$, $i = 1, 2, 3, 5, 6, 8$. Since J_4, J_7, and J_9 satisfy the condition of Lemma 11, we can obtain

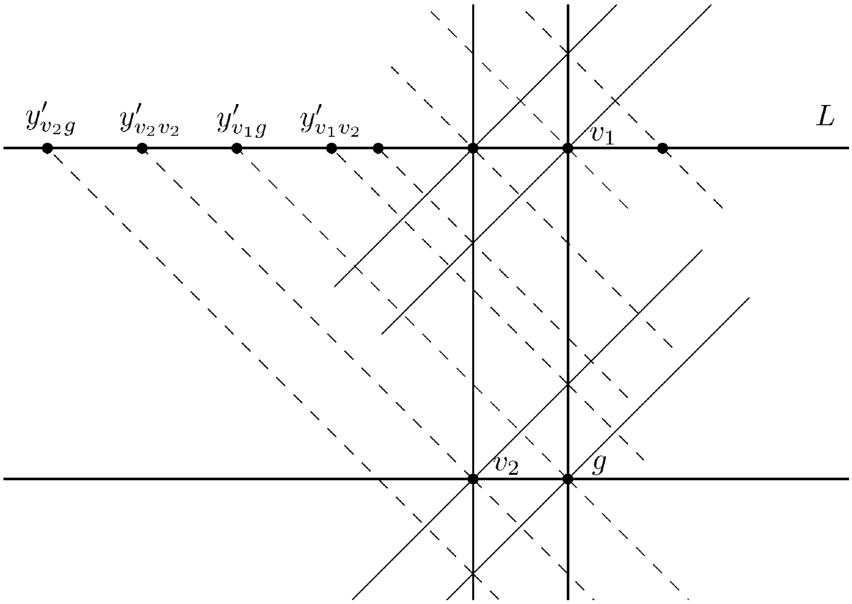

Fig. 3. An illustration for the set J_1.

in $O(n^2 \log n)$ time localization intervals that do not contain any points of J_4, J_7, J_9. According to the previous discussion, it remains to show that each one of the sets $J_i, i = 1, 2, 3, 5, 6, 8$ can be represented as a matrix M satisfying the conditions of Lemma 13.

Consider set J_1. For any $v \in V$ and $g \in N(G_1)$ let y'_{vg} denote the point of intersection of L with the line symmetrical to line $l'(g)$ about the vertical line going through v. Then $J_1 = \{y'_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nonincreasing values of $x_1(g) - x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of their first coordinates $x_1(v)$ (this can be done in $O(n^2 \log n)$ time because $|C| = n$ and $|N(G_1)| = O(n^2)$). Then the matrix $M = \|y'_{vg}\|$ satisfies the conditions of Lemma 13.

Consider set J_2. For any $v \in V$ and $g \in N(G_1)$, let y''_{vg} denote the point of intersection of L with the line symmetrical to line $l'(g)$ about the horizontal line going through v. Then $J_2 = \{y''_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nondecreasing values of $x_1(g) - x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of their second coordinates $x_2(v)$. Then the matrix $M = \|y''_{vg}\|$ satisfies the conditions of Lemma 13.

Consider set J_3. For any $v \in V$ and $g \in N(G_1)$, let y'''_{vg} denote the point of intersection of L with the line symmetrical to line $l'(g)$ about the positive diagonal line going through v. Then $J_3 = \{y'''_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nonincreasing values of $x_1(g) - x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of $x_1(v) - x_2(v)$. Then the matrix $M = \|y'''_{vg}\|$ satisfies the conditions of Lemma 13.

The cases of sets $J_5,\ J_6$, and J_8 are clearly similar.

Thus, sets $J_i, i = 1, \ldots, 9$ are covering sets for the set of critical points K_L, and the line test for L can be conducted in $O(n^2 \log n)$ time.

Case 2: Line L belongs to grid G_2. Suppose that line L is positive diagonal (the case where L is a negative diagonal line is analogous). We will consider L as a number line with the positive direction defined by vector (1,1). Observe that $|J_9| = \mathrm{O}(n)$, $J_3 = J_4 = J_5 = J_6 = \emptyset$, $|J_7| = \mathrm{O}(n^2)$. According to the previous discussion, it remains to show that each one of the sets J_1, J_2, J_8 can be represented as a matrix M satisfying the conditions of Lemma 13.

For any $v \in V$ and $g \in N(G_1)$, let y'_{vg} and y''_{vg} be defined as in Case 1.

Consider set J_1, $J_1 = \{y'_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nonincreasing values of $x_1(g) - x_2(g)$ and sort the points $v \in V$ in the order of nonincreasing values of their first coordinates $x_1(v)$. Then the matrix $M = \|y'_{vg}\|$ satisfies the conditions of Lemma 13.

Consider set J_2, $J_2 = \{y''_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nondecreasing values of $x_1(g) - x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of their second coordinates $x_2(v)$. Then the matrix $M = \|y''_{vg}\|$ satisfies the conditions of Lemma 13.

Consider set J_8. For any $v \in V$ and $g \in N(G_1)$, let y''''_{vg} denote the point of intersection of L with the line symmetrical to line $l''(g)$ about the negative diagonal line through v. Then $J_8 = \{y''''_{vg},\ v \in V,\ g \in N(G_1)\}$. Sort the points $g \in N(G_1)$ in the order of nonincreasing values of $x_1(g) + x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of $x_1(v) + x_2(v)$. Then the matrix $M = \|y''''_{vg}\|$ satisfies the conditions of Lemma 13.

Thus, sets J_i, $i = 1, \ldots, 9$ are covering sets for the set of critical points K_L, and the line test for L can be conducted in $\mathrm{O}(n^2 \log n)$ time.

We have proven

Theorem 2. *The line test for a line from grid G_1 or grid G_2 can be performed in $\mathrm{O}(n^2 \log n)$ time.*

To obtain an algorithm for Problem ROBMED with the time complexity of $\mathrm{O}(n^2 \log^2 n)$, it remains to show that for each one of the groups G_{1v}, G_{1h}, G_{2+}, G_{2-} it is possible to perform the group search in $\mathrm{O}(n^2 \log^2 n)$ time using the line test. This is clear for the groups G_{1v}, G_{1h}, because $|G_{1h}| = \mathrm{O}(n)$, $|G_{1v}| = \mathrm{O}(n)$, and the lines of G_{1v} (of G_{1h}) can be sorted in $\mathrm{O}(n \log n)$ time; subsequent application of the standard binary search uses $\mathrm{O}(\log n)$ line tests and, therefore, takes $\mathrm{O}(n^2 \log^2 n)$ time. The situation with the groups G_{2+} and G_{2-} is somewhat more complicated because $|G_{2+}| = \mathrm{O}(n^3)$, $|G_{2-}| = \mathrm{O}(n^3)$, and we cannot afford to obtain explicitly and sort the lines in the groups in order to perform binary search. However, special structure of these groups allows us to avoid generating explicitly and sorting all the lines of these groups. We use the same ideas (based on Lemmas 12 and 13) that allowed us to reduce complexity for the line test.

Consider, for example, group G_{2+}. Let G'_{2+} be the set of positive diagonal lines that go through the points of $N(G_1)$, and let G''_{2+} be the set of lines symmetrical to the lines of G'_{2+} about the positive diagonal lines that go through the points $v \in V$. Clearly $|G'_{2+}| = \mathrm{O}(n^2)$, $|G''_{2+}| = \mathrm{O}(n^3)$, $G_{2+} \subset (G'_{2+} \cup G''_{2+})$.

The area between two parallel lines (not including the lines themselves) is called a *localization strip* if it contains all points of X^*. To perform the group search for G_{2+}, we need to either find a line that has a common point with X^*, or to find a localization strip that does not contain any lines of G_{2+}.

The lines of G'_{2+} can be ordered in $\mathrm{O}(n^2 \log n)$ time; then, applying the regular binary search based on the line test, we can perform in $\mathrm{O}(n^2 \log^2 n)$ time the group search for G'_{2+}, that is, to find in $\mathrm{O}(n^2 \log^2 n)$ time a localization strip that does not contain any lines of G'_{2+} (or to find a line of G'_{2+} that has common points with X^*).

It remains to perform the group search for G''_{2+} in $\mathrm{O}(n^2 \log^2 n)$ time; then, either a line of G_{2+} that contains a point of X^* will be found, or two localization strips that do not contain lines from G'_{2+} and G''_{2+}, respectively, will be found, and the intersection of these strips will give a localization strip that does not contain lines from G_{2+}.

For any $v \in V$ and $g \in N(G_1)$, let l_{vg} denote the line symmetrical to the positive diagonal line that goes through g about the positive diagonal line that goes through v. Then, $G''_{2+} = \{l_{vg} \mid v \in V,\ g \in N(G_1)\}$. Any positive diagonal line is uniquely defined by the value $x_1 - x_2$ for the points of the line (which is the same for all points of any positive diagonal line). Therefore, lines $l_{vg},\ v \in V,\ g \in N(G_1)$ can be considered as real numbers (equal to the corresponding values of $x_1 - x_2$). Sort the points $g \in N(G_1)$ in the order of nonincreasing values of $x_1(g) - x_2(g)$, and sort the points $v \in V$ in the order of nonincreasing values of $x_1(v) - x_2(v)$ (this can be done in $\mathrm{O}(n^2 \log n)$ time); then, the matrix $M = \|l_{vg}\|$ satisfies the condition of Lemma 12 with $k_1 = \mathrm{O}(n)$ and $k_2 = \mathrm{O}(n^2)$. So, the group search for G''_{2+} can be performed in $\mathrm{O}(n^2 \log^2 n)$ time using the binary search technique discussed in the proof of Lemma 13 (where instead of the point test we use the line test, which accounts for the additional logarithm in the order of complexity). Thus, the group search for G_{2+} can be performed in $\mathrm{O}(n^2 \log^2 n)$ time.

The case of group G_{2-} is completely analogous. We obtain

Theorem 3. *Problem* ROBMED *can be solved in* $\mathrm{O}(n^2 \log^2 n)$ *time.*

4. Rectilinear center

In this section, we consider the case of $F(s, x) = F_2(s, x)$ and rectilinear distances. In Section 4.1, we consider the case where uncertainty is only in weights of customers (but not in locations). First, we observe that it is sufficient to consider only n "efficient" scenarios. Next, we observe that the problem decomposes into two one-dimensional problems, which can be solved in linear time if all values $F^*(s)$ for the n "efficient" scenarios are known. Last, using computational geometry, we show that all n values $F^*(s)$ for the "efficient" scenarios s can be computed in $\mathrm{O}(n \log n)$ time, which results in $\mathrm{O}(n \log n)$ algorithm for Problem ROBCEN.

In Section 4.2, we consider the case of uncertainty in both weights and locations of customers. The general logic in this case is rather similar to that of the previous case, but details are much more complicated. First, we introduce an auxiliary problem without uncertainty that is an extension of the standard weighted 1-center problem and can be solved in $\mathrm{O}(n)$ time. Next, we show that if we have optimal objective values of $4n$ auxiliary problems, then Problem ROBCEN can be formulated as a linear programming problem with 3 variables and $16n$ constraints, and therefore can be solved in linear time. Last, using computational geometry arguments and techniques, we show that the optimal objective values of the $4n$ auxiliary problems can be obtained in $\mathrm{O}(n \log n)$ total time, which results in $\mathrm{O}(n \log n)$ algorithm for Problem ROBCEN.

4.1. Uncertain weights, certain locations

First, let us consider the case of uncertainty in weights without uncertainty in locations, that is, $a_c^- = a_c^+ = a_c$, $b_c^- = b_c^+ = b_c$, $v_c = (a_c, b_c)$, $c \in C$, because this case will provide useful insights, and the algorithm in this case is much simpler.

Let s_c be the scenario that assigns weight w_c^+ to customer c and weight $w_{c'}^-$ to any other customer c', and let $S_3 = \{s_c, c \in C\}$.

Lemma 14. (a) Set S_3 *is globally sufficient.*
(b) $Z(x) = \max_{c\in C}\{w_c^+ d(v_c, x) - F^*(s_c)\}$.

Proof. Consider an $x \in R^2$. Let $\{w'_c, c \in C\}$ be the weights corresponding to a worst-case scenario s' for x, and let $y' \in R^2$ be an optimal solution to Problem CEN(s'). Let $c' \in \text{argmax}_{c\in C}\, w'_c d(v_c, x)$ (i.e. $w'_{c'} d(v_{c'}, x) \geqslant w'_c d(v_c, x)$ for any $c \in C$). Observe that $d(v_{c'}, x) \geqslant d(v_{c'}, y')$ (since $w'_{c'} d(v_{c'}, x) = F(s', x) \geqslant F(s', y') = F^*(s') \geqslant w'_{c'} d(v_{c'}, y')$). Therefore, value $F(s', x) - F(s', y')$ cannot decrease if we replace scenario s' with scenario $s_{c'}$. It cannot increase either, and y' must be an optimal solution to Problem CEN($s_{c'}$) (otherwise s' cannot be a worst-case scenario for x). Therefore, $s_{c'}$ is also a worst-case scenario for x, which proves part (a). Now, $Z(x) = F(s_{c'}, x) - F^*(s_{c'}) = w_{c'}^+ d(v_{c'}, x) - F^*(s_{c'})$. Since $w_c^+ d(v_c, x) - F^*(s_c) \leqslant Z(x)$ for any $c \in C$, we have part (b) as well. □

Remark. The lemma is a direct corollary of the general results on MRO problems with a minimax type of objective function from [3].

For convenience of presentation we will consider l_∞ distances instead of l_1 distances, i.e. we assume that for any $x = (x_1, x_2)$ and $y = (y_1, y_2)$, $d(x, y) = \max\{|x_1 - y_1|, |x_2 - y_2|\}$. (We notice that if there is no uncertainty in locations, the case of l_1 distances $d(x, y) = |x_1 - y_1| + |x_2 - y_2|$ can be reduced to the case of l_∞ distances by the change of coordinates $x'_1 = x_1 + x_2$, $x'_2 = x_1 - x_2$.)

For any $x = (x_1, x_2)$, according to Lemma 14 we have $Z(x) = \max\{Z_1(x_1), Z_2(x_2)\}$, where $Z_1(x_1) = \max_{c\in C}\{w_c^+(|x_1 - a_c|) - F^*(s_c)\}$, $Z_2(x_2) = \max_{c\in C}\{w_c^+(|x_2 - b_c|) - F^*(s_c)\}$. If values $F^*(s_c)$, $c \in C$ are known, then Problem ROBCEN decomposes into two one-dimensional problems: since $Z_1(x_1)$ depends only on x_1 and $Z_2(x_2)$ depends only on x_2, it is sufficient to find x_1^* that minimizes $Z_1(x_1)$ and x_2^* that minimizes $Z_2(x_2)$, and the point $x^* = (x_1^*, x_2^*)$ will be an optimal solution to Problem ROBCEN. The problem of minimizing function $Z_i(x_i)$, $i = 1, 2$, is the problem of finding a minmax of $2n$ linear functions which can be solved in $O(n)$ time [28]. Therefore, having values $F^*(s_c)$, $c \in C$, Problem ROBCEN can be solved in $O(n)$ time. Now it remains to find values $F^*(s_c)$, $c \in C$ efficiently. Next, we show that all these values can be found in $O(n \log n)$ total time, which results in $O(n \log n)$ time complexity for solving Problem ROBCEN in the case of no uncertainty in locations.

We have that for any $x = (x_1, x_2) \in R^2$ and $s \in S$, $F(s, x) = \max\{F'(s, x_1), F''(s, x_2)\}$, where $F'(s, x_1) = \max_{c\in C}\{w_c^{(s)}(|x_1 - a_c|)\}$, $F''(s, x_2) = \max_{c\in C}\{w_c^{(s)}(|x_2 - b_c|)\}$. Therefore, Problem CEN(s) decomposes into two one-dimensional problems;

$F^*(s) = \max\{F'(s), F''(s)\}$, where

$$F'(s) = \min_{x_1 \in R} \max_{c \in C}\{w_c^{(s)}(|x_1 - a_c|)\}, \quad F''(s) = \min_{x_2 \in R} \max_{c \in C}\{w_c^{(s)}(|x_2 - b_c|)\}. \tag{12}$$

According to (12), each one of $F'(s)$ and $F''(s)$ is a minmax of $2n$ linear functions and therefore can be found in $O(n)$ time; however, we have to solve these problems for n scenarios from S_3. We show that all values $F'(s), F''(s), s \in S_3$ can be obtained in $O(n \log n)$ total time.

To be specific, consider obtaining values $F'(s), s \in S_3$ (the case of values $F''(s), s \in S_3$ is completely similar). Consider scenario $s^- = \{w_c^-, c \in C\}$ that sets all weights to be equal to the corresponding lower bounds. Consider function $F^-(x_1) = \max_{c \in C} \{w_c^-(|x_1 - a_c|)\}$. The piecewise-linear nonnegative convex function $F^-(x_1)$ is the upper envelope of $2n$ linear functions and therefore can be obtained in $O(n \log n)$ time [31] (it can be represented by means of listing all its $O(n)$ corner points in increasing order, along with the values of the function at the corner points and the slopes of linear pieces). Let x_1^- be the point where $F^-(x_1)$ has its minimum value. Notice that $F'(s_c) = \min_{x_1 \in R} \max\{F^-(x_1), w_c^+(|x_1 - a_c|)\}$ because $w_c^+(|x_1 - a_c|) \geqslant w_c^-(|x_1 - a_c|)$ for any x_1.

Having obtained function $F^-(x_1)$, value $F'(s_c)$ can be found as follows. Find $r^- = \max\{x_1 \mid F^-(x_1) = w_c^+(|x_1 - a_c|),\ x_1 \leqslant a_c\}, r^+ = \min\{x_1 \mid F^-(x_1) = w_c^+(|x_1 - a_c|),\ x_1 \geqslant a_c\}$ (notice that it is possible that $r^- = -\infty$ or $r^+ = +\infty$, if the corresponding equation has no solutions). Values r^- and r^+ can be found in $O(\log n)$ time using binary search on the corner points of $F^-(x_1)$. Now, if $x_1^- \in [r^-, r^+]$, then $F'(s_c) = F^-(x_1^-)$; if $x_1^- \notin [r^-, r^+]$, then $F'(s_c) = \min\{F^-(r^-), F^-(r^+)\}$. So, having obtained function $F^-(x_1)$, value $F'(s_c)$ can be found in $O(\log n)$ time for any $c \in C$; therefore, all values $F'(s_c), c \in C$ can be found in $O(n \log n)$ time.

We have shown that Problem ROBCEN in the case of rectilinear distances and uncertainty only in weights can be solved in $O(n \log n)$ time.

4.2. Uncertain weights and uncertain locations

Consider now the case of uncertainty in both weights and locations, $a_c^- \leqslant a_c^+,\ b_c^- \leqslant b_c^+$, $c \in C$. We again consider l_1 distances $d(x, y) = |x_1 - y_1| + |x_2 - y_2|$ for any $x = (x_1, x_2)$, $y = (y_1, y_2)$. (When locations of customers are uncertain, it is no longer true that the case of l_1 distances is equivalent to the case of l_∞ distances.) For any compact set $U \subset R^2$ and any point $x \in R^2$, let us define $d(x, U) = \min\{d(x, y) \mid y \in U\}$. Suppose that for each $c \in C$, a rectangle $T_c \subset R^2$ with sides parallel to the coordinate axes is fixed (in the following, these rectangles will be either the rectangles of uncertainty Γ_c or single points), and let $T = \{T_c, c \in C\}$. For any point $x \in R^2$ and a vector of weights $W = \{w_c, c \in C\}$, let us define

$$\tilde{F}(x, W, T) = \max_{c \in C} w_c d(x, T_c)$$

That is, $\tilde{F}(x, W, T)$ is the maximum of weighted distances from x to the corresponding rectangles T_c. Consider the following auxiliary problem

Problem CEN1(W, T). Find $x \in R^2$ so as to minimize $\tilde{F}(x, W, T)$.

Problem CEN1(W, T) is an extension of the standard weighted 1-center problem to the case where customers are represented by rectangles T_c rather than points.

Since for any $c \in C$, function $d(x, T_c)$ is the upper envelope of at most 9 linear functions of $x = (x_1, x_2)$, $\tilde{F}(x, W, T)$ as a function of x_1, x_2 is the upper envelope of at most $9n$ linear functions of two variables x_1, x_2; so, Problem CEN1(W, T) can be written as a linear programming problem with 3 variables and at most $9n$ constraints and solved in O(n) time according to [28].

For any $\alpha \in A$ and $c \in C$, let us define value $\tilde{F}^*(\alpha, c)$ as follows: $\tilde{F}^*(\alpha, c)$ is the optimal objective function value for Problem CEN1(W, T), where in the vector of weights W customer c has weight w_c^+ and any other customer $c' \in C$ has weight $w_{c'}^-$, T_c is the point v_c^α, and for any other $c' \in C$, $T_{c'}$ is the corresponding rectangle of uncertainty $\Gamma_{c'}$. These W and T will be denoted $W(c)$ and $T(\alpha, c)$, respectively, and the corresponding Problem CEN1(W, T) will be referred to as Problem CEN1($W(c), T(\alpha, c)$); therefore, $\tilde{F}^*(\alpha, c) = \min_{x \in R^2} \tilde{F}(x, W(c), T(\alpha, c))$.

Lemma 15. *For any* $x \in R^2$, $Z(x) = \max_{\alpha \in A,\ c \in C}(w_c^+ d(v_c^\alpha, x) - \tilde{F}^*(\alpha, c))$.

Proof. For any $\alpha \in A$ and $c \in C$, let us define $y_{\alpha,c} \in R^2$ and $s_{\alpha,c} \in S$ as follows:

$y_{\alpha,c}$ is the optimal solution to Problem CEN1($W(c), T(\alpha, c)$) (if this problem has more than one optimal solution, an arbitrary rule, e.g. lexicographic, can be used to choose one of them);

$s_{\alpha,c}$ is the scenario where customer c is located at v_c^α and has weight w_c^+, and for any $c' \in C$, $c' \neq c$, customer c' has weight $w_{c'}^-$ and is located at the point of $\Gamma_{c'}$ closest to $y_{\alpha,c}$.

Let us fix an arbitrary $x \in R^2$. First, observe that for any $\alpha \in A$ and $c \in C$, $w_c^+ d(v_c^\alpha, x) - \tilde{F}^*(\alpha, c) \leqslant Z(x)$ (because $w_c^+ d(v_c^\alpha, x) \leqslant F(s_{\alpha,c}, x)$, $\tilde{F}^*(\alpha, c) = F(s_{\alpha,c}, y_{\alpha,c})$, and therefore $w_c^+ d(v_c^\alpha, x) - \tilde{F}^*(\alpha, c) \leqslant F(s_{\alpha,c}, x) - F(s_{\alpha,c}, y_{\alpha,c}) \leqslant Z(x)$). To prove the lemma, it remains to show that there exist $\alpha \in A$ and $c \in C$ such that $w_c^+ d(v_c^\alpha, x) - \tilde{F}^*(\alpha, c) = Z(x)$.

A pair (s', y'), $s' \in S$, $y' \in R^2$, is called a *worst-case pair for x*, if $Z(x) = F(s', x) - F(s', y')$. Let (s', y') be a worst-case pair for x. Let $c' \in \arg\max_{c \in C} \{w_c^{(s')} d(x, v_c^{(s')})\}$. Then, $Z(x) = F(s', x) - F(s', y') = w_{c'}^{(s')} d(x, v_{c'}^{(s')}) - F(s', y')$. Suppose that $v_c^{(s')}$ belongs to the α'-quadrant for x for some $\alpha' \in A$ (for the definition of an α'-quadrant for x, see Section 2). Then, value $F(s', x) - F(s', y')$ will not decrease if in the scenario s' we replace $v_{c'}^{(s')}$ with $v_{c'}^{\alpha'}$, $w_{c'}^{(s')}$with $w_{c'}^+$, and $w_c^{(s')}$ with w_c^- for any $c \neq c'$ (this can be shown using an argument similar to that used in the proof of Lemma 14). It cannot increase either because (s', y') is a worst-case pair for x. The scenario obtained after this modification of s' will be called s''. We see that (s'', y') is also a worst-case pair for x and $F(s'', x) = w_{c'}^+ d(x, v_{c'}^{\alpha'})$. Now, observe that

$$F(s'', y') \geqslant \tilde{F}^*(\alpha', c') = F(s_{\alpha',c'}, y_{\alpha',c'}) \tag{13}$$

and

$$F(s'', x) = w_{c'}^+ d(x, v_{c'}^{\alpha'}) \leqslant F(s_{\alpha',c'}, x). \tag{14}$$

Therefore

$$F(s_{\alpha',c'}, x) - F(s_{\alpha',c'}, y_{\alpha',c'}) \geqslant F(s'', x) - F(s'', y') = Z(x).$$

Recalling the definition of $Z(x)$, we see that $F(s_{\alpha',c'}, x) - F(s_{\alpha',c'}, y_{\alpha',c'}) = Z(x) = F(s'', x) - F(s'', y')$ and therefore $(s_{\alpha',c'}, y_{\alpha',c'})$ is a worst-case pair for x. Taking into account (13) and (14), we see that $F(s_{\alpha',c'}, x) = F(s'', x) = w^+_{c'} d(x, v^{\alpha'}_{c'})$. Using $F(s_{\alpha',c'}, y_{\alpha',c'}) = \tilde{F}^*(\alpha', c')$, we have $Z(x) = w^+_{c'} d(x, v^{\alpha'}_{c'}) - \tilde{F}^*(\alpha', c')$. The lemma is proven. □

Function $w^+_c d(v^\alpha_c, x)$ is the upper envelope of four linear functions of $x = (x_1, x_2)$, and therefore $Z(x)$ is the upper envelope of $16n$ linear functions. Thus, if all values $\tilde{F}^*(\alpha, c),\ \alpha \in A,\ c \in C$ are known, Problem ROBCEN can be formulated as a linear programming problem with 3 variables and $16n$ constraints and solved in $O(n)$ time [28]. (Another way to show that given values $\tilde{F}^*(\alpha, c),\ \alpha \in A,\ c \in C$ Problem ROBCEN can be solved in $O(n)$ time is to decompose the problem into two one-dimensional problems using Lemma 15 and an argument similar to that used in the previous subsection). Since Problem CEN1(W, T) can be solved in $O(n)$ time (as observed above), all $4n$ values $\tilde{F}^*(\alpha, c)$ can be obtained in $O(n^2)$ time. Below, we show that in fact all $4n$ values $\tilde{F}^*(\alpha, c)$ can be obtained in $O(n \log n)$ total time, which will result in $O(n \log n)$ time complexity for Problem ROBCEN.

Let $W^- = \{w^-_c, c \in C\}$ and $\tilde{T} = \{\Gamma_c,\ c \in C\}$. Function $H(x) = \tilde{F}(x, W^-, \tilde{T})$ is the upper envelope of at most $9n$ linear functions (because function $d(x, \Gamma_c)$ for any $c \in C$ is the upper envelope of at most 9 linear functions). Vertices, edges, and faces of this envelope (we will refer to them as to vertices, edges, and faces of $H(x)$) along with all "vertex-face" and "face-vertex" incidence lists can be found in $O(n \log n)$ time using convex hull algorithms [17,32]. The total number of vertices, edges, and faces is $O(n)$ [31]. We will refer to values of function $H(x)$ as to z-coordinates of the corresponding points of the upper envelope.

We say that a function $f(x)$ defined on R^2 satisfies the *8-gon property* if for any $z \in R$, the set $\{x \in R^2 \mid f(x) \leqslant z\}$ is a convex k-gon with $k \leqslant 8$ such that the angles between the sides of this k-gon and the coordinate axes are multiples of $\pi/4$ ($k = 0$ if the set $\{x \in R^2 \mid f(x) \leqslant z\}$ is empty or consists of a single point). For any $c \in C$, function $d(x, \Gamma_c)$ satisfies the 8-gon property; so does function $H(x)$ as it inherits the 8-gon property from functions $d(x, \Gamma_c)$, $c \in C$.

Let $z_1, z_2, \ldots, z_q$ be the distinct z-coordinates of the vertices of $H(x)$ in the increasing order ($q = O(n)$). We will refer to values $z_1, z_2, \ldots, z_q$ as to "levels". Let H_i denote the polygon $\{x \mid H(x) \leqslant z_i\}$. Having obtained $H(x)$ with its vertex-faces incidence structure, all polygons H_i, $i \in \{1, 2, \ldots, q\}$ can be obtained in $O(n)$ time as follows. For each face of $H(x)$ we find the levels of its lowest and highest vertices, say z_i and z_j, respectively, and assign the face to levels $z_i, z_{i+1}, \ldots, z_{j-1}$. (If a face is unbounded and does not have a highest vertex, then the face is assigned to levels $z_i, z_{i+1}, \ldots, z_q$, where z_i is the level of the lowest vertex of the face. If $H(x)$ has a horizontal face, then this face is at the bottom of $H(x)$ and is not assigned to any level.) The lowest and highest vertices of all faces can be found in $O(n)$ total time because there are $O(n)$ pairs of incident vertices and faces [31]. Because $H(x)$ satisfies the 8-gon property, no more than 8 faces can be assigned to each level z_i, $i = 1, 2, \ldots, q$. The faces assigned to a level z_i define the polygon H_i.

We need to show that all values $\tilde{F}^*(\alpha, c)$, $\alpha \in A$, $c \in C$ can be obtained in $O(n \log n)$ time. Let us fix some $\alpha \in A$ and $c \in C$. Notice that $\tilde{F}(x, W(c), T(\alpha, c)) = \max\{H(x), w_c^+ d(v_c^\alpha, x)\}$ and that $\tilde{F}^*(\alpha, c) = \min_{x\in R^2} \tilde{F}(x, W(c), T(\alpha, c))$ is the smallest value $z \in R$ such that the polygon $\{x \mid H(x) \leqslant z\}$ (which is a k-gon with $k \leqslant 8$) and the square $\{x \mid w_c^+ d(v_c^\alpha, x) \leqslant z\}$ have nonempty intersection. For any $i \in \{1, 2, \ldots, q\}$, the intersection of H_i and the square $\{x \mid w_c^+ d(v_c^\alpha, x) \leqslant z_i\}$ can be found in O(1) time. This intersection is nonempty if and only if $\tilde{F}^*(\alpha, c) \leqslant z_i$.Thus, for any $i \in \{1, 2, \ldots, q\}$ it takes O(1) time to check whether $\tilde{F}^*(\alpha, c) \leqslant z_i$;notice also that $z_1 \leqslant \tilde{F}^*(\alpha, c) < +\infty$. Applying binary search on $z_1, z_2, \ldots, z_q$, in $O(\log n)$ time we either discover that $\tilde{F}^*(\alpha, c) = z_1$, or we find the largest $i \in \{1, 2, \ldots, q\}$ such that $\tilde{F}^*(\alpha, c) > z_i$;let it be i^*. In the latter case value $\tilde{F}^*(\alpha, c)$ is defined by the faces of $H(x)$ assigned to level z_{i^*} (no more than 8 faces) and the four faces of function $w_c^+ d(v_c^\alpha, x)$ (which is the upper envelope of four linear functions), and can be found in O(1) time using a linear programming problem with 3 variables and at most 12 constraints. Therefore, all values $\tilde{F}^*(\alpha, c)$, $\alpha \in A$, $c \in C$ can be obtained in $O(n \log n)$ time. We have proven

Theorem 4. *Problem* ROBCEN *in the case of rectilinear distances can be solved in* $O(n \log n)$ *time.*

5. Euclidean median

Suppose that distances are Euclidean, that is, for any $x = (x_1, x_2) \in R^2$ and $y = (y_1, y_2) \in R^2$, $d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$, and suppose that $F(s, x) = F_1(s, x)$. For the Euclidean case, we assume that there is no uncertainty in locations (i.e. $a_c^- = a_c^+ = a_c$, $b_c^- = b_c^+ = b_c$, $v_c = (a_c, b_c)$, $c \in C$), and all uncertainty is in weights. Therefore, a scenario corresponds to assigning weights to customers. The reasons for this assumption were stated in Section 2.

For any $x, y \in R^2$, let us define scenario $s^*(x, y) = \{w_c^*(x, y), c \in C\}$ as follows:

$$w_c^*(x, y) = \begin{cases} w_c^- & \text{if } d(x, v_c) \leqslant d(y, v_c), \\ w_c^+ & \text{if } d(x, v_c) > d(y, v_c). \end{cases}$$

Then, it is clear that $REGR(x, y) = F(s^*(x, y), x) - F(s^*(x, y), y)$. Let $\hat{S}_1$ be the set of scenarios $s \in S$ such that $s = s^*(x, y)$ for some $x, y \in R^2$. It is well-known that $|\hat{S}_1| = O(n^2)$ (e.g. [17, Theorem 3.1, p. 47]; different scenarios of $\hat{S}_1$ correspond to different ways to partition n points on the plane into two disjoint sets by a straight line).

Lemma 16. *Set $\hat{S}_1$ is globally sufficient.*

Proof. For any $x \in R^2$, there exists a worst-case scenario s for x such that $s \in \hat{S}_1$; this is straightforward to show using an argument similar to that used in the proof of Lemma 3. The statement of the lemma follows immediately. □

In the case of Euclidean distances, we will be interested in solving Problem ROBMED only approximately, because even Problem MED(s) for a specific scenario s, which is the classical Weber problem [36] and a special case of Problem ROBMED, is not well-solvable exactly, and available methods find only approximate solutions to Problem MED(s) (see., e.g., [9,13,22,25]). For an $\varepsilon > 0$, a feasible solution x for an optimization problem is called an ε-*optimal solution* for the problem, if the objective function value at x differs from the optimal objective function value by no more than ε.

A natural approach to solving Problem ROBMED is as follows:

(1) Obtain the set $\hat{S}_1$.
(2) For each $s \in \hat{S}_1$, solve Problem MED(s) approximately with a precision $\varepsilon_1 > 0$; that is, obtain values $\hat{F}^*(s)$, $s \in \hat{S}_1$ such that $0 \leqslant \hat{F}^*(s) - F^*(s) \leqslant \varepsilon_1$.
(3) Solve the following problem with a precision $\varepsilon_2 > 0$:

$$\underset{x \in R^2}{\text{minimize}}\ \hat{Z}(x), \tag{15}$$

where

$$\hat{Z}(x) = \max_{s \in \hat{S}_1} (F(s, x) - \hat{F}^*(s)). \tag{16}$$

It is clear that $\hat{Z}(x) \leqslant Z(x)$ and $Z(x) - \hat{Z}(x) \leqslant \varepsilon_1$ for any $x \in R^2$; therefore, we have the following

Lemma 17. *An ε_2-optimal solution x^* to* (15) *will be an* $(\varepsilon_1 + \varepsilon_2)$*-optimal solution to Problem* ROBMED.

$\hat{Z}(x)$ is a convex function, and (15) can be solved approximately by iterative methods of convex optimization. Each iteration of such methods typically consists of computing value of $\hat{Z}(x)$ and its subgradient at some specific $x \in R^2$.

Lemma 18. *For any $x \in R^2$, all* $O(n^2)$ *values $F(s, x)$, $s \in \hat{S}_1$ can be obtained in* $O(n^2)$ *total time.*

Proof. Clearly, each $s \in \hat{S}_1$ corresponds to a partition of the set of customers by a straight line passing through the location of a customer. Rotating this straight line about the customer location on the line generates $O(n)$ scenarios from $\hat{S}_1$, and values $F(s, x)$ for these $O(n)$ scenarios (for a fixed $x \in R^2$) are obtained in $O(n)$ total time by updating dynamically the value $F(s, x)$ for the current scenario during such a rotation (for details, see [6]). Since there are n customers, all values $F(s, x)$, $s \in \hat{S}_1$ are generated in $O(n^2)$ time. □

If values $\hat{F}^*(s)$, $s \in \hat{S}_1$ have already been obtained, then according to Lemma 18 the maximizer in (16) can be obtained in $O(n^2)$ time. If a scenario s' is a maximizer in (16), then a subgradient of $F(s', x)$ at x will also be a subgradient of $\hat{Z}(x)$ at x. Therefore, we have

Lemma 19. *When values $\hat{F}^*(s)$, $s \in \hat{S}_1$ have been obtained, the value of function $\hat{Z}(x)$ and its subgradient at any point $x \in R^2$ can be obtained in $O(n^2)$ time.*

Thus, finding an $(\varepsilon_1 + \varepsilon_2)$-optimal solution to Problem ROBMED has been reduced to solving $O(n^2)$ regular Weber problems (Problems MED(s), $s \in S'$) with precision ε_1 and minimizing convex function $\hat{Z}(x)$ with precision ε_2; the value and a subgradient of $\hat{Z}(x)$ at any point $x \in R^2$ can be obtained in $O(n^2)$ time.

6. Euclidean center

As in the previous section, suppose that distances are Euclidean and uncertainty is only in weights. Suppose that $F(s, x) = F_2(s, x)$. Let scenarios s_c, $c \in C$ be as defined in Section 4.1.

Lemma 20. $Z(x) = \max_{c \in C} \{w_c^+ d(v_c, x) - F^*(s_c)\}$.

Proof. The proof is completely similar to the proof of Lemma 14, part (b). □

Since Problem CEN(s) can be solved in $O(n)$ time [16,29,1], all values $F^*(s_c)$, $c \in C$ can be obtained in $O(n^2)$ time.

Having obtained values $F^*(s_c)$, $c \in C$, Problem ROBCEN is reduced to the problem of minimizing $Z(x)$. For any $c \in C$, let $Z_c(x) = w_c^+ d(v_c, x) - F^*(s_c)$ and let $\bar{Z}_c = \{(x, z) \mid x \in R^2,\ z \in R,\ z \geqslant Z_c(x)\}$. The graph of function $Z(x)$ is the upper envelope E of the functions $Z_c(x)$, $c \in C$. E is the boundary of the intersection of the cones $\bar{Z}_c$, $c \in C$. The problem of minimizing $Z(x)$ seems very similar to the classical Euclidean weighted 1-center problem (Problem CEN(s)), with the exception that in Problem CEN(s) vertices of all cones are at the same height. So, it seems natural to try to extend known efficient algorithms for Problem CEN(s) to the problem of minimizing $Z(x)$. The fastest algorithms for solving the classical Euclidean weighted 1-center problem are the linear-time algorithms of Dyer [16] and Megiddo [29] based on the multidimensional search technique of [28]. Unfortunately, they do not seem to be applicable to our case; the property that the vertices of the cones are at the same height in Problem CEN(s) is essential for these algorithms. It appears that the parametric-search based algorithm of Megiddo [27] can be extended to the problem of minimizing $Z(x)$ without major modifications in an appropriate model of computation (the model of computation should allow infinite-precision real arithmetic, finding in constant time intersection points and tangency points of certain curves, etc.); the complexity of the algorithm is $O(n \log^3 n (\log\ \log n)^2)$. Since $O(n^2)$ time has been spent on obtaining values $F^*(s_c)$, $c \in C$, this will result in $O(n^2)$ time complexity for Problem ROBCEN. However, since $O(n^2)$ time is already used for obtaining values $F^*(s_c)$, $c \in C$, it is not attractive to use complicated and difficult to implement techniques such as the parametric search for minimizing $Z(x)$ if $Z(x)$ can be minimized in almost quadratic time using simpler approaches. Below, we describe a geometric algorithm that minimizes $Z(x)$ in time $O(n^2 2^{\alpha(n)} \log^2 n)$, where $\alpha(n)$ is the inverse Ackermann function [2] that grows very slowly and can be considered as a constant for any practical purposes (for any possible “practical” values of n, $\alpha(n) < 4$ and therefore $2^{\alpha(n)} < 16$). The algorithm uses classical results and constructions from computational geometry, and its logic is much simpler than that of the parametric approach. For simplicity of presentation and to avoid tedious consideration of practically unimportant special cases, we assume that the points $v_a, a \in C$ are in *general position*, that is, they are all distinct and no three of them belong to the same straight line.

We need the following:

Lemma 21. *Suppose a finite set $K \subset R$ of distinct "candidate" values is known such that the optimal objective function value $z^* = \min_{x \in R^2} Z(x)$ for Problem* ROBCEN *belongs to K. Then, given the set K and values $F^*(s_c), c \in C$, Problem* ROBCEN *can be solved in* $O(n \log n \log |K| + |K|)$ *time.*

Proof. Suppose a finite set K that satisfies the condition of the lemma and the values $F^*(s_c), c \in C$ are given. For any value $z \in R$, it is possible to check in $O(n \log n)$ time whether $z \geqslant z^*$. This can be done as follows. Consider the circles $\Psi_c(z) = \{x \in R^2 \mid Z_c(x) \leqslant z\}, c \in C$. The circles $\Psi_c(z)$ correspond to horizontal slices of the cones $\bar{Z}_c$ at height z. Observe that $z \geqslant z^*$ if and only if the intersection of the circles $\Psi_c(z), c \in C$ is not empty. The intersection of n circles can be computed in $O(n \log n)$ time [15]; therefore, for any $z \in R$ it is possible to check in $O(n \log n)$ time whether $z \geqslant z^*$.

Since $z^* \in K$, z^* can be found now using binary search over the elements of K. At each iteration, we find the median z' of the current set K and check whether $z' \geqslant z^*$ using the test described above. If $z' \geqslant z^*$, the elements of K which are greater than z' can be discarded; if $z' < z^*$, the elements of K which are smaller or equal to z' can be discarded. After $O(\log |K|)$ iterations, value z^* is found. The total time spent on finding medians and discarding elements is $O(|K| + \frac{1}{2}|K| + \frac{1}{4}|K| + \cdots) = O(|K|)$, so the total time spent on finding z^* is $O(n \log n \log |K| + |K|)$. After finding z^*, an optimal solution to Problem ROBCEN can be found by obtaining the intersection of the circles $\Psi_c(z^*), c \in C$. □

Now it remains to find a reasonably small set K of candidate values that would contain z^*. Below, we discuss how to find such a set K of cardinality $O(n^2 2^{\alpha(n)})$ in $O(n^2 2^{\alpha(n)} \log^2 n)$ time, where $\alpha(n)$ is the inverse Ackermann function. Together with Lemma 21, this will provide an $O(n^2 2^{\alpha(n)} \log^2 n)$ algorithm for solving Problem ROBCEN.

The upper envelope E of functions $Z_c(x), c \in C$ represents a two-dimensional manifold and consists of vertices, edges, and faces. Let $x^* = (x_1^*, x_2^*)$ be an optimal solution to Problem ROBCEN. The point $p^* = (x^*, z^*) \in R^3$ is a lowest point of the envelope E (where height is associated with z-coordinate); it can be a vertex, an interior point of an edge, or an interior point of a face of E. Let us consider separately all three cases.

Face: Suppose that p^* is an interior point of a face of E. Then x^* is a local minimum of the function $Z_c(x)$ that corresponds to the face. The function $Z_c(x)$ has a unique local (which is also global) minimum at $x = v_c$. So, in this case $z^* = -F^*(s_c)$ for some $c \in C$. We include the n values $-F^*(s_c), c \in C$ in K; this makes sure that $z^* \in K$ if there is an optimal solution to Problem ROBCEN that corresponds to an interior point of a face of the envelope E.

Edge: Suppose p^* is an interior point of the edge defined by functions $Z_a(x)$ and $Z_b(x)$ for some $a, b \in C$. Then, $z^* = \min\{z \mid z = Z_a(x) = Z_b(x),\ x \in R^2\}$. For any particular $z_0 \in R$, the set of solutions of equation $Z_a(x) = z_0$ (of equation $Z_b(x) = z_0$) is the circumference of the circle $\Psi_a(z_0)$ ($\Psi_b(z_0)$) (defined in the proof of Lemma 21). The minimum value of z_0 such that the two circumferences intersect is achieved when the two circumferences are tangent. (In the degenerate case $v_a = v_b$ the two circumferences may coincide, but we assumed that points v_c, $c \in C$ are in general position, so $v_a \neq v_b$). Then x^* lies on the line l that passes through v_a and v_b. Let $x = x(t)$ be a parameterization of the line l. Then function $Z_a(x(t))$ (function $Z_b(x(t))$ is a piecewise linear function of t with 2 linear pieces, and x^* (and, therefore, z^*) can be found in O(1) time by solving the equation $Z_a(x(t)) = Z_b(x(t))$ and choosing the best of the solutions. We conclude that for any pair of customers $a, b \in C$, it is possible to find in O(1) time a candidate value $z'_{a,b}$ that is equal to z^* if there is an optimal solution to Problem ROBCEN that corresponds to an interior point of the edge defined by functions $Z_a(x)$ and $Z_b(x)$. We include the O(n^2) values$z'_{a,b}$, $a \in C, b \in C$ into K; this makes sure that $z^* \in K$ if there is an optimal solution to Problem ROBCEN that corresponds to an interior point of an edge of the envelope E.

Vertex: The case where p^* is a vertex of E is more difficult because a vertex is defined by a triple of functions $Z_a(x), Z_b(x), Z_c(x)$, $a, b, c \in C$, and there may be $\Theta(n^3)$ such triples. We say that a customer $a \in C$ is dominated by a customer $b \in C$ if $Z_a(x) \leqslant Z_b(x)$ for all $x \in R^2$. A customer $a \in C$ is called *dominated* if it is dominated by at least one other customer. Notice that customer a is dominated by customer b if and only if $w_a^+ \leqslant w_b^+$ and $Z_a(v_b) \leqslant Z_b(v_b)$ (that is, $Z_a(v_b) \leqslant -F^*(s_b)$). Therefore, it takes O(1) time to check whether a customer a is dominated by a customer b, and therefore it takes O(n^2) time to identify all dominated customers. Notice that deleting all dominated customers from C does not affect the envelope E. So, we will assume that there are no dominated customers in C.

Lemma 22. *Suppose that* $c, a \in C$ *and* $x' \in R^2$, *and suppose that* $w_c^+ \geqslant w_a^+$ *and* $Z_c(x') \geqslant Z_a(x')$. *Then for any* $t \geqslant 1$, $Z_c(v_c + t(x' - v_c)) \geqslant Z_a(v_c + t(x' - v_c))$.

Proof. For any $t \geqslant 1$, let $x'(t) = v_c + t(x' - v_c)$. Let $\Delta_c = d(v_c, x'(t)) - d(v_c, x')$ and $\Delta_a = d(v_a, x'(t)) - d(v_a, x')$. It suffices to show that $Z_c(x'(t)) - Z_c(x') \geqslant Z_a(x'(t)) - Z_a(x')$ or $w_c^+ \Delta_c \geqslant w_a^+ \Delta_a$. This inequality follows from $\Delta_c \geqslant \Delta_a$ since $w_c^+ \geqslant w_a^+ > 0$ and $\Delta_c = (t-1)d(v_c, x') \geqslant 0$. The inequality $\Delta_c \geqslant \Delta_a$ is equivalent to the triangle inequality $d(x'(t), x') + d(v_a, x') \geqslant d(v_a, x'(t))$. □

Consider a lexicographical order "$\prec$" of customers $c \in C$ determined by $(w_c^+, F^*(s_c))$, $c \in C$. That is, for any $a, b \in C$, we write $a \prec b$ if either $w_a^+ < w_b^+$ or $w_a^+ = w_b^+$ and $-F^*(s_a) > -F^*(s_b)$. (That is, $a \prec b$ either if the angle of cone $\bar{Z}_b$ is sharper than the angle of cone $\bar{Z}_a$, or if both cones have equal angles but the vertex of cone $\bar{Z}_a$ is higher than the vertex of cone $\bar{Z}_b$.) Notice that there may be lexicographical ties, that is, customers $a, b \in C$ such that $w_a^+ = w_b^+$ and $-F^*(s_a) = -F^*(s_b)$. For any $a, b \in C$, we write $a \preceq b$ (a is lexicographically not greater than b) if it is not true that $b \prec a$. For any customer $c \in C$, let $C_c = \{a \in C \mid a \preceq c\}$, and let E_c be the upper envelope of functions $Z_a(x)$, $a \in C_c$. Suppose that p^* is a vertex of E defined by functions $Z_a(x), Z_b(x), Z_c(x)$ for some $a, b, c \in C$ (that is, $Z_a(x^*) = Z_b(x^*) = Z_c(x^*) = z^*$), see Fig. 4. Without loss of generality suppose that $a \preceq c$ and $b \preceq c$. Since $p^* \in E$, we also have $p^* \in E_c$; thus, p^* is a vertex of E_c. By Lemma 22 and taking into account the definition of C_c, all points of the ray

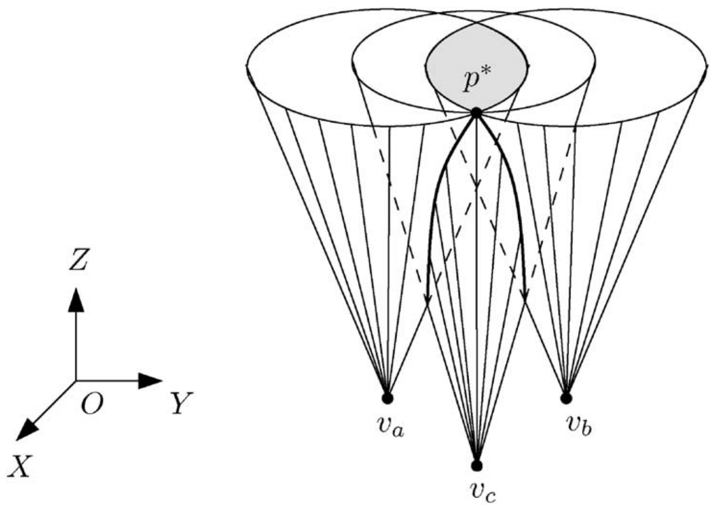

Fig. 4. The vertex p^* of the upper envelope of functions $Z_a(), Z_b(), Z_c()$. The boundary of the shaded region is the intersection of the upper envelope with a plane $z = const$.

$\{(x^*(t), Z_c(x^*(t))) \mid t \geqslant 1\}$, where $x^*(t) = v_c + t(x^* - v_c)$ belong to E_c. Therefore, the envelope E_c has an unbounded face defined by function $Z_c(x)$. We have the following

Lemma 23. *If p^* is a vertex of E, then for some $c \in C$, p^* is a vertex of an unbounded face of E_c that is defined by function $Z_c(x)$.*

Lemma 23 indicates that envelopes E_c are more convenient to work with than the envelope E, as for them it is easier to "catch" a face that may contain p^*. The lexicographical order implies also another important property of the envelopes E_c, $c \in C$.

Lemma 24. *If the numbers w_c^+, $c \in C$ are all distinct, then for any $c \in C$ the envelope E_c has a unique unbounded face, and this face is defined by function $Z_c(x)$. If the numbers w_c^+, $c \in C$ are not all distinct, then for any $c \in C$ the envelope E_c has at most one unbounded face defined by function $Z_c(x)$ (although in this case E_c may have more than one unbounded face, and may not have an unbounded face defined by $Z_c(x)$).*

Proof. Let us fix some $c \in C$. For any $z \in R$, let $\Phi_c(z) = \bigcap_{a \in C_c} \Psi_a(z)$ (where $\Psi_a(z)$ was defined in the proof of Lemma 21). If all w_a^+, $a \in C$ are distinct, then for any $a \in C_c \backslash \{c\}$, $w_a^+ < w_c^+$. Therefore, for all sufficiently large z, $\Phi_c(z) = \Psi_c(z)$, which implies the first statement of the lemma.

Suppose now that not all w_a^+, $a \in C$ are distinct. Let $C_c' = \{a \in C_c : w_a^+ = w_c^+\}$. Then for all sufficiently large z, $\Phi_c(z)$ is defined only by functions $Z_a(x)$, $a \in C_c'$, and the boundary of $\Phi_c(z)$ consists of arcs of some circles $\Psi_a(z)$, $a \in C_c'$. Notice that the lexicographical order implies that for any $a \in C_c'$, the radius of the circle $\Psi_a(z)$ is not larger than the radius of the circle $\Psi_c(z)$. Using this property and induction on the cardinality of C_c', it is straightforward to see that for all sufficiently large z the boundary of $\Phi_c(z)$ has no more than one arc of the circle $\Psi_c(z)$. This implies the second statement of the lemma. □

Lemma 25. *For any $c \in C$, it is possible to check in $O(n)$ time whether there exists an unbounded face f_c of the envelope E_c that is defined by function $Z_c(x)$. If such a face exists, it is unique (according to Lemma 24) and has $O(n2^{\alpha(n)})$ vertices, where $\alpha(n)$ is the inverse Ackermann function, and all these vertices can be computed in $O(n2^{\alpha(n)}\log^2 n)$ time.*

Proof. Let us fix $c \in C$. For any $a \in C_c\backslash\{c\}$, let $\mathscr{L}_a$ be the intersection curve of the boundaries of the cones $\bar{Z}_a$ and $\bar{Z}_c$. Observe that for any $a, b \in C_c\backslash\{c\}$, the curves $\mathscr{L}_a$ and $\mathscr{L}_b$ can intersect at most four times. Indeed, an intersection point $(x, z) \in R^3$ of the curves $\mathscr{L}_a$ and $\mathscr{L}_b$ must satisfy the equations

$$\begin{cases} (z + F^*(s_c))/w_c^+ + = d(v_c, x), \\ (z + F^*(s_a))/w_a^+ = d(v_a, x), \\ (z + F^*(s_b))/w_b^+ = d(v_b, x). \end{cases} \tag{17}$$

Let us square both sides of each equation in (17), and then let us subtract the first equation from the second and from the third. The left sides of all three new equations are quadratic functions of z. We obtain linear functions of the coordinates of point $x \in R^2$ in the right sides of the second and the third equations. Then, the second and the third equations represent a system of two linear equations with two unknowns with respect to the coordinates of x. Solving this system for the coordinates of x will give their expressions as (at most) quadratic functions of z. (It is not difficult to see that since we assumed that the points v_a, $a \in C$ are in general position, the system cannot be degenerate.) Substituting these expressions into the first equation $(z + F^*(s_c))^2 = (w_c^+ d(v_c, x))^2$ we obtain a quartic equation with variable z, which has at most four solutions.

Since every pair of curves $\mathscr{L}_a$, $\mathscr{L}_b$ can intersect at most four times, the unbounded face f_c of the envelope E_c that is defined by function $Z_c(x)$ (if it exists) has $O(\lambda_4(n))$ vertices and edges, where $\lambda_4(n) = O(n2^{\alpha(n)})$ is the maximum size of an $(n, 4)$-Davenport–Schinzel sequence [2]. (Davenport–Schinzel sequences are powerful combinatorial structures that play a central role in many geometric problems; see, e.g., [2] for a comprehensive introduction.) All vertices of the face f_c can be computed in $O(\lambda_4(n)\log^2 n) = O(n2^{\alpha(n)}\log^2 n)$ time [2] if a point inside the face is known, or if a direction $r \in R^2$ is known such that for all sufficiently large $t \in R$ the point $(v_c + tr, Z_c(v_c + tr))$ belongs to the face f_c. Here we assume a model of computation with infinite precision real arithmetic that allows us to compute the intersection points of any pair of curves $\mathscr{L}_a$, $\mathscr{L}_b$ in $O(1)$ time (this assumption is common in computational geometry [2]).

To complete the proof, it is sufficient to show that we can check in $O(n)$ time whether there exists an unbounded face f_c of the envelope E_c that is defined by function $Z_c(x)$, and that if such a face f_c exists, we can find in $O(n)$ time a direction $r \in R^2$ such that for all sufficiently large $t \in R$ the point $(v_c + tr, Z_c(v_c + tr))$ belongs to the face f_c (that is, $Z_c(v_c + tr) \geqslant Z_a(v_c + tr)$, $a \in C_c$).

Consider the set $C'_c = \{a \in C_c : w_a^+ = w_c^+\}$. If $C'_c = \{c\}$, then for any $a \in C_c\backslash\{c\}$, $w_a^+ < w_c^+$, and therefore the unbounded face f_c exists and any nonzero vector from R^2 can be taken as r. Suppose that $C'_c \neq \{c\}$. Consider any $a \in C'_c\backslash\{c\}$. Since $w_a^+ = w_c^+$ and $F^*(s_c) \geqslant F^*(s_a)$, it is not difficult to see that for any unit vector $r \in R^2$ such that $\langle v_a - v_c, r\rangle \leqslant \frac{F^*(s_c)-F^*(s_a)}{w_c^+}$ we have $Z_c(v_c + tr) < Z_a(v_c + tr)$ for all sufficiently large $t \in R$, and for any unit vector $r \in R^2$ such that $\langle v_a - v_c, r\rangle > \frac{F^*(s_c)-F^*(s_a)}{w_c^+}$ we have $Z_c(v_c + tr) > Z_a(v_c + tr)$ for all sufficiently large $t \in R$. (Here $\langle\cdot,\cdot\rangle$ is the regular dot product.) Hence, the unbounded face f_c exists if and only if there exists a unit vector $r \in R^2$ such that for all $a \in C'_c \setminus \{c\}$, $\langle v_a - v_c, r\rangle > \frac{F^*(s_c)-F^*(s_a)}{w_c^+}$. Clearly in $O(n)$ time we can either find such a vector r or to show that it does not exist. If we find such a vector r, then for any $a \in C_c$ and all sufficiently large $t \in R$, $Z_c(v_c + tr) \geqslant Z_a(v_c + tr)$. The lemma is proven. □

Lemmas 23–25 suggest the following approach. For each $c \in C$, check in $O(n)$ time whether there is an unbounded face f_c of the envelope E_c defined by function $Z_c(x)$. If such a face exists, obtain all its $O(n2^{\alpha(n)})$ vertices in $O(n2^{\alpha(n)} \log^2 n)$ time, and include their z-coordinates into set K. This makes sure that $z^* \in K$ if there is an optimal solution to Problem ROBCEN that corresponds to a vertex of the envelope E. Combining all the results, we have

Theorem 5. *The algorithm described above solves Problem* ROBCEN *in* $O(n^2 2^{\alpha(n)} \log^2 n)$ *time.*

7. Conclusion

In this paper, we studied interval data minmax regret single facility location problems on a plane. For the case of rectilinear distances, our models incorporated uncertainty in both customers' weights and location coordinates. For the case of Euclidean distances, we considered only uncertainty in weights. The problems are generalizations of the classical (without uncertainty) single facility location problems on a plane.

We presented an $O(n^2 \log^2 n)$ algorithm for the minmax regret rectilinear 1-median problem and an $O(n \log n)$ algorithm for the minmax regret rectilinear weighted 1-center problem. We have also discussed possibilities of solving approximately the minmax regret Euclidean 1-median problem, and presented an $O(n^2 2^{\alpha(n)} \log^2 (n))$ algorithm for solving the minmax regret Euclidean weighted 1-center problem.

The algorithms developed in the paper use special geometric properties of the considered problems and therefore seem unlikely to be easily extendable to other models. However, we believe that the underlying methodological ideas are sufficiently general to be useful for other geometric location problems. For example, the approach of Section 3 combined local optimality and convexity arguments, identification of grids that allowed to simplify the structure of the problem, and computational geometry techniques to speed up the solution procedure. It appears that such a combination of tools is effective for many minisum location problems with rectilinear distances; e.g., a similar combination of techniques was used in [18] to compute a 1-median for a continuum of customer points.

It appears that the ideas of Section 3 can be generalized to the case of block norms [35], although we doubt that this line of research would produce elegant results.

A possible direction for future research is to study minmax regret location problems with multiple facilities on a plane.

Appendix

Proof of Lemma 8. Let an $x \in R^2$ be fixed. Since Problem MED(s) can be solved in O(n) time (e.g., [26]) for any $s \in S$, it is straightforward to compute all values $F(s, x) - F^*(s)$, $s \in S^*(x)$ in O(n^3) time. This can be done by obtaining scenario $s_\alpha(x, y)$ according to (5), solving Problem MED($s_\alpha(x, y)$), and checking whether y is a 1-median for the scenario $s_\alpha(x, y)$, for all $y \in N(G_1)$ and $\alpha \in A$. We will show that the necessary computations can be performed in O(1) amortized time per scenario, that is, in O(n^2) total time.

A horizontal line of grid G_1 will be called a *row*. The set of nodes of G_1 that belong to a row L is denoted as $N(L)$. A point z of a row L is called a *change point* corresponding to $v \in V$ (or produced by $v \in V$), if $d(x, v) = d(z, v)$. Change points produced by different $v \in V$ are considered different even if they coincide. It follows from (5) that when y moves along a row L, scenario $s_\alpha(x, y)$ can change only at change points corresponding to v_c^α, $c \in C$. There are at most 8n change points on each row (at most 2 change points for each $v \in V$).

The structure of the algorithm is as follows. As a preprocessing, for each row L the algorithm computes all the change points on the row (recording the points $v \in V$ that produced them), and sorts them. We will show that the preprocessing can be implemented in O(n^2) total time. Then the algorithm processes the rows of grid G_1, spending O(n) time per row. For each row, the algorithm processes nodes y of the row from left to right, using the information obtained during the preprocessing to efficiently update scenarios $s_\alpha(x, y)$, $\alpha \in A$ and some auxiliary values as y changes and spending O(1) amortized time per update. The auxiliary values obtained for every node y and $\alpha \in A$ include values $F(s_\alpha(x, y), x)$, $F(s_\alpha(x, y), y)$, and some values that allow to check in O(1) time whether y is a 1-median for the scenario $s_\alpha(x, y)$. If y is a 1-median for the scenario $s_\alpha(x, y)$, then $F^*(s_\alpha(x, y)) = F(s_\alpha(x, y), y)$, and value $F(s_\alpha(x, y), x) - F^*(s_\alpha(x, y))$ is recorded. When all nodes of all rows have been processed, values of all functions $F(s, x) - F^*(s)$, $s \in S^*(x)$ have been obtained.

Let us discuss the preprocessing. For a $v \in V$, let $Q(v, x) = \{r \in R^2 \mid d(v, r) = d(v, x)\}$. Set $Q(v, x)$ is a "diagonal" square with the center at v and the point x on its boundary. Notice that a change point corresponding to a $v \in V$ must belong to $Q(v, x)$. Therefore, the change points lie on O(n) diagonal lines that contain sides of the squares $Q(v, x)$, $v \in V$. Lines that correspond to different squares are considered different even if they coincide. We break up these lines into two groups with slopes 1 and −1, respectively. The lines in each group are sorted (it takes O(n log n) time). This allows to find in O(n) time for any row L the order of intersection points of L with the lines in the group. The two sorted lists of intersection points for L (corresponding to the two groups of diagonal lines) can be merged in O(n) time, producing one sorted list of intersection points of L with the lines containing sides of squares $Q(v, x)$, $v \in V$. For each intersection point, the corresponding $v \in V$ is recorded. Then, the intersection points that are not change points are removed (it takes O(1) time to check whether an intersection point is a change point for the corresponding $v \in V$), which results in a sorted list of change points for the row L. Since there are O(n) rows, the preprocessing takes O(n^2) time.

Now we discuss how to process rows spending O(n) time per row. Suppose that a row L of grid G_1 is fixed. When y is equal to the leftmost node of L, scenarios $s_\alpha(x, y)$, $\alpha \in A$ are computed in $O(n)$ time using (5). When y moves through the nodes of L from left to right, scenario $s_\alpha(x, y)$ is updated using the scenario $s_\alpha(x, y')$ for the previous node y' and the change points between y' and y. Since there are $O(n)$ change points in any row, only $O(n)$ weights will be updated when y travels along the row L. Therefore, maintaining scenarios $s_\alpha(x, y)$, $\alpha \in A$ when y moves through a row takes $O(n)$ time.

For any $y \in R^2$ and a scenario s, let us define the following values. Let $W_1(y, s)$ ($W_2(y, s)$) be the sum of weights of all customers located strictly to the left (strictly to the right) of y under the scenario s, that is, whose first coordinate is strictly smaller (strictly greater) than $x_1(y)$. Let $W_3(y, s)$ ($W_4(y, s)$) be the sum of weights of all customers that are located higher (lower) than y under the scenario s, that is, whose second coordinate is strictly greater (strictly smaller) than $x_2(y)$. Let $W(s)$ be the sum of weights of all customers under scenario s. It is well known that y is a 1-median for a scenario s if and only if $W_i(y, s) \leqslant \frac{1}{2} W(s)$, $i = 1, 2, 3, 4$ [26]. Thus, given values $W(s)$, $W_i(y, s)$, $i = 1, 2, 3, 4$, it takes $O(1)$ time to check whether y is a 1-median for the scenario s.

For every node y of row L and every $\alpha \in A$, we will obtain values $W_i(y, s_\alpha(x, y))$, $(i = 1, 2, 3, 4$, $W(s_\alpha(x, y))$, $F(s_\alpha(x, y), x)$, $F(s_\alpha(x, y), y)$, and check whether y is a 1-median for the scenario $s_\alpha(x, y)$ using the test discussed above. For the leftmost node of row L, values $W_i(y, s_\alpha(x, y))$, $i = 1, 2, 3, 4$, $W(s_\alpha(x, y))$, $F(s_\alpha(x, y), x)$, $F(s_\alpha(x, y), y)$ are obtained directly (this takes $O(n)$ time). For any other node y of the row, these values are computed using the corresponding values for the previous node (nodes of a row are processed from left to right). Given the previous discussion, it is straightforward to see that these values for all nodes of a row can be computed in $O(n)$ total time.

If y is a 1-median for scenario $s_\alpha(x, y)$, then $F^*(s_\alpha(x, y)) = F(s_\alpha(x, y), y)$, and value $F(s_\alpha(x, y), x) - F^*(s_\alpha(x, y))$ is recorded. When all nodes of all rows are processed, values of all functions $F(s, x) - F^*(s)$, $s \in S^*(x)$ have been obtained and recorded. Since there are $O(n)$ rows and processing each row takes $O(n)$ time, the complexity of the algorithm is $O(n^2)$.

The directional derivatives of functions $F(s, x) - F^*(s)$, $s \in S^*(x)$ at x in a given direction can be computed in $O(n^2)$ total time in a similar way. □

References

[1] P. Agarwal, M. Sharir, Efficient algorithms for geometric optimization, Comput. Surveys 30 (4) (1998) 412–458.

[2] P. Agarwal, M. Sharir, Davenport–Schinzel sequences and their geometric applications, in: J.-R. Sack, J. Urrutia (Eds.), Handbook of Computational Geometry, Elsevier, Amsterdam, 1999, pp. 1–47, (Chapter 1).

[3] I. Averbakh, Minmax regret solutions for minimax optimization problems with uncertainty, Oper. Res. Lett. 27 (2) (2000) 57–65.

[4] I. Averbakh, On the complexity of a class of combinatorial optimization problems with uncertainty, Math. Programming 90 (2001) 263–272.

[5] I. Averbakh, Complexity of robust single facility location problems on networks with uncertain edge lengths, Discrete Appl. Math. 127 (2003) 505–522.

[6] I. Averbakh, S. Bereg, Facility location problems with uncertainty on the plane, Working paper, University of Toronto, 2003.

[7] I. Averbakh, O. Berman, Minmax regret median location on a network under uncertainty, INFORMS J. Comput. 12 (2) (2000) 104–110.

[8] I. Averbakh, O. Berman, Algorithms for the robust 1-center problem on a tree, European J. Operat. Res. 123 (2000) 292–302.
[9] C. Bajaj, The algebraic degree of geometric optimization problems, Discrete Comput. Geom. 3 (1988) 177–191.
[10] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Math. Oper. Res. 23 (1998) 769–805.
[11] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Math. Programming B 98 (2003) 43–71.
[12] J.R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer, Berlin, 1997.
[13] R. Chandrasekaran, A. Tamir, Algebraic optimization: the Fermat–Weber location problem, Math. Programming 46 (1990) 219–224.
[14] B. Chen, C. Lin, Robust one-median location problem, Networks 31 (1998) 93–103.
[15] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, E. Welzl, Combinatorial complexity bounds for arrangements of curves and spheres, Discrete Comput. Geom. 5 (1990) 99–160.
[16] M.E. Dyer, On a multidimensional search technique and its applications to the Euclidean one-center problem, SIAM J. Comput. 15 (3) (1986) 725–738.
[17] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer, Berlin, 1987.
[18] S.P. Fekete, J.S.B. Mitchell, K. Weinbrecht, On the continuous Weber and k-median problems, Proceedings of the 16th annual ACM symposium on Computational Geometry, 2000, pp. 70–79.
[19] F.R. Fernandez, S. Nickel, J. Puerto, A.M. Rodriguez-Chia, Robustness in the Pareto-solutions for the multicriteria minisum location problem, J. Multicriteria Decision Anal. 10 (2001) 191–203.
[20] G. Frederickson, D. Johnson, Generalized selection and ranking: sorted matrices, SIAM J. Comput. 13 (1) (1984) 14–30.
[21] P. Kall, S. Wallace, Stochastic Programming, Wiley, New York, 1994.
[22] I. Katz, Local convergence in Fermat's problem, Math. Programming 6 (1974) 89–104.
[23] P. Kouvelis, G. Vairaktarakis, G. Yu, Robust 1-median location on a tree in the presence of demand and transportation cost uncertainty, Working paper 93/94-3-4, Department of Management Science and Information Systems, Graduate School of Business, The University of Texas at Austin, 1994.
[24] P. Kouvelis, G. Yu, Robust Discrete Optimization and Its Applications, Kluwer Academic Publishers, Boston, 1997.
[25] H. Kuhn, A note on Fermat's problem, Math. Programming 4 (1973) 98–107.
[26] R. Love, J. Morris, G. Wesolowsky, Facilities Location: Models and Methods, North-Holland, New York, 1988.
[27] N. Megiddo, The weighted Euclidean 1-center problem, Math. Oper. Res. 8 (1983) 498–504.
[28] N. Megiddo, Linear programming in linear time when the dimension is fixed, J. ACM 31 (1984) 114–127.
[29] N. Megiddo, On the ball spanned by balls, Discrete Comput. Geom. 4 (1989) 605–610.
[30] J.M. Mulvey, R.J. Vanderbei, S.A. Zenious, Robust optimization of large-scale systems, Oper. Res. 43 (2) (1995) 264–281.
[31] F. Preparata, M. Shamos, Computational Geometry: An Introduction, Springer, New York, 1985.
[32] R. Seidel, Convex hull computations, in: J. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press, Boca Raton, FL, 1997, pp. 361–376, (Chapter 19).
[33] B. Tansel, G. Scheuenstuhl, Facility location on tree networks with imprecise data, Research Report IEOR-8819, Bilkent University, Turkey, 1988.
[34] G. Vairaktarakis, P. Kouvelis, Incorporating dynamic aspects and uncertainty in 1-median location problem, Naval Res. Logist. 46 (1999) 147–168.
[35] J.E. Ward, R.E. Wendell, Using block norms for location modelling, Oper. Res. 33 (1985) 1074–1090.
[36] G. Wesolowsky, The Weber problem: history and perspective, Location Sci. 1 (1993) 5–23.