國立交通大學

資訊科學與工程研究所

碩 士 論 文

競爭性區位設施之防禦問題

The Defensive Competition Problem

研 究 生：黃敬強

指導教授：李德財　教授

中 華 民 國 九十五 年 六 月

競爭性區位設施之防禦問題

The Defensive Competition Problem

研 究 生 ： 黃敬强　　Student : Ching-Chiang Huang

指 導 教 授 ： 李德財　　Advisor: D. T. Lee

國 立 交 通 大 學

碩 士 論 文

A Thesis

Submitted to Department of Departement of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

June, 2006

Taipei, Taiwan

中華民國九十五年 六月

Contents

Title **ii**

Tabel of Contents **ii**

Abstract in Chinese **iv**

Abstract **v**

Acknowledgements **vi**

1 Introduction **1**

2 Defensive Competition Problem **4**

2.1 Competitive Location Problems 4

2.2 One-to-one Defensive Competition Problem 5

3 Problem Transformation **6**

4 K-th Depth Contour and K^+-th Depth Contour **9**

4.1 K-th Depth Contour 9

4.2 K^+-th Depth Contour 11

5 The Proposed Algorithm **13**

5.1 Dual Mapping 13

5.2 Topological Sweep 16

5.3 Minkowski Sum 18

6 Problem Relationships **21**

6.1 One-to-one Defensive Competition Problem 21

6.2 one-to-k Defensive Competition Problem 22

6.3 k-to-one Defensive Competition Problem 22

6.4 Conclusion 23

競爭性區位設施之防禦問題

學生： 黃敬强　　　　　　　　　　　　　指 導 教 授 ： 李德財

國立交通大學

摘　　要

競爭性區位設施問題 是個著名並且尚未被完全解決的問題。此題目基本的型態是給定 n 個需求點，及一區位設施之集合。目標是要找到另一區位設施來搶走最多的需求點。

本論文我們加入了防禦之概念。也提出了競爭性區位設施之防禦問題。此問題的基本型態是給定 n 個需求點，找出一防禦區位設施之集合來抵抗任何之進攻區位設施之集合。我們主要專注於一對一競爭性區位設施之防禦問題也就是只需要防禦一個進攻區位設施。我們提供一$O(n^2)$ 時間以及 $O(n^2)$ 空間複雜度之解法。

同時我們也討論了競爭性區位設施之防禦問題與其他計算幾何領域內的著名題目例如，$Tukey$ 中間值及最小圓盤集合之間的關係。最後我們延伸一對一競爭性區位設施之防禦問題並證明一對k競爭性區位設施之防禦問題是NP-hard而k對一競爭性區位設施之防禦問題則是還沒有解的問題。

The Defensive Competition Problem

Student: Ching-Chiang Huang   Advisor: D. T. Lee

Department of Departement of Computer Science
National Chiao Tung University

ABSTRACT

The *competitive location problem* is a well-known problem of great interest and not yet fully explored. The basic setting is: given a set of demand points and a set of existing facilities, find a new set of facilities that will attract the most demand points.

In this thesis we incorporate the idea of *defense* and introduce a new model called the *defensive competition problem*. The basic setting is: given a set of demand points, find a set of *defending* facilities that will defend the most demand points from any set of *attacking* facilities. Our main focus is on the *one-to-one defensive competition problem* where there is only one attacking facility to defend from. We give an $O(n^2)$ time and $O(n^2)$ space solution for this problem.

We also show that the one-to-one defensive competition problem is a generalization of the *Tukey median problem* and the *smallest enclosing circle problem* which are two famous problems in computational geometry. Last, we generalize the one-to-one defensive competition problem and show that the *one-to-k defensive competition problem* is a generalization for the (r, X_p)-*medianoid problem* which is NP-hard while the *k-to-one defensive competition problem* remains an open problem.

誌　　　　謝

在這裡要謝謝實驗室的大家!

所長精闢的見解總是能在很短的個別 meeting 內給我很大的啓發.

博士班的 kero, shou, ccli, alassi 學長.

與 kero 老大的討論總是很有收穫, kero 學長讓我感受到他對研究的熱情和學術研究是應該如何來進行. 廖崇碩學長負責帶領我們理論組的運作, 謝謝你讓我們兩年過得那麼愉快及順暢而且充實. 李政崇老大每次都在 msn 上面聊聊天讓人放鬆. alassi 學長在我的題目上幫助非常大, plus contour 的概念就是來自於你.

碩士班的同學 eieio, bbmouth, waitingN, neutrino

eieio 每次 meeting 偷偷跟我聊橋牌, I'm going to miss that.

bbmouth OS project 罩我我不會忘記的!

waitingN 跟你聊天相當之愉快啊.

neutrino 不要在要宅了啊!!!

學弟 mong,魔獸繼續衝啊!

碩士兩年能在這個實驗室實在太幸運了, 也謝謝大家容忍我的賤嘴啊!

Chapter 1

Introduction

Suppose there are ice cream vendors on the beach. People on the beach would prefer to buy ice cream from the nearest vendor possible. Thus, the placement of the ice cream vendors would determine how much business each vendor would be able to attract. Deciding where to place new vendors to compete against existing vendors and attract the most business is considered an instance of the competitive location problem.

We incorporate the idea of *defense* into the *competitive location problem*[2] and introduce a new model, called the *defensive competition problem.* Our main focus is on the subcase *one-to-one defensive competition problem* for which we present an $O(n^2)$ time and $O(n^2)$ space solution. We discuss generalizations of the *defensive competition problem* and many famous computational geometry problems such as the *Tukey median problem*[6], the smallest enclosing circle problem and the (r, X_p)-*medianoid problem*[21] are found to be closely related to the defensive competition problem.

In 1929, Hotelling[1] described the univariate median as the point which minimizes the maximum number of data points on one of its sides. This notion was generalized to higher dimensions many years later by Tukey[6]. The Tukey median, or *half-space* median, is perhaps the most widely studied and used multivariate median in recent years.

The Tukey median can be better understood by considering the *half-space depth* for a query point x with respect to a set P of n demand points in $\Re^d$. For each half-space that contains x, count the number of demand points in P which are in the half-space. Take the minimum number found over all half-spaces to be the half-space depth of x.

For example in $\Re^2$, place a line through x so that the number of demand points on one side of the line is minimized. The half-space median of a data set is any point in $\Re^d$ which has maximum half-space depth. We shall give the formal definition of the Tukey median in later sections.

Notice that any point outside of the convex hull of P has depth zero. To find the region of maximum depth in $\Re^2$, we can use the fact that its boundaries are segments of lines passing through pairs of demand points. In other words, the vertices of the desired region must be intersection points through pairs of demand points. There are $O(n^4)$ intersection points, and it is a straightforward task to find the depth of a query point in $O(n^2)$ time(for each line defined by the query point and a demand point, count the number of demand points above and below the line). Thus, in $O(n^6)$ time, we can find the intersection points of maximum depth.

An improvement upon this was made by Rousseeuw and Ruts[7], who showed how to compute the half-space depth of a point in $O(n \log n)$ time. Thus using this as a subroutine an $O(n^5 \log n)$ time algorithm is obtained to compute the deepest point. Later they also gave an $O(n^2 \log n)$ time algorithm which is more complicated and provided an implementation[8]. Before that, Matoušek[9] presented an $O(n \log^5 n)$ algorithm for computing the half-space median. Matoušek showed how to compute any point with depth greater than some constant k in $O(n \log^4 n)$. With this as a subroutine, a binary search on k can be used to find the median. Matoušek's algorithm was improved by Langerman and Steiger[10], whose algorithm computes the median in $O(n \log^4 n)$ time. A further improvement and the current best result is an $O(n \log^3 n)$ time algorithm that appeard in Langerman's Ph.D. thesis[11]. A recent implementation by Miller et al.[12] uses $O(n^2)$ time and space to find all depth contours, after which it is easy to compute the median. Recently, Chan[13] gave an optimal randomized algorithm to compute the half-space median, with time complexity $O(n \log n)$.

The rest of the thesis is organized as follows. First we define the defensive competition problem formally, and from it the one-to-one defensive competition problem. To solve the one-to-one defensive competition problem, we perform a problem transformation and define the k^+-depth contour in the following two chapters. From that, we propose an $O(n^2)$ time and $O(n^2)$ space algorithm that solves the one-to-one defensive competition problem. Last we show that many important computational geometry problems are closely related and generalize the one-to-one defensive problem to the k-to-one defensive competition problem and one-to-k defensive competition problem. We show that the one-to-k variation is NP-hard and propose some open problems.

Chapter 2

Defensive Competition Problem

In this chapter we first introduce the *competitive location problem*, then from it introduce the formal definition of the *defensive competition problem*.

2.1 Competitive Location Problems

The competitive location problem has long been an interesting part of competitive location theory[2] in the plane. The general setting of the competitive location theory is: given a set of demand points in the plane and a set of existing facilities, we want to locate a new set of facilities so that they will *compete* favorably against existing facilities under some distance criteria. We give the formal definition of the problem:

Definition 1: Given n demand points $P = \{p_1, p_2, \ldots, p_n\}$ in $\Re^2$ and a set of j facilities $X = \{x_1, x_2, \ldots, x_j\}$, find a new set of k facilities $Y = \{y_1, y_2, \ldots, y_k\}$ that maximizes $\|\{p_i | d(p_i, Y) \leqq d(p_i, X)\}\|$, where $d(A, B) = min\{d(a_i, b_i), a_i \in A, b_i \in B\}$ with $d(a_i, b_i)$ being a distance criterion.

The *one-on-one competition problem* which is the special case where $j = k = 1$ has been studied extensively by Drezner[3] and is defined as follows: given a set of demand points $P = \{p_1, p_2, \ldots, p_n\}$ in $\Re^2$ and an existing facility x, locate a new facility y that maximizes $\|\{p_i | d(p_i, y) \leqq d(p_i, x)\}\|$. In other words, find a new facility y that attracts the most demand points from facility x. Drezner gave an $O(n \log n)$ algorithm, and was later proved by Lee and Wu[5] to be optimal under the algebraic computational tree model of Ben-Or[4].

2.2 One-to-one Defensive Competition Problem

The competitive location problem can be considered to be trying to place attacking facilities to take away customers from existing facilities. From the defender's point of view we propose the defensive competition problem. In simple words, we try to place facilities so as to defend the demand points from attacking facilities.

We also add a radius constraint to the problem. The radius constraint restricts the attacking facilities from coming too close to the defending facilities. This constraint is motivated from zone properties that facilities possess in real life.

Now we give the definition of the *defend* relation.

Definition 2: Facility x is said to *defend* point $p_i \in P$ from facility y if $d(p_i, x) < d(p_i, y)$, where $d(a, b)$ is the Euclidean distance between a and b.

With this definition, we can now define the *defensive competition problem*. The general setting of the *defensive competition problem* is: given a set of demand points $P = \{p_1, p_2, \ldots, p_n\}$ in $\Re^2$ and a radius r, find locations to place a set of facilities $X = \{x_1, x_2, \ldots, x_j\}$ that minimizes $\max \|\{p_i \in P | d(p_i, Y) \leq d(p_i, X)\}\|$ from any set of attacking facilities $Y = \{y_1, y_2, \ldots, y_k\}$ such that $d(y_i, X) \geq r$ for all i. In other words, to find a set of defending facilities X that defends the most demand points from any set of attacking facilities Y under the constraint that attacking facilites cannot come within distance r of the defending facilities. Note that the problem is the competitive facility problem from the defender's point of view. We shall first focus on solving a special case where $j = k = 1$, specifically the *one-to-one defensive competition problem*. We can see that the case when $r = 0$ is exactly the famous Tukey median problem.

The formal definition of the one-to-one defensive competition problem are as follows: Given a set of n demand points $P = \{p_1, p_2, \ldots, p_n\}$ in $\Re^2$ and radius r, find the location to place facility x, satisfying $\min_{x \in \Re^2} \max_{d(x,y) \geq r} \|\{p_i \in P | d(y, p_i) \leq d(x, p_i)\}\|$.

Chapter 3

Problem Transformation

With some observations we can transform the one-to-one defensive competition problem to the r^+location depth problem which we shall define in this chapter. For convenience, we denote $C(x,r)$ to be the circle with the center at x and the radius r and denote Y_x^* to be the set of locations to place y that maximizes $\|\{p_i \in P | d(y,p_i) \leq d(x,p_i)\}\|$.

Lemma 1: $Y_x^* \bigcap C(x,r) \neq \emptyset$.

Proof: For any attacking facility y placed outside $C(x,r)$, moving y towards x along $\overrightarrow{yx}$ will attract more demand points. Therefore an optimal attacking facility can always be found on $C(x,r)$. See figure 3.1. □

Thus, there exists an optimal attacking facility on $C(x,r)$.

Lemma 2: $\min\limits_{x \in \Re^2} \max\limits_{d(x,y) \geq r} \|\{p_i \in P | d(y,p_i) \leq d(x,p_i)\}\| = \min\limits_{\mathcal{H}^* \in H} \|\{i | p_i \in H^*\}\|$, where $\mathcal{H}$ is the set of all closed half-spaces containing $C(x, r/2)$.

Proof: This is apparent from lemma 1, since the optimal attacking facility can be found on $C(x,r)$, therefore the closed half-space that y will attract would be determined by a tangent line on $C(x,r/2)$ and since y is optimal, it will attract the maximum demand points inside a closed half-space containing y and determined by a tangent line on $C(x,r/2)$. Hence the other side of the half-space (the half-space that contains $C(x,r/2)$) determines the number of demand points that facility x can defend. □

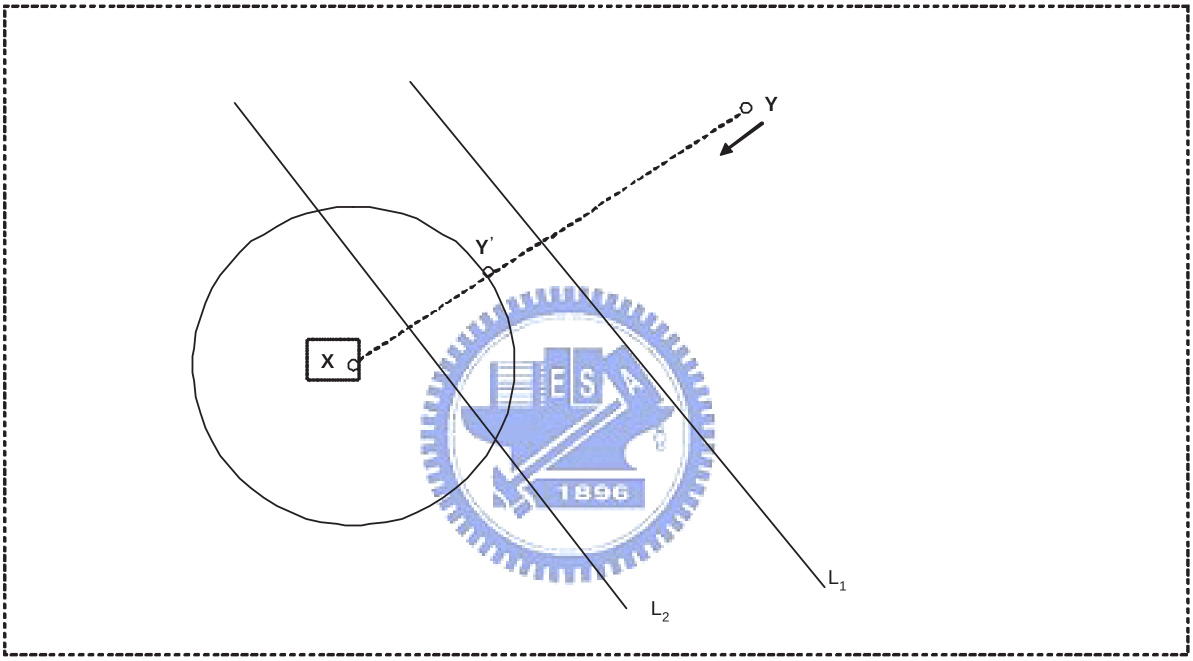


Figure 3.1: When facility Y is outside of $C(X, r)$, then the demand points that Y will attract are the demand points on the same side of Y in the half-space determined by L_1. If Y moves along the dotted line towards X until Y' which is on $C(X, r)$, then the demand points that Y' will attract are the demand points on the same side of Y in the half-space determined by L_2. Since L_1 is parallel to L_2, clearly Y' attracts at least as many demand points as Y.

In other words, the defending facility attracts the minimal number of demand points inside a half-space containing $C(x, r/2)$.

Note that the above observation is quite similar to the concept of location depth suggested by Tukey[6]. We shall give the definiton of the location depth used by Miller et al.[12]:

Definition 3: Let $P = \{p_1, \ldots, p_n\}$ be a finite set of points in $\Re^2$ and a be an arbitrary point, not necessarily in P. The *location depth* of a relative to P is the minimum number of points of P lying in any closed half-space determined by a line through a.

Using these observations, we can redefine the problem to be the r^+ location depth problem:

Given n demand points in $\Re^2$ and a radius r, find location(s) to place defending facility x that maximize $\min_{H^* \in \mathcal{H}} \|\{i|p_i \in H^*\}\|$, where $\mathcal{H}$ is the set of all closed half-spaces containing $C(x, r/2)$. With this definition we can see that when $r = 0$, $C(x, r/2)$ contracts to a point x, thus $\min_{H^* \in \mathcal{H}} \|\{i|p_i \in H^*\}\|$, where $\mathcal{H}$ is the set of all closed half-spaces containing $C(x, r/2)$ equals the location depth of x. Therefore when $r = 0$ the r^+ location depth problem is equivalent to finding the Tukey median.

Chapter 4

K-th Depth Contour and K^+-th Depth Contour

To understand our proposed algorthm, a background knowledge on the k-th depth contour would be very helpful. We shall give a brief description of the k-th depth contour in the first section and use it to define the new k^+-th depth contour. We will also prove in this chapter that the one-to-one defensive competition problem is equivalent to finding the deepest k^+-th depth contour.

4.1 K-th Depth Contour

For a fixed integer k, the set of points in the plane with location depth $\geq k$ is a convex polygonal region, whose boundary is the *k-th depth contour* (referred to as the *k-th hull* in the computational geometry literature).[12]

The depth contour concept has also been suggested by Tukey[6]. It was proposed to be used as a graphical display of data. Intuitively speaking, the k-th contour is the intersection of all halfplanes that contain $n - k - 1$ demand points. See figure4.1. Note that every depth contour is a convex polygon, which is apparent because the intersection of halfplanes are always a convex polygon. Suppose that the deepest contour has depth k^*, then by Helly's theorem we can prove that $k^* \leq \lfloor n/3 \rfloor$. Thus the existence of a centerpoint in a plane is also proved.

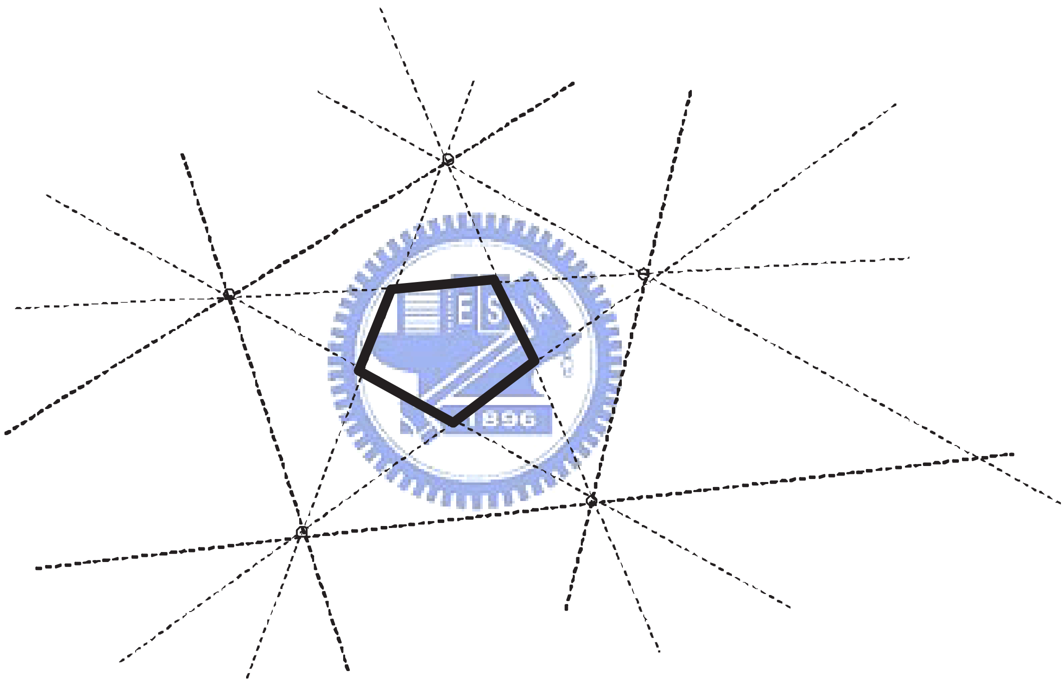


Figure 4.1: The 2nd depth contour is shown in bold lines.

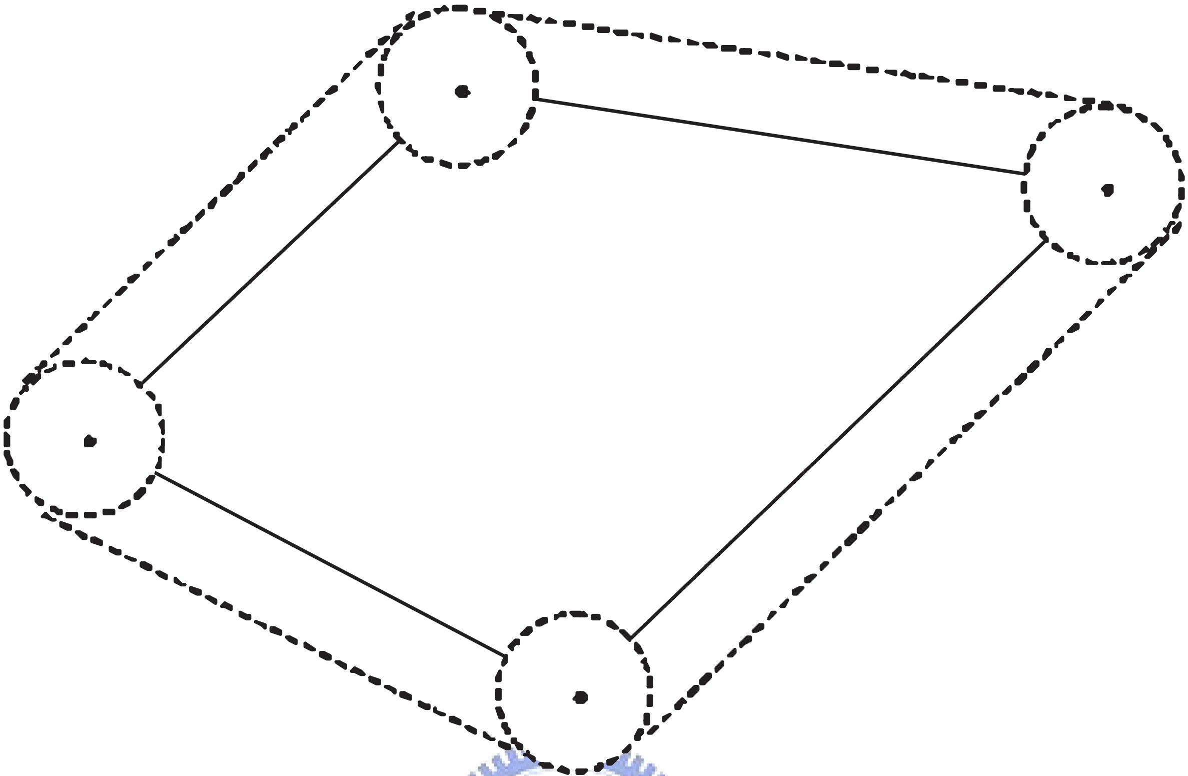


Figure 4.2: Tracing a disk of radius $r/2$ around the k-th depth contour, forming the k^+-th depth contour.

4.2 K^+-th Depth Contour

The k^+-th depth contour is defined as the boundary traced out by moving a disk of radius $r/2$ whose center is on the boundary of the k-th depth contour, with r being the radius given in the one-to-one defensive competition problem. See figure 4.2.

The above description of the k^+-th depth contour is somewhat vague, and can be more precisely described by using the notion of the *Minkowski sum* of two point sets[14].

Let A and B be two sets of points in the plane. If we establish a coordinate system, then the points can be viewed as vectors in that coordinate system. Define the *sum* of A and B in the most natural manner possible: $A \bigoplus B = \{x + y | x \in A, y \in B\}$, where $x + y$ is the vector sum of the two points. This is known as the *Minkowski sum* of A and B. With the knowledge of Minkowski sum, we can define the k^+-th depth contour as follows:

Definition 4: The k^+-th depth contour is defined as the Minkowski sum of the k-th contour and a disk or radius $r/2$, with r being the radius given in the one-to-one defensive comeptition problem.

We now prove an important lemma in this thesis to show that the one-to-one defensive competition problem is actually the same as finding the deepest k^+-th depth contour.

Lemma 3: Facility x can defend at least k demand points if and only if facility x is placed inside the k^+-th depth contour.

Proof: For the if part, placing x inside the k^+-th depth contour guarantees that $C(x, r/2)$ would intersect with the k-th depth contour. Therefore every tangent line would either pass through the k-th depth contour which would guarantee that at least k demand points are defended or the halfplane determined by a tangent line that does not pass through the k-th depth contour would be the side that does not include the k-th depth contour thus still promising that X will be able to defend at least k demand points.

As for the only if part, suppose we place x outside of the k^+-th depth contour. By definition, $C(x, r/2)$ would lie outside of the k-th depth contour. We choose a tangent line that parallels one of the sides of the k-th depth contour which does not pass through the contour. This half-space would attract at least $n - k$ demand points by definition, hence X can defend at most k demand points.

□

Chapter 5

The Proposed Algorithm

From Lemma 3 we proved that in order to solve the one-to-one defensive competition problem, we need to find the deepest k^+-th contour. In this chapter we present an $O(n^2)$ time and $O(n^2)$ space complexity algorithm for this problem.

The algorithm involves a couple of steps.

1. A dual mapping(of a point to a line) on the demand points is made to form an arrangement of lines.

2. Using the topological sweep, we can efficiently find all intersection of lines within the arrangement. Each intersection in the dual corresponds to a line between two points in the primal. As we find each intersection we can determine the depth contour to which the corresponding halfplane potentially belongs.

3. Peform the Minkowski sum on these halfplanes with a disk of radius $r/2$.

4. Combine the upper and lower hulls to form the k^+-th contour for every k.

The basic approach was first suggested by Cole, Sharir, and Yap[15]. We shall describe each step in detail next.

5.1 Dual Mapping

For the first step, a standard dual mapping[16] $p(a, b) \rightarrow l : y = -ax + b$ with a and b being the x and y coordinates of a demand point respectively is used. This dual mapping yields many properties that we desire. The mapping preserves the ordering of demand points along the x-axis as the slope of each mapped lines. A line through two points in the primal maps to the intersection point of the corresponding lines in the arrangement. Most importantly, the mapping preserves the above/below relationship: if a point is above a line formed by two points in the primal, then the corresponding line is above the point in the dual.

Since each intersection inside the dual mapping represents a line formed by two demand points in the original arrangement and that the above/below relation of lines and points are preserved, the number of lines above/below an intersection in the dual mapping represents the number of demand points above/below the corresponding line that maps to the intersection. Thus the depth contour that the line formed by two demand points is determined by the number of demand points either above or below the line, whichever is smaller. Also note that the line is determined to be in the lower/upper hull of the respective depth contour in the process.

For every intersection point in the arrangement, we determine the contour to which the corresponding halfplane in the primal contributes. Let L be a line through two points in the primal corresponding to the intersecting point I in the dual. The vertical line V through I in the dual intersects every line of the arrangement, some above I and some below I. The number of lines intersected above/below I exactly equals the number of points lying above/below L respectively in the primal.

Each depth contour is the interior intersection of the halfplanes that potentially belong to this contour. During the sweep, the halfplanes for each contour are split into upper and lower sets. Let m be the number of crossings above I. Let r be the number of crossings below I. The intersection would belong in the upper set of the $m + 1$ contour and the lower set of the $r + 1$ contour. See figure5.1.

The lines containing the edges of the lower convex hull of the upper set of dual intersections correspond to the vertices of the upper boundary of the contour in the primal. The lines containing the edges of the upper convex hull of the lower set of dual intersection correspond to the vertices of the lower boundary of the contour. The intersection of the lower hull of the upper set and the upper hull of the lower set is the complete contour.

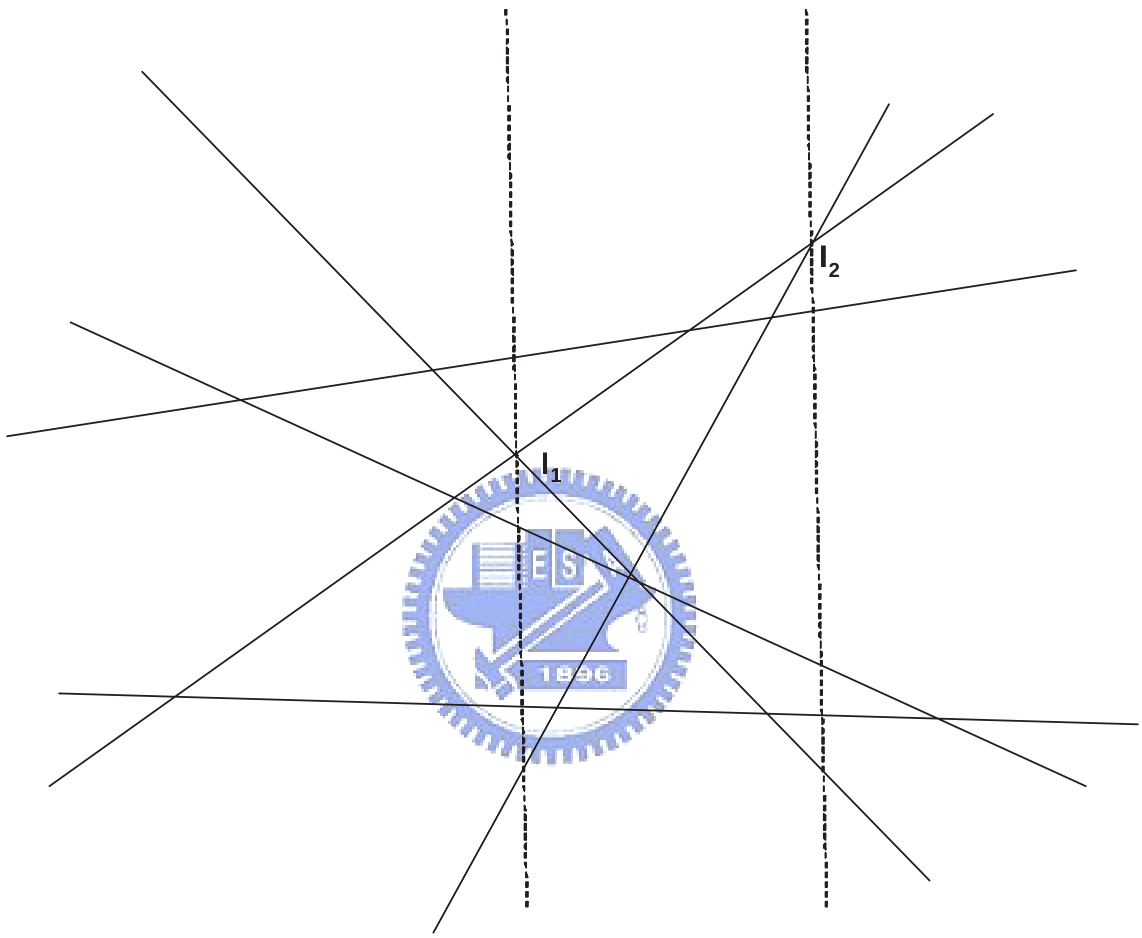


Figure 5.1: We can see there are one line above and three lines below intersection I_1, hence it belongs to the upper hull of the second contour and the lower hull of the fourth contour. Similarly, there are no lines above and four lines below intersection I_2, hence it belongs to the upper hull of the first contour and the lower hull of the fifth contour.

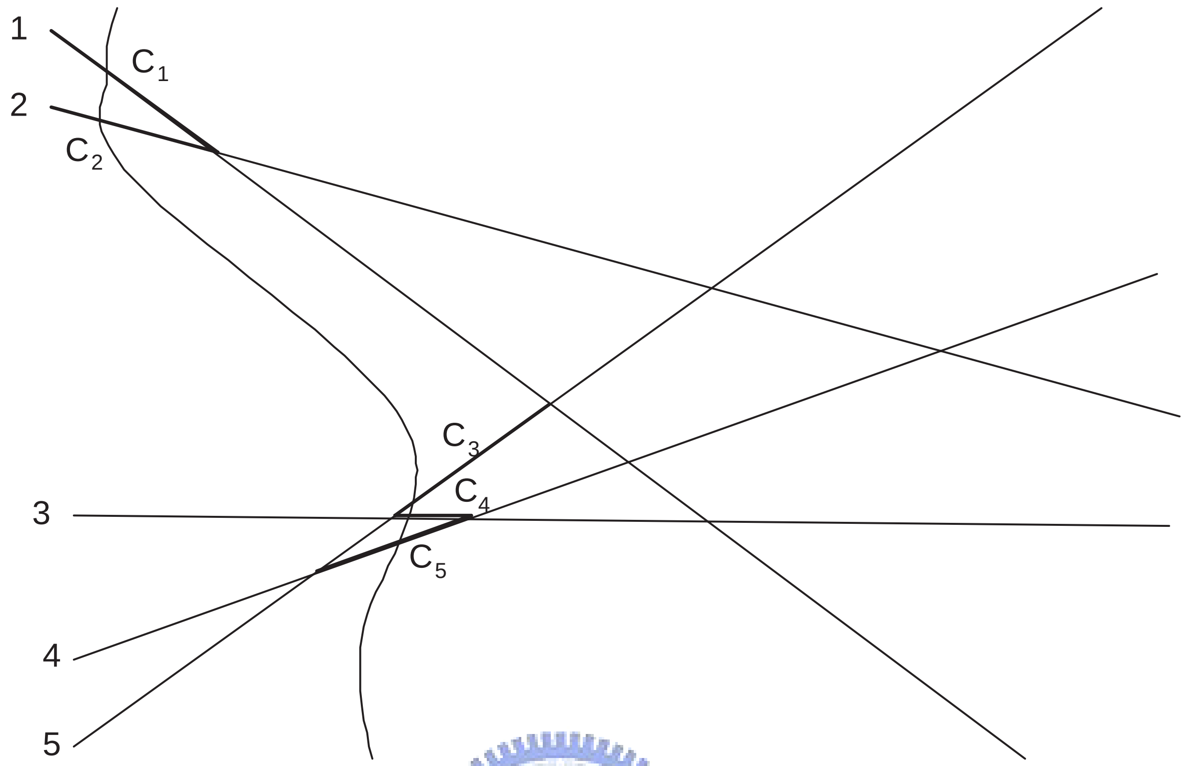


Figure 5.2: A topological line and the cut are emphasized in bold lines. The (vertical) cut in simple words denote the order of edges that the topological line intersects in the vertical direction. The formal definition of cut can be found in [17].

5.2 Topological Sweep

To obtain the needed information, we perform the topological sweep. For clarity, we give a brief introduction on the topological sweep. A detailed analysis regarding the topological sweep can be found in reference [17].

Topological sweep allows the traversal of all vertices of an arrangement in $O(n^2)$ time and linear space. The typical vertical sweepline method maintains a priority queue (heap) of potential next points in the vertical sweep, at a cost of $O(\log n)$ per update and a total time complexity of $O(n^2 \log n)$ and linear space. Explicit construction of the arrangement as a planar graph uses $O(n^2)$ time and space.

Topological sweep improves the time complexity by sweeping not with a straight line, but with a topological line and a *cut*. See figure 5.2.

Using these two concepts, the topological sweep of the arrangement will be imple-

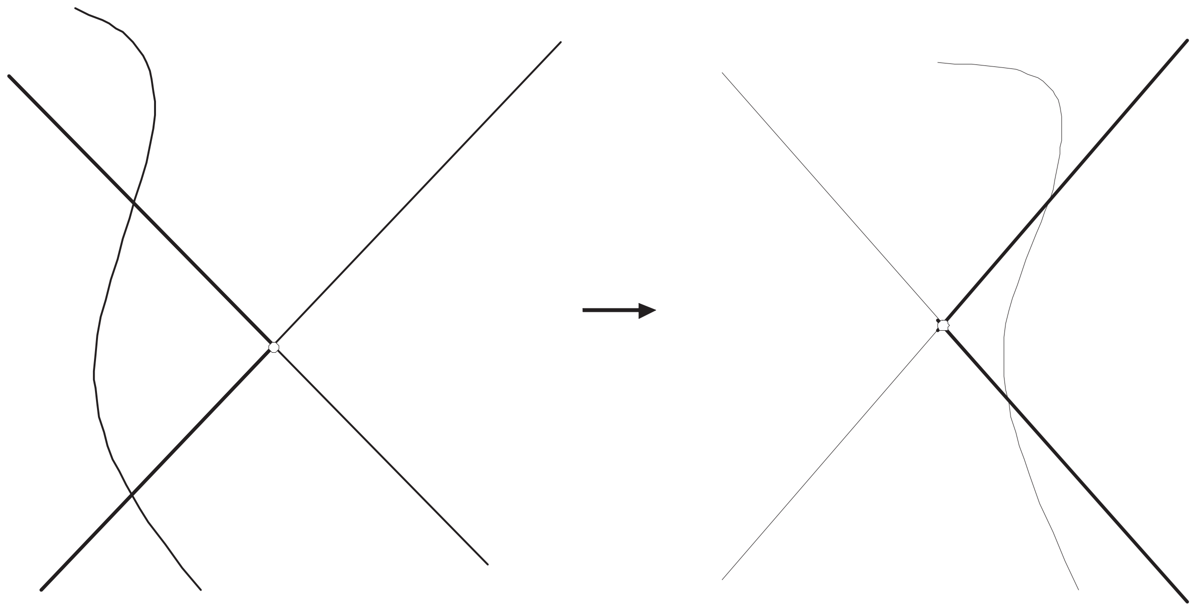


Figure 5.3: An elementary step.

mented by starting with the leftmost cut and pushing it to the right till it becomes the rightmost cut, in a series of elementary steps. See figure5.3.

An elementary step is performed when the topological line sweeps past a vertex of the arrangement; it corresponds to a transposition in the underlying numbering of the lines as defined by the order in which they are intersected by the sweeping topological line. $O(n^2)$ elementary steps will be required to sweep the arrangement.

The major difficulty in implementing the topological sweep is how to discover where in a cut an elementary step can be applied. An auxiliary notion of horizon tree is introduced. Let $(m_1, m_2, \ldots, m_n)$ denote the lines containing the edges $(c_1, c_2, \ldots, c_n)$ respectively.

The *upper horizon tree* $T^+(C)$ of cut C is constructed by starting with the edges of the cut and extending them to the right. When two edges come together at an intersection point, only the one of higher slope continues on to the right; the other one stops at that point and is removed from further consideration. More formally, the upper horizon tree consists of one segment from each of the lines m_i, where a point p of m_i belongs to $T^+(C)$ if

- p is above all lines m_j with $j > i$, and
- p is below all lines m_k satisfying both $k < i$ and having slope greater than the slope

of m_i.

Figure5.4 shows $T^+(C)$ for the cut of figure5.2, as well as the symmetrically defined *lower horizon tree* $T^-(C)$ (where lines of lower slope are the winners).

The horizon tree stores one line segment per level of the arragement and uses a stack that contains all points of intersection of current line segments on adjacent levels, represented as array indices. The algorithm sweeps a curved line across the arrangement, over the intersection points of currently incident line segments. Each line segment has at most two neighbors, so the stack remains linear in size. However, all $O(n^2)$ intersection points of the arrangement are saved to generate a halfplane for the computation of contours, producing quadratic space complexity.

5.3 Minkowski Sum

To find the k^+-th depth contour, we perform the Minkowski sum on the upper and lower hulls of the depth contour with a disk of radius $r/2$. Therefore, we are finding the Minkowski sum for a convex polygon and a disk. This can be done in $O(n)$. Here we present a simple approach that can compute the Minkowski sum in $O(n)$ time.

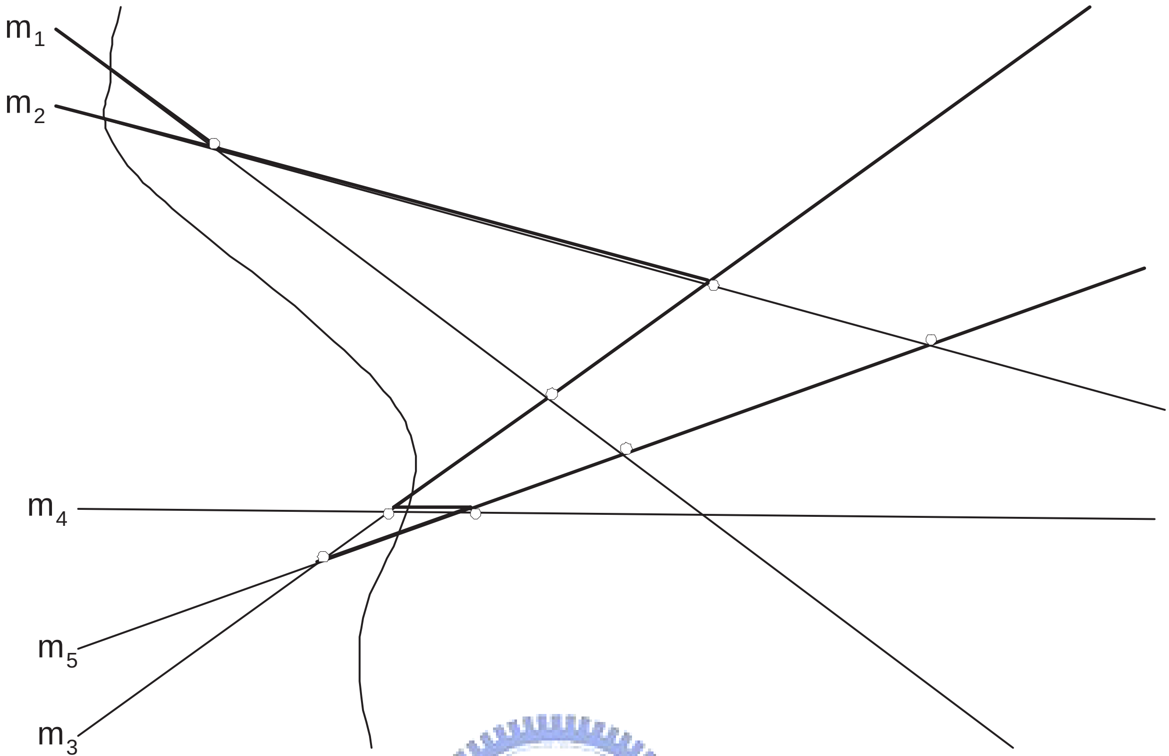
First, we add width $r/2$ to the boundaries of the depth contour. This simple procedure clearly can be done in $O(n)$ time. Next we find the incircles of radius $r/2$ for each adjacent boundaries. This also can be achieved in $O(n)$ time, since the calculation for each incircle takes $O(1)$ time and there are $O(n)$ adjacent boundaries. These incircles round the edges on the boundaries to form the Minkowski sum.

The time complexity of each step is discussed as follows:

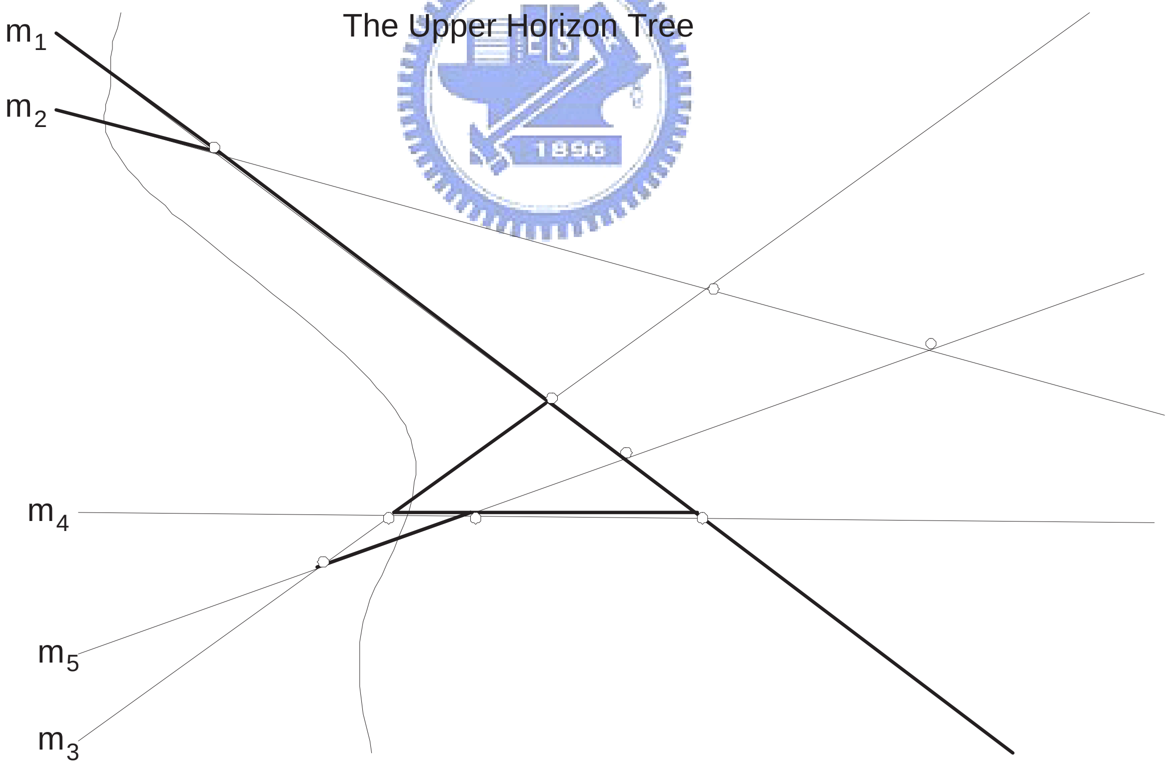
1. The dual mapping of each demand point takes $O(1)$ time, and since there are n demand points, the total time to perform dual mapping is $O(n)$.

2. A detailed analysis regarding the topological sweep can be found in reference [17], and has been briefly explained in the previous paragraphs. The time complexity for this step is $O(n^2)$.

3. Peforming the Minkowski sum is done in $O(n)$ time.



The Upper Horizon Tree



The Lower Horizon Tree

Figure 5.4: The horizon trees of a cut.

4. The upper and lower hulls of a set of points can be computed in linear time, under the assumption that the points are in sorted order. Since we can obtain a sorted order of the lower and upper hull after the topological sweep, the combining of lower and upper hulls can be done in $O(n)$ time.

We conclude with the following theorem:

Theorem 4: The one-to-one defensive compeition problem can be solved in $O(n^2)$ time and space.

Chapter 6

Problem Relationships

We discuss some interesting relationships between the defensive competition problem and various other problems in this chapter. To our surprise, the defensive competition problem yields as a middle ground for two important problems in the computational geometry. As before, we begin with the one-to-one defensive competition problem and generalize it afterwards.

6.1 One-to-one Defensive Competition Problem

The one-to-one defensive competition problem is actually a generalization of the famous Tukey median problem [6]. The Tukey median is defined to be the region which has the largest location depth and has been studied extensively over the years. We show that the Tukey median problem is the special case of the one-to-one defensive competition problem with $r = 0$.

As described in previous chapters, the one-to-one defensive competition problem can be defined to be trying to maximize the minimal number of demand points inside a closed half-space containing $C(x, r/2)$. If $r = 0$, we would actually be maximizing the minimal number of demand points inside a closed half-space containinig x. Therefore, the most demand points that x can defend is the minimum demand points that lie in any half-space containing x. This is exactly the location depth of x. So, when $r = 0$, finding x is actually trying to locate the region that has the greatest location depth which is also what the Tukey median problem attempts to solve.

When $r = 0$, the one-to-one defensive competition problem is equivalent to the Tukey median problem. When r is sufficiently large, the problem reduces to the *smallest enclosing circle problem*. Specifically speaking, when $r/2$ is greater than the radius of the minimum enclosing circle, then the one-to-one defensive competition problem is equivalent to finding the smallest enclosing circle.

We show that placing x on the center of the smallest enclosing circle when $r/2$ is greater than the radius of the minimum circle is an optimal solution for the one-to-one defensive competition problem. Since x is on the center of the smallest enclosing circle and $C(x, r/2)$ has a larger radius, thus every demand point lies inside $C(x, r/2)$. Therefore x will be able defend every demand point from any opposing facility y, and is optimal.

6.2 one-to-k Defensive Competition Problem

Generaliztion of the one-to-one defensive competition problem to one-to-k, is similarly defined as follows:

Definition 5: Given n demand points $P = p_1, p_2, \ldots, p_n$ in $\Re^2$, a constant k and a radius r, the one-to-k defensive competition problem is to place k facilities $X = \{x_1, x_2, \ldots, x_k\}$ satisfying $\min\limits_{X \in \Re^2} \max\limits_{d(X,y) \geq r} \|\{p_i \in P | d(y, p_i) \leq d(X, p_i)\}\|$.

The special case when $r = 0$ is exactly the (k, y)-medianoid problem[21]: given n demand points $P = \{p_1, p_2, \ldots, p_n\}$ find locations to place k sites(facilities)$X = \{x_1, x_2, \ldots, x_k\}$ so as to $\min\limits_{X \in \Re^2} \max\limits_{d(X,y) \geq r} \|\{p_i \in P | d(y, p_i) \leq d(X, p_i)\}\|$.

This problem has been proven to be NP-hard, hence the one-to-k defensive competition problem is also NP-hard.

6.3 k-to-one Defensive Competition Problem

We now generalize the one-to-one defensive competition problem to k-to-one, defined as follows:

Definition 6: Given n demand points $P = \{p_1, p_2, \ldots, p_n\}$ in $\Re^2$, a constant k and a radius r, the k-to-one defensive competition problem is to find a place a facility x satisfying $\min\limits_{x \in \Re^2} \max\limits_{d(x,Y) \geq r} \|\{p_i \in P | d(Y, p_i) \leq d(x, p_i)\}\|$.

This is still an open problem currently.

6.4 Conclusion

In this thesis, we introduce a defensive competition problem which is a generalization of a well-known competitive location problem. An $O(n^2)$ time and space solution has been proposed for the special case of one-to-one defensive competition problem. Furthermore, it is shown that the one-to-k defensive competition problem is NP-hard.

The k-to-one defensive competition problem however, still remains an open problem and deserves further research.

Bibliography

[1] H. Hotelling: Stability in competition, *Economical Journal*39, 1929, 41-57

[2] S. L. Hakimi: On locating new facilities in a competitive enviroment, *ISOLDE II*, June 1981

[3] Z.Drezner: Competitive location strategies for two facilities, *Regional Science and Urban Economics*12, 1982, 485-493

[4] M.Ben-Or: Lower Bounds for algebraic computation trees, *Proc. 15th Annual Symp on Theory of Computation*, 1983, 80-86

[5] D. T. Lee and Y. F. Wu: Geometric complexity of some location problems, *Algorithmica*1, 1985, 193-211

[6] J. Tukey: Mathematics and the picturing of data, *Proceedings of the International Congress of Mathematicians(Vancouver)*, 1975, 523-531

[7] P. Rousseeuw and I. Ruts: Bivariate location depth, *Applied Statistics*45, 1996, 516-526

[8] P. Rousseeuw and I. Ruts: Constructing the bivariate Tukey median, *Statistica Sinica*8, 1998, 828-839

[9] J. Matoušek: Computing the center of planar point sets, *Computational Geometry: Papers from the DIMACS special year*6, American Mathematical Society, 1991, 221-230

[10] S. Langerman and W. Steiger: The complexity of hyperplane depth in the plane, *Japan Conference on Discrete and Computational Geometry*, November 2000

[11] S. Langerman: Algorithms and data structures in computational geometry, *Ph.D. thesis, Department of Computer Science, Rutgers University, New Brunswick*, 2001

[12] K. Miller, S. Ramaswami, P. Rousseeuw, T. Sellarés, D. Souvaine, I. Streinu, and A. Struyf: Fast implementation of depth contours using topological sweep, *Proc. 12th Symposium on Discrete Algorithms(SODA)(Washington D.C.)*, 2001

[13] T. Chan: An optimal randomized algorithm for maximum Tukey depth, *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, 2004, 430-436

[14] J. O'Rourke: Computational Geometry in C, *Cambridge University Press, New York*, 1994

[15] R. Cole, M. Sharir, and C. K. Yap: On k-hulls and related problems, *SIAM Journal on Computing*, 15(1), 1987, 61-77

[16] H. Edelsbrunner: Algorithms in combinatorial geometry, *Springer-Verlag, Berlin*, 1987

[17] H. Edelsbrunner and L. G. Guibas: Topologically sweeping an arrangement, *Journal of Computer and System Science*, 38, 1989, 165-194

[18] D. T. Lee, M. Sarrafzadeh and Y. F. Wu: Minimum cut for circular-arc graphs, *SIAM Journal on Computing*, 19(6), 1990, 1041-1050

[19] D. T. Lee and K. H. Tsai: k-best circular-arc graphs *Algorithmica*, 18(2), 1997, 198-216

[20] D. S. Hochbaum and W. Maass: Approximation schemes for covering and packing problems in image processing and VLSI *J. ACM*, 32, 130-146

[21] S. L. Hakimi: Locations with spatial interactions: Competitive locations and games, *Discrete Location Theory, Wiley , New York*, 1990, 439-478