On Some Optimization Problems in Obnoxious Facility Location*

Zhongping Qin[1], Yinfeng Xu[2], and Binhai Zhu[3]

[1] Dept. of Mathematics, Huazhong University of Science and Technology, China
[2] School of Management, Xi'an Jiaotong University, Xi'an, China
[3] Dept. of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.
bhz@cs.cityu.edu.hk

Abstract. In this paper we study the following general MaxMin-optimization problem concerning undesirable (obnoxious) facility location: Given a set of n sites S inside a convex region P, construct m garbage deposit sites V_m such that the minimum distance between these sites V_m and the union of S and V_m, $V_m \cup S$, is maximized. We present a general method using Voronoi diagrams to approximately solve two such problems when the sites S's are points and weighted convex polygons (correspondingly, V_m's are points and weighted points and the distances are L_2 and weighted respectively). In the latter case we generalize the Voronoi diagrams for disjoint weighted convex polygons in the plane. Our algorithms run in polynomial time and approximate the optimal solutions of the above two problems by a factor of 2.

1 Introduction

In the area of environmental operations management we usually face the problem of locating sites to deposit (nuclear and/or conventional) wastes. Because of the radiation and pollution effect we do not want to locate such a site too close to a city. Moreover, we cannot even locate two sites too close to each other (even if they are far from cities where human beings reside) — this is especially the case when the sites are used to deposit nuclear wastes as the collective radiation becomes stronger.

Another related application is something to do with satellite placement. In the sky we already have quite a lot of satellites and certainly the number is increasing year by year. When we launch a new satellite, we certainly need to place it at such a location which is far from the existing ones as long as there is no other constraints — putting them too close would affect the communication quality.

This problem arises in many other applications in practice. However, how to find efficient solutions for these kinds of problems is not an easy task and this

* The authors would like to acknowledge the support of research grants from NSF of China, Research Grants Council of Hong Kong SAR, China (CERG Project No. CityU1103/99E) and City University of Hong Kong

D.-Z. Du et al. (Eds.): COCOON 2000, LNCS 1858, pp. 320–329, 2000.

presents a new challenge for computational geometers. In this paper we study several versions of this problem. We call this general problem the Undesirable Facility Location Problem (UFL, for short) and formulate two versions of the problem as follows.

I. Undesirable Point Placement Problem (UPP). Given a convex polygonal domain P and a set of point sites S in P, find the location of m undesirable or obnoxious facilities in P, which consist of a point set V_m, so as to

$$\texttt{Maximize}\ \{\hat{d}(V_m, V_m \cup S)\}$$

where the maximum is over all the possible positions of the points in V_m and $\hat{d}(V_m, V_m \cup S)$ defined as

$$\hat{d}(V_m, V_m \cup S) = \min_{\{x \neq q\} \wedge \{x \in V_m\} \wedge \{q \in V_m \cup S\}} \{d(x, q)\}$$

with $d(x, q)$ being the Euclidean distance between x and q.

Note that the Undesirable Point Placement problem (UPP) is evidently a MaxMin-optimization problem. We can see that the solution maximizes the distances between the undesirable facilities and the distances between the ordered pairs of points from the undesirable facilities and the sites. Therefore, the damage or pollution to the sites is low.

In reality, we know that the importance, areas, or the capacity of enduring damages of various cities are different. We take this into consideration and make the problem more general and practical, we introduce the weighted function $w(q)$ for point q, extend the point sites to disjoint weighted convex polygons and permit the metric to be either L_2 or L_1. Our objective is to distinguish the difference of importance of various sites, capture the real geographic conditions and to satisfy various needs in practice. We therefore formulate the following more general problem.

II. Undesirable Weighted Point Placement Problem (UWPP). Given a convex polygonal domain P and a set of disjoint convex polygonal sites S in P where each point q in a polygon $s \in S$ has weight $w(q) = w(s)$ ($w(s) \geq 1$ is associated with the polygon s), find the location of m undesirable facilities in P, which consist of a point set V_m and have the same weight $1 \leq w_0 \leq w(s), s \in S$, so as to

$$\texttt{Maximize}\ \{\hat{d}_w(V_m, V_m \cup S)\}$$

where the maximum is over all the possible positions of the points in V_m and $\hat{d}_w(V_m, V_m \cup S)$ is defined as

$$\hat{d}_w(V_m, V_m \cup S) = \min_{\{x \neq q\} \wedge \{x \in V_m\} \wedge \{q \in V_m \cup S\}} \{d_w(x, q)\}$$

with $d_w(x, q) = \frac{1}{w(q)} d(x, q)$, $d(x, q)$ being the Euclidean distance in between x and q.

We remark that the Undesirable Weighted Point Placement Problem (UWPP) is a much more complex MaxMin-optimization problem. If set V_m is chosen such that the objective function of UWPP reaches its optimal value (which is called the m-optimal value of UWPP, denoted as d^*_m) then we call the thus chosen set V_m a m-extreme set of UWPP, denoted as V^*_m.

It is obvious that in a solution to UWPP the undesirable facilities are farther away from those sites with larger weights. Hence the sites with larger weights can get more protect against the undesirable facilities. Although we formulate the problem as the Undesirable Weighted Point Placement Problem in the plane, it can be formulated as problem in other fields like packing satellites in the sky or packing transmitters on the earth.

It should be noted that finding the optimal solutions, V^*_m, for either UPP or UWPP is not an easy task. The exhaustive search method is not possible as the spaces of the solutions to the problems are continuous regions in R^{2m} and the minimizing step of the objective function concerns some continuous regions in R^2. So far we have not been able to obtain an algorithm to find an optimal solution for either of the two problems even if it is of exponential time complexity. The difficulty of the problem can also be seen from the following. It is not yet known how to pack as many as possible unit circles in a given special polygon P, like a square. In this setting, S is empty and we want to determine the maximum set V_m such that the distance $\hat{d}(V_m, V_m)$ is at least two [13].

Although the problem is hard to solve, in practice, usually an approximate solution is acceptable as long as the solution is efficient. In this paper we present a general incremental Voronoi diagram algorithm which approximate the optimal solution of the two problems by a factor of 2.

2 The Incremental Voronoi Diagram Algorithm for UPP

In this section we present an incremental Voronoi diagram algorithm for approximating the Undesirable Point Placement Problem (UPP). The basic ideas of the algorithm are as follows: (1). We successively choose the points of V_m from a discrete set. At i-th step we choose the point which is the farthest away from the points in V_{i-1} chosen previously and all sites in S. (2). The point in $V_i - V_{i-1}$ which is the farthest from $V_{i-1} \cup S$ is always among the Voronoi vertices of $Vor(V_{i-1} \cup S)$, vertices of polygon P and the intersection points of the Voronoi edges of $Vor(V_{i-1} \cup S)$ and P.

Basically, the algorithm uses a strategy of computing points in V_m from a finite discrete field rather than from a continuous field. The algorithm and analysis is as follows.

Algorithm UPP

Input: A convex polygon P, a set of points S in P.

Output: The set of points V_m in P.

Procedure:

1. Initialize $V := \emptyset$.
2. Compute the Voronoi diagram of $(V \cup S)$, $Vor(V \cup S)$.

3. Find the set B consisting of the Voronoi vertices of $(V \cup S)$, the vertices of P and the intersection points between Voronoi edges and the edges of P. Among the points in B, choose the point v which maximizes $d(v, q_v), v \in B$, where $q_v \in V \cup S$ and $Vor\_region(q_v)$ contains point v.
4. Update $V := V \cup \{v\}$ and return to Step 2 when $|V| \leq m$.

We have the following theorem regarding the above algorithm.

Theorem 1. *The output V_m generated by Algorithm UPP presents a 2-approximation to the optimal solution for the Undesirable Point Placement Problem (UPP).*

The proof of Theorem 1 is based on the following lemma, which is similar to the methods used in [6,8].

Lemma 2. *Given any convex polygon P and any set S of point sites in P, for any set V_t of t points in P there exists a point $p \in P$ such that $d(p, V_t \cup S) \geq \frac{d^*_{t+1}}{2}$, where d^*_{t+1} is the $(t+1)$-optimal value of the UPP, where $d(p, V_t \cup S)$ denotes the Euclidean distance between point p and set $V_t \cup S$.*

Proof. Suppose that the lemma is not true. Then there exists a point set V_t in P such that for any $p \in P$,

$$\max d(p, V_t \cup S) < \frac{d^*_{t+1}}{2}.$$

Let r be the value of the left-hand side and we have $r < \frac{d^*_{t+1}}{2}$. For each point $q \in V_t \cup S$, draw a circle centered at q with radius r. By the optimality of r, every pair of points in P and $V_t \cup S$ must be at most r distance away. Therefore, it is obvious that the union of these circles must cover the whole area of P. This implies that one of these circles must cover two points of $V^*_{t+1} \cup S$. Consequently, $d^*_{t+1} = \hat{d}(V^*_{t+1}, V^*_{t+1} \cup S) \leq 2r$. This contradicts the fact that $r < \frac{d^*_{t+1}}{2}$. □

Proof for Theorem 1. For any $k \leq m$ let point $p \in P$ maximizes $d(x, V_k \cup S), x \in P$. We assert that p must be in the set of points consisting of the Voronoi vertices of $Vor(V_k \cup S)$, the vertices of P and the intersection points between Voronoi edges of $Vor(V_k \cup S)$ and the edges of P. In other words p must be a boundary point of some Voronoi region of $Vor(V_k \cup S) \cap P$. If it is not the case then p would fall in the interior of $Vor\_region(q) \cap P$ for some $q \in V_k \cup S$. Hence there is one point $p\,'$ on the boundary of $Vor\_region(q) \cap P$ such that $d(p\,', q) > d(p, q)$ which contradicts the definition of p. Hence for point v_k, the k-th $(k \leq m)$ chosen point for set V_m by step 3 of Algorithm UPP, $d(v_k, V_{k-1} \cup S) = \max_{\{x \in P\}}\{d(x, V_{k-1} \cup S)\}$ (recall that $d(v_k, V_{k-1} \cup S)$ denotes the distance between point v_i and set $V_{i-1} \cup S$).

Also, we have the relation

$$d(v_k, V_{k-1} \cup S) \leq d(v_{k-1}, V_{k-2} \cup S), \; k = 2, 3, \cdots, m.$$

The reason is that all $Vor\_region(q), q \neq v_k$, which belong to $Vor(V_k \cup S) \cap P$ will not grow relative to $Vor\_region(q) \in Vor(V_{k-1} \cup S) \cap P$ after point v_k is inserted. In addition, the point on the boundary of $Vor\_region(v_k) \in Vor(V_k \cup S) \cap P$ which is the farthest to v_k is always on the boundary of $Vor\_region(q) \in Vor(V_k \cup S) \cap P, q \in V_{k-1} \cup S$, which is adjacent to $Vor\_region(v_k) \in Vor(V_k \cup S) \cap P$. So from this relation by induction we can obtain the result that $d(v_k, V_{k-1} \cup S) = d(V_k, V_k \cup S)$. Hence by the above assertion and Lemma 2 $\hat{d}(V_m, V_m \cup S) = d(v_m, V_{m-1} \cup S) = \max_{\{x \in P\}}\{d(x, V_{m-1} \cup S)\} \geq \frac{d^*_m}{2}$. This shows that V_m, the output of Algorithm UPP, is a 2-approximation of V^*_m (the m-extreme set of UPP). □

Time complexity. The algorithm runs in $O(mN \log N)$ time, where $N = |S| + |P| + m$. At each of the m iterations, Step 2 takes $O(N \log N)$ time, Step 3 involves finding the nearest neighbor of each vertex v hence takes $O(N \log N)$ time — we need to perform $O(N)$ point locations each taking logarithmic time. It is possible to improve Step 2 by using the dynamic Voronoi diagram (Delaunay triangulation) algorithm of [3,4] so that over all of the m iterations [1], Step 2 takes $O(N \log N + m \log N)$ time instead of $O(mN \log N)$ time. But this will not change the overall time complexity of the algorithm as Step 3 takes $O(N \log N)$ time at each of the m iterations.

3 A Generalization to the Weighted Plane

In this section, we consider yet another generalization of the problem to the weighted (L_2 or L_1) plane. We make some necessary adjustment to Algorithm UPP so as to approximately solve the Undesirable Weighted Point Placement Problem (UWPP). The basic idea is the same as Algorithm UPP. But the set B in step 3 should be change to the set B' consisting of the Voronoi vertices and dividing points of $Vor(V_t \cup S)$, vertices of P, the intersection points between Voronoi segments of $Vor(V \cup S)$ and the edges of P.

3.1 The Weighted Voronoi Diagram of Convex Polygons in the Plane

We generalize the concept of Voronoi diagram of points in the plane in three aspects: (1) change point sites to disjoint convex polygons, (2) each site has a positive weight which is at least one, (3) under both metric L_2 and L_1 (we will focus on L_2). To the best knowledge of the authors, in metric L_2 the Voronoi diagram of weighted points in the plane is well studied [1], but the Voronoi diagram of weighted convex polygon objects is seldomly studied. In metric L_1

[1] In practice, we advocate the use of [7] to update the Voronoi diagram when a new point is added and the point location algorithm of [5,12].

there are only results on the Voronoi diagram of points in the plane [10,11] before this work.

Let S be a set of disjoint convex polygons in the plane which are associated with weights $w(s) \geq 1, s \in S$. The *weighted Voronoi diagram of S* (for short WVD(S)) is a subdivision of the plane consisting of Voronoi regions, Voronoi faces, Voronoi edges, Voronoi segments, Voronoi vertices and Voronoi dividing points. *Vor_region*(s) is defined as

$$Vor\_region(s) = \{x | \frac{1}{w(s)} d(x, s) \leq \frac{1}{w(t)} d(x, t), t \in S\}.$$

A *Voronoi face* of the WVD(S) is a connected component of a thus defined Voronoi region. A *Voronoi edge* is the intersection of two Voronoi faces. A *Voronoi segment* is the maximal portion of a Voronoi edge which can be described by a curve equation. A *Voronoi dividing point* is an endpoint of a Voronoi segment and a *Voronoi vertex* is the intersection of three Voronoi edges. We now focus on L_2 and present the necessary details. In Figure 1, we show an example of three convex polygons with weights 1, 2 and 3 (for convenience, we use 1, 2 and 3 to represent them as well). Note that the Voronoi edge between 2 and 3 which is inside P is composed a set of Voronoi segments.

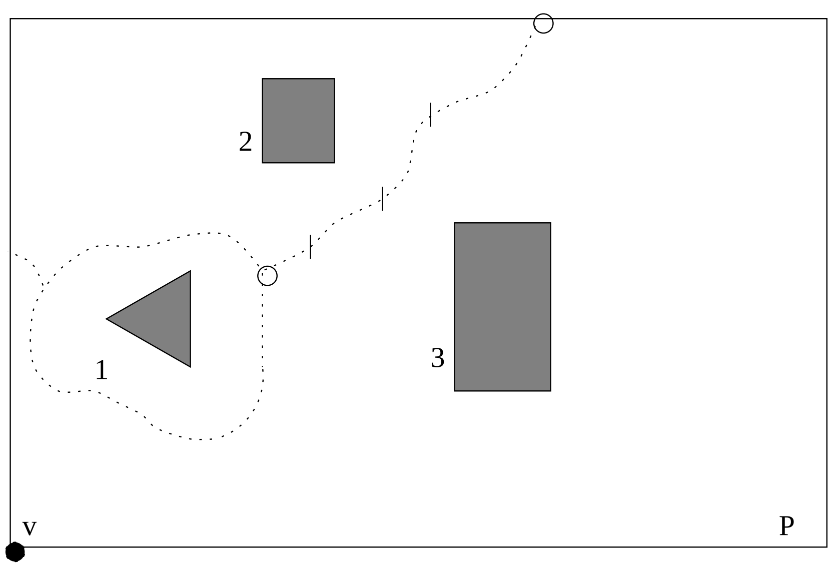

Fig. 1. A weighted Voronoi diagram of three convex polygons.

Lemma 3. *Suppose $S = \{s_1, s_2\}$ consist of two weighted disjoint convex polygons in the plane and $1 \leq w(s_1) < w(s_2)$, then Vor_region(s_1) is a limitary connected region. The boundary of Vor_region(s_1) (which is a Voronoi edge) consists of some Voronoi segments which are parts of conic curves, respectively, i.e.*

1. A Voronoi segment is a part of a circle determined by equation

$$\frac{1}{w(p)}d(x,p) = \frac{1}{w(q)}d(x,q)$$

where x is a point of the Voronoi segment and p, q are a pair of vertices of two convex polygons s_1 and s_2.

2. A Voronoi segment is a part of a hyperbola, parabola or ellipse determined by equation

$$\frac{1}{w(p)}d(x,p) = \frac{1}{w(e)}d(x,e)$$

where x is a point of the Voronoi segment and p (e) is vertex (edge) convex polygon s_1 (s_2). The segment is a part of a hyperbola, parabola or ellipse when the weight of the vertex p is bigger than, equal to or less than the weight of the edge e.

3. A Voronoi segment is a straight line segment determined by equation

$$\frac{1}{w(e_1)}d(x,e_1) = \frac{1}{w(e_2)}d(x,e_2)$$

where x is a point of the Voronoi segment and e_1, e_2 are a pair of edges of two convex polygons s_1 and s_2.

There are some difference between the WVD(S) and the classic Voronoi diagram of points (for short VD(P)) or the weighted Voronoi diagram of points (for short WVD(P)): an Voronoi edge in VD(P) or WVD(P) has just one Voronoi segment which is a part of a straight line (in VD(P)) or a part of circle (in WVD(P)), but a Voronoi edge in WVD(S) contains some Voronoi segments which are of different conic curves. Therefore, in WVD(S) Voronoi dividing points are different from Voronoi vertices. Similar to [2], we make the following definition.

Definition 1. *Given two weighted disjoint convex polygons s_1, s_2 in the plane and $1 \le w(s_1) < w(s_2)$, $Vor\_region(s_1)$ is defined as the dominance of s_1 over s_2, for short $dom(s_1, s_2)$, and the closure of the complement of $Vor\_region(s_1)$ is called the dominance of s_2 over s_1, for short $dom(s_2, s_1)$.*

The following lemma follows from the definition of Voronoi diagrams and is the basis for constructing WVD(S).

Lemma 4. *Let S be a finite set of weighted convex polygons in the plane and $s \in S$.*

$$Vor\_region(s) = \bigcap_{t \in S - \{s\}} dom(s,t).$$

In general, $Vor\_region(s)$ in WVD(S) may have $O(|S|)$ Voronoi faces, i.e., the region does not need to be connected and its connected parts do not need to be simply connected. Overall, the WVD(S) can be computed in $O(|S|^3)$ time and $O(|S|^2)$ space, using standard techniques in computational geometry [14].

3.2 Algorithm

We now present the approximation algorithm for UWPP.

Algorithm UWPP

Input: A convex polygon P, a set S of disjoint weighted convex polygons in P and each polygon $s \in S$ has a weight $w(s) \geq 1$.
Output: The set of points V_m, all with weight $1 \leq w_0 \leq w(s), s \in S$.
Procedure:

1. Initialize $V := \emptyset$.
2. Compute the weighted Voronoi diagram of $(V \cup S)$, WVD$(V \cup S)$.
3. Find the set B' consisting of the Voronoi vertices and dividing points of $WVD(V \cup S)$, the vertices of P and the intersection points between Voronoi segments of $WVD(V \cup S)$ and the edges of P. Among the points in B', choose the point v which maximizes $\frac{1}{w(q_v)}d(v, q_v)$, where $v \in B'$, $q_v \in V \cup S$ and $v \in Vor\_region(q_v)$.
4. Update $V := V \cup \{v\}$ and return to Step 2 when $|V| \leq m$.

In Figure 1, if we only add one obnoxious facility then it would be placed at the low-left corner of P. We have the following theorem.

Theorem 5. *The output V_m of Algorithm UWPP is a 2-approximation to the Undesirable Weighted Point Placement Problem.*

First we need to generalize Lemma 2, which is important to the proof of Theorem 1, to the following Lemma 6.

Lemma 6. *Let P and S be as in the definition of UWPP. For any set V_t of t points in P with weight $1 \leq w_0 \leq w(s), s \in S$ there exists a point $x \in P$ such that*

$$\max \frac{1}{w(q_x)} d(x, q_x) \geq \frac{d^*_{t+1}}{2}, x \in P,$$

where $q_x \in V_t \cup S, x \in Vor\_region(q_x)$ and d^*_{t+1} is the $(t+1)$-optimal value of UWPP.

Proof. Suppose the lemma is not true, then there exists a point set V_t in P (note that points in V_t have the same weight $1 \leq w_0 \leq w(s), s \in S$) such that

$$\max \frac{1}{w(q_x)} d(x, q_x) < \frac{d^*_{t+1}}{2}, x \in P$$

where $q_x \in V_t \cup S$ and x is contained in $Vor\_region(q_x)$. Let r be the value of the left-hand side, so we have $r < \frac{d^*_{t+1}}{2}$. For $q \in V_t \cup S$ ($V_t \cup S$ is either a point in V_t or a convex polygon in S), we expand q to $\bar{q}$ such that $\bar{q}$ contains all the points within $w(q)r$ distance (in L_2) to q. By the optimality of r, $\bigcup_{q \in V_t \cup S} \{\bar{q}\}$

contains all the points in P. So there are two points of $V^*_{t+1} \cup S$ (V^*_{t+1} is the $(t+1)$-extreme set of UWPP) contained in some $\bar{q}$ ($q \in V_t$). This implies that the $(t+1)$-optimal value of the UWPP $d^*_{t+1} \leq 2r$, a contradiction with $r < \frac{d^*_{t+1}}{2}$. □

Proof for Theorem 5. Let v_k be the k-th chosen point for set V_m by step 3 of Algorithm UWPP. Similar to the proof of Theorem 1 we also have the result that v_k maximizes $\frac{1}{w(q_x)}L(x, q_x),\ x \in P$ where $q_x \in V_{k-1} \cup S$ and $x \in \mathit{Vor\_region}(q_x)$. We assert that $\frac{1}{w(q_{v_k})}d(v_k, q_{v_k}) = \hat{d}_w(V_k, V_k \cup S)$. In fact, we have

$$\hat{d}_w(v_{k-1}, q_{v_{k-1}}) \geq \hat{d}_w(v_k, q_{v_k}), k = 1, 2, \cdots, m.$$

The reason is that as $w_0 \leq w(s), s \in S$, all $\mathit{Vor\_region}(q), q \neq v_k$, which belong to $\mathit{Vor}(V_k \cup S)$ will not grow relative to $\mathit{Vor\_region}(q) \in \mathit{Vor}(V_{k-1} \cup S)$ after point v_k is inserted. Also, the point on the boundary of $\mathit{Vor\_region}(v_k) \in \mathit{Vor}(V_k \cup S)$ which is the farthest to v_k is always on the boundary of $\mathit{Vor\_region}(q) \in \mathit{Vor}(V_k \cup S), q \in V_{k-1} \cup S$, which is adjacent to $\mathit{Vor\_region}(v_k) \in \mathit{Vor}(V_k \cup S)$. Based on this fact we can easily prove the above assertion by induction. Therefore, following Lemma 6

$$\begin{aligned}\hat{d}_w(V_m, V_m \cup S) &= \tfrac{1}{w(q_{v_m})}d(v_m, q_{v_m}) \\ &= \max_{\{x \in P\}}\{\tfrac{1}{w(q_x)}d(x, q_x)\}, \\ &\quad \mathit{where}\ q_x \in V_{m-1} \cup S\ \mathit{and}\ x \in \mathit{Vor\_region}(q_x) \\ &\geq \tfrac{d^*_m}{2}.\end{aligned}$$

□

Time complexity. The algorithm runs in $O(mN^3)$ time, where $N = |S| + |P| + m$. At each of the m iterations, Steps 2 and 3 take $O(N^3)$ time and $O(N^2)$ space.

We comment that the WVD(S) can be generalized to L_1 metric. Moreover, the facilities can have different weights (but we then should first place the facility with the largest weight, among those unplaced ones). The details are omitted.

Finally, we comment that it might be possible for us to trade the running time of this algorithm with the approximation factor. We can use the geometric Voronoi diagram of S in L_2, which is planar and can be constructed in $O(N \log N)$ time and $O(N)$ space [9,15,16].

4 Concluding Remarks

In this paper we present a method to solve a MaxMin-optimization problem in obnoxious facility location. The method is general and can be generalized to many interesting cases when the sites are not necessary points. It is an open question whether we can obtain an optimal solution for the problem (even it is of exponential time complexity). Another question is whether the $O(N^3)$ time for computing WVD(S) in Section 3 can be further improved. Finally, all the facilities we consider are points (or weighted points); however, in practice sometimes the facilities might be of some size as well. It is not know whether our method can be generalized to these situations. For example, what if the facilities are unit circles?

References

[1] F. Aurenhammer, Voronoi diagrams: a survey of a fundamental geometric data structures, *ACM Comput. Surveys*, 23(3), pp. 343-405, 1991.

[2] F. Aurenhammer and H. Edelsbrunner, An optimal algorithm for constructing the weighted Voronoi diagram in the plane, *Pattern Recognition*, 17, pp. 251-257, 1984.

[3] J. Boissonnat, O. Devillers, R. Schott, M. Teillaud and M. Yvinec, Applications of random sampling to on-line algorithms in computational geometry, *Disc. Comp. Geom.* 8, pp. 51-71, 1992.

[4] O. Devillers, S. Meiser and M. Teillaud, Fully dynamic Delaunay triangulation in logarithmic expected time per operation, *Comp. Geom. Theory and Appl.* 2, pp. 55-80, 1992.

[5] L. Devroye, E. Mücke and B. Zhu, A note on point location in Delaunay triangulations of random points, *Algorithmica*, special issue on Average Case Analysis of Algorithms, *22(4)*, pp. 477-482, Dec, 1998.

[6] T. Feder and D. Greene, Optimal algorithms for approximate clustering, *Proc. 20th STOC*, pp. 434-444, 1988.

[7] L. Guibas, D. Knuth and M. Sharir, Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica*, 7, pp. 381-413, 1992.

[8] T. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theo. Comput. Sci.*, 38, pp. 293-306, 1985.

[9] R. Klein, K. Mehlhorn and S. Meiser, Randomized incremental construction of abstract Voronoi diagrams, *Comp. Geom. Theory and Appl.* 3, pp. 157-184, 1993.

[10] D.T. Lee and C.K. Wong, Voronoi diagrams in the L_1 (L_∞) metric with two-dimensional storage applications, *SIAM. J Comput.*, 9, pp. 200-211, 1980.

[11] D.T. Lee, Two-dimensional Voronoi diagrams in the L_p metric, *J. ACM*, 27, pp. 604-618, 1980.

[12] E. Mücke, I. Saias and B. Zhu, Fast Randomized Point Location Without Preprocessing in Two and Three-dimensional Delaunay Triangulations, *Comp. Geom. Theory and Appl*, special issue for SoCG'96, *12(1/2)*, pp. 63-83, Feb, 1999.

[13] K. Nurmela and P. Ostergard, Packing up to 50 equal circles in a square, *Disc. Comp. Geom.* 18, pp. 111-120, 1997.

[14] F. Preparata and M. Shamos, Computational Geometry, Springer-Verlag, 1985.

[15] P. Widmayer, Y.F. Wu, and C.K. Wong, On some distance problems in fixed orientations. *SIAM J. Comput.*, 16, pp. 728–746, 1987.

[16] C.K. Yap, An O($n \log n$) algorithm for the Voronoi diagram of a set of curve segment, *Disc. Comp. Geom.* 2, pp. 365-393, 1987.