

Computing the Rectilinear Center of Uncertain Points in the Plane^{*}

Haitao Wang and Jingru Zhang

Department of Computer Science, Utah State University, Logan, UT 84322, USA
haitao.wang@usu.edu, jingruzhang@aggiemail.usu.edu

1 Introduction

In the real world, data is inherently uncertain due to many facts, such as the measurement inaccuracy, sampling discrepancy, resource limitation, and so on. Problems on uncertain data have been studied extensively, e.g., [1–3, 5]. In this abstract, we consider the one-center problem on uncertain points in the plane with respect to the rectilinear distance.

Let $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ be a set of n uncertain points in the plane, where each uncertain point $P_i \in \mathcal{P}$ has m possible locations $p_{i1}, p_{i2}, \dots, p_{im}$ and for each $1 \leq j \leq m$, p_{ij} is associated with a probability $f_{ij} \geq 0$ for P_i being at p_{ij} (which is independent of other locations).

For any (deterministic) point p in the plane, we use x_p and y_p to denote the x - and y -coordinates of p , respectively. For any two points p and q , we use $d(p, q)$ to denote the *rectilinear distance* between p and q , i.e., $d(p, q) = |x_p - x_q| + |y_p - y_q|$.

Consider a point q in the plane. For any uncertain point $P_i \in \mathcal{P}$, the *expected rectilinear distance* between q and P_i is defined as

$$\text{Ed}(P_i, q) = \sum_{j=1}^m f_{ij} \cdot d(p_{ij}, q).$$

Let $\text{Ed}_{\max}(q) = \max_{P_i \in \mathcal{P}} \text{Ed}(P_i, q)$. A point q^* is called a *rectilinear center* of \mathcal{P} if it minimizes the value $\text{Ed}_{\max}(q^*)$ among all points in the plane. Our goal is to compute q^* . Note that such a point q^* may not be unique, in which case we let q^* denote an arbitrary such point.

2 Our Results

We assume that for each uncertain point P_i of \mathcal{P} , its m locations are given in two sorted lists, one by x -coordinates and the other by y -coordinates. To the

best of our knowledge, this problem has not been studied before. We present an $O(mn)$ time algorithm. Since the input size of the problem is $\Theta(nm)$, our algorithm essentially runs in linear time, which is optimal.

Further, our algorithm is applicable to the weighted version of this problem in which each $P_i \in \mathcal{P}$ has a weight $w_i \geq 0$ and the *weighted expected distance*, i.e., $w_i \cdot \text{Ed}(P_i, q)$, is considered. To solve the weighted version, we can first reduce it to the unweighted version by changing each f_{ij} to $w_i \cdot f_{ij}$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$, and then apply our algorithm for the unweighted version. The running time is still $O(mn)$.

3 Related Work

The problem of finding one-center among uncertain points on a line has been considered in our previous work [7], where an $O(nm)$ time algorithm was given. An algorithm for computing k centers for general k was also given in [7] with the running time $O(mn \log mn + n \log n \log k)$. In fact, in [7] we considered the k -center problem under a more general uncertain model where each uncertain point can appear in m intervals. We also studied the one-center problem for uncertain points on tree networks in [6], where a linear-time algorithm was proposed.

4 The Main Techniques

Consider any uncertain point $P_i \in \mathcal{P}$ and any (deterministic) point q in the plane \mathbb{R}^2 . We first show that $\text{Ed}(P_i, q)$ is a convex piecewise linear function with respect to $q \in \mathbb{R}^2$. More specifically, if we extend a horizontal line and a vertical line from each location of P_i , these lines partition the plane into a grid G_i of $(m+1) \times (m+1)$ cells. Then, $\text{Ed}(P_i, q)$ is a linear function (in both the x - and y -coordinates of q) in each cell of G_i . In other words, $\text{Ed}(P_i, q)$ defines a plane surface patch in 3D on each cell of

^{*} The full paper of this abstract can be found at arXiv:1509.05377.

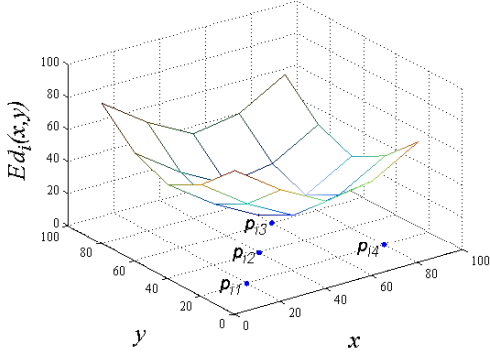


Figure 1. Illustrating the function $\text{Ed}_i(x, y)$ of an uncertain point P_i with $m = 4$.

G_i (see Fig. 1). Then, finding $q^* \in \mathbb{R}^2$ is equivalent to finding a lowest point p^* in the upper envelope of the n graphs in 3D defined by $\text{Ed}(P_i, q)$ for all $P_i \in \mathcal{P}$ (specifically, q^* is the projection of p^* onto the xy -plane). As a basic computational geometry problem, it may be of interest in its own right.

The problem of finding p^* can be solved in $O(nm^2)$ time by the linear-time algorithm for the 3D linear programming (LP) problem [4]. Indeed, for a plane surface patch, we call the plane containing it the *supporting plane*. Let \mathcal{H} be the set of the supporting planes of the surface patches of the functions $\text{Ed}(P_i, q)$ for all $P_i \in \mathcal{P}$. Since each function $\text{Ed}(P_i, q)$ is convex, p^* is also a lowest point in the upper envelope of the planes of \mathcal{H} . Thus, finding p^* is an LP problem in \mathbb{R}^3 and can be solved in $O(|\mathcal{H}|)$ time [4]. Note that $|\mathcal{H}| = \Theta(nm^2)$ since each grid G_i has $(m + 1)^2$ cells.

We give an $O(mn)$ time algorithm without computing the functions $\text{Ed}(P_i, q)$ explicitly. We use a prune-and-search technique that extends Megiddo’s technique for the 3D LP problem [4]. In each recursive step, we prune at least $n/32$ uncertain points from \mathcal{P} in linear time. In this way, q^* can be found after $O(\log n)$ recursive steps.

Unlike Megiddo’s algorithm [4], each recursive step of our algorithm itself is a recursive algorithm of $O(\log m)$ recursive steps. Therefore, our algorithm has $O(\log n)$ “outer” recursive steps and each outer recursive step has $O(\log m)$ “inner” recursive steps. In each outer recursive step, we maintain a rectangle R that always contains q^* in the xy -plane. Initially, R is the entire plane. Each inner recursive step shrinks R with the help of a *decision algorithm*. The key idea is that after $O(\log m)$ steps, R is so

small that there is a set \mathcal{P}^* of at least $n/2$ uncertain points such that R is contained inside a single cell of the grid G_i of each uncertain point P_i of \mathcal{P}^* (i.e., R does not intersect the extension lines from the locations of P_i). At this point, with the help of our decision algorithm, we can use a pruning procedure similar to Megiddo’s algorithm [4] to prune at least $|\mathcal{P}^*|/16 \geq n/32$ uncertain points of \mathcal{P}^* . Each outer recursive step is carefully implemented so that it takes only linear time.

In particular, our decision algorithm is for the following *decision problem*. Let R be a rectangle in the plane and R contains q^* (but the exact location of q^* is unknown). Given an arbitrary line l that intersects R , the decision problem is to determine which side of l contains q^* . Megiddo’s technique [4] gave an algorithm that can solve our decision problem in $O(m^2n)$ time. We give a decision algorithm of $O(mn)$ time. In fact, in order to achieve the overall $O(mn)$ time for computing q^* , our decision algorithm has the following performance. For each $1 \leq i \leq n$, let a_i and b_i be the number of columns and rows of the grid G_i intersecting R , respectively. The running time of our decision algorithm is bounded by $O(\sum_{i=1}^n (a_i + b_i))$.

References

1. P.K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *Proc. of the 28th Symposium on Principles of Database Systems (PODS)*, pages 137–146, 2009.
2. P.K. Agarwal, S. Har-Peled, S. Suri, H. Yıldız, and W. Zhang. Convex hulls under uncertainty. In *Proc. of the 22nd Annual European Symposium on Algorithms (ESA)*, pages 37–48, 2014.
3. P. Kamousi, T.M. Chan, and S. Suri. Stochastic minimum spanning trees in Euclidean spaces. In *Proc. of the 27th Annual Symposium on Computational Geometry (SoCG)*, pages 65–74, 2011.
4. N. Megiddo. Linear-time algorithms for linear programming in \mathbb{R}^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.
5. S. Suri and K. Verbeek. On the most likely voronoi diagram and nearest neighbor searching. In *Proc. of the 25th International Symposium on Algorithms and Computation (ISAAC)*, pages 338–350, 2014.
6. H. Wang and J. Zhang. Computing the center of uncertain points on tree networks. In *Proc. of the 14th Algorithms and Data Structures Symposium (WADS)*, pages 606–618, 2015.
7. H. Wang and J. Zhang. One-dimensional k -center on uncertain data. *Theoretical Computer Science*, online first, 2015.