J Comb Optim (2006) 11: 203–218
DOI 10.1007/s10878-006-7132-y

A PTAS for a disc covering problem using width-bounded separators*

Zhixiang Chen · Bin Fu · Yong Tang · Binhai Zhu

Received: xx xx xxx / Revised: xx xx xxx / Accepted: xx xx xxxx

Abstract In this paper, we study the following disc covering problem: Given a set of discs of various radii on the plane, find a subset of discs to maximize the area covered by exactly one disc. This problem originates from the application in digital halftoning, with the best known approximation factor being 5.83 (Asano et al., 2004). We show that if the maximum radius is no more than a constant times the minimum radius, then there exists a polynomial time approximation scheme. Our techniques are based on the width-bounded geometric separator recently developed in Fu and Wang (2004), Fu (2006).

1. Introduction

In real life we are always dealing with the problem of mixed technology; for instance maintaining COBOL and JAVA compilers at the same time. It is also not uncommon that sometimes we have to print some colored fancy images onto a black/white tone printer. Digital-halftoning is exactly such a technology, it converts a continuous, possibly colored

*This research is supported by Louisiana Board of Regents fund under contract number LEQSF(2004–07)-RD-A-35.

Z. Chen (✉)
Department of Computer Science, University of Texas - Pan American, Edinburg, TX 78539-2999
e-mail: chen@cs.panam.edu

B. Fu
Department of Computer Science, University of New Orleans, New Orleans, LA 70148; Research Institute for Children, 200 Henry Clay Avenue, New Orleans, LA 70118
e-mail: fu@cs.uno.edu

Y. Tang
Dept. of Computer Science, Sun Yat-Sen University, Guangzhou 510275, China
e-mail: issty@zsu.edu.cn

B. Zhu
Dept. of Computer Science, Montana State University, MT 59717-3880, USA
e-mail: bhz@cs.montana.edu

image into a binary image (Ostromoukhov, 1993; Ostromoukhov and Hersch, 1999). In the cluster-dot halftoning, dots form clusters whose sizes are determined by their corresponding intensity level. Given a continuous-tone image, one computes spatial frequency distribution by Laplacian. Each grid point is then assigned a disc of radius reflecting the Laplacian value at the corresponding position. This results in a set of discs of different radii. The problem is then to find a subset of discs to maximize the area that belongs to exactly one disc.

We study the approximation algorithm for the above disc covering problem with applications in digital halftoning (Asano et al., 2000; Ostromoukhov, 1993; Ostromoukhov and Hersch, 1999; Sasahara and Asano, 2003; Asano et al., 2004). Given a set of discs of various radii, find a subset of discs from them to maximize the area covered by exactly one disc. This seems computationally hard although there is not yet a proof about NP-hardness. We show that if the maximum radius is no more than a constant times the minimum radius, there exists a polynomial time approximation scheme. If the centers of the discs are at the grid points and the radii are between two positive constants, there exists a constant factor approximation which runs in almost linear time.

In Asano et al. (2004), a polynomial time approximation algorithm was designed with approximation ratio 5.83. In their algorithm, no condition is specified that the maximum radius is no more than a constant times the minimum radius. However, the empirical data used in Asano et al. (2004) shows that not only such a constant stands, it is also always relatively small (i.e., 3–5). We believe that this assumption is practically reasonable since each disc reflects the intensity level of a local point.

Geometric separator has applications in many problems. It plays important role when we develop divide and conquer algorithm for geometric problems. Lipton and Tarjan (1979) presented the well known geometric separator for planar graphs. They proved that every n-vertex planar graph has at most $\sqrt{8n}$ vertices whose removal separates the graph into two disconnected parts of size at most $\frac{2}{3}n$. Their $\frac{2}{3}$-separator was improved to $\sqrt{6n}$ by Djidjev (1982), $\sqrt{5n}$ by Gazit (1986), and $\sqrt{4.5n}$ by Alon et al. (1990). Spielman and Teng (1996) showed a $\frac{3}{4}$-separator with size $1.82\sqrt{n}$ for planar graph.

Some other forms of the separators were studied in Miller et al. (1991), Smith and Wormald (1998). They let each input point be covered by a regular geometric object such as circle, rectangle, etc. If every point on the plane is covered by at most k objects, it is called k-thick. Some separators of size $c \cdot \sqrt{k \cdot n}$ were proved in Miller et al. (1991), Smith and Wormald (1998), where c is a constant. Fu and Wang (2004) developed a method for deriving sharper upper bound separator for grid points via controlling the distance to the separator line. They proved that for a set of n grid points on the plane, there is a separator that has $\leq 1.129\sqrt{n}$ points and each side has $\leq \frac{2}{3}n$ points. Fu (2006) introduced the concept of width-bounded geometric separator and applied it to a class of NP-complete geometric problems to improve their computational time from $n^{O(\sqrt{n})}$ to $2^{O(\sqrt{n})}$. In this paper we use the width-bounded geometric separator to develop a polynomial time approximation scheme for the halftoning problem.

Section 2 explains a simple width-bounded geometric separator that is used in our approximation algorithm. Section 3 describes the approximation algorithm based on the width-bounded separator. Section 4 gives a randomized almost linear time algorithm for finding the separator used in Section 3. The description of the randomized algorithm is almost self-contained except the well known fact Lemma 12 for the existence of the center point. A linear time algorithm for finding the width-bounded geometric separator is described in Section 5, which depends on some non-trivial results from Fu (2006), Jadhar (1993).

2. Separators on the plane

Definition 1. For two points p_1, p_2 in the plane R^2, $\text{dist}(p_1, p_2)$ is the Euclidean distance between p_1 and p_2. For a set $A \subseteq R^2$, $\text{dist}(p_1, A) = \min_{q \in A} \text{dist}(p_1, q)$. Let P be a set of points on the plane, and $w > 0$ be a constant. A w-*wide-separator* is determined by a line L, called the center line of the separator, on the plane. It has two measurements for its quality of separation: (1) $\text{balance}(L, P) = \frac{\max(|P_1|, |P_2|)}{|P|}$, where P_1 and P_2 are the two subsets of P on the two sides of L; and (2) $\text{measure}(L, P, \frac{w}{2})$, which is the number of elements of P with distance $\leq \frac{w}{2}$ to L. The w-width separator area is all points with distance $\leq \frac{w}{2}$ to L. For constants $0 < b_0 < 1$, $z_0 \geq 0$, $w \geq 0$, and a set of n grid points P on the plane, a (b_0, z_0)-w-width-separator (for P) is a w-width separator L with $\text{balance}(L, P) \leq b_0$ and $\text{measure}(L, P, \frac{w}{2}) \leq \frac{z_0 w}{2}\sqrt{n}$.

From the definition of width-bounded separator, its quality is measured by two numbers. One measures the balance of the separation. A well balanced separator can reduce the problem size efficiently during the application to divide and conquer algorithm. This brings that the algorithm runs in a polynomial time. The other number measures the number of points inside the separator area. The small number of points in the separator area ($O(\sqrt{n})$) is used to control the accuracy of our approximation algorithm.

Theorem 2. *Fu and Wang (2004), Fu (2006) Let constant $w > 0$ be a constant and $\delta > 0$ be a small constant. Let P be a set of n grid points. Then there is an $O(n^3)$ time algorithm that finds a separator line L such that each side of L has $\leq \frac{2}{3}n$ points from P, and the number of points of P with distance $\leq w$ to L is $\leq (\frac{4}{\sqrt{\pi}} + \delta)w \cdot \sqrt{n}$ for all large n.*

3. The approximation scheme

Definition 3. For constant $c > 0$, the input is a set of discs $D_1, \cdots, D_n$ on the plane with $r(D_i) \leq c \cdot r(D_j)$ for all $1 \leq i, j \leq n$, where $r(D_i)$ is the radius of D_i. The H_c *problem* P is to find a subset $Q \subseteq P$ with the maximal area covered by exactly one disc in Q. Define $\text{opt}(P)$ to be the subset of discs of P in an optimal solution. The H'_c *problem* P is a special H_c problem such that the distance between every pair of disc centers in P is at least $c' \times r(D_i)$ for any D_i in the P, where $c' > 0$ is a fixed constant. This problem studied by Asano et al. (2004) requires that every center is a grid point. If the radii are between two positive constants then it is covered by our definition. For a grid point $p = (i, j)$ (i and j are integers) on the plane, define $\text{grid}(p) = \{(x, y) | i - \frac{1}{2} \leq x < i + \frac{1}{2}, j - \frac{1}{2} < y \leq j + \frac{1}{2}\}$, which is a half close and half open 1×1 square. The *net* $g(P)$ for a H_c problem P is a set of grid points such that (1) for each point $p \in g(P)$, $\text{grid}(p)$ contains the center for some disc in P; and (2) for each disc D of P, $\text{center}(D) \in \text{grid}(p)$ for some point p in $g(P)$, where $\text{center}(D)$ is the center point of disc D. For a set of discs Q on the plane, define $s(Q)$ to be the size of the area covered by exactly one disc in Q.

In the theorem below, the function $f_P(e)$ controls the number of disc centers in the area with e grid points. The purpose of the function f_P is to unify the algorithms for both H_c and H'_c problems. For an H_c problem, $f_P(O(1))$ is up to $|P|$, but for an H'_c problem, $f_P(O(1)) = O(1)$. Our approximation scheme depends on the algorithm to find the width-bounded separator for a set of grid points on the plane. Theorem 2 gives $O(n^3)$ time algorithm for finding the width-bounded separator. An $O(n(\log n)^4)$ time randomized algorithm for finding separator is presented at section 4. Our Theorem 4 shows how the time of our approximation algorithm depends on the time for the separator detection. This is why it assumes there exists an $O(n^a(\log n)^b)$ time algorithm for finding separator, where a, b are constants.

Theorem 4. *Let $0 < b_0 < 1$, $0 \leq z_0$, and $0 < \epsilon$ be constants. Let P be an H_c problem and f_P be an non-decreasing function from N to N such that $|Q| \leq f_P(|g(Q)|)$ for every $Q \subseteq P$. Assume that there exists an $O(n^a(\log n)^b)$ time algorithm for computing the (b_0, z_0)-$O(1)$-width-bounded separator for some constants $a \geq 1$ and $b \geq 0$. Then there exists an $O(f_P(\frac{E_1}{\epsilon^{\frac{1}{1-\alpha}}})^{\frac{E_2}{\epsilon^{\frac{1}{1-\alpha}}}} n^a(\log n)^{b+1})$ time approximation algorithm to output $Q \subseteq P$ with $s(Q) \geq (1-\epsilon)s(\text{opt}(P))$, where $\alpha = 0.6$, E_1 and E_2 are constants.*

Proof: We first give an overview about our method. Assume the minimum radius of the input discs is 1. The radius of every disc of P is $\leq c$. For a set of discs $P = \{D_1, \cdots, D_n\}$ on the plane, the net $g(P)$ shows that the optimal solution of P has $\Omega(|g(P)|)$. Apply a separator with width $\geq 2c$. The discs on the different sides of the separator do not intersect each other. The two sub-problems on the left and right sides of the separator can solved independently. Our separator can control there are only $O(\sqrt{|g(P)|})$ points from $g(P)$ to stay in the separator area. The discs on the separator area only affect the overall solution by $O(\sqrt{|g(P)|})$, which does not affect its total accuracy much. Our algorithm is based on such a divide and conquer approach by using width-bounded geometric separator.

Let $\epsilon > 0$ be a constant that determines the accuracy of our approximation algorithm. Let P be the H_c problem, which consists of a set of discs on the plane. Select some constants: $w_0 = c + \frac{\sqrt{2}}{2}$, $\delta = 0.01$, $b_1 = 1 - b_0$, $\delta_1 = \min(0.08, \frac{b_1}{4})$, $c_2 = \pi(\frac{\sqrt{2}}{2} + c)^2$ and $c_3 = \frac{1}{\pi(2\sqrt{2}+2c+\frac{\sqrt{2}}{2})^2}$, $\alpha = 0.6$, and e_1 is a constant that satisfies the inequalities:

$$\frac{z_0 w_0}{\sqrt{e_1}} \leq \delta_1, \tag{1}$$

$$\epsilon(c_3(b_1 - 2\delta_1)e_1) > ((b_1 - 2\delta_1)e_1)^{\alpha}, \text{ and} \tag{2}$$

$$c_2 z_0 w_0 \sqrt{e_1} \leq \delta_1 e_1^{\alpha}. \tag{3}$$

We can choose constant E_1 big enough and let $e_1 = \frac{E_1}{\epsilon^{\frac{1}{1-\alpha}}}$. Then e_1 satisfies the conditions (1)–(3).

Algorithm

Input: a set of discs $P = \{D_1, \cdots, D_n\}$ on the plane

Output: A subset $A(P) \subseteq P$ with $s(A(P)) \geq (1-\epsilon)s(\text{opt}(P))$.

- If $|g(P)| \leq e_1$, then find $A(P) = \text{opt}(P)$ using the brute-force method and return $A(P)$.
- Find a $2w_0$-width separator center line L for $g(P)$ such that balance$(L, g(P)) \leq b_0$ and measure$(L, g(P), w_0) \leq z_0 w_0 \sqrt{|g(P)|}$ (see Theorem 2).
- Let P_0 be all the discs D of P with dist(center$(D), L) \leq c$.
- Let P_1 be all the discs D of centers on the one side of the separator and dist(center$(D), L) > c$.
- Let P_2 be all the discs D of centers on the other side of the separator and dist(center$(D), L) > c$.

Solve P_1 to get the approximate solution $A(P_1)$.
Solve P_2 to get the approximate solution $A(P_2)$.
Merge the solutions for P_1 and P_2 to output $A(P) = A(P_1) \cup A(P_2)$.
End of Algorithm

Lemma 5. *Every $\delta \times \delta$-square has $\leq K$ disc centers from P in the optimal solution, where $K = 20$.*

Proof: Assume that opt(P) has more than K centers in a $\delta \times \delta$ square. Let $\eta = \frac{c-1}{K}$. All of the K radii are in the range $[1, c]$, which can be partitioned into the union of K intervals of format $[1 + (i-1)\eta, 1 + i\eta]$ for $i = 1, 2, \cdots, K$. At least two discs in opt(P) have radii in an interval $[1 + (i-1)\eta, 1 + i\eta]$ for some $i \in \{1, 2, \cdots, K\}$.

Let C_1 and C_2 be the two discs (in opt(P)) whose centers are in the same $\delta \times \delta$-square and radii are in the same interval $[1 + (i-1)\eta, 1 + i\eta]$. For a region R, let $v(R)$ be the area size of R. The two centers of discs C_1 and C_2 are close. So are their radii. It is easy to verify that $v(C_1 - C_2) \leq 0.2 \cdot v(C_1)$ and $v(C_2 - C_1) \leq 0.2 \cdot v(C_1)$. Let $R_0 \subseteq C_1$ be the maximal sub-region of C_1 such that every point in R_0 is covered by exactly one disc in opt$(P) - \{C_1, C_2\}$. We check the following two cases:

Case I $v(R_0) \geq 0.6 \cdot v(C_1)$. Since C_1 and C_2 are in opt(P), every point in $C_1 \cap C_2$ is covered by at least two discs in opt(P). We have that $s(\text{opt}(P) - \{C_1, C_2\}) \geq s(\text{opt}(P)) + v(R_0) - v(C_1 - C_2) - v(C_2 - C_1) \geq s(\text{opt}(P)) + 0.6v(C_1) - 0.2v(C_1) - 0.2v(C_1) > s(\text{opt}(P))$. This contradicts that opt(P) is the optimal solution.

Case II $v(R_0) < 0.6 \cdot v(C_1)$. We have that $s(\text{opt}(P) - \{C_2\}) \geq s(\text{opt}(P)) + (v(C_1) - v(R_0)) - v(C_2 - C_1) \geq s(\text{opt}(P)) + 0.4v(C_1) - 0.2v(C_1) > s(\text{opt}(P))$. This is also a contradiction.

□

Lemma 6. *Let P be a H_c problem. Then (1) $s(\text{opt}(P)) \leq c_2|g(P)|$, and (2) $c_3|g(P)| \leq s(\text{opt}(P))$.*

Proof: (1) For every point q in a disc of P, there is a grid point $p \in g(P)$ with dist$(p, q) \leq \frac{\sqrt{2}}{2} + c$. Therefore, $s(\text{opt}(P)) \leq |g(P)|\pi(\frac{\sqrt{2}}{2} + c)^2$. (2) We prove this by induction. It is clearly true when $|g(P)| \leq 1$. Assume it is true for $|g(P)| < k$. Let $k = |g(P)|$. Select a grid point $p \in g(P)$. Let M_1 be the set of all discs D in P such that center$(D) \in$ grid(p). Let M_2 be the set of all discs D' in P such that $D' \cap D \neq \emptyset$ for some $D \in M_1$. Let $P' = P - M_1 \cup M_2$. The problem P is adjusted to the problem P'. For every point $p' \in g(P) - g(P')$, dist$(p, p') \leq 2(\frac{\sqrt{2}}{2} + c)$. The number of grid points with distance $\leq 2(\frac{\sqrt{2}}{2} + c)$ to p is $\leq \pi(2\sqrt{2} + 2c + \frac{\sqrt{2}}{2})^2 = \frac{1}{c_3}$. So, we have $|g(P')| \geq |g(P)| - \frac{1}{c_3}$. For $D \in M_1$, $s(\text{opt}(P)) \geq s(\{D\} \cup \text{opt}(P')) \geq s(\text{opt}(P')) + \pi \geq c_3|g(P')| + \pi \geq c_3(|g(P)| - \frac{1}{c_3}) + \pi \geq c_3|g(P)|$. □

Lemma 7. *The algorithm has solution with $s(A(P)) \geq (1 - \epsilon)s(\text{opt}(P)) + (|g(P)|)^\alpha$ if $|g(P)| \geq (b_1 - 2\delta_1)e_1$.*

Proof: We prove by induction. If $(b_1 - 2\delta_1)e_1 \leq |g(P)| \leq e_1$, $s(A(P)) = s(\text{opt}(P)) \geq (1 - \epsilon)s(\text{opt}(P)) + (g(|P|))^\alpha$ by the inequality (2) and part (2) of Lemma 6. Assume that $|g(P)| \geq e_1$ and let L be the center line of the $2w_0$-width separator for $g(P)$. Let P_0, P_1 and P_2 are the sub-problems derived from P in the algorithm.

It is easy to see that $s(\text{opt}(P)) \le s(\text{opt}(P_1)) + s(\text{opt}(P_2)) + s(\text{opt}(P_0))$. Therefore, $s(\text{opt}(P_1)) + s(\text{opt}(P_2)) \ge s(\text{opt}(P)) - s(\text{opt}(P_0))$. Clearly, $g(P_0)$ is the subset of $g(P)$ with distance $\le (c + \frac{\sqrt{2}}{2}) \le w_0$ to L. Therefore, $|g(P_0)| \le z_0 w_0 \sqrt{|g(P)|}$. By Lemma 6, $s(\text{opt}(P_0)) \le c_2 |g(P_0)| \le c_2 \cdot z_0 w_0 \sqrt{|g(P)|}$.

Let G_1 (G_2) be the set of grid points of $g(P)$ on the left (right resp.) of the center line L of the separator. Let S be the set of grid points of $g(P)$ inside the separator area (with distance $\le w_0$ to L). Thus, $|S| \le z_0 w_0 \sqrt{|g(P)|}$. We have $|G_1|, |G_2| \le b_0 |g(P)|$ (Notice that b_0 is the balance upper bound for the separator).

For each $p \in g(P_1)$, there exists a disc $D \in P_1$ with $\text{dist}(p, \text{center}(D)) \le \frac{\sqrt{2}}{2}$. Since center$(D)$ is on one side of L, p can not stay on the other side of L and has distance more than $\frac{\sqrt{2}}{2} (\le w_0)$ to L. Thus, $p \in G_1 \cup S$. Therefore, $g(P_1) \subseteq G_1 \cup S$. For a grid point $q \in G_1 - S$, there exists $D \in P$ such that center$(D) \in \text{grid}(q)$. Since q has distance $> w_0$ to L, center(D) has distance $> w_0 - \frac{\sqrt{2}}{2} = c$ to L. So, $D \notin P_0 \cup P_2$, which implies $D \in P_1$. We have $G_1 - S \subseteq g(P_1)$. We have proven that $G_1 - S \subseteq g(P_1) \subseteq G_1 \cup S$. Similarly, $G_2 - S \subseteq g(P_2) \subseteq G_2 \cup S$. The set $G_1 \cup G_2$ contains all of the grid points in $g(P)$ except those in the line L. So, $g(P) \subseteq G_1 \cup G_2 \cup S$.

Thus, we have the following inequalities: $|g(P)| \le |G_1| + |G_2| + |S|$; $|G_1| \le b_0 |g(P)|$; $|G_2| \le b_0 |g(P)|$; $|G_1| - |S| \le |g(P_1)| \le |G_1| + |S|$; and $|G_2| - |S| \le |g(P_2)| \le |G_2| + |S|$. Since $\frac{|S|}{|g(P)|} \le \frac{z_0 w_0 \sqrt{|g(P)|}}{|g(P)|} \le \frac{z_0 w_0}{\sqrt{|g(P)|}} \le \frac{z_0 w_0}{\sqrt{e_1}} \le \delta_1$ (by (1)), we have

$$|g(P_1)| \ge (b_1 - 2\delta_1)|g(P)| \tag{4}$$

$$|g(P_2)| \ge (b_1 - 2\delta_1)|g(P)| \tag{5}$$

$$|g(P_1)| + |g(P_2)| \ge (1 - 3\delta_1)|g(P)| \tag{6}$$

By our inductive assumption, (4) and (5), $s(A(P_1)) \ge (1-\epsilon)s(\text{opt}(P_1)) + (|g(P_2)|)^\alpha$, and $s(A(P_2)) \ge (1-\epsilon)s(\text{opt}(P_2)) + (|g(P_2)|)^\alpha$. Let $|g(P_1)| = \beta_1 |g(P)|$ and $|g(P_2)| = \beta_2 |g(P)|$. We have $\beta_1 + \beta_2 \ge 1 - 3\delta_1$ and $\beta_1, \beta_2 \ge b_1 - 2\delta_1$. By the standard method in calculus, $\beta_1^\alpha + \beta_2^\alpha$ is minimal when $\beta_1 = \beta_2 = \frac{1-3\delta_1}{2}$. So, $\beta_1^\alpha + \beta_2^\alpha \ge 2(\frac{1-3\delta_1}{2})^\alpha = 2^{1-\alpha}(1-3\delta_1)^\alpha > 2^{1-\alpha}(1 - 3\delta_1\alpha) > 1.12 > 1 + \delta_1$. So, $|g(P_1)|^\alpha + |g(P_2)|^\alpha > (1+\delta_1)|g(P)|^\alpha$. Since $|g(P)| \ge e_1$, $|g(P_1)|^\alpha + |g(P_2)|^\alpha - c_2 z_0 w_0 \sqrt{|g(P)|} > |g(P)|^\alpha$ by inequality (3). Therefore, $s(A(P)) \ge s(A(P_1)) + s(A(P_2)) \ge (1-\epsilon)(s(\text{opt}(P_1)) + s(\text{opt}(P_2)) + (|g(P_1)|)^\alpha + (|g(P_2)|)^\alpha \ge (1-\epsilon)(s(\text{opt}(P)) - s(\text{opt}(P_0)) + (|g(P_1)|)^\alpha + (|g(P_2)|)^\alpha \ge (1-\epsilon)s(\text{opt}(P)) - c_2 \cdot z_0 \cdot w_0 \sqrt{|g(P)|} + (|g(P_1)|)^\alpha + (|g(P_2)|)^\alpha \ge (1-\epsilon)s(\text{opt}(P)) + (|g(P)|)^\alpha$. □

Lemma 8. *The optimal solution* opt(P) *can be computed in* $O(|P|^{\frac{2|g(P)|K}{\delta^2}})$ *time by the brute force method.*

Proof: For each disc D in P, center$(D) \in \text{grid}(q)$ for some $q \in g(P)$. All centers of discs in P stay in the area of size $\le |g(P)|$. By Lemma 5, opt(P) has $\le \frac{2|g(P)|K}{\delta^2}$ discs. The lemma follows since each disc in the optimal solution has $\le |P|$ choices. □

Lemma 9. *The total time of the algorithm is* $O(M \cdot n^a (\log n)^{b+1})$, *where* $M = f_P(e_1)^{\frac{2e_1 K}{\delta^2}}$.

Let $m = |g(P)|$ and $T(m)$ be the time complexity of the algorithm. Clearly, $m \le n$, where $n = |P|$. Assume that C_4 is a positive constant such that finding the separator takes $\le C_4 m^a (\log m)^b$ steps. By Lemma 8 and $|P| \le f(|g(P)|)$, $T(m) \le M$ for $m \le e_1$. We have $T(m) \le C_5 M T(\gamma_1 m) + C_5 M T(\gamma_2 m) + C_4 m^a (\log m)^b$, where $0 \le \gamma_1, \gamma_2 \le b_0$,

$\gamma_1 + \gamma_2 \leq 1$, and C_5 is a constant that is selected big enough so that we have following:

$$\begin{aligned} T(m) &\leq C_5 M T(\gamma_1 m) + C_5 M T(\gamma_2 m) + C_4 m^a (\log m)^b \\ &\leq C_5 M(\gamma_1 m)^a (\log \gamma_1 m)^{b+1} + C_5 M(\gamma_2 m)^a (\log \gamma_2 m)^{b+1} + C_4 m^a (\log m)^b \\ &\leq C_5 M m^a (\log m)^{b+1}. \end{aligned}$$

Since $e_1 = \frac{E_1}{\epsilon^{\frac{1}{1-\alpha}}}$, we let $E_2 = \frac{2E_1 K}{\delta^2}$. The theorem follows from Lemma 9 and Lemma 7. □

Corollary 10. *Let $0 < b_0 < 1$, $0 \leq z_0$, and $0 < \epsilon$ be constants. Let P be an H_c problem. Assume that there exists an $O(n^a(\log n)^b)$ time algorithm for computing the (b_0, z_0)-$O(1)$-width-bounded separator with constants $a \geq 1$ and $b \geq 0$. Then there exists an $O((n^{\frac{E_2}{\epsilon^{\frac{1}{1-\alpha}}}})n^a(\log n)^{b+1})$ time approximation algorithm to output $Q \subseteq P$ with $s(Q) \geq (1 - \epsilon)s(\mathrm{opt}(P))$, where $\alpha = 0.6$, and E_2 is a constant.*

Corollary 11. *Let $0 < b_0 < 1$, $0 \leq z_0$, and $0 < \epsilon$ be constants. Let P be an H'_c problem. Assume that there exists an $O(n^a(\log n)^b)$ time algorithm for computing the (b_0, z_0)-$O(1)$-width-bounded separator with constants $a \geq 1$ and $b \geq 0$. Then there exists an $O(n^a(\log n)^{b+1})$ time approximation algorithm to output $Q \subseteq P$ with $s(Q) \geq (1 - \epsilon)s(\mathrm{opt}(P))$.*

4. A randomized algorithm to find the separator

From corollary 10 and corollary 11, the separator algorithm affects the speed of our approximation. In this section, we will give an $O(n(\log n)^4)$-time randomized algorithm for finding the width-bounded separator on the plane. We will use the following well known fact that can be easily derived from Helly theorem (see Graham et al., 1996; Pach and Agarwal, 1995). Section 5 gives a deterministic linear time algorithm for finding the width-bounded separator, but it highly depends on the results from other papers (Fu, 2006; Jadhar, 1993). This section shows the reader about the existence and algorithm of the separator.

Lemma 12. *For an n-element set P in d-dimensional space, there is a point q with the property that any half-space that does not contain q, covers at most $\frac{d}{d+1}n$ elements of P. Such a point q is called a centerpoint of P. The point q is called $\frac{2}{3}$-center at the case $d = 2$.*

Let $c \geq 3$ be a constant. For a set of n grid points P, we first sort them by their x-coordinates. Now let $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ be all points of P and their x-coordinates are sorted by increasing order: $x_1 \leq x_2 \leq \cdots \leq x_n$. Let $i_1, \cdots, i_k$ be the positions such that $|x_{i_j} - x_{i_j+1}| \geq n^{c-1}$ $(i = 1, \cdots, k)$. Partition P into $P_1, \cdots, P_k$, where $P_t = \{(x_j, y_j) | i_t \leq j < i_{t+1})\}(t = 1, 2, \cdots, k)$. Since $|P| = n$, $|x_{j_1} - x_{j_2}| \leq n \cdot n^{c-1} = n^c$ for every two points $(x_{j_1}, y_{j_1}), (x_{j_2}, y_{j_2})$ in the same set P_t. On the other hand, $|x_{j_1} - x_{j_2}| \geq n^{c-1}$ for every two points $(x_{j_1}, y_{j_1}), (x_{j_2}, y_{j_2})$ in the different sets P_{t_1} and P_{t_2}, respectively. We act the same on each P_i by their y-coordinates. Then P is partitioned into $\cup_{i,j} P_{i,j}$ such that each $P_{i,j}$ is inside an square of size $n^c \times n^c$, and the distance between two points in two different subsets sets P_{i_1,j_1} and P_{i_2,j_2} is at least n^{c-1}. This can be done in $O(n \log n)$ steps. The gap n^{c-1} between two different P_{i_1,j_1} and P_{i_2,j_2} is sufficient for the divide and conquer application for the disc covering problem in the last section since each disc radius is between 1 and another constant. We only design the algorithm a set of n grid points set P in an $n^c \times n^c$ region. It is not meaningful to consider the width $w \geq \sqrt{n}$ as our upper bound $w\sqrt{n}$ is even larger than the total number of points.

Definition 13. Let P be a set of grid points on the plane. A $\frac{2}{3}$*-boundary* is a line L such that the number of points of P on one side of L is in the interval $(\frac{2}{3}|P|, \frac{2}{3}|P| + 1]$. For a $\frac{2}{3}$-boundary L, if L' is another $\frac{2}{3}$-boundary for P such that L and L' are parallel each other, and there are $\geq \frac{1}{3}|P|$ points between them, we call L and L' are *a pair of* $\frac{2}{3}$*-boundaries*. For a line L and vector v, if L can be expressed by the equation $p(t) = p_0 + t \cdot v$, then we say that the line L is *along direction* v. A set of vectors $v_1, v_2, \cdots, v_m$ is called a *m-star vectors* if the angle between v_i and v_{i+1} is $\frac{\pi}{m}$ for $i = 1, 2, \cdots, m-1$. If $L_1, L_2, \cdots, L_m$ are m lines through a same point and each L_i is along v_i, we call $L_1, L_2, \cdots, L_m$ *m-star* for the m-star vectors $v_1, v_2, \cdots, v_m$.

It is easy to see that each $\frac{2}{3}$ center point is between every pair of $\frac{2}{3}$-boundaries. Assume that P is a set of n grid points in an $n^c \times n^c$ area S, where c is a constant. The function $f(L, S, P)$ computes the number of points of P on the two sides of the line L. For a vector v, if $p_i p_j$ is not parallel to v for any two points $p_i \neq p_j$ in P, it always exists a pair of $\frac{2}{3}$-boundaries along the direction v. If the angle between v and $p_i p_j$ is $> \frac{1}{n^{100}}$ for any $p_i \neq p_j$ in P, such a pair of boundaries can be found by binary search via checking the number of points of P on two sides of each line, which can be done by calling functin $f(L, S, P)$. It only checks $O(\log n)$ lines along the vector v. The idea of our algorithm is to find a m-star such that each line of the m-star is between a pair of $\frac{2}{3}$-boundaries. Therefore, each of them gives a balanced partition for the point set P. With high probability, each line also has angle $> \frac{1}{n^{100}}$ with any $p_i p_j$ for every $p_i \neq p_j$ in P. Select one of the m-lines L that has the least number of points from P to close L.

4.1. Intersection between a polygon and a strip area

We use a linked list to store the vertices of a convex polygon in counterclockwise order. A strip area is an area between two parallel lines on the plane. For two parallel lines L_1 and L_2 on the plane, we use $[L_1, L_2]$ to represent the strip region between L_1 and L_2. Each node of the linked list holds a vertex of the polygon. Throughout the algorithm, we often compute the intersection of a strip and a polygon. If the polygon has m nodes, such an intersection can be computed in $O(m)$ steps. For each line segment in the polygon, we check if there is a intersection between it and the strip boundary lines. Record the area of the polygon inside the strip area.

4.2. Count the number of points on the two sides of a line

Assume P is a set of n points in $n^c \times n^c$ square S_0. The square S_0 is partitioned into 4 squares S_1, S_2, S_3, S_4 of the same size. Each S_i is partitioned into smaller and smaller squares until the square size is less than 1×1. We obtain a tree of squares which has the largest square S_0 as root and all the squares in the same level have the same size. The depth of the tree is $O(\log n)$. The squares in this tree are called simple square. Each simple square S is assigned a counter denoted by *count(S)*, which counts the number of points in it.

Lemma 14. *For a set of grid points P of n points on the plane, there is an $O(n \log n)$-time algorithm to computer $count(S)$ for all of those simple squares S that contains at least one point.*

Proof: For each square S with at least one point from the set P, set up a counter for it. For each point p, start from the bottom-most square which contains $p \in P$, increase the counter by one for each simple square which contains p. Since each point only has $O(\log n)$ simple squares that contain it, it takes $O(n \log n)$ steps to set up those counters. □

Algorithm
Input: a line L, a square S_0 of size $n^c \times n^c$, a set of n grid points P inside S_0.
Output: n_1 and n_2 that are the numbers of points of P on the left side and
  the right side of L, respectively.
$f(L, S_0, P)$
  $n_1 = n_2 = 0$;
  for the 4 sub-squares S_1, S_2, S_3, S_4 of S_0
    if ($S_i \cap L = \emptyset$) then
      if S_i is on the left of L, then $n_1 = count(S_i) + n_1$.
      else $n_2 = count(S_i) + n_2$.
    let $S_{i_1}, \cdots, S_{i_k} (k \leq 4)$ be all squares from S_1, S_2, S_3, S_4 that
      $S_{i_j} \cap L \neq \emptyset$ and $count(S_{i_j}) > 0$ $(j = 1, \cdots, k)$.
    $(n_{i_j,1}, n_{i_j,2}) = f(L, S_{i_j})$ for $(j = 1, \cdots, k)$.
    $n_1 = n_1 + (n_{j_1,1} + \cdots + n_{j_k,1})$ and $n_2 = n_2 + (n_{j_1,2} + \cdots + n_{j_k,2})$
  return (n_1, n_2).
End of Algorithm

Lemma 15. *The running time for $f(L, S_0, P)$ is $O(t)$, where t is the number of simple squares $s \in S_0$ that touch L and have $count(s) > 0$.*

Proof: Going through the recursion, we only go to the next level of squares that touch the line L. □

4.3. The algorithm and its time complexity

Definition 16. For two lines L_1 and L_2, share(L_1, L_2) is the number of simple squares that intersect both L_1 and L_2.

Let $\delta > 0$ be a small constant and $m = c_0\sqrt{n}$ for some constant $c_0 > 0$, which will be fixed at the end of the proof for Lemma 23. The algorithm below finds the separator for a set of n grid points in $n^c \times n^c$ region.

Algorithm
  select a random 2D m-star vectors $v_1, v_2, \cdots, v_m$
  find the pairs of $\frac{2}{3}$-boundaries $(L_{1,1}, L_{1,2})$ and $(L_{2,1}, L_{2,2})$ along
    the directions v_1 and v_2 respectively
  let S be the intersection of two strips $[L_{1,1}, L_{1,2}]$ and $[L_{2,1}, L_{2,2}]$
  for $(i = 3$ to $m)$ do
    find the pair of $\frac{2}{3}$-boundaries $(L_{i,1}, L_{i,2})$ along direction v_i
    let S be the intersection between S and the strip region $[L_{i,1}, L_{i,2}]$

$m_0 = \infty$
select a point $p \in S$
for $i = 1$ to m
  let L_i be a line through p
  if (measure($L_i, P, a) < m_0$) then $m_0 =$ measure(L_i, P, a) and $L = L_i$
return L
End of Algorithm

Lemma 17. *During the first loop, S is an nonempty polygon all the time.*

Proof: The intersection between a convex polygon and a strip area is still convex polygon. By Lemma 12, S is nonempty all the time. □

Lemma 18. *For two lines L_1 and L_2 with angle $0 < \theta \leq \frac{\pi}{2}$ between them, they share at most $\frac{c_1 \log n}{\sin\theta}$* simple square*s for some constant c_1.*

Proof: Let p be the intersection point of the two lines L_1 and L_2. If s_1 and s_2 are intersections between L_1, L_2 and a $t \times t$ square respectively, then dist$(s_1, s_2) \leq \sqrt{2}t$. It is easy to see that $dist(s_1, p) \leq \frac{\sqrt{2}t}{\sin\theta}$ and dist$(s_2, p) \leq \frac{\sqrt{2}t}{\sin\theta}$. Every point q in a $t \times t$ square that touches both L_1 and L_2 has distance $\leq \frac{\sqrt{2}t}{\sin\theta} + \sqrt{2}t$ to p. Furthermore, the point q has distance $\leq \sqrt{2}t$ to the middle line (through p) between L_1 and L_2 . Since those $t \times t$ squares do not overlap one other, the total number of them is $\leq \frac{2(\frac{\sqrt{2}t}{\sin\theta} + \sqrt{2}t)2\sqrt{2}t}{t^2} = 4\sqrt{2}(\frac{\sqrt{2}}{\sin\theta} + \sqrt{2}) \leq \frac{16}{\sin\theta}$. For some constant c_3, there are at most $c_3 \log n$ possible different sizes for the simple squares. Thus, L_1 and L_2 can share at most $\frac{16 c_3 \log n}{\sin\theta}$ simple squares. □

Lemma 19. *Let $v_1, v_2, \cdots, v_m$ be a m-star vectors. Each vector v_i has at most k lines along it (the line set along direction v_i is denoted by $L(v_i)$). Then for each line L_j in $L(v_j)$, $\sum_{i=1, i\neq j}^{m} \sum_{L_i \in L(v_i)}$* share$(L_j, L_i) \leq c_4 k \cdot m \cdot (\log m) \cdot (\log n)$ *for some constant $c_4 > 0$.*

Proof: For $L_i \in L(v_i)$, the angle between L_i and L_j is $\frac{\pi|i-j|}{m}$. By Lemma 18, share$(L_j, L_i) \leq \frac{c_1 \log n}{\sin\frac{|i-j|\pi}{m}} \leq \frac{c_2 m \log n}{\pi|i-j|}$ for some constant c_2. Therefore,

$$\sum_{i=1, i\neq j}^{m} \sum_{L_i \in L(v_i)} \text{share}(L_j, L_i) \leq \sum_{i=1, i\neq j}^{m} \sum_{L_i \in L(v_i)} \frac{c_2 m \log n}{\pi|i-j|} \leq \sum_{i=1, i\neq j}^{m} \frac{k c_2 m \log n}{\pi|i-j|}$$

$$\leq \frac{k c_2 m \log n}{\pi} \sum_{i=1, i\neq j}^{m} \frac{1}{|i-j|} < \frac{2 k c_2 m \log n}{\pi} \sum_{i=1}^{m} \frac{1}{i} \leq c_4 \cdot k \cdot m \cdot (\log m) \cdot (\log n),$$

where c_4 is a constant $> \frac{2c_2}{\pi}$. □

Lemma 20. *Let $\theta \leq \frac{\pi}{4m}$. Let $M_1, \cdots, M_t$ be t fixed line. Let $L_1, \cdots, L_m$ be the m lines along the m directions in a random m-star vectors $v_1, \cdots, v_m$, respectively. Then with probability $\leq \frac{4\theta \cdot m \cdot t}{\pi}$, one of $M_1, M_2, \cdots, M_t$ has angle $\leq \theta$ with some line from $L_1, \cdots, L_m$.*

Proof: We assume that the vector v_1 has an angle between 0 to $\frac{\pi}{m}$ with x-axis. Each M_j can have angle $\leq \theta$ with at most one line among $L_1, \cdots, L_m$. For a line L_i with angle to x-axis between $\frac{k\pi}{m}$ and $\frac{(k+1)\pi}{m}$, it has probability $\leq \frac{2\theta}{\frac{\pi}{m}} = \frac{2\theta m}{\pi}$ to have angle $\leq \theta$ with M_j. Therefore, the probability is $\leq \frac{4\theta m}{\pi} \cdot t$ to have one line $M_i \in \{M_1, \cdots, M_t\}$ such that that M_i has angle $\leq \theta$ with one of the vectors $L_1, L_2, \cdots, L_m$. □

Lemma 21. *Let v be a vector and P be a set of n grid points in a $n^c \times n^c$. The vector v has angle $\geq \theta$ with any line $p_i p_j$ for every $p_i \neq p_j$ in P. It generate $O(\log n + \log \frac{1}{\theta})$ lines L along v (to query $f(L, S_0, P)$) to find out a pair of $\frac{2}{3}$ boundaries at direction v.*

Proof: We assume that v is along the direction of y-axis. For each point p on the plane, let $p(x)$ be the x-coordinate of p. Since the angle between y-axis and $p_i p_j$ is $\geq \theta$ and $\mathrm{dist}(p_i, p_j) \geq 1$, we have $|p(x_i) - p(x_j)| \geq \sin\theta$. Let L_1 and L_2 be two vertical lines of distance $\leq n^c$ such that all points of P are between them. Let L be the middle vertical lines between L_1 and L_2. Let $(n_1, n_2) = f(L, S_0, P)$. If $n_1 < \frac{n}{3}$, then let $L_1 = L$. Otherwise, let $L_2 = L$. Repeat the binary search until one $\frac{2}{3}$-boundary line is found. After $O(\log n + \log \frac{1}{\theta})$ queries the function $f(L, S_0, P)$, the distance between two lines L_1 and L_2 is $< \sin\theta$. □

Lemma 22. *Let $v_1, v_2, \cdots, v_m$ be a random m-star vectors. Let $h_0 > 2$ be a constant and $\theta = \frac{\pi}{4m^{h_0}}$. If for every two points $p_i, p_j \in P$, $p_i p_j$ has angle $\geq \theta$ with $v_k (k = 1, \cdots, m)$. Then the algorithm spends $O(n(\log n)^4)$ for finding the separator.*

Proof: In order to compute measure(L, P, a), we let L' and L'' be two lines on the left and right sides of L respectively, and both of them are parallel to L. Furthermore, both L' and L'' have distance a to L. Let $(n_1', n_2') = f(L', S_0, a)$ and $(n_1'', n_2'') = f(L'', S_0, a)$. Since all points of P with distance $\leq a$ to L are between L' and L'', measure$(L, P, a) = n - n_1' - n_2''$.

Let $L(v_i)$ be the set of all lines L along v_i that are used to query the function $f(L, S_0, P)$ in the algorithm. The set $L(v_i)$ includes the lines (along v_i) for finding the the pair of $\frac{2}{3}$-boundaries along the v_i and also the line L_i' and L_i'' for computing measure(L_i, P, a). It is easy to see that the computational time of the algorithm is propositional to the number times that the lines in $\cup_{i=1}^m L(v_i)$ touch the simple squares s with *count*$(s) > 0$.

For a square s, assume s is touched by the lines in $U_1 \cup U_2 \cdots U_m$, where $U_i \subseteq L(v_i)$ $(i = 1, \cdots, m)$. If $U_1 = U_2 = \cdots = U_m = \emptyset$, s is called of type 0. If there exists only one i $(1 \leq i \leq m)$ with $U_i \neq \emptyset$, s is called of type 1. Otherwise, s is of type 2 (there exist $i \neq j$ with $U_i \neq \emptyset$ and $U_j \neq \emptyset$). For each v_i, $|L(v_i)| \leq c_5 \log n$ for some constant c_5. This is because that $L(v_i)$ is generated during the binary search for a pair of $\frac{2}{3}$-boundaries and the set $L(v_i)$ has $O(\log n)$ lines the along v_i (by Lemma 21 with $m = O(\sqrt{n})$ and $\theta = \frac{1}{m^{O(1)}}$). Define touch$(s)$ to be the number of lines in $\cup_{i=1}^m L(v_i)$ that intersects the simple square s.

There are only $O(n \log n)$ simple squares s that has points in P (*count*$(s) > 0$). Since $|L(v_i)| \leq c_5 \log n$, $\sum_{s \text{ is of type 1 and } count(s) > 0} \mathrm{touch}(s) = O(n(\log n)^2)$. For the set of of all type 2 simple squares,

$$\sum_{s \text{ is of type 2 and } count(s) > 0} \mathrm{touch}(s)$$

$$\leq 2 \sum_{j=1}^{m} \sum_{L_j \in L(v_j)} \Big(\sum_{i=1, i \neq j}^{m} \sum_{L_i \in L(v_i)} \mathrm{share}(L_j, L_i) \Big)$$

P1: NAE
SJNW719-08-NO00007132 styleA.cls (2005/11/30 v1.0 LaTeX Springer document class) February 10, 2006 16:14

$$\leq 2\sum_{j=1}^{m}\sum_{L_j\in L(v_j)} c_5\cdot\log n\cdot c_4\cdot m\cdot(\log m)(\log n)(\text{ by Lemma 19 with } k\leq c_5\log n)$$

$$\leq 2|\cup_{j=1}^{m} L(v_j)|\cdot c_5\cdot\log n\cdot c_4\cdot m\cdot(\log m)(\log n)$$

$$\leq 2m\cdot(c_5\log n)\cdot c_5\cdot c_4\cdot m\cdot(\log n)^3 = O(n\cdot(\log n)^4).$$

Combining the two cases above, we conclude that

$$\sum_{s \text{ is a simple square}} \text{touch}(s)$$

$$= \sum_{s \text{ is of type } 0} \text{touch}(s) + \sum_{s \text{ is of type 1 and } count(s)>0} \text{touch}(s) + \sum_{s \text{ is of type 2 and } count(s)>0} \text{touch}(s)$$

$$= 0 + O(n\log n)^2 + O(n(\log n)^4) = O(n(\log n)^4).$$

□

Lemma 23. *Let $L_1, L_2, \cdots, L_m$ be a m-star through the same point o. There is a line L_i such that P has $\leq (\frac{4a}{\sqrt{\pi}})\cdot\sqrt{n}+\delta\sqrt{n}$ grid points from P with distance $\leq a$ to L_i.*

Proof: For a grid point p, the number of lines that p has $\leq a$ distance to them is $\leq 2\arcsin\frac{a}{\text{dist}(p,o)}\cdot\frac{m}{\pi}+1$. The total number of cases is $T=\sum_{i=1}^{n}(2\arcsin\frac{a}{\text{dist}(p_i,o)}\cdot\frac{m}{\pi}+1)=\frac{2m}{\pi}\sum_{i=1}^{n}(\arcsin\frac{a}{\text{dist}(p_i,o)})+n$. We present an upper bound for $\sum_{i=1}^{n}(\arcsin\frac{a}{\text{dist}(p_i,o)})$ by using the method as Fu and Wang (2004).

Let $\epsilon>0$ be a small constant which will be determined later. Select r_0 to be large enough such that for every point p with $dist(o,p)\geq r_0$, $\arcsin\frac{a}{\text{dist}(o,p)}<(1+\epsilon)\frac{a}{\text{dist}(o,p)}$ and $\frac{1}{\text{dist}(o,p')}<\frac{1+\epsilon}{\text{dist}(o,p)}$ for every point p' with $\text{dist}(p',p)\leq\frac{\sqrt{2}}{2}$. Let P_1 be the set of all points p in P such that $\text{dist}(o,p)<r_0$. The number of grid points in P_1 is no more than $\pi(r_0+\frac{\sqrt{2}}{2})^2$. For each point $p\in P_1$, $\arcsin\frac{a}{\text{dist}(o,p)}\leq\frac{\pi}{2}$. Let r be the minimum radius of a circle C with center at o and contains n grid points. Let $r'=r+\frac{\sqrt{2}}{2}$. The circle C' of radius r' contains all the 1×1 unit grid squares with center at points of P. Therefore, $\sum_{i=1}^{n}\arcsin\frac{a}{\text{dist}(p_i,o)}=\sum_{p\in P_1}\arcsin\frac{a}{\text{dist}(p,o)}+\sum_{p\in P-P_1}\arcsin\frac{a}{\text{dist}(p,o)}\leq\sum_{p\in P_1}\frac{\pi}{2}+\sum_{p\in P-P_1}\arcsin\frac{a}{\text{dist}(o,p)}<\frac{\pi^2}{2}(r_0+\frac{\sqrt{2}}{2})^2+\sum_{p\in P-P_1}\frac{(1+\epsilon)a}{\text{dist}(o,p)}\leq\frac{\pi^2}{2}(r_0+\frac{\sqrt{2}}{2})^2 + a(1+\epsilon)^2\int\int_{C'}\frac{1}{\text{dist}(o,p)}d_x d_y=a(1+\epsilon)^2\int_0^{2\pi}\int_0^{r'}\frac{\rho}{\rho}d_\rho d_\theta+\frac{\pi^2}{2}(r_0+\frac{\sqrt{2}}{2})^2 = 2a\pi(1+\epsilon)^2 r'+\frac{\pi^2}{2}(r_0+\frac{\sqrt{2}}{2})^2$.

It is easy to verify that $r\leq\frac{1}{\sqrt{\pi}}\sqrt{n}+4\sqrt{2}$ (see Lemma 9 in Fu and Wang (2004)). Therefore, there is a line L_i that has $\leq\frac{T}{m}\leq\frac{\frac{2m}{\pi}(2a\pi(1+\epsilon)^2r'+\frac{\pi^2}{2}(r_0+\frac{\sqrt{2}}{2})^2)+n}{m}\leq(\frac{4a}{\sqrt{\pi}})\cdot\sqrt{n}+\delta\sqrt{n}$ grid points from P with distance $\leq a$ if ϵ is selected small enough and c_0 is big enough. □

Theorem 24. *For constant $a>0$ and small constant $\delta>0$, there is an $O(n(\log n)^4)$-time randomized algorithm for finding a-width separator for a set of n grid points set P in a*

$n^{O(1)} \times n^{O(1)}$ region such that each side has $\leq \frac{2}{3}|P| + 1$ points of P, and the number of points with distance to the center line of the separator is $\leq (\frac{4a}{\sqrt{\pi}}) \cdot \sqrt{n} + \delta\sqrt{n}$.

Proof: Let $\theta = \frac{\pi}{4m^{h_0}}$ for constant $h_0 > 2$. By Lemma 20, it has probability $\geq 1 - \frac{1}{m^{h_0-1}}$ that for every two points $p_i, p_j \in P$, the line $p_i p_j$ has angle $\geq \theta$ with any v_k among the random m-star $v_1, \cdots, v_m$. By Lemma 22, the computational time is $O(n(\log n)^4)$. By Lemma 23, we can find a line L_i that satisfies the requirements of the theorem. □

This theorem implies the corollary below by combining with corollary 11.

Corollary 25. *Let $\epsilon > 0$ be a constant and P be a H_c' problem. There exists an $O(n(\log n)^5)$ time randomized approximation algorithm to output $Q \subseteq P$ with $s(Q) \geq (1-\epsilon)s(\text{opt}(P))$.*

5. Linear time deterministic algorithm for 2D separator

Using the linear time algorithm for finding the center point for a set of 2D points by Jadhar (1993) and the existence of width-bounded separator by Fu (2006), we derive a deterministic linear time algorithm for 2D width-bounded geometric separator. The width-bounded geometric separator studied in this section is more general than that in the previous sections. This version was applied in developing $2^{O(\sqrt{n})}$-time exact algorithms (Fu, 2006) for a class of geometric NP-hard problems whose previous exact algorithm take $n^{O(\sqrt{n})}$-time.

The *diameter* of any $P \subseteq R^2$ is $\max_{p_1, p_2 \in P} \text{dist}(p_1, p_2)$. For $a > 0$ and a set A of points in R^2, if the distance between every two points in A is at least a, then A is called *a-separated*. For $\epsilon > 0$ and a set Q of points in R^2, an *ϵ-sketch* of Q is another set P of points in R^2 such that each point in Q has distance $\leq \epsilon$ to some point in P. We say P is a sketch of Q if P is an ϵ-sketch of Q for some constant $\epsilon > 0$ (ϵ does not necessarily depend on the size of Q). A sketch set is usually a 1-separated set such as a grid point set. A weight function $w : P \to [0, \infty)$ is often used to measure the density of Q near each point in P. Let $f : R^2 \to R$ be a smooth function. Its *curve* is the set $L(f) = \{v \in R^2 | f(v) = 0\}$. A *line* in R^2 through a fixed point $p_0 \in R^2$ is defined by the equation $(p - p_0) \cdot v = 0$, where v is the normal vector of the plane and "." is the usual vector inner product ($u \cdot v = \sum_{i=1}^{d} u_i v_i$ for $u = (u_1, \ldots, u_d)$ and $v = (v_1, \ldots, v_d)$). A line in R^2 is determined by $L(f)$ for some linear function $f : R^2 \to R$.

Definition 26. Given any $Q \subseteq R^2$ with sketch $P \subseteq R^2$, a constant $a > 0$, and a weight function $w : P \to [0, \infty)$, an *a-wide-separator* is determined by the curve $L(f)$ for some linear function $f : R^2 \to R$. The separator has two measurements for its quality of separation: (1) $\text{balance}(L(f), Q) = \frac{\max(|Q_1|, |Q_2|)}{|Q|}$, where $Q_1 = \{q \in Q | f(q) < 0\}$ and $Q_2 = \{q \in Q | f(q) > 0\}$; and (2) $\text{measure}(L(f), P, \frac{a}{2}, w)$, where in general $\text{measure}(A, P, x, w) = \sum_{p \in P, \text{dist}(p, A) \leq x} w(p)$ for any $A \subseteq R^2$ and $x > 0$. When f is fixed or no confusion arises, we use $\text{balance}(L, Q)$ and $\text{measure}(L, P, \frac{a}{2}, w)$ to stand for $\text{balance}(L(f), Q)$ and $\text{measure}(L(f), P, \frac{a}{2}, w)$, respectively.

Definition 27. A (b, c)*-partition* of the 2-dimensional plane R^2 divides the plane into a disjoint union of regions $P_1, P_2, \ldots$, such that each P_i, called a *regular region*, has an area size of b and a diameter $\leq c$. A (b, c)*-regular point set* A is a set of points in R^2 with a

(b, c)-partition $P_1, P_2, \ldots$, such that each P_i contains at most one point from A. For two regions A and B, if $A \subseteq B$ ($A \cap B \neq \emptyset$), we say B *contains* (*intersects* resp.) A.

Definition 28. Let $a > 0$, p and o be two points in R^2. Define $Pr_2(a, p_0, p)$ to be the probability that the point p has $\leq a$ perpendicular distance to a random line L through the point p_0. Define function $f_{a,p,o}(L) = 1$ if p has a distance $\leq a$ to the line L through o, or 0 otherwise. The expectation of function $f_{a,p,o}(L)$ is $E(f_{a,p,o}(L)) = Pr_d(a, o, p)$. Assume $P = \{p_1, p_2, \ldots, p_n\}$ is a set of n points in R^2 and each p_i has weight $w(p_i) \geq 0$. Define function $F_{a,P,o}(L) = \sum_{p \in P} w(p) f_{a,p,o}(L)$.

We give an upper bound for the expectation $E(F_{a,P,o}(L))$ for $F_{a,P,o}(L)$ in the lemma below.

Lemma 29. *Fu (2006) Let $d \geq 2$. Let o be a point in R^2, $a, b, c > 0$ be constants and $\epsilon, \delta > 0$ be small constants. Assume that $P_1, P_2, \ldots,$ form a (b, c)-partition for R^2, and the weights $w_1 > \cdots > w_k > 0$ satisfy $k \cdot \max_{i=1}^{k}\{w_i\} = O(n^\epsilon)$. Let P be a set of n weighted (b, c)-regular points in a 2-dimensional plane with $w(p) \in \{w_1, \ldots, w_k\}$ for each $p \in P$. Let n_j be the number of points $p \in P$ with $w(p) = w_j$ for $j = 1, \ldots, k$. We have $E(F_{a,P,o}(L)) \leq (k_2 \cdot (\frac{1}{b})^{\frac{1}{2}} + \delta) \cdot a \cdot \sum_{j=1}^{k} w_j \cdot \sqrt{n_j} + o(n^\epsilon)$, where $k_2 = \frac{4}{\sqrt{\pi}}$.*

Definition 30. A set of vectors $v_1, v_2, \cdots, v_m$ is called a *m-star vectors* if the angle between v_i and v_{i+1} is $\frac{\pi}{m}$ for $i = 1, 2, \cdots, m-1$. If $L_1, L_2, \cdots, L_m$ are m lines through a same point and each L_i is along the direction of the vector v_i, we call $L_1, L_2, \cdots, L_m$ *m-star* for the m-star vectors $v_1, v_2, \cdots, v_m$.

Theorem 31. *Jadhar (1993) There exists an $O(n)$ time algorithm to find a center point for a finite set of points on the plane.*

Theorem 32. *Let $a, a_1, a_2 > 0$ be constants and $\epsilon, \delta > 0$ be small constants. Let P be a set of n (a_1, a_2)-grid points in R^2, and Q be another set of m points in R^2 with sketch P. Let $w_1 > w_2 \cdots > w_k > 0$ be positive weights with $k \cdot \max_{i=1}^{k}\{w_i\} = O(n^\epsilon)$, and w be a mapping from P to $\{w_1, \cdots, w_k\}$. There exists a deterministic $O(n+m)$ time algorithm to find a hyper plane L such that (1) each half plane has $\leq \frac{2}{3}m$ points from Q, and (2) for the subset $A \subseteq P$ containing all points in P with $\leq a$ distance to L has the property $\sum_{p \in A} w(p) \leq \left(k_d \cdot \frac{1}{\sqrt{a_1 \cdot a_2}} + \delta\right) \cdot a \cdot \sum_{j=1}^{k} w_j \cdot \sqrt{n_j} + O(n^\epsilon)$ for all large n.*

Proof: Let $b = a_1 \cdot a_2$ and $\delta_1 = \frac{\delta}{2}$. By Lemma 29, $E(F_{a,P,o}) \leq \left(\frac{k_2}{\sqrt{b}} + \delta_1\right) \cdot a \cdot \sum_{j=1}^{k} w_j \cdot \sqrt{n_j} + O(n^\epsilon)$. In particular, we have $\sum_{p \in P} Pr_2(p, o, a) \leq \left(\frac{k_d}{\sqrt{b}} + \delta_1\right) \cdot a\sqrt{n}$ when we let each weight be equal to 1. Each point p of P has format $< (x, y), w(p) >$, where (x, y) is the coordinates for p and $w(p)$ is the weight of p.

Algorithm: find separator on the plane

(a) Input: A set of points Q and a set of weighted points P on the plane.

(b) find a center point o for the set Q (see Theorem 31).

(c) let $m = \frac{\sqrt{n}}{\delta_1 a}$.

(d) select a m-star $l_1, \cdots, l_m$ with center at o.

(e) let $N(l_i) = 0$ for $i = 1, \cdots, m$.

P1: NAE
SJNW719-08-NO00007132 styleA.cls (2005/11/30 v1.0 LaTeX Springer document class) February 10, 2006 16:14

(f) for each $p \in P$
(e) for each l_i with $\text{dist}(p, l_i) \le a$, $N(l_i) = N(l_i) + w(p)$
(k) Output the line l_i with the least $N(l_i)$.
End of the Algorithm

We analyze the algorithm. By Theorem 31, the center point can be found in $O(m)$ steps. The probability that a point p has distance $\le a$ to L is $Pr_2(p, o, a) = \frac{2 \arcsin \frac{a}{\text{dist}(o,p)}}{\pi}$. For a grid point p, the number of lines that p has $\le a$ distance to them is $\le m \cdot Pr_2(p, o, a) + 1$. Now we have those $N(l_i)(i = 1, \cdots, m)$ after running the algorithm. Each $N(l_i)$ is the sum of weights of the points of P with distance $\le a$ to the line l_i. In other words, $N(l_i) = F_{a,P,o}(l_i)$. For each point $p \in P$, its weight is added to the $N(l_i)$s for at most $m Pr_2(p, o, a) + 1$ lines. We conclude that $\sum_{i=1}^{m} N(l_i) = \sum_{p\in P} w(p) \cdot (m \cdot Pr_2(p, o, a) + 1) = m(\sum_{p\in P} w(p) Pr_2(p, o, a)) + \sum_{p\in P} w(p) = m \cdot \text{E}(F_{P,o,a}) + \sum_{j=1}^{k} w_j n_j$. We also have $\frac{\sum_{j=1}^{k} w_j n_j}{m} = \sum_{j=1}^{k} w_j \frac{n_j}{m} = \sum_{j=1}^{k} w_j \frac{n_j}{\frac{\sqrt{n}}{\delta_1 a}} \le \sum_{j=1}^{k} w_j \delta_1 a \sqrt{n_j}$. Therefore, one of the m lines has the sum of weights $N(l_i) \le (\sum_{i=1}^{m} N(l_i))/m \le (\frac{k_d}{\sqrt{b}} + 2\delta_1) \cdot a \cdot \sum_{j=1}^{k} w_j \cdot \sqrt{n_j} + O(n^{\epsilon})$ for all large n.

After the center is found, the total number of operations is propositional to $\sum_{p\in P} m Pr_2(p, o, a) + 1 \le m \sum_{p\in P} Pr_2(p, o, a) + n \le \frac{\sqrt{n}}{a\delta_1}(\frac{k_d}{\sqrt{b}} + \delta_1) \cdot a\sqrt{n} + n \le (\frac{(\frac{k_d}{\sqrt{b}} + \delta_1)}{\delta_1} + 1) n = O(n)$ □

Corollary 33. *Let $\epsilon > 0$ be a constant and P be a H_c' problem. There exists an $O(n(\log n))$ time approximation algorithm to output $Q \subseteq P$ with $s(Q) \ge (1 - \epsilon) s(\text{opt}(P))$.*

Acknowledgments We would like to thank the anonymous referees from COCOON'05 for their helpful comments. We are also grateful to Lusheng Wang for his much help.

References

Alon N, Seymour P, Thomas R (1990) Planar Separator, SIAM J. Discr. Math., 7(2):184–193

Asano T, Brass P, Sasahara S (2004) Disc covering problem with application to digital halftoning, International Conference on Computational Science and Applications, Assisi, Italy, Proceedings, Part III. Lecture Notes in Computer Science 3045, Springer 2004, pp. 11–21

Asano T, Matsui T, Tokuyama T (2000) Optimal roundings of sequences and matrices. Nordic Journal of Computing, 3(7):241–256

Djidjev HN (1982) On the problem of partitioning planar graphs, SIAM Journal on Discrete Mathematics, 3(2):229–240

Fu B (2006) Theory and Application of Width Bounded Geometric Separator, The draft was posted at Electronic Colloquium on Computational Complexity 2005, TR05-13. Accepted by the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS 2006), Marseille, France

Fu B, Wang W (2004) A $2^{O(n^{1-1/d} \log n)}$-time algorithm for d-dimensional protein folding in the HP-model, In proceedings of 31st International Colloquium on Automata, Languages and Programming, pp. 630–644

Gazit H (1986) An improved algorithm for separating a planar graph, manuscript, USC

Graham R, Grötschel M, Lovász L (1996) Handbook of Combinatorics (volume I), MIT Press

Jadhar S (1993) Computing a center of a finite planar set of points in linear time, The 9th annual ACM symposium on computational geometry, pp. 83–90

Lipton RJ, Tarjan R (1979), A separator theorem for planar graph, SIAM J. Appl. Math. 36:177–189

Miller GL, Teng S-H, Vavasis SA (1991) An unified geometric approach to graph separators. In 32nd Annual Symposium on Foundation of Computer Science, IEEE, pp. 538–547

Ostromoukhov V (1993) Pseudo-random halftone screening for color and black & white printing, Proceedings of the 9th congress on advances in non-impact printing technologies, Yohohama, pp. 579–581

Ostromoukhov V, Hersch RD (1999) Stochastic clustered-dot dithering, Journal of Electronic Imaging, pp. 4(8):439–445

Pach J, Agarwal PK (1995) Combinatorial Geometry, Wiley-Interscience Publication

Sasahara S, Asano T (2003) Adaptive cluster arrangement for cluster-dot halftoning using bubble packing method, Proceeding of 7th Japan joint workshop on algorithms and computation, Sendai, pp. 87–93

Smith WD, Wormald NC (1998) Application of geometric separator theorems, FOCS, pp. 232–243

Spielman DA, Teng SH (1996) Disk packings and planar separators, 12th Annual ACM Symposium on Computational Geometry, pp. 349–358