Mobile Facility Location*

Sergei Bespamyatnikh† Binay Bhattacharya‡ David Kirkpatrick† Michael Segal†

Abstract

In this paper we investigate the location of mobile facilities (in L_∞ and L_2 metric) under the motion of clients. In particular, we present lower bounds and efficient algorithms for exact and approximate maintenance of 1-center and 1-median for a set of moving points in the plane. Our algorithms are based on the *kinetic* framework introduced by Basch et. al [4].

1 Introduction

The goal of this paper is to introduce and study problems related to the location of mobile facilities serving a set of points moving continuously in d-dimensional space, $d \geq 1$. Examples of such problems are the maintenance of the k-center and k-median for moving points under the L_p metric. This has applications, for example in mobile wireless communication networks when the broadcast range should contain all the customers so as to providing service to the cellular phones. This is only one example of the many proximity problems that can be posted for a system of moving points. Another application of these problems is locating a welding robot in an automobile manufacturing plant. Both problems are interesting from both theoretical and practical points of view.

Facility location is a classical problem of operations research that has also been examined in the computational geometry community. Most of the problems described in the facility location literature are concerned with finding a "desirable" facility location: the goal is to *minimize* a distance function between the facility (e.g. a service) and the sites (e.g. the customers). To our best knowledge nothing has been done in the area of facility location for continuously moving points. Surprisingly, the data structures and algorithms that have been developed for the *static* problems (i.e. customers are not moving) are not directly applicable to the setting of moving points when the motion of the facilities must satisfy natural constraints. We consider mobile versions of two classical facility location problems:

1. **k-center:** Given a set S of n demand points in d-dimensional space ($d \geq 1$), find a set P of k supply points so that the *radius* defined as the maximum L_p distance between a demand point and its nearest supply point in P is minimized. Note that for some metrics such a set P is not necessarily unique even for $k = 1$ and $d = 2$.

2. **k-median:** Given a set S of n demand points in d-dimensional space ($d \geq 1$), find a set P of k supply points so that the *dispersion* defined as the sum of the L_p distances between demand points and their nearest supply points in P is minimized. Again, it is possible that for some metrics the solution is not unique.

These problems have been well studied in both the exact [1, 5, 6, 7, 8, 9, 11, 12, 15, 19] and approximate [8, 10, 14, 17] versions. In approximate versions a set P provides $(1+\varepsilon)$-approximate k-center (k-median) if the associated radius (dispersion) is at most $(1+\varepsilon)$ times of the optimal radius (dispersion), for any $\varepsilon > 0$.

*This work has been supported by a NSERC grant. Work by Michael Segal has been also supported by the Pacific Institute for Mathematical Studies. Emails: {besp, kirk, msegal}@cs.ubc.ca, binay@cs.sfu.ca.

†Department of Computer Science, University of British Columbia Vancouver, B.C. Canada V6T 1Z4.

‡School of Computing Science, Simon Fraser University, Burnaby, B.C. Canada, V5A 1S6.

We define the *mobile k-center* (*mobile k-median*) problem as follows. Given a set $S = \{p_1, p_2, \ldots, p_n\}$ of n continuously moving points specified by a piecewise differentiable functions $\{g_1, g_2, \ldots, g_n\}$, where $g_i, 1 \leq i \leq n$ maps time interval $[0, T]$ to $\mathbb{R}^d$, we want to determine k continuous functions $f_1, \ldots f_k$; $f_i : [0, T] \rightarrow \mathbb{R}^d$ such that at any given moment $t \in [0, T]$, the points $f_1(t), \ldots, f_k(t)$ form a k-center (k-median) for the points at locations $g_1(t), \ldots, g_n(t)$. The *mobile approximate k-center (k-median)* problem is defined similarly. In this paper we focus on the simplest instances of these mobile facility location problems, when $k = 1, d = 2, p = \{1, 2, \infty\}$. Even these simplest instances pose several challenging algorithmic and geometric questions. We also consider *velocity restricted* versions of these instances, i.e. when the velocity of the facility may vary in some range.

Our algorithms for these problems are based on a new class of data structures – *kinetic data structures (KDS)* – aimed at keeping track of attributes of interest in systems of moving objects [4, 13]. In the kinetic setting, a set of points is assumed to be continuously changing, or moving. Each point follows a posted *flight plan*, but a plan which can change at any moment through a *flight plan update.* A KDS maintains a *configuration function* of interest (in our case will be the set P) by watching for critical events as the objects move. A KDS for computing a particular attribute for a set of points in motion maintains a set of *certificates.* A certificate based on a tuple of points is a continuous function that associates a real number with each configuration of these points. For example, the certificates for a convex hull KDS are the collection of the triple of points, each with a particular orientation. At any one time, the conjunction of all the certificates being maintained over by the kinetic data structure proves the combinatorial correctness of its output. As the points move, some certificates may become invalid, e.g. a triple of points changes the orientation. When a certificate fails, the proof structure needs to be modified and the combinatorial description of the configuration function may need to be updated. Certificates are stored in a priority queue, ordered by failure time, and processed in order as they fall. A good kinetic data structure will take advantage of the continuity of the point motion to select certificate structures which are easy to update at these critical events; a structure satisfying this condition is called *responsive.* Other criteria for a KDS are *efficiency,* i.e. the number of events processed by KDS is not much greater than the number of combinatorial changes in the configuration function itself, *compactness*, i.e. the number of active certificates at any one time is roughly linear in the complexity of the moving points, and *locality,* i.e. a flight plan update for any one points affects only a small number of certificate.

This paper is organized as follows. In next Section we show several results regarding the mobile rectilinear and Euclidean 1-center problems. Section 3 deals with the mobile 1-median problem. Section 4 provides some concluding remarks and questions for future investigation.

2 Mobile 1-center problems

In this section we investigate the complexity of mobile 1-center problems under the L_∞ and L_2 metrics. We first show how to deal with the case of the L_∞ metric and then extend our result for Euclidean case.

2.1 Rectilinear 1-center

Exact algorithm. We are given a set S of n moving points (customers) in the plane and we want to maintain a 1-center of these points (under the L_∞ metric), i.e. a point c with the property that the maximum distance between a point of S and c is minimized. We assume, for simplicity, that the motion of each point of S (i.e. its flight plan) is determined by a linear function. We assume that the velocities of points are bounded by 1. During a flight plan update the velocity of the point can change but still not exceed 1.

Observation 1 *Any point described by a fixed linear combination of the customer points moves with velocity at most 1.*

Next, we observe that the required center c can move faster than all the points. See Figure 1.

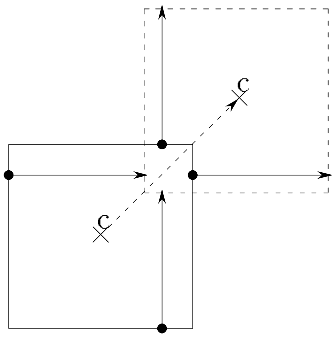

Figure 1: Rectilinear one center moves with velocity $\sqrt{2}$.

Observation 2 *For any instance of mobile* 1*-center problem there is a rectilinear* 1*-center whose velocity is bounded by* $\sqrt{2}$*. Furthermore, there is an instance of the problem with unique solution moving with velocity* $\sqrt{2}$*.*

Proof. The bounding box of point set S is defined by four points: leftmost p_l, rightmost p_r, topmost p_t and bottommost p_b (some of them may coincide). The center of bounding box has coordinates $((x(p_l) + x(p_r))/2, (y(p_b) + y(p_t))/2)$. The center of the bounding box can serve as a rectilinear 1-center. The x-coordinate of the center of bounding box cannot move faster than velocity 1; the same holds for the y-coordinate of the center of bounding box. Hence, the center of bounding box can move with velocity at most $\sqrt{2}$. On the other hand, Figure 1 shows an example when the rectilinear 1-center moves with velocity $\sqrt{2}$. ∎

The center of the bounding box and , thus, the rectilinear 1-center can be computed using elementary data structures. The idea is to maintain points p_l,p_r, p_t, p_b under motion of points. The problem is equivalent to maximum maintenance problem [4]. Basch et al. [4] present several data structures in order to solve different variants of the maximum maintenance problem. For our purposes it suffices to use the kinetic swapping heap with $O(\log n)$ time responsiveness, $O(1)$ locality and $O(n \log^3 n)$ efficiency.

Lemma 3 *A rectilinear* 1*-center can be maintained under the motion of points in an efficient KDS.*

A natural question now is: What approximation factor can be achieved once we restrict the velocity of facility f to some constant between 1 and $\sqrt{2}$? Recall that the approximation factor at moment t is defined as a relation between the radius determined by the facility f and the radius determined by the optimal solution at this moment. Our goal is to minimize the approximation factor over all the time. First, we consider the case of unit velocity. Notice that with restricted velocity we are not able to track the exact location of the rectilinear 1-center and, thus, kinetic versions of static 1-center algorithms do not suffice. Moreover, the approximation factor now depends on the initial position of facility. In what follows we provide simple algorithms with guaranteed approximation factor 2 and then present a lower bound for the approximation factor when facility moves with velocity 1.

2-approximation factor. If at the beginning of the motion of points we are allowed to put our facility f on any point in the plane, the simple way to achieve 2-approximation factor is to put our facility on any demand point and follow its flight plan. Obviously, our facility will always be inside of the bounding box of S, providing 2-approximation factor. As a matter of fact the same approximation factor can be achieved even in the case when the starting location of facility is restricted to be any point inside of the convex hull of S.

Lemma 4 *Starting with the arbitrary location of facility inside of the convex hull of* S *there is a flight plan for facility that guarantees a* 2*-approximation factor and can be maintained efficiently.*

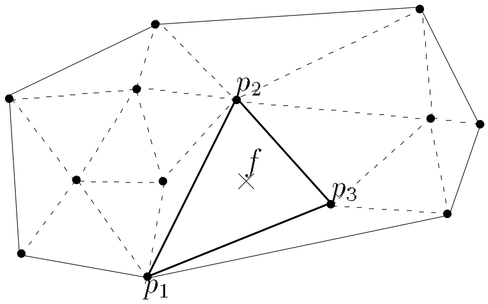

Figure 2: Facility lies inside of triangle Δ with vertices p_1, p_2, p_3.

Proof. At the preprocessing step we compute the convex hull and triangulation T of S. We find a triangle Δ in T that contain our facility f. Let p_1, p_2, p_3 be the vertices of this triangle. See Figure 2. Our goal is to stay inside of Δ. Initial velocity of f is 0. We maintain three certificates; each one proves the fact that f is to the left of an edge of Δ. When a certificate fails, the facility f becomes collinear with the other two points, e.g. p_1 and p_2. If f coincides with either p_1 or p_2 we are done since f can follow this point as described above. Otherwise we change our strategy and certificates – since the endpoints p_1 and p_2 of the edge p_1p_2 move with velocity at most 1 our facility f can always follow the motion and stay on this edge (the certificate proves the collinearity of p_1, p_2, f). If during this motion f will coincide with either p_1 or p_2 it can simply follow this point providing 2-approximation factor. ∎

Remark. Being inside of convex hull (initially) is crucial for achieving a guaranteed approximation factor. Otherwise, points of S can move according to the following strategy. They will move towards some point r in the plane such that the distance from f to r is larger than any distance from r to points in S. This point always exists since f is outside of convex hull. Thus, at point r the radius determined by the optimal 1-center is equal to 0 and the radius is determined by f is greater than 0 which violates any approximation factor.

Improvement of approximation factor. Now we present a new algorithm with a slightly improved approximation factor for a facility moving with unit velocity. The key idea is to use the notion of the center of mass. Given a set S of n points we can define mass in various ways:

- **Point mass.** The weight of $\frac{1}{n}$ is assigned to each point.
- **Region mass.** The mass is uniformly distributed in convex hull.
- **Boundary mass.** The mass is uniformly distributed on the boundary of convex hull.

By Observation 1, the center of mass of the points moves with velocity at most 1. In contrast, the center mass of the convex hull and the center mass of the boundary of convex hull can move with velocity exceeding the maximum velocity of customers (See Figure 3(a)).

Figure 3(a) illustrates an example of four points p_1, p_2, p_3, p_4 moving with unit velocity over one unit of time. Initially, the center of mass of the convex hull (or its boundary) is on the segment (p_2, l) and its final location is on the segment (p_3, m). Any segment that connects between (p_2, l) and (p_3, m) should have a length larger than 1. Hence, the center of mass of the convex hull (or its boundary) moves with the velocity exceeding 1.

We have shown already an example (Figure 1) where the rectilinear 1-center moves with velocity $\sqrt{2}$ while the points move with unit velocity. When we allow the facility to move with unit velocity the best approximation factor that can be achieved in this example is $2 - \frac{1}{\sqrt{2}}$. This approximation factor is defined by the facilities f and f' on the segment cc' in Figure 3(b) such that the length of the segment connecting f and f' is 1. It is interesting to note that the center of mass the boundary of the convex hull achieves this approximation factor. Clearly, that the center of mass the boundary of the convex hull is always on the

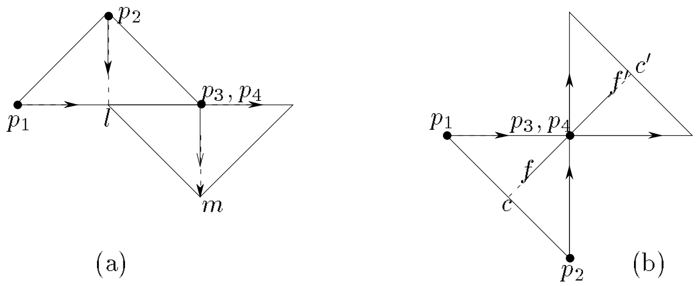

Figure 3: (a) Center of mass of the convex hull (or the boundary of convex hull) moves with velocity exceeding 1. (b) Center of mass of the boundary of convex hull gives best strategy for a given example.

segment cc'. Using the definition of the center of mass one can show that the x-coordinate of the center of mass the curve $f(x)$ defined on interval $[a, b]$ is $\frac{\int_a^b x\sqrt{1+(f'(x))^2}dx}{l}$, where l is the length of the curve. It can be verified that the initial position of the center of mass of the triangle $\Delta p_1p_2p_3$ coincides with the position of facility f and finally with the position of f'.

Lemma 5 *The center of mass of points guarantees the approximation factor of* $2 - \frac{2}{n}$ *and can be maintained efficiently under the motion of points.*

Proof (sketch). By Observation 1 the velocity of the center of mass of the points does not exceed the maximal velocity of the customers. The worst possible case is when $n-1$ points coincide with the origin and the other point has coordinates $(1, 0)$. The center of mass of the points has coordinates $(\frac{1}{n}, 0)$. The relation between the radius determined by the center of mass and the radius determined by the optimal solution is $(1 - \frac{1}{n})/\frac{1}{2} = 2 - \frac{2}{n}$. In our kinetic model we maintain the queue of events: changing velocity of the points and changing flight plan of the points. We also maintain the location and velocity of the center of mass. Given a new event we can calculate in constant time the current location of the center of mass and its new velocity. ∎

Surprisingly, the approximation factor stated in the previous lemma is, in fact, asymptotically optimal.

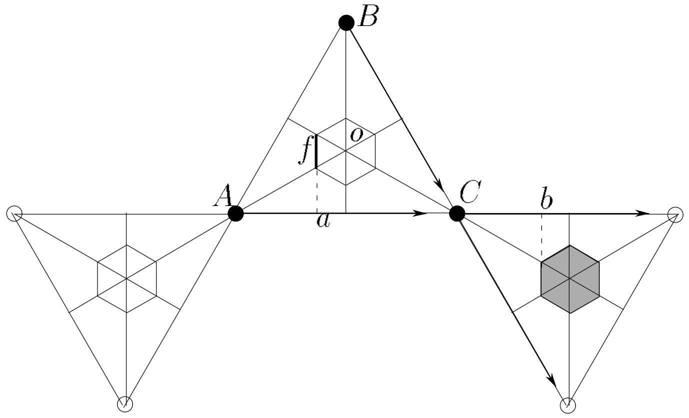

Figure 4: Adversary strategy for achieving lower bound for approximation factor.

Theorem 6 *Any algorithm that starts with the facility inside the convex hull and moves the facility with at most unit velocity achieves an approximation factor of at most 2 (asymptotically) in the worst case.*

Proof. We prove this theorem by an adversary argument. Adversary picks the set S that contains points at vertices A, B, C of equilateral triangle in Figure 4. Without loss of generality we assume that there are arbitrary many points at each vertex of triangle. The edge size of the triangle is 2. An algorithm starts with some initial location of facility f inside of triangle ΔABC. In order to prove a lower bound, the adversary will force an algorithm to put a facility f to the boundary of the bounding box of S. This goal can be achieved doing the following strategy. The adversary checks which one of the triangles AoB, BoC, CoA contains facility, where o is the center of ΔABC. Depending on location of the facility adversary moves points into the next equilateral triangle. For example, if facility is located in triangle AoB adversary moves all the points to the right triangle in Figure 4. All points at vertices A and B move towards vertex C and the points at vertex C are split into two halves that move to the two remaining vertices of right triangle in Figure 4. Similarly, if facility is located in triangle BoC adversary moves the points to the left triangle (the symmetric case of triangle CoA is not shown on the Figure 4). Notice that all points of S move distance 2 from one triangle to another. The adversary continues to play this game until it forces the facility to coincide with the vertex of the current triangle (possible infinite number of steps, each step getting closer to the vertex). To show this we define the distance of facility to the center of current triangle using a hexagon centered at the center of triangle. Assume that facility is located in ΔABC on the bold side of hexagon centered at o. We show that the next location of facility will be outside of the hexagon of the same size, shaded in Figure 4. This follows easily from the fact that the distance between point a and b is 2. ∎

Motion with a bounded velocity of facility. Suppose, now, that the velocity of the facility is bounded by $v_{max} \in [1, \sqrt{2}]$. Recall that for values of $v_{max} = 1, \sqrt{2}$ the best approximation factor are 2 and 1, respectively. We *mix* our two strategies: the center of mass of the points and the center of the bounding box in the following way. Let (f_1, v_1) be the location (vector in the plane) and velocity of the maintained center of mass of the points. Similarly, (f_2, v_2) are the location and velocity of the maintained center of the bounding box. The algorithm maintains the *mixing center* (f, v) defined as $(\alpha f_1 + (1-\alpha)f_2, \alpha v_1 + (1-\alpha)v_2)$, where $\alpha = \frac{\sqrt{2}-v_{max}}{\sqrt{2}-1}$.

Lemma 7 *The velocity of the mixing center is bounded by v_{max} and the approximation factor is*

$$1 + \frac{\sqrt{2} - v_{max}}{\sqrt{2} - 1}(1 - \frac{2}{n}).$$

Proof. First we show that velocity of the mixing center is bounded by v_{max}. We want to prove the inequality $\alpha v_1 + (1-\alpha)v_2 \le v_{max}$. In order to do this, we replace v_1 by 1 and v_2 by $\sqrt{2}$. It can be easily seen that the value of the expression $\alpha + (1-\alpha)\sqrt{2}$ when $\alpha = \frac{\sqrt{2}-v_{max}}{\sqrt{2}-1}$ is equal to v_{max}.

Regarding the approximation factor we consider the case where the center of mass is at largest L_∞ distance from the center of the bounding box. This distance is $1 - \frac{2}{n}$ if the radius that is determined by the center of the bounding box is 1. Then the L_∞ distance between the mixing center and the center of the bounding box is $\frac{\sqrt{2}-v_{max}}{\sqrt{2}-1}(1 - \frac{2}{n})$. The approximation factor bound follows. ∎

Variants of mixing strategy. We consider two different ways to improve the mixing strategy from a practical point of view. First we notice that the mixing strategy described above constraints the initial location of facility to some particular point. We avoid this by replacing the center of mass of the points in the mixing strategy by the algorithm described in Lemma 4. In other words, the initial location of facility is the linear combination of any point p inside of the convex hull and the center c of the bounding box . The approximation factor, however, grows slightly to $1 + \frac{\sqrt{2}-v_{max}}{\sqrt{2}-1}$.

The second improvement concerns the approximation factor. The mixing center in the previous strategy sometimes may move with velocity less than v_{max}. In this case, we may use the excess of the velocity to improve the approximation factor by moving mixing center f towards the center of the bounding box. When the mixing center f moves it makes coefficient α non-valid in the linear combination of p and c. We would like to preserve the initial value of α. It can be viewed as a new motion of point p. However, the point p, that initially lies in some triangle $\Delta p_i p_j p_k$ inside of the convex hull of S, may move outside of this triangle.

We can calculate this event a put it into the queue of events. When this event happens point p is still inside of the convex hull and we need to compute a new triangle that contains p. It can be done as following. We may assume that only points on convex hull triangulation T. The mobile convex hull can be maintained kinetically [3] with $O(\log n)$ locality and $O(\log^2 n)$ time responsiveness. We introduce new type of events: when one of the vertices of $\Delta p_i p_j p_k$ no longer serves as a vertex of the convex hull. In this case we can update T in $O(1)$ time and find a new triangle that contains p and calculate the new three certificates providing the proof of p inside this triangle. If p moves from one triangle Δ in T to the adjacent triangle Δ_1, the vertices of Δ_1 can be found in constant time using the maintained convex hull. From a practical point of view this strategy is more useful than the strategies described above.
Remark. It is well known that the metrics L_1 and L_∞ are dual in the plane, in the sense that nearest neighbors under L_1 in a given coordinate system are also nearest neighbors under L_∞ in a 45 degrees rotated coordinate system (and vice versa). The distances, however, are different by a multiplicative factor of $\sqrt{2}$. Thus, the results for L_∞ carry over to the L_1 metric and vise versa.

2.2 Euclidean 1-center

Although the static Euclidean 1-center problem like the static rectilinear 1-center problem can be solved in linear time [18], the mobile versions of these problems are quite different. In contrast to the rectilinear 1-center problem we show an example providing that Euclidean 1-center may move with unbounded velocity. See Figure 5.

Lemma 8 *For any velocity $V \geq 0$ there is a motion of three points p_1, p_2, p_3 whose Euclidean 1-center moves with velocity faster than V.*

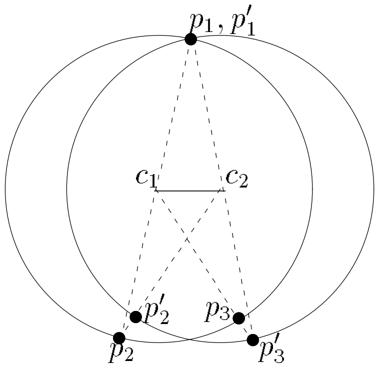

Figure 5: Euclidean 1-center may move with unbounded velocity.

Proof. We show an example (see Figure 5) with three points p_1, p_2, p_3 that initially define a left circle of radius 1 and move to points p_1', p_2', p_3', respectively. The points p_1', p_2', p_3' define the right circle in Figure 5 also of radius 1. Assume that the distances between centers c_1 and c_2 of these two circles is $x > 0$, i.e. the exact mobile 1-center moves distance x. We want to determine the distance d between p_2 and p_2'. Lemma follows if $Vd < x$ (*). Clearly $d = |p_2c_2| - 1$. Applying the cosines theorem to triangles $p_2c_1c_2$ and $p_1c_1c_2$ we obtain:

$$|p_2c_2|^2 = x^2 + 1 - 2x\cos\angle p_2c_1c_2 \text{ and } x^2 = 2 - 2\cos\angle c_1p_1c_2.$$

Notice that $\angle p_2c_1c_2 = \pi - \frac{\pi - \angle c_1p_1c_2}{2} = \frac{\pi + \angle c_1p_1c_2}{2}$. Thus, $\cos\angle p_2c_1c_2 = -\sin\frac{\angle c_1p_1c_2}{2}$. On the other hand, $\sin\frac{\angle c_1p_1c_2}{2} = \sqrt{\frac{1-\cos\angle c_1p_1c_2}{2}}$. Therefore, after algebraic manipulations we have $d = \sqrt{2x^2+1} - 1$. The inequality (*) follows when $x > V\sqrt{2x^2+1} - 1$ or $x < \frac{2/V}{2-1/V^2}$. ∎

Fortunately, the results analogous to Lemma 4 and Lemma 5 for the Euclidean 1-center can be easily obtained. Furthermore, we can achieve an approximation factor of $\sqrt{2}$ when our facility f follows the trajectory of the center of the bounding box of S. The velocity of f is bounded , though, by $\sqrt{2}$ (see Observation 1).

The same mixing strategy described in Lemma 7 can be applied for Euclidean 1-center problem for the facility moving with velocity $v_{max} \in [1, \sqrt{2}]$. The approximation factor from Lemma 7 grows slightly to $1 + \sqrt{2}(1 - \frac{1}{\sqrt{2}} - v_{max})(1 - \frac{2}{n})$. Due to lack of space we omit the proof of this fact in this extended abstract.

3 Mobile 1-median problems

Similarly to 1-center problem, we consider 1-median problem under L_∞ and L_2 metric. Intuitively, The maintenance of 1-median seems harder than of 1-center because now all points are participating in definition of median (only four points define center).

3.1 Rectilinear 1-median

We consider the problem where the distances are measured by the L_1 metric. We create a grid M by drawing a horizontal and a vertical line through each point of S. Assume the points of S are sorted according to their x coordinates and according to their y coordinates. Denote by $M(i,j)$ the grid point that was generated by the i^{th} horizontal line and the j^{th} vertical line in the y and x orders of S respectively. Bajaj [2] observed that the solution to the 1-median problem should be a grid point. As a matter of fact it has been shown that for this problem the point $M(\lfloor n/2 \rfloor, \lfloor n/2 \rfloor)$ is the required point. (Where for an even n the solution is not unique and there is a whole grid rectangle whose points can be chosen as the solution.) We are interested in maintaining $M(\lfloor n/2 \rfloor, \lfloor n/2 \rfloor)$ under the motion of points of S with unit velocity. As in the case of the 1-center the velocity of the 1-median can be as large as $\sqrt{2}$ times the maximum velocity of the points.

Lemma 9 *The rectilinear* 1*-median can be maintained under the motion of points in efficient KDS.*

Proof. We show how to maintain efficiently the x-coordinate of the median. The same approach works for the y-coordinate. Assume that indices of the points of S correspond to the sorted x order of points. The idea is to split the points into two sets S_1 and S_2 such that $S_1 = \{p_1, \ldots, p_{\lfloor n/2 \rfloor}\}$ and $S_2 = \{p_{\lfloor n/2 \rfloor + 1}, \ldots, p_n\}$. We maintain S_1 in the kinetic swapping heap H_1 (of the maximum maintenance problem) and S_2 in the kinetic swapping heap H_2 (of the minimum maintenance problem). We also need an additional certificate that states that $x(p_{\lfloor n/2 \rfloor}) \le x(p_{\lfloor n/2 \rfloor + 1})$. As we update the maximal value of H_1 and the minimal value of H_2 we check whether this certificate is valid, and if not, we exchange two points defining these two values between H_1 and H_2. Similarly to the maximum maintenance case the responsiveness is $O(\log n)$ and locality is $O(1)$. ∎

Approximating median by the center of mass.

Lemma 10 *The center of mass of n points provides an approximation factor of $2 - \frac{2}{n}$ for the rectilinear* 1*-median problem and this bound is tight.*

Proof. Let (c_x, c_y) be the coordinates of the center c of mass of the points of S and (m_x, m_y) be the coordinates of the rectilinear 1-median m. Want to prove that $\Sigma_{i=1}^{n} |cp_i|_1 \le (2 - \frac{2}{n}) \Sigma_{i=1}^{n} |mp_i|_1$, where $| * |_1$ stands for the L_1 distance. Following the definition of the L_1 distance it is sufficient to prove that $\Sigma_{i=1}^{n} |c_x - x(p_i)| \le (2 - \frac{2}{n}) \Sigma_{i=1}^{n} |m_x - x(p_i)|$ and $\Sigma_{i=1}^{n} |c_y - y(p_i)| \le (2 - \frac{2}{n}) \Sigma_{i=1}^{n} |m_y - y(p_i)|$. We will show the first inequality. The second inequality follows analogously. Notice that the first inequality can be viewed as 1-dimensional case of the Lemma for the points $\{x(p_1), x(p_2), \ldots, x(p_n)\}$ because their center of mass is the point c_x and their 1-median is the point m_x.

Assume that $x(p_1) \le x(p_2) \le \ldots \le x(p_n)$. For simplicity, we also assume that n is even. Let $M = \sum_{i=1}^{n} |m_x - x(p_i)|$. Rewriting M we obtain $M = \sum_{i=n/2+1}^{n} x(p_i) - \sum_{i=1}^{n/2} x(p_i)$. Let $C = \sum_{i=1}^{n} |c_x - x(p_i)|$. Our goal is to show that $C \le (2 - 2/n)M$. Let x_l and x_r be the centers of masses of $\{x(p_1), \ldots, x(p_{n/2})\}$

and $\{x(p_{n/2+1}), \ldots, x(p_n)\}$, respectively, i.e. $x_l = 2(\Sigma_{i=1}^{n/2} x(p_i))/n$ and $x_r = 2(\Sigma_{i=n/2+1}^{n} x(p_i))/n$. Clearly, $c_x = (x_l + x_r)/2$ and $x_r - x_l = 2M/n$. Let $D = \sum_{i=n/2+1}^{n}(x(p_i) - x_l) + \sum_{i=1}^{n/2}(x_r - x(p_i))$. We can rewrite D as follows:

$$D = \sum_{i=n/2+1}^{n} x(p_i) - \sum_{i=1}^{n/2} x(p_i) + n/2(x_r - x_l) = M + n/2(x_r - x_l) = 2M.$$

First we show that $C \leq D$. Let $x(p_i)$ and $x(p_j)$ be two arbitrary points such that $1 \leq i \leq n/2$ and $n/2 + 1 \leq j \leq n$. Obviously, $x(p_j)$ lies to the right of x_l. Without any loss of generality we assume that $c_x \geq x(p_{n/2})$. We show that the contribution of $x(p_i)$ and $x(p_j)$ in D is greater than their contribution in C. We now consider the following two cases:

Case 1: $x(p_j) \geq c_x$. In this case we have $(x_r - x(p_i)) + (x(p_j) - x_l) = (c_x - x(p_i)) + (x(p_j) - c_x) + (x_r - x_l) \geq (c_x - x(p_i)) + (x(p_j) - c_x)$.

Case 2: $x(p_j) < c_x$. In this case $(x_r - x(p_i)) + (x(p_j) - x_l) = (x(p_j) - x_l) + (x_r - c_x) + (c_x - x(p_i)) \geq (c_x - x(p_j)) + (c_x - x(p_i))$, because $x_r - c_x = c_x - x_l \geq c_x - x(p_j)$.

From the above two cases we can conclude that $C \leq D$. However, we can make this inequality a bit more tight. We notice that if we have k pairs of points satisfying Case 1, then actually $D = 2M \geq C + k(x_r - x_l)$. Since $x_r - x_l = 2M/n$, we have in this case $(2 - 2k/n)M \geq C$. The number k is defined by the number of points lying in interval $[x_r, \infty)$. Obviously there is at least one such point, i.e. $k \geq 1$. Thus, $(2-2/n)M \geq C$.

The tightness of this bound follows from the Example in Lemma 5. It can be easily verified that in this case $M = 1$ and $C = \frac{n-1}{n} + 1 - \frac{1}{n} = 2 - \frac{2}{n}$. ∎

3.2 Euclidean 1-median

This problem, also known as Fermat-Weber problem received a lot of attention during last century. Only iterative algorithms (see e.g. [16]) are known for this problem. The 1-median problem differs from 1-center problem not only by its difficulty but also in terms of approximation. Contrary to the 1-center case now we approximate the sum of the distances and not the distance between the facility f and the exact 1-median. Surprisingly, the rectilinear 1-median provides a constant approximation for the Euclidean 1-median.

Lemma 11 *The rectilinear 1-median provides a $\sqrt{2}$-approximation factor for the Euclidean 1-median.*

Proof. Let m_1 and m_2 be the locations of exact L_1 and L_2 medians, respectively, for a given set of points S. Let r_1 be the sum of the L_1 distances from m_1 to points in S and r_2 be the sum of the L_2 distances from m_2 to points in S, i.e. $r_1 = \Sigma_{i=1}^{n}|m_1 p_i|_1$ and $r_2 = \Sigma_{i=1}^{n}|m_2 p_i|_2$. Define $r' = \Sigma_{i=1}^{n}|m_1 p_i|_2$. Note that for any two points a, b in the plane the following holds:

$$|ab|_2 \leq |ab|_1 \leq \sqrt{2}|ab|_2. \qquad (**)$$

We want to show that $r' \leq 2r_2$. Using (**) we obtain $r_2 = \Sigma_{i=1}^{n}|m_2 p_i|_2 \geq \frac{1}{\sqrt{2}}\Sigma_{i=1}^{n}|m_2 p_i|_1 \geq \frac{r_1}{\sqrt{2}}$. On the other hand, $r' = \Sigma_{i=1}^{n}|m_1 p_i|_2 \leq \sqrt{2}\Sigma_{i=1}^{n}|m_1 p_i|_1 = r_1 \leq \sqrt{2} r_2$. The last inequality follows the bound on r_2. ∎

Based on this result and Lemma 10 we immediately conclude

Corollary 12 *The center of mass of the points gives $\sqrt{2}(2 - \frac{2}{n})$ approximation factor for the Euclidean 1-median problem.*

We can use again the mixing strategy from Lemma 7 for the facility moving with velocity $v_{max} \in [1, \sqrt{2}]$ by mixing the strategies of the center of mass of the points and the rectilinear 1-median. It provides approximation factor of $3 + \frac{a - v_{max}}{a-1}$, where $a = \frac{1+\sqrt{2}}{2}$. This result follows from the fact the function $s(x, y) = \Sigma_{i=1}^{n}\sqrt{(x - x_i)^2 + (y - y_i)^2}$ defining the sum of the distances from the point (x, y) in the plane to points of S is concave. Details are provided in the full version.

4 Conclusion

In this paper we investigated the complexity of the mobile 1-center and 1-median problems. Possible future directions for the research in this area include: providing tighter bounds for mobile 1-median problem, obtaining precise trade-offs between the velocities of the facilities and approximation factors and generalization of our results to the mobile k-center and k-median problems for $k \geq 2$.

References

[1] P. Agarwal and M. Sharir. "Planar geometric location problems", *Algorithmica*, 11, pp. 185–195, 1994.

[2] C. Bajaj "Geometric optimization and computational complexity", Ph.D. thesis. Tech. Report TR-84-629. Cornell University (1984).

[3] J. Basch, "Kinetic Data Structures", Ph.D. thesis, Stanford University, USA, 1999.

[4] J. Basch, L. Guibas and J. Hershberger "Data structures for mobile data", In *Proc. 8th ACM-SIAM Sympos. Descrete Algorithms*, pp. 747–756, 1997.

[5] S. Bespamyatnikh, K. Kedem and M. Segal "Optimal facility location under various distance functions", *Workshop on Algorithms and Data Structures (WADS'99)* Lecture Notes in Computer Science 1663, Springer-Verlag, pp. 318–329.

[6] S. Bespamyatnikh and M. Segal "Rectilinear static and dynamic center problems", *Workshop on Algorithms and Data Structures (WADS'99)* Lecture Notes in Computer Science 1663, Springer-Verlag, pp. 276–287.

[7] J. Brimberg and A. Mehrez "Multi-facility location using a maximin criterion and rectangular distances", *Location Science* 2, pp. 11–19, 1994.

[8] Z. Drezner "The p-center problem: heuristic and optimal algorithms", *Journal of Operational Research Society*, 35, pp. 741–748, 1984.

[9] Z. Drezner "On the rectangular p-center problem", *Naval Res. Logist. Q.*, 34, pp. 229–234, 1987.

[10] M. Dyer and M. Frieze "A simple heuristic for the p-center problem", *Oper. Res. Lett.*, 3:285-288,1985.

[11] H. Elgindy and M. Keil "Efficient algorithms for the capacitated 1-median problem", *ORSA J. Comput*, 4, pp. 418–424, 1982.

[12] D. Eppstein "Faster construction of planar two-centers", *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms*, pp. 131–138, 1997.

[13] L. Guibas "Kinetic data structures: A state of the art report", In *Proc. 1998 Workshop Algorithmic Found. Robot.*, pp. 191–209, 1998.

[14] D. Hochbaum and D. Shmoys "A best possible approximation algorithm for the k-center probem", *Math. Oper. Res.*, 10:180-184, 1985.

[15] J. Jaromczyk and M. Kowaluk, "An efficient algorithm for the Euclidian two-center problem", *Proc. 10th ACM Sympos. Comput. Geom.*, pp. 303-311, 1994.

[16] H. Kuhn, "A note on Fermat's problem", *Mathematical Programming*, 4, pp. 98–107, 1973.

[17] J. Lin and J. Vitter, "Approximation algorithm for geometric median problems", *Inform. Proc. Lett.*, 44:245–249, 1992.

[18] N. Megiddo "Linear time algorithms for linear programming in R^3 and related problems", *SIAM J. Comput.*, 12, pp. 759–776, 1983.

[19] N. Megiddo and A. Tamir "New results on the complexity of p-center problems", *SIAM J. Comput.*, 12(4), pp. 751–758, 1983.