Dynamic Programming Based Approximation Scheme for Locating Disks within Convex Polygons*

Hirofumi Aota, Takuro Fukunaga, Hiroshi Nagamochi

Department of Applied Mathematics and Physics,
Graduate School of Informatics, Kyoto University, Japan
{hirofumi,takuro,nag}@amp.i.kyoto-u.ac.jp

Abstract

This paper considers a problem of locating the given number of disks into a container so that the area covered by the disks is maximized. In the problem, the radii of disks can be changed arbitrarily unless they overlap outside of the container, and the disks are allowed to overlap each other. We present an approximation scheme for this problem assuming that the container is a convex polygon. Our scheme gives a $(0.78 - \epsilon)$-approximation algorithm for any small $\epsilon > 0$. Since the computation time of our algorithm depends on the number of corners of the convex polygon exponentially, we also give a heuristic to reduce the number of corners.

1 Introduction

In this paper, we consider a problem of locating circular disks within a convex polygon so that the area covered by the disks is maximized. More formally, the problem is defined as follows. Given a 2-dimensional convex polygon P with k vertices and a positive integer n, we are asked to decide the positions and the radii of n disks such that the disks are located inside the polygon P and the area covered by the disks is maximized. The located disks are allowed to overlap each other, but they cannot overlap outside P. We call the problem *disk covering problem* in this paper.

The problem of locating disks is a fundamental geometric problem, and has many variations on the problem setting; The sizes of disks are fixed or not, disks can overlap each other or not, the disks must be located inside a given container or can overlap outside of it, and so on. Because of their importance, the problems attract many researchers and there is a long history of researches on them. One example of such variations is the problem of packing identical disks into a container (e.g., circle, square, triangle) without overlap each other. Since there are too many papers on it, we refer to only [2, 4, 5, 11, 13] here. The problem of packing given non-identical disks into the smallest circle without overlap is also considered [14, 15, 16]. In particular, Sugihara [14] and Sugihara et al. [15] mentioned its application to wire bundling. Another example is the problem of covering an object by disks. In this problem, disks can overlap outside the object in contrast with our problem. Previous

*Technical Report 2010-017, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University. Updated on December 14, 2010.

works on this problem (e.g., [6, 9, 12]) discuss covering entire objects such as rectangles and triangles by a specified number of identical disks so that the radii of the disks are minimized.

As these variations, the disk covering problem, we consider in this paper, is formulated naturally, and has many practical applications. We here would like to mention two examples of applications. One application is design of wireless sensor locations in a region. The area covered by a sensor can be represented by a disk. The radius of the disk can be set arbitrarily by changing the strength of signal whereas it is prohibited to send so strong signal that it leaks to outside of the region. In this situation, the disk covering problem formulates deciding locations of the given number of sensors inside a region with the objective to maximize the covered area. A similar application is discussed in [8], which considered a problem of choosing a specified number of disks covering all given points from a set of disks.

Another application is approximation of a polygon by disks. Imamichi et al. [10] proposed a scheme for packing given polygons into a container without overlap, which has many important applications especially to industrial fields. The key idea of their scheme is to approximate polygons by sets of disks for making it easy to judge whether located two polygons overlap or not. By solving the disk covering problem with a polygon as a container, we can obtain a good approximation of it.

The disk covering problem is a special case of the *maximum coverage problem*; In the maximum coverage problem, given a finite set U, a family $\mathcal{S}$ of subsets of U and an integer k, we are required to find a subfamily $\mathcal{S}'$ of $\mathcal{S}$ such that $|\mathcal{S}'| \leq k$ and $|\cup_{X \in \mathcal{S}'} X|$ is maximized. If U consists of all points in the polygon P and each element of $\mathcal{S}$ is a set of points included in a disk inside P, then the maximum coverage problem is equivalent to the disk covering problem. It is known that a greedy algorithm is a good approximation algorithm for the maximum coverage problem. Here, when an algorithm is guaranteed to output a solution of objective value at least α times the optimum for a maximization problem and some constant $\alpha \leq 1$, the algorithm is called *α-approximation algorithm*, and α *approximation factor*. Hochbaum [7] showed that a greedy algorithm achieves approximation factor $1-1/e \approx 0.63$ for the maximum coverage problem. Moreover Feige [3] proved that no polynomial-time algorithm achieves approximation factor $1 - 1/e + \epsilon$ for any $\epsilon > 0$ unless P=NP.

For applying the greedy algorithm presented by Hochbaum [7] to the disk covering problem, we need to repeat computing a place the maximal disk located at which covers the maximum area uncovered by the disks that have been already located inside a polygon. This can be done by maximizing one-variable concave functions. However it is hard to maximize the functions precisely because the solutions cannot be derived analytically. Moreover algorithms for the maximum coverage problem can exploit no geometric information in the disk covering problem. Hence it may be possible to achieve better approximation factor for the disk covering problem than for the maximum coverage problem. These are the reasons why we need a new algorithm for the disk covering problem.

In this paper, we propose a $(0.78 - \epsilon)$-approximation algorithm for the disk covering problem with any small $\epsilon > 0$. Roughly speaking, our algorithm first divides the given polygon P into small pieces, and then solves the problem of locating disks within the pieces. Our algorithm applies the dynamic programming (DP) to solve the problem of locating disks within pieces. Since the disk covering problem is defined geometrically, it may not seem appropriate to apply DP for the disk covering problem. We show that a wise discretization of the problem makes it possible to apply DP while introducing small approximation factor.

The computation time of our algorithm depends on the number k of vertices of P exponentially, and on a function h on the maximum inner angle θ of P that approaches $+\infty$

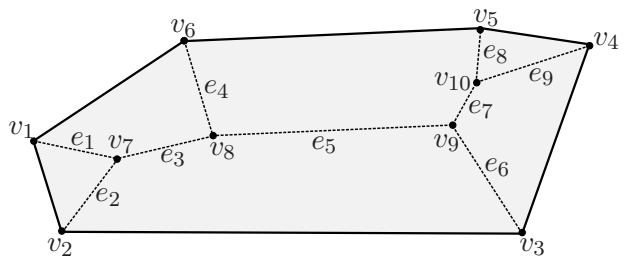

Figure 1: An example of medial axis, where $V_p = \{v_1, v_2, \ldots, v_6\}$, $V_m = \{v_7, v_8, v_9, v_{10}\}$, $E_p = \{e_1, e_2, e_4, e_6, e_8, e_9\}$, and $E_m = \{e_3, e_5, e_7\}$.

as θ approaches π. We also show that this is not a big drawback by presenting a method to transform a convex polygon P into another of fewer vertices and smaller maximum inner angle such that solving the disk covering problem with the obtained one gives an approximate solution to the problem with P. If P has a constant number of vertices and its maximum inner angle is a constant smaller than π, the computation time of our algorithm is a polynomial in n and $1/\epsilon$.

The rest of this paper is organized as follows. Section 2 introduces definitions and fundamental theorems. Sections 3 and 4 define problems of locating disks within pieces of a convex polygon, and present algorithms to solve them. Section 5 describes the entire our algorithm, and analyses it. It also explains a method to transform a convex polygon into another of fewer vertices and smaller maximum inner angle. Section 6 concludes the paper.

2 Preliminaries

We denote a segment between points p and q by $\overline{pq}$. For an object O, $A(O) \in \Re$ denotes the area covered by O. For a non-negative real number α and an object O, αO stands for the object obtained by scaling up or down O α times. We have $A(\alpha O) = \alpha^2 A(O)$. Let O and O' be two located objects. We represent the union of O and O' by $O \cup O'$, and the intersection of O and O' by $O \cap O'$. If O covers the area covered by O', then we write $O' \subseteq O$.

We define the *position* of a disk as the position of its center. The *medial axis* of an object is defined as the set of positions of disks tangent to the object in two or more points from the inside. Dotted lines in Figure 1 shows the medial axis of a convex polygon as an example. The medial axis of a convex polygon P is a tree whose leaves are the vertices of P. We represent the medial axis by a set (V_p, V_m, E_p, E_m) of point sets V_p and V_m, and segment sets E_p and E_m. V_p consists of the vertices of P, and V_m denotes the set of positions of disks tangent to P in more than two points. E_p denotes the set of segments between points in V_p and those in V_m, and E_m denotes the set of segments whose both ends are in V_m. For each segment $e \in E_p \cup E_m$, $v_1(e)$ and $v_2(e)$ denote the end points of e. When e is in E_p, $v_1(e)$ denotes the point in V_p, and $v_2(e)$ denotes the one in V_m. For example, $e_1 \in E_p$ in Figure 1 satisfies $v_1(e_1) = v_1$ and $v_2(e_1) = v_7$. If $e \in E_m$, we let $v_2(e)$ denote the end point on which a larger disk can be put than on $v_1(e)$. When $e \in E_m$, $E[e]$ denotes the set of segments in $E_p \cup E_m \setminus \{e\}$ that are connected to $v_1(e)$ after removing e from the medial axis. For $e_5 \in E_m$ in Figure 1, $v_1(e_5) = v_8$, $v_2(e_5) = v_9$, and $E[e_5] = \{e_1, e_2, e_3, e_4\}$ hold.

The medial axis of a convex polygon can be computed as shown in the following theorem.

Theorem 1 (Aggarwal et al. [1]). *The medial axis* (V_p, V_m, E_p, E_m) *of a convex polygon with* k *vertices can be computed in* $O(k)$ *time. In particular,* $|V_p| + |V_m| = O(k)$ *and* $|E_p| + |E_m| = O(k)$.

Let us observe basic properties of optimal solutions in the disk covering problem. We say that a set of located disks is *connected* if for any two disk D and D', there exists a sequence $(D_1 = D, D_2, \ldots, D_i = D')$ of disks in the set such that D_j and D_{j+1} overlap or are tangent for $j = 1, 2, \ldots, i-1$.

Theorem 2. *The disk covering problem with a convex polygon has an optimal solution consisting of connected disks located on the medial axis of the polygon.*

Proof. Let us consider an optimal solution of the disk covering problem. If the solution is not connected, it contains a connected set of disks which can move inside the polygon preserving relative positions of the disks. By repeating moving such sets, we can obtain another optimal solution which is connected.

Suppose that a disk D in the obtained optimal solution is not on the medial axis. That is to say, D is tangent to the polygon in exactly one point, or not tangent to the polygon. Then there exists another disk, call D', containing D and tangent to the polygon in two or more points. Replacing D by D' increases the covered area, which contradicts the optimality. □

Our algorithm locates disks only on the medial axis. Once the position of a disk on the medial axis is decided, its radius can be defined immediately so that the disk is maximal (i.e., no disk at the same position has larger radius without overlapping outside the polygon). Hence in the remainder of this paper, we concentrate on deciding positions of disks on the medial axis.

3 Dynamic programming to locate disks around a corner

In this section, we define a problem of locating disks only on a segment in E_p of the medial axis, and present a DP-based algorithm to solve it.

First of all, let us define the problem. Let l_1 and l_2 be two lines crossing at a point o with angle 2θ $(0 < \theta < \pi/2)$. Let D be a disk tangent to both l_1 and l_2, and the distance between its center c and o be 1. Figure 2 illustrates these definitions. The problem considered in this section asks to locate n disks $D_1, D_2, \ldots, D_n$ on the segment $\overline{oc}$ so that the area covered by $D_1, D_2, \ldots, D_n$ and D (i.e., $A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D)$) is maximized. The radii of D_i, $i = 1, 2, \ldots, n$ are set so that they are tangent to l_1 and l_2.

Define the coordinate of a disk on $\overline{oc}$ as the distance between its center and o. The radius of the disk with coordinate x is $x \sin\theta$. By this, $A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D)$ is a function on the coordinates $x_1, x_2, \ldots, x_n$ of $D_1, D_2, \ldots, D_n$. In the following, we denote $A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D)$ simply by $A(x_1, x_2, \ldots, x_n)$, or by $A(X)$ where $X = \{x_1, x_2, \ldots, x_n\}$.

Let $\mathrm{OPT} = \{D_1^*, D_2^*, \ldots, D_n^*\}$ be an optimal solution of the problem. We let x_i^* be the coordinate of D_i^* for $i = 1, 2, \ldots, n$, $X_n^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$, and A_n^* stand for $A(X_n^*)$. Without loss of generality, suppose that $0 < x_n^* < x_{n-1}^* < \cdots < x_1^* < 1$. For convenience, we sometimes let x_0^* denote 1.

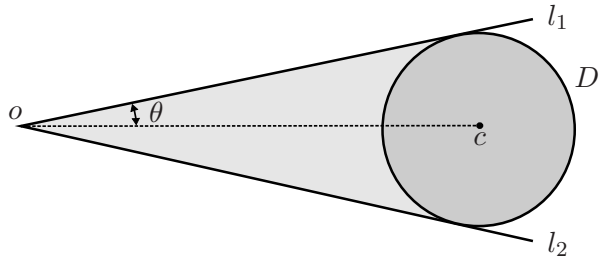

Figure 2: Definitions of o, l_1, l_2, θ, D and c.

3.1 Definition of our algorithm

Performance of our algorithm is controlled by a parameter ϵ where $0 < \epsilon < 0.5$. As ϵ is smaller, the approximation factor is better, but the algorithm needs more computation time. We define $f(n, \theta)$ as a lower bound of $(x_i^* - x_{i+1}^*)/x_i^*$ for all $i = 0, 1, \ldots n-1$, and $g(n, \theta)$ as a lower bound of x_n^*. These functions will be analyzed in Section 3.3.

Let

$$m = \left\lceil \frac{\log(g(n,\theta))}{\log(1 - \epsilon f(n,\theta))} \right\rceil, \tag{1}$$

and define m points on $\overline{oc}$ such that the j-th point has the coordinate

$$p_j := (1 - \epsilon f(n,\theta))^{j-1} \tag{2}$$

for $j = 1, 2, \ldots, m$. We call a point of coordinate in $\{p_1, p_2, \ldots, p_m\}$ *candidate point*. For $i = 1, 2, \ldots, n+1$ and $j = 1, 2, \ldots, m$, consider locating i disks at candidate points of coordinates in $\{p_j, p_{j+1}, \ldots, p_m\}$ such that one of them is located at p_j and the covered area is maximized. Let $X(i, j)$ be the set coordinates of the located disks in an optimal solution of the problem. Then $X(i, j)$ has a recursion formula

$$A(X(i,j)) = \max_{j+1 \le j' \le m} A(X(i-1, j') \cup \{p_j\}).$$

If the maximization in the right-hand side is attained by j^*, then $X(i, j) = X(i-1, j^*) \cup \{p_j\}$. Our algorithm computes $X(i, j)$ for all $i = 1, 2, \ldots, n+1$ and $j = 1, 2, \ldots, m$ by applying this recursion formula, and output $X(n+1, 1) \setminus \{p_1\}$ as an approximate solution to the original problem.

Algorithm CORNER

Input: Angle θ ($0 < \theta < \pi/2$), integer n, parameter ϵ ($0 < \epsilon < 0.5$)
Output: Set of n coordinates

```
Define m and p_j (j = 1, 2, ..., m) as (1) and (2)
X(1, j) := {p_j} for j = 1, 2, ..., m
for i = 2 to n + 1 do
    for j = 1 to m do
        Compute j* that attains max_{j+1 ≤ j' ≤ m} {A(X(i − 1, j') ∪ {p_j})}
        X(i, j) := X(i − 1, j*) ∪ {p_j}
    end for
end for
Output X(n + 1, 1) \ {p_1}
```

3.2 Performance analysis of Algorithm CORNER

Now we present the next theorem.

Theorem 3. *Algorithm CORNER achieves approximation factor* $(1-2\epsilon)(1-\epsilon f(n,\theta))$ *in* $O(nm^2) = O\left(n\lceil\log(g(n,\theta))/\log(1-\epsilon f(n,\theta))\rceil^2\right)$ *time.*

In Theorem 3, the approximation factor and the computation time depend on $f(n,\theta)$ and $g(n,\theta)$. The result is refined by analyzing these functions in Section 3.3. Here we prove Theorem 3.

Since Algorithm CORNER computes $X(i,j)$ in $O(m)$ time for each $i = 1,2,\ldots,n+1$ and $j = 1,2,\ldots,m$, it is easy to see that CORNER runs in $O(nm^2)$ time. Hence in the following, we discuss only the approximation factor.

From x_i^*, define x_i' as the unique coordinate p_j that satisfies $p_j \le x_i^* < p_{j-1}$. Recall that $(x_i^* - x_{i+1}^*)/x_i^* > f(n,\theta)$. Since $(1-\epsilon f(n,\theta))x_i^* \ge (1-f(n,\theta))x_i^* > x_{i+1}^*$, some p_j satisfies $x_{i+1}^* \le p_j < x_i^*$ for any i, which means that $x_i' \ne x_j'$ holds for $i \ne j$. Let D_i' denote the disk located at x_i', and let $X' = \{x_1', x_2', \ldots, x_n'\}$.

Lemma 1. *The solution computed by Algorithm CORNER covers the area larger than or equal to* $A(X')$.

Proof. Notice that Algorithm CORNER computes a solution covering the maximum area in which each disk is located at some candidate point. Since $X' \subseteq \{p_1, p_2, \ldots, p_m\}$, this completes the proof. □

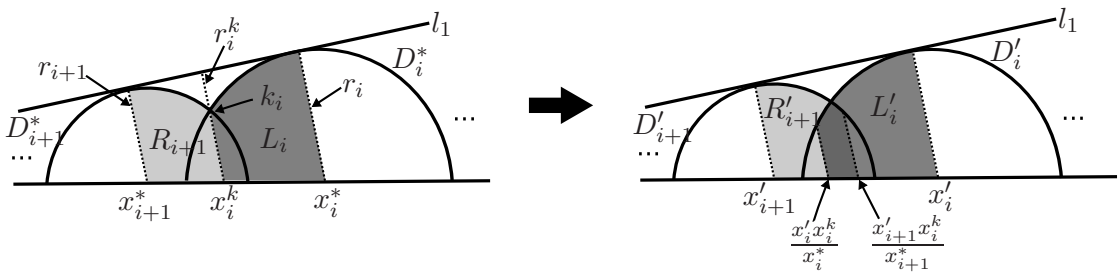

Figure 3: Definitions of R_i, L_i, L_i' and R_{i+1}'.

Now we estimate the ratio of $A(X')$ to $A(X^*)$. Since disks are symmetric about segment $\overline{oc}$, we discuss only the area above the segment. Call the intersection point of boundaries of D_i^* and D_{i+1}^* by k_i. Let r_i denote the line which is vertical to l_1 and passes the center of D_i. Let r_i^k denote the line which is vertical to l_1 and passes k_i where we let x_i^k denote the coordinate of the intersection point of r_i^k and $\overline{oc}$. By elementary calculation, we can observe that

$$x_i^k < \frac{x_i^* + x_{i+1}^*}{2}. \tag{3}$$

Lines $r_1, r_2, \ldots, r_n$ and $r_1^k, r_2^k, \ldots, r_{n-1}^k$ divide the upper-half of the area covered by $D_1^* \cup D_2^* \cup \cdots \cup D_n^* \cup D$ into several parts. Call the area between r_i and r_i^k by L_i for $i = 1,2,\ldots,n-1$,

and the area between r_i and r_{i-1}^k by R_i for $i = 2, 3, \ldots, n$. Define L_n (resp., R_1) as the upper-half of $D_n^* \setminus R_n$ (resp., $D_1^* \setminus L_1$). The left figure in Figure 3 illustrates these definitions. By the definition, it holds that

$$\frac{A(X^*) - A(D \setminus D_1^*)}{2} = \sum_{i=1}^{n} (A(L_i) + A(R_i)). \tag{4}$$

Scale down L_i by ratio x_i'/x_i^*, and move it in parallel so that its right-bottom point is at x_i'. Define L_i' as the area covered by the obtained object. Similarly, scale down R_i by ratio x_i'/x_i^*, and move it in parallel so that its left-bottom point is at x_i'. Define R_i' as the area covered by the obtained object. The right figure in Figure 3 explains these definitions. Note that R_{i+1}' and L_i' may overlap. Since these areas are covered by the upper-half of $(\cup_{i=1}^n D_i') \setminus (D \setminus D_1^*)$, we have

$$\frac{A(X') - A(D \setminus D_1^*)}{2} \geq \sum_{i=1}^{n} (A(L_i') + A(R_i')) - \sum_{i=1}^{n-1} A(L_i' \cap R_{i+1}'). \tag{5}$$

Lemma 2.

$$\frac{A(R_i')}{A(R_i)} \geq (1 - \epsilon f(n, \theta))^2 \quad \textit{for } i = 1, 2, \ldots, n,$$

and

$$\frac{A(L_n')}{A(L_n)} \geq (1 - \epsilon f(n, \theta))^2.$$

Proof. By the definition, $A(R_i')/A(R_i) = (x_i'/x_i^*)^2$ holds. Recall that x_i' is defined as p_j that satisfies $p_j \leq x_i^* < p_{j-1}$. Hence

$$\frac{x_i'}{x_i^*} \geq \frac{p_j}{p_{j-1}} = 1 - \epsilon f(n, \theta)$$

holds for all $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. Hence $A(R_i')/A(R_i) \geq (1 - \epsilon f(n, \theta))^2$. $A(L_i')/A(L_i) \geq (1 - \epsilon f(n, \theta))^2$ is also proven similarly. □

Lemma 3.

$$\frac{A(L_i') - A(L_i' \cap R_{i+1}')}{A(L_i)} \geq (1 - 2\epsilon)(1 - \epsilon f(n, \theta)) \quad \textit{for } i = 1, 2, \ldots, n-1.$$

Proof. Observe that the coordinate of the right-bottom point of R_{i+1}' is $x_{i+1}' x_i^k / x_{i+1}^*$ and that the coordinate of the left-bottom point of L_i' is $x_i' x_i^k / x_i^*$.

If $x_{i+1}'/x_{i+1}^* \leq x_i'/x_i^*$, L_i' and R_{i+1}' do not overlap, i.e., $A(L_i' \cap R_{i+1}') = 0$. In this case,

$$\frac{A(L_i') - A(L_i' \cap R_{i+1}')}{A(L_i)} = \frac{A(L_i')}{A(L_i)} \geq (1 - \epsilon f(n, \theta))^2 \geq (1 - 2\epsilon)(1 - \epsilon f(n, \theta))$$

can be proven as in Lemma 2. Hence in what follows, we suppose that $x_{i+1}'/x_{i+1}^* > x_i'/x_i^*$

Draw a line vertical to l_1 so that it intersects segment $\overline{oc}$ at the right-bottom point of R_{i+1}'. By the line, L_i' is divided into two parts as shown in Figure 4. Let G_i stand for the left one, and H_i stand for the right one. Obviously $A(G_i) + A(H_i) = A(L_i')$. $A(L_i' \cap R_{i+1}') \leq A(G_i)$ holds because $L_i' \cap R_{i+1}' \subseteq G_i$. Thus

$$\frac{A(L_i') - A(L_i' \cap R_{i+1}')}{A(L_i)} \geq \frac{A(L_i') - A(G_i)}{A(L_i)} = \frac{A(H_i) A(L_i')}{A(L_i)(A(G_i) + A(H_i))}.$$

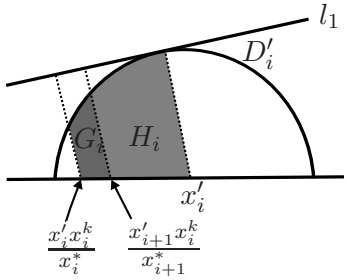

Figure 4: Definitions of G_i and H_i.

$A(L'_i)/A(L_i) \geq (1-\epsilon f(n,\theta))^2$ can be proven as in Lemma 2. Hence it suffices to show that

$$\frac{A(H_i)}{A(G_i)+A(H_i)} \geq \frac{1-2\epsilon}{1-\epsilon f(n,\theta)}. \tag{6}$$

The length of the bottom side of G_i is $x'_{i+1}x^k_i/x^*_{i+1} - x'_i x^k_i/x^*_i$, and the length of the bottom side of H_i is $x'_i - x'_{i+1}x^k_i/x^*_{i+1}$. $A(G_i)/A(H_i)$ is at most the ratio of the former length to the latter one. It means that

$$\frac{A(H_i)}{A(G_i)+A(H_i)} \geq 1 - \frac{(x'_{i+1}/x^*_{i+1} - x'_i/x^*_i)x^k_i}{x'_i - x'_i x^k_i/x^*_i}. \tag{7}$$

Let us estimate the second term of the right-hand side of (7).

Since $1-\epsilon f(n,\theta) \leq x'_{i+1}/x^*_{i+1} \leq 1$ for all i,

$$\left(\frac{x'_{i+1}}{x^*_{i+1}} - \frac{x'_i}{x^*_i}\right) x^k_i \leq (1-(1-\epsilon f(n,\theta)))\, x^k_i = \epsilon f(n,\theta) x^k_i. \tag{8}$$

On the other hand, (3) indicates that

$$x'_i - \frac{x'_i x^k_i}{x^*_i} > x'_i - \frac{x'_i}{x^*_i}\frac{x^*_i + x^*_{i+1}}{2} = \frac{x'_i}{2}\left(\frac{x^*_i - x^*_{i+1}}{x^*_i}\right) > \frac{x'_i}{2} f(n,\theta). \tag{9}$$

By (8) and (9),

$$\frac{(x'_{i+1}/x^*_{i+1} - x'_i/x^*_i)x^k_i}{x'_i - x'_i x^k_i/x^*_i} < \frac{\epsilon f(n,\theta) x^k_i}{x'_i f(n,\theta)/2} = \frac{2\epsilon x^k_i}{x'_i}. \tag{10}$$

By (3),

$$\frac{2\epsilon x^k_i}{x'_i} < \epsilon\left(\frac{x^*_i}{x'_i} + \frac{x^*_{i+1}}{x'_i}\right) = \epsilon\frac{x^*_i}{x'_i}\left(2 - \frac{x^*_i - x^*_{i+1}}{x^*_i}\right). \tag{11}$$

By the definition of $f(n,\theta)$,

$$\epsilon\frac{x^*_i}{x'_i}\left(2 - \frac{x^*_i - x^*_{i+1}}{x^*_i}\right) \leq \epsilon\frac{2-f(n,\theta)}{1-\epsilon f(n,\theta)}. \tag{12}$$

Combining (7), (10), (11), and (12) proves (6). □

Lemma 4.

$$\frac{A(X')}{A(X^*)} \geq (1-2\epsilon)(1-\epsilon f(n,\theta)).$$

Proof. (4) and (5) indicate that

$$\frac{A(X')}{A(X^*)} \geq \min\left\{\frac{A(R'_i)}{A(R_i)}, \frac{A(L'_j)-A(L'_j \cap R'_{j+1})}{A(L_j)}, \frac{A(L'_n)}{A(L_n)} \mid i=1,2,\ldots,n, j=1,2,\ldots,n-1\right\}.$$

Lemmas 2 and 3 show that the right-hand side of this inequality is at least $(1-2\epsilon)(1-\epsilon f(n,\theta))$. This proves the lemma. □

Note that Theorem 3 is immediate from Lemmas 1 and 4.

3.3 Analysis of $f(n,\theta)$ and $g(n,\theta)$

Here we analyze how $f(n,\theta)$ and $g(n,\theta)$ depend on n and θ. Recall that $A^*_n = \max\{A(D_1 \cup D_2 \cup \cdots D_n \cup D) \mid 0 < x_n < x_{n-1} < \cdots < x_1 < 1\}$ where x_i denotes the coordinate of disk D_i for $i = 1,2,\ldots,n$. Define $f_{\text{int}}(x)$ as the area of intersection of D and the disk at coordinate x. By elementary calculation, we can see that $\frac{\partial}{\partial x} f_{\text{int}}(x)$ is concave and monotone non-decreasing for $(1-\sin\theta)/(1+\sin\theta) \leq x \leq 1$ and any fixed θ.

A key observation in our analysis is that relative positions of disks in optimal solutions are independent from the position of disk D. That is to say, x^*_i/x^*_j $(1 \leq i < j \leq n)$ does not change even if we change the coordinate of D. Hence if D_i is located at x_i, then each D_j, $j = i+1, i+2, \ldots, n$ must be located at $x^*_j x_i / x^*_i$ to maximize the covered area. When the positions of D_j, $j = i+1, i+2, \ldots, n$ are fixed as this, $A((D_i \setminus D_{i-1}) \cup D_{i+1} \cup \cdots \cup D_n) = x_i^2 A^*_{n-i} - x_{i-1}^2 f_{\text{int}}(x_i/x_{i-1})$ holds. Therefore the relative position of D^*_i to D^*_{i-1}, i.e., x^*_i/x^*_{i-1}, attains $\max\{x^2 A^*_{n-i} - f_{\text{int}}(x) \mid (1-\sin\theta)/(1+\sin\theta) \leq x \leq 1\}$. $x^2 A^*_{n-i} - f_{\text{int}}(x)$ is a concave function on x for $(1-\sin\theta)/(1+\sin\theta) \leq x \leq 1$. Therefore

$$2A^*_{n-i}\frac{x^*_i}{x^*_{i-1}} = \frac{\partial}{\partial x} f_{\text{int}}(x)\bigg|_{x=\frac{x^*_i}{x^*_{i-1}}} \tag{13}$$

holds for all $i = 1,2,\ldots,n$.

First, we present an upper-bound of A^*_n.

Lemma 5. *Define*

$$\tilde{A}_n = \begin{cases} (n+1)\pi\sin^2\theta & \text{if } n \leq \frac{\cos\theta-(0.5\pi-\theta)\sin\theta}{4\sin\theta}, \\ \sin\theta\cos\theta + (\theta+0.5\pi)\sin^2\theta - \frac{(\cos\theta-(0.5\pi-\theta)\sin\theta)^4}{2048n^3\sin^2\theta} & \text{otherwise.} \end{cases}$$

Then $A^*_n \leq \tilde{A}_n$ holds.

Proof. We define an object F as the union of all disks of radius $\sin\theta$ whose centers are on a segment of length $(\cos\theta + (\theta - 0.5\pi)\sin\theta)/2$ (see Figure 5). Notice that $A(F)$ is equal to the area surrounded by l_1, l_2, and D, which is colored by dark and light gray in Figure 2, i.e., $A(F) = \sin\theta\cos\theta + (\theta+0.5\pi)\sin^2\theta$. First, we show that A^*_n is at most the maximum area covered by $n+1$ disks inside F. Next, we prove that $\tilde{A}_n$ is an upper-bound of the area covered by $n+1$ disks inside F.

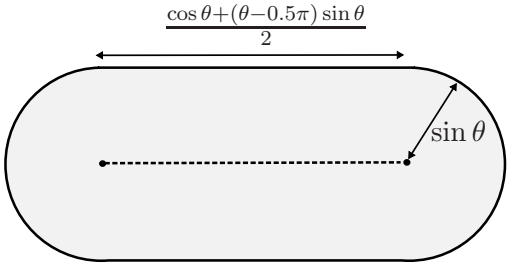

Figure 5: Object F.

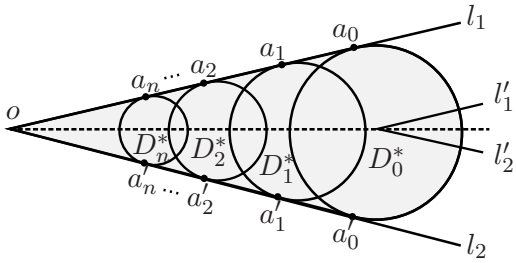

Figure 6: Definitions of a_i and a'_i $(i = 0, 1, 2, \ldots, n)$, l'_1 and l'_2.

Let D^*_0 denote D for convenience. For $i = 0, 1, 2, \ldots, n$, call the point on l_1 (resp., l_2) to which disk D^*_i is tangent by a_i (resp., a'_i). From the center of D^*_0, draw the segment l'_i parallel to l_i for $i = 1$ and 2 as in Figure 6. By l'_1, l'_2 and $\overline{oc}$, the gray area in Figure 6 is divided into three parts as in the upper-left figure of Figure 7. Rearrange them as in the lower-left figure of Figure 7.

Then put $n+1$ disks $D'_0, D'_1, \ldots, D'_n$ of radius $\sin\theta$ so that D'_i is tangent to l_1 at a_i and to l_2 at a'_i. In the lower-center figure of Figure 7, their outlines are represented by dotted lines. Observe that these $n+1$ disks cover the area originally covered by $D^*_0 \cup D^*_1 \cup \cdots \cup D^*_n$. Moreover, they also cover outside the object, which is represented by dark gray in the lower-center figure of Figure 7. We add this dark gray area to the object.

Transform the shape of the area to the left of D'_n inside the object as in the lower-right figure of Figure 7, and call the obtained object F'. We can see that $A(D'_1 \cup D'_2 \cup \cdots \cup D'_{n+1})/A(F')$ is larger than or equal to the ratio of $A(D^*_1 \cup D^*_2 \cup \cdots \cup D^*_n \cup D)$ to the gray area in Figure 6.

Notice that $A(F')$ may be larger than $A(F)$ since the dark gray area in the lower-center figure of Figure 7, which is not contained by the object originally, is added to F'. Divide F' into three parts as in the upper-right figure of Figure 7, and scale down the width of only the middle part so the area of entire the object becomes $(0.5\pi + \theta)\sin^2\theta$. F is the obtained object. It is possible to locate $n+1$ disks of radius $\sin\theta$ inside F so that they cover the area originally covered by $D'_0 \cup D'_1 \cup \cdots \cup D'_n$. This means that the maximum area covered by $n+1$ disks inside F is larger than $A(D^*_0 \cup D^*_1 \cup \cdots \cup D^*_n)$.

Next, let us estimate the maximum area covered by $n+1$ disks inside F. We have mentioned that $\frac{\partial}{\partial x} f_{\text{int}}$ is monotone non-decreasing. Hence f_{int} is a convex function. By this convexity, it is optimal to put $n+1$ disks with equal intervals. In other words, the distance between centers of two adjacent disks is $(\cos\theta + (\theta - 0.5\pi)\sin\theta)/2n$.

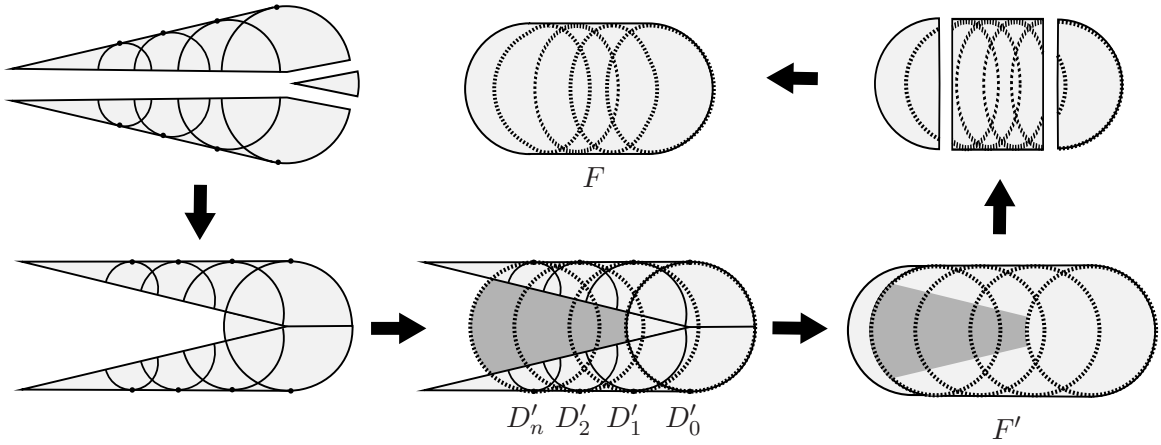

Figure 7: Construction of F.

Consider the case where $2\sin\theta \le (\cos\theta + (\theta - 0.5\pi)\sin\theta)/2n$. In this case, we can put $n+1$ disks of radius $\sin\theta$ without overlapping inside F. The area covered by such disks is $(n+1)\pi\sin^2\theta$. Thus we are done in this case. Suppose the otherwise. In this case, the area inside F not covered by the $n+1$ disks is geometrically separated into $2n$ parts. In each part, it is possible to put a disk of radius $r = (\cos\theta - (0.5\pi - \theta)\sin\theta)^2)/(64n^2\sin\theta)$. Hence the area covered by $n+1$ disks is at most

$$A(F) - 2n\pi r^2 = \sin\theta\cos\theta + (\theta + 0.5\pi)\sin^2\theta - \frac{(\cos\theta - (0.5\pi - \theta)\sin\theta)^4}{2048n^3\sin^2\theta},$$

which completes the proof. □

As we see in the next theorem, $f(n,\theta)$ can be defined from $\tilde{A}_n$.

Theorem 4. *Define $f(n,\theta)$ as*

$$\frac{2\sin\theta\{2\cos\theta + (\pi + 2\theta)\sin\theta\} - 4\tilde{A}_{n-1}}{(1+\sin\theta)\{2\cos\theta + (\pi + 2\theta)\sin\theta\} - 4\tilde{A}_{n-1}}$$

where $\tilde{A}_{n-1}$ is defined as in Lemma 5. Then it satisfies

$$\frac{x^*_{n-1} - x^*_n}{x^*_{n-1}} \ge \frac{x^*_{n-2} - x^*_{n-1}}{x^*_{n-2}} \ge \cdots \ge \frac{x^*_0 - x^*_1}{x^*_0} \ge f(n,\theta) > 0.$$

Proof. It is obvious that A^*_{n-i} is monotone decreasing on i. As we have already mentioned, $\frac{\partial}{\partial x} f_{\text{int}}(x)$ is monotone non-decreasing on x. Hence by (13), x^*_i/x^*_{i-1} is monotone non-increasing on i. This proves that $(x^*_{i-1} - x^*_i)/x^*_{i-1} \ge (x^*_{i-2} - x^*_{i-1})/x^*_{i-2}$ for any $2 \le i \le n$.

Now we prove $(x^*_0 - x^*_1)/x^*_0 = 1 - x^*_1 \ge f(n,\theta)$. By (13), $x^*_1/x^*_0 = x^*_1$ is the solution to the equation

$$2A^*_{n-1}x = \frac{\partial}{\partial x} f_{\text{int}}(x).$$

Since $\tilde{A}_{n-1} \ge A^*_{n-1}$ and $\frac{\partial}{\partial x} f_{\text{int}}(x)$ is monotone non-decreasing on x, x^*_1 is smaller than or equal to the solution of

$$2\tilde{A}_{n-1}x = \frac{\partial}{\partial x} f_{\text{int}}(x). \tag{14}$$

Elementary calculations show that $\frac{\partial}{\partial x} f_{\text{int}}(x)\big|_{x=\frac{1-\sin\theta}{1+\sin\theta}} = 0$ and $\frac{\partial}{\partial x} f_{\text{int}}(x)\big|_{x=1} = 2\sin\theta\cos\theta + (2\theta+\pi)\sin^2\theta$ hold. Define $\tilde{f} : [\frac{1-\sin\theta}{1+\sin\theta}, 1] \to \Re$ as the linear function such that $\tilde{f}(\frac{1-\sin\theta}{1+\sin\theta}) = \frac{\partial}{\partial x} f_{\text{int}}(x)\big|_{x=\frac{1-\sin\theta}{1+\sin\theta}}$ and $\tilde{f}(1) = \frac{\partial}{\partial x} f_{\text{int}}(x)\big|_{x=1}$, i.e.,

$$\tilde{f}(x) = \frac{(1+\sin\theta)(2\cos\theta + (2\theta+\pi)\sin\theta)}{2}x + \frac{(\sin\theta - 1)(2\cos\theta + (2\theta+\pi)\sin\theta)}{2}$$

for $(1-\sin\theta)(1+\sin\theta) \le x \le 1$. Since $\frac{\partial}{\partial x} f_{\text{int}}(x)$ is concave for $(1-\sin\theta)(1+\sin\theta) \le x \le 1$, the solution of

$$2\tilde{A}_{n-1}x = \tilde{f}(x) \tag{15}$$

is at least the solution of (14). It is easy to see that $1 - f(n,\theta)$ satisfies (15). Therefore $x_1^* \le 1 - f(n,\theta)$, as required. □

Theorem 5. *Define $g(n,\theta)$ as $\left(\frac{1-\sin\theta}{1+\sin\theta}\right)^n$. Then it satisfies $x_n^* \ge g(n,\theta) > 0$.*

Proof. We show that $x_i^* \ge \left(\frac{1-\sin\theta}{1+\sin\theta}\right)^i$ holds for any $1 \le i \le n$ by the induction on i. This proves the theorem.

Let us consider the disk which is tangent to C_i^* from the left side of C_i^*. The coordinate of this disk is $x_i^*\left(\frac{1-\sin\theta}{1+\sin\theta}\right)$. C_{i+1}^* is located between this disk and C_i^* by Theorem 2. Hence $x_{i+1}^* \ge x_i^*\left(\frac{1-\sin\theta}{1+\sin\theta}\right)$, which and the induction hypothesis prove the required inequality for $i+1$. □

Recall that computation time of Algorithm CORNER is $O(nm^2) = O\left(n\left\lceil\frac{\log(g(n,\theta))}{\log(1-\epsilon f(n,\theta))}\right\rceil^2\right)$. By Theorem 5, $-\log(g(n,\theta)) = n\log\left(\frac{1+\sin\theta}{1-\sin\theta}\right)$ holds. By Theorem 4,

$$\begin{aligned}
-\frac{1}{\log(1-\epsilon f(n,\theta))} &< \frac{1}{\epsilon f(n,\theta)} \\
&= \frac{1}{\epsilon}\cdot\frac{(1+\sin\theta)(2\cos\theta + (\pi+2\theta)\sin\theta) - 4\tilde{A}_{n-1}}{2\sin\theta(2\cos\theta + (\pi+2\theta)\sin\theta) - 4\tilde{A}_{n-1}} \\
&< \frac{1}{\epsilon}\cdot\frac{(1+\sin\theta)(2\cos\theta + (\pi+2\theta)\sin\theta)}{2\sin\theta(2\cos\theta + (\pi+2\theta)\sin\theta) - 4\tilde{A}_{n-1}}
\end{aligned}$$

holds. The definition of $\tilde{A}_{n-1}$ implies that the right-most hand side is at most

$$\frac{1}{\epsilon}\cdot\frac{(1+\sin\theta)(2\cos\theta + (\pi+2\theta)\sin\theta)}{(4-\pi)\sin\theta\cos\theta + (0.5\pi^2 + (2-\theta)\pi + 4\theta)\sin^2\theta}$$

if $n \le \frac{\cos\theta - (0.5\pi-\theta)\sin\theta}{4\sin\theta}$, and

$$\frac{1}{\epsilon}\cdot\frac{512n^3\sin^2\theta(1+\sin\theta)(2\cos\theta + (\pi+2\theta)\sin\theta)}{(\cos\theta - (0.5\pi-\theta)\sin\theta)^4}$$

otherwise. With these, the computation time of Algorithm CORNER is $O(n^9 h(\theta)/\epsilon^2)$, where $h(\theta)$ is a function on θ which approaches $+\infty$ as θ approaches $\pi/2$.

Corollary 1. *Algorithm CORNER achieves approximation factor $(1-2\epsilon)(1-\epsilon)$ in $O(n^9 h(\theta)/\epsilon^2)$ time.* □

4 Dynamic programming to locate disks between two fixed disks

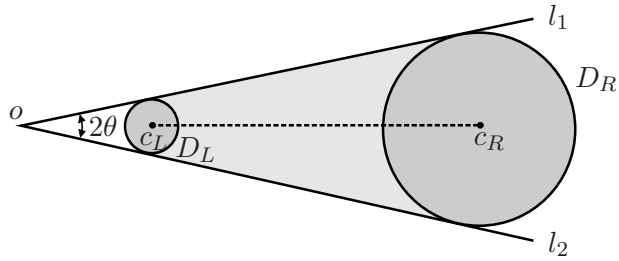

Figure 8: Definitions of o, l_1, l_2, θ, D_L, D_R, c_L and c_R.

Define o, l_1, l_2 and θ as in Section 3. Consider two disks tangent to both l_1 and l_2. Let D_L be the left one, and D_R be the right one. Call the center of D_L by c_L, and the center of D_R by c_R. In this section, we consider a problem of locating n disks $D_1, D_2, \ldots, D_n$ on segment $\overline{c_L c_R}$ where the radii of the disks are set so that they are tangent to l_1 and l_2. The objective is to maximize $A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D_L \cup D_R)$.

We define the coordinate of a disk on segment $\overline{c_L c_R}$ as the distance between its center and o, supposing that the coordinate of D_R is 1. Let x_L denote the coordinate of D_L. Let $D_1^*, D_2^*, \ldots, D_n^*$ stand for the disks in an optimal solution while X^* denotes the set of coordinates of these disks.

Now we described our algorithm.

Algorithm TWODISKS

Input: Angle θ $(0 < \theta < \pi/2)$, integer n, real x_L $(0 < x_L < 1)$, parameter ϵ $(0 < \epsilon < 0.5)$

Output: Set of n coordinates

1: Execute Algorithm CORNER with θ, n and ϵ, and let $X = \{x_1, x_2, \ldots, x_n\}$ be the obtained output
2: **for** $i = 1, 2, \ldots, x_n$ **do**
3: If $x_i < x_L$, set $x_i := x_L$
4: **end for**
5: Output X

The computational time of Algorithm TWODISKS is same with that of Algorithm CORNER. In what follows, we discuss the approximation factor of Algorithm TWODISKS. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the solution obtained by Algorithm TWODISKS. Define $A(X)$ as $A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D_L \cup D_R)$ where D_i denote the disk of coordinate x_i for $i = 1, 2, \ldots, n$.

First, let us consider the case where the obtained solution contains no disk overlapping D_L. In this case, the solution is same with the solution output by Algorithm CORNER. Hence Theorem 3 tells that

$$A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D_R) \geq (1 - 2\epsilon)(1 - \epsilon f(n, \theta)) A(D_1^* \cup D_2^* \cup \cdots \cup D_n^* \cup D_R^*).$$

Obviously $A(D_L) \geq A(D_L \setminus D_n^*)$. Therefore,

$$\begin{aligned} A(X) &= A(D_L) + A(D_1 \cup D_2 \cup \cdots \cup D_n \cup D_R) \\ &\geq A(D_L \setminus D_n^*) + (1-2\epsilon)(1-\epsilon f(n,\theta))A(D_1^* \cup D_2^* \cup \cdots \cup D_n^* \cup D_R^*) \\ &\geq (1-2\epsilon)(1-\epsilon f(n,\theta))A(X^*). \end{aligned}$$

Thus in this case, Algorithm TWODISKS achieves the same approximation factor with Algorithm CORNER.

In what follows, let us assume that some disks in $D_1, D_2, \ldots, D_n$ overlap D_L. Let $\bar{A}$ denote the area colored by dark and light gray in Figure 8. Since $\bar{A}$ contains the area covered by an optimal solution, $\bar{A} \geq A(X^*)$ holds. We estimate $A(X)/\bar{A}$ for bounding $A(X)/A(X^*)$.

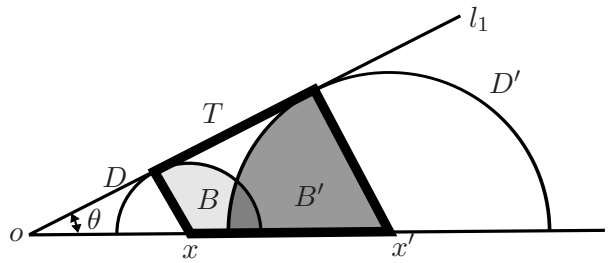

Figure 9: Definitions of D, D', T, B and B'.

Lemma 6. *If some disks in $D_1, D_2, \ldots, D_n$ overlap D_L, then $A(X)/\bar{A} \geq 0.78$ holds.*

Proof. Now every two adjacent disks in $D_1, D_2, \ldots, D_n, D_L$ and D_R overlap or are tangent. Let us consider two adjacent disks D and D' on segment $\overline{c_L c_R}$. We let x (resp., x') denote the coordinate of D (resp., D'). We only discuss the area above $\overline{c_L c_R}$. From the centers of D and D', draw the lines vertical to l_1. Let T denote the trapezoid whose sides are these two lines, l_1, and $\overline{c_L c_R}$. The outline of this trapezoid is represented by bold lines in Figure 9. Let B (resp., B') denote $T \cap D$ (resp., $T \cap D'$). See Figure 9 for these definitions. We show that

$$\frac{A(B \cup B')}{A(T)} \geq 0.78. \tag{16}$$

By applying this argument to each two adjacent disks in $D_1, D_2, \ldots, D_n, D_L$ and D_R, we obtain the required inequality.

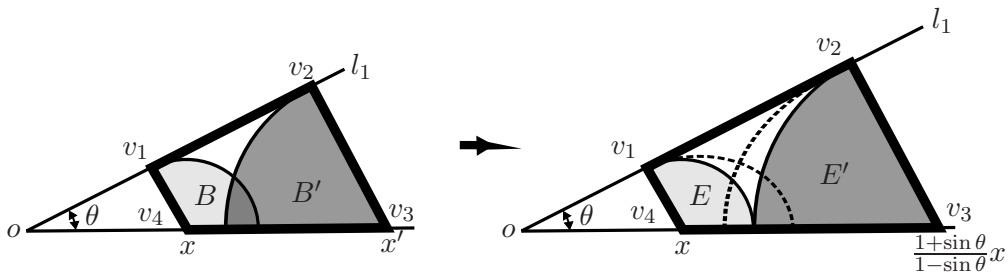

Figure 10: Stretching T.

We first observe that it suffices to prove (16) only for the case where D and D' are tangent. Call the vertices of trapezoid T by v_1, v_2, v_3 and v_4 as in the left figure of Figure 10. The distance between v_1 and v_2 is $(x' - x)\cos\theta$. Stretch T in direction along l_1 so that the distance between v_1 and v_2 becomes $2x\sin\theta\cos\theta/(1-\sin\theta)$. Notice that the stretching does not change the ratio of $A(B \cup B')$ to $A(T)$. By this operation, the coordinate of v_3 becomes $x(1+\sin\theta)/(1-\sin\theta)$. Consider the two disks the centers of which are respectively on v_3 and on v_4 after stretching while we let E (resp., E') denote the area in T covered by the former (resp., latter) disk. Notice that these disks are tangent. The area covered by $E \cup E'$ are originally covered by $B \cup B'$. Hence $A(B \cup B')/A(T) \geq A(E \cup E')/A(T)$. This tells that if (16) holds under the assumption that D and D' are tangent, then it also holds under the assumption that D and D' are not tangent.

Now we suppose that D and D' are tangent. In this case, $x' = x(1+\sin\theta)/(1-\sin\theta)$. Then we know an exact representation of each $A(T)$, $A(B)$, and $A(B')$ as functions on x and θ, from which we have

$$\frac{A(B \cup B')}{A(T)} = \frac{A(B) + A(B')}{A(T)} = \frac{\pi + \pi\sin^2\theta - 4\theta\sin\theta}{4\cos\theta}.$$

This function is monotone increasing for $0 < \theta < \pi/2$, and it returns the minimum value $\pi/4 > 0.78$ when $\theta = 0$. □

Theorem 6. *Algorithm TWODISKS achieves approximation factor* $\min\{0.78, (1-2\epsilon)(1-\epsilon)\}$ *in* $O(n^9 h(\theta)/\epsilon^2)$ *time.* □

5 Algorithm for the disk covering problem

In this section, putting algorithms in Sections 3 and 4 together, we design an algorithm to the disk covering problem. Performance of this algorithm is controlled by two parameters $0 < \epsilon < 1/2$ and $0 < \delta < 1$.

Let P be a convex polygon with k vertices. Let (V_p, V_m, E_p, E_m) denote the medial axis of P, and $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ be a solution consisting of maximal disks on the medial axis. For each $e \in E_p \cup E_m$, let $v_0(e)$ be the intersecting point of two lines obtained by extending the two sides of P to which maximal disks on e are tangent, and $2\theta_e$ be the angle of these two lines ($0 < \theta_e < \pi/2$). Notice that $v_0(e)$ coincides with $v_1(e)$ if $e \in E_p$. We define the coordinate of a point on e as the distance from $v_0(e)$. Let $y_i(e)$ denote the coordinate of $v_i(e)$ for $i = 1$ and 2. When $e \in E_p$, a disk on e in $\mathcal{D}$ is called the *end disk* of e if it has the largest coordinate among those on e in $\mathcal{D}$. When $e \in E_m$, a disk in $\mathcal{D}$ on e is called an *end disk* of e if it has the smallest or largest coordinate among those on e in $\mathcal{D}$. $e \in E_m$ may have two end disks while $e \in E_p$ has at most one end disk.

Once allocations of disks to segments in $E_p \cup E_m$ and locations of all end disks are decided, the disk covering problem can be reduced to the problems discussed in Sections 3 and 4. Our algorithm proposed in this section is based on this observation.

We define an *allocation* of n as a function $u : E_p \cup E_m \to \mathcal{N}_+$ such that $\sum_{e \in E_p \cup E_m} u(e) = n$ where $\mathcal{N}_+$ denotes the set of non-negative integers. Define $\mathcal{U}_n$ as the set of all allocations of n.

Let $e \in E_p \cup E_m$. For $i = 0, 1, 2, \ldots, \lceil 2/(1-\delta) \rceil$, define $p_e(i)$ as the point on e of coordinate

$$\max\left\{ y_1(e),\ y_2(e) - \frac{2i(1-\delta)y_2(e)\sin\theta_e}{(1+\sin\theta_e)^2} \right\}.$$

Define $\mathcal{Z}$ as the set of all functions $z : E_p \cup E_m \to \{0, 1, 2, \ldots, \lceil 2/(1-\delta) \rceil\}$.

For $e \in E_m$, we also define $p'_e(i)$ as the point on e of coordinate

$$\min \left\{ y_2(e),\ y_1(e) + \frac{2i(1-\delta)y_1(e)\sin\theta_e}{1-\sin^2\theta_e} \right\}$$

for $i = 0, 1, \ldots, \lceil 2/(1-\delta) \rceil$. Define $\mathcal{Z}'$ as the set of all functions $z' : E_m \to \{0, 1, \ldots, \lceil 2/(1-\delta) \rceil\}$. We call $p_e(i)$ ($e \in E_p \cup E_m$, $i = 0, 1, \ldots, \lceil 2/(1-\delta) \rceil$) and $p'_e(i)$ ($e \in E_m$, $i = 0, 1, \ldots, \lceil 2/(1-\delta) \rceil$) by *candidate points*.

Now we describe entire our algorithm. We let $A(\mathcal{D})$ denote $A(\cup_{D \in \mathcal{D}} D)$ for a set $\mathcal{D}$ of located disks.

Algorithm POLYGON

Input: Convex polygon P, integer n, parameters ϵ ($0 < \epsilon < 0.5$) and δ ($0 < \delta < 1$)

Output: Set of n locations on the medial axis of P

$\mathcal{D} := \emptyset$
for each set of $u \in \mathcal{U}_n$,$z \in \mathcal{Z}$ and $z' \in \mathcal{Z}'$ **do**
  $\mathcal{D}' := \emptyset$
  for each $e \in E_p$ **do**
    By using Algorithm CORNER, locate $n(e)$ disks on e so that the end disk is at $p_e(z(e))$, and add those locations to $\mathcal{D}'$
  end for
  for each $e \in E_m$ **do**
    if $n(e') = 0$ for all $e' \in E[e]$ **then**
      By using Algorithm CORNER, locate $n(e)$ disks on e so that a disk is at $p_e(z(e))$ and the other disks are at points of smaller coordinate than $p_e(z(e))$; If some disks are located at points of smaller coordinate than $y_1(e)$, locate them at arbitrary positions on e; Add those locations to $\mathcal{D}'$
    else
      By using Algorithm TWODISKS, locate $n(e)$ disks on e so that the end disks are at $p_e(z(e))$ and $p'_e(z'(e))$, and add those locations to $\mathcal{D}'$
    end if
  end for
  if $A(\mathcal{D}) < A(\mathcal{D}')$ **then**
    $\mathcal{D} := \mathcal{D}'$
  end if
end for
Output $\mathcal{D}$

Theorem 7. *Let θ be the half of the maximum inner angle of P. Algorithm POLYGON achieves approximation factor* $\min\left\{0.78\delta^2, \delta^2(1-\epsilon)(1-2\epsilon)\right\}$ *in* $O\left(k2^k n^{k+9} h(\theta)/(\epsilon^2(1-\delta)^k)\right)$ *time.*

Proof. Algorithm POLYGON enumerates all combinations of $u \in \mathcal{U}_n$, $z \in \mathcal{Z}$ and $z' \in \mathcal{Z}'$, the number of which is $O(2^k n^k/(1-\delta)^k)$. For each combination, it calls Algorithm CORNER at most $|E_p| + |E_m|$ times and Algorithm TWODISKS at most $|E_m|$ times. By Theorem 1, $|E_p| + 2|E_m| = O(k)$. Thus we can give the computation time.

Now we discuss the approximation factor. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ denote a set of disks outputted by Algorithm POLYGON, and $\mathcal{D}^* = \{D_1^*, D_2^*, \ldots, D_n^*\}$ denote an optimal set of disks. We estimate $A(\mathcal{D})/A(\mathcal{D}^*)$.

Define $\mathcal{D}' = \{D_1', D_2', \ldots, D_n'\}$ as a set of n disks that covers maximum area among those in which every end disk is on some candidate point or overlaps no disks on the other segments in $E_p \cup E_m$. Notice that Algorithm POLYGON approximates such sets by deciding locations of non-end disks with Algorithms CORNER and TWODISKS. Hence

$$\frac{A(\mathcal{D})}{A(\mathcal{D}')} \geq \min\{0.78, (1-2\epsilon)(1-\epsilon f(n,\theta))\}$$

holds by Theorems 3 and 6.

We now show that

$$\frac{A(\mathcal{D}')}{A(\mathcal{D}^*)} \geq \delta^2.$$

For each $i = 1, 2, \ldots, n$, define D_i'' as the disk which is obtained by scaling down D_i^* δ times and which is located at the same position with D_i^*. Then the set $\mathcal{D}''$ of such disks satisfies

$$A(\mathcal{D}'') \geq \delta^2 A(\mathcal{D}^*).$$

In the following, we prove that $A(\mathcal{D}') \geq A(\mathcal{D}'')$.

Let D_i^* be the disk in $\mathcal{D}^*$ that has the largest coordinate x_i^* among those on $e \in E_p \cup E_m$. By Theorem 2, $y_2(e)(1-\sin\theta_e)/(1+\sin\theta_e) \leq x_i^* \leq y_2(e)$. ($y_2(e)(1-\sin\theta_e)/(1+\sin\theta_e)$ is the coordinate of the maximal disk tangent to the disk on $v_2(e)$.) A maximal disk D on e contains the area covered by the disk D_i'' obtained from D_i^* if D is located at a position of the coordinate in $[x_i^*(1+\delta\sin\theta_e)/(1+\sin\theta_e), x_i^*(1-\delta\sin\theta_e)/(1-\sin\theta_e)]$. Observe that the length of this interval is

$$\begin{aligned}
\frac{1-\delta\sin\theta_e}{1-\sin\theta_e}x_i^* - \frac{1+\delta\sin\theta_e}{1+\sin\theta_e}x_i^* &= \frac{2(1-\delta)\sin\theta_e}{1-\sin^2\theta_e}x_i^* \\
&\geq \frac{2(1-\delta)\sin\theta_e}{1-\sin^2\theta_e}\cdot\frac{1-\sin\theta_e}{1+\sin\theta_e}l_2(e) \\
&= \frac{2(1-\delta)l_2(e)\sin\theta_e}{(1+\sin\theta_e)^2}.
\end{aligned}$$

Hence there exists $j \in \{0, 1, \ldots, \lceil 2/(1-\delta)\rceil\}$ such that the maximal disk on $p_e(j)$ contains the area covered by D_i''. Accordingly we can replace such D_i'' by a maximal disk on some candidate point.

In addition, consider the disk D_i^* in $\mathcal{D}^*$ that has the smallest coordinate x_i^* among those on $e \in E_m$. If D_i^* overlaps some disk on other segments in $E_p \cup E_m$, then $l_1(e) \leq x_i^* \leq l_1(e)(1+\sin\theta_e)/(1-\sin\theta_e)$. As stated above, a maximal disk D on e contains the area covered by D_i'' if the coordinate of D is in $[x_i(1+\delta\sin\theta_e)/(1+\sin\theta_e), x_i(1-\delta\sin\theta_e)/(1-\sin\theta_e)]$. The length of this interval satisfies

$$\frac{1-\delta\sin\theta_e}{1-\sin\theta_e}x_i^* - \frac{1+\delta\sin\theta_e}{1+\sin\theta_e}x_i^* = \frac{2(1-\delta)\sin\theta_e}{1-\sin^2\theta_e}x_i^* \geq \frac{2(1-\delta)l_1(e)\sin\theta_e}{1-\sin^2\theta_e}.$$

Hence there exists $j \in \{0, 1, \ldots, \lceil 2/(1-\delta)\rceil\}$ such that the maximal disk on $p_e'(j)$ contains D_i''. Accordingly we can replace such D_i'' by a maximal disk on some candidate point, again.

By applying the above operations to each end disk in $\mathcal{D}''$, $\mathcal{D}''$ becomes satisfying the conditions satisfied by $\mathcal{D}'$. Consequently we have $A(\mathcal{D}') \geq A(\mathcal{D}'')$. □

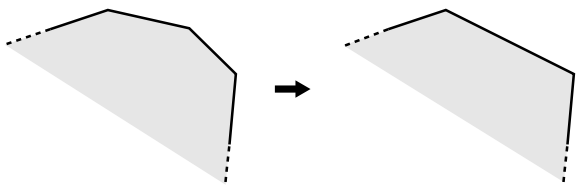

Figure 11: Cutting off a corner.

The computation time of Algorithm POLYGON depends on the number of vertices of P exponentially. Moreover, the computation time becomes larger when θ approaches $\pi/2$. In the rest of this section, we present a method to reduce these defects.

For $\alpha \leq 1$, an object P is called to be *α-approximated* by another object Q if $\alpha Q \subseteq P \subseteq Q$.

Theorem 8. *Suppose that an object P is α-approximated by another object Q. Let $\mathcal{D}$ be a β-approximate solution for the disk covering problem with αQ. Then $\mathcal{D}$ is an $\alpha^2\beta$-approximate solution for the disk covering problem with P.*

Proof. Since $\alpha Q \subseteq P$, each disk in $\mathcal{D}$ is contained by P, which means that $\mathcal{D}$ is feasible for P. Let $\mathcal{D}^*$ denote an optimal solution for P. Since $P \subseteq Q$, $\alpha\mathcal{D}^*$ is contained by αQ. Hence $\alpha^2\beta A(\mathcal{D}^*) = \beta A(\alpha\mathcal{D}^*) \leq A(\mathcal{D})$ holds, which means that $\mathcal{D}$ is an $\alpha^2\beta$-approximate solution for P. □

Let P be a convex polygon with k vertices which is given in an instance of the disk covering problem, and P' be another convex polygon with k' vertices obtained by cutting of $k - k'$ corners as in Figure 11 where $k' < k$. Theorem 8 tells that, solving the instance with P' provides an approximate solution to the original instance of the disk covering problem although it needs to pay additional approximation factor. This gives a heuristic to reduce the number of vertices of polygons. It also reduces θ, which is the half of the maximum inner angle of the polygon, by cutting off corners of large inner angles. As we set k' larger, we can find P' that approximates P better, and hence the additional approximation factor approaches 1.

6 Conclusion

We have presented an approximation scheme for the disk covering problem with convex polygons. Its computation time is not good. Although an idea to reduce this defect has been presented in Section 5, we think that it is expected to reduce the running time further. We also note that several related interesting problems remain. For example, we believe that the disk covering problem with non-convex polygons is important.

References

[1] A. Aggarwal, L.J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. Discrete and Computational Geometry, 4:591–604, 1989.

[2] H. Akeb and Y. Li. A basic heuristic for packing equal circles into a circular container. Computers and Operations Research, 33:2125–2142, 2006.

[3] U. Feige. A threshold of $\ln n$ for approximating set cover. Journal of the ACM, 45:634–652, 1998.

[4] R.L. Graham and B.D. Lubachevsky. Dense packings of equal disks in an equilateral triangle: from 22 to 34 and beyond. Electronic Journal of Combinatorics, page #A1, 1995.

[5] R.L. Graham, B.D. Lubachevsky, K.J. Nurmela, and P.R.J. Östergård. Dense packing of congruent circles in a circle. Discrete Mathematics, 181:139–154, 1998.

[6] A. Heppes and H. Melissen. Covering a rectangle with equal circles. Periodica Mathematica Hungarica, 34:65–81, 1997.

[7] D.S. Hochbaum. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In Approximation algorithms for NP-hard problems, pages 94–143. PWS Publishing Company, 1997.

[8] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. Mobile Networks and Applications, 10:519–528, 2005.

[9] K.J. Nurmela. Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles. Experimental Mathematics, 9:241–250, 2000.

[10] T. Imamichi and H. Nagamochi. A Multi-sphere scheme for 2D and 3D packing problems. Proceedings of the Engineering Stochastic Local Search (SLS2007). Lecture Notes in Computer Science, 4638:207–211, 2007.

[11] S. Kravitz. Packing cylinders into cylindrical containers. Mathematics Magazine, 40:65–71, 1967.

[12] J.B.M. Melissen and P.C. Schuur. Covering a rectangle with six and seven circles. Discrete Applied Mathematics, 99:149–156, 2000.

[13] G.E. Reis. Dense packing of equal circles within a circle. Mathematics Magazine, 48:33–37, 1975.

[14] K. Sugihara. Robust geometric computation based on digital topology. K.-Y. Chwa and J.I. Munro, editors, Proceedings of the 10th International Computing and Combinatorics Conference, Lecture Notes in Computer Science, 3106:3–12, 2004.

[15] K. Sugihara, M. Sawai, H. Sano, D.-S. Kim, and D. Kim. Disk packing for the estimation of the size of a wire bundle. Japan Journal of Industrial and Applied Mathematics, 21:259–278, 2004.

[16] H. Wang, W. Huang, Q. Zhang, and D. Xu. An improved algorithm for the packing of unequal circles within a larger containing circle. European Journal of Operational Research, 141:440–453, 2002.