Algorithmica (2008) 50: 58–71
DOI 10.1007/s00453-007-9043-4

An Optimal Algorithm for the Minimum Disc Cover Problem

Min-Te Sun · Chih-Wei Yi · Chuan-Kai Yang · Ten-Hwang Lai

Received: 13 October 2005 / Accepted: 24 May 2006 / Published online: 10 November 2007

Abstract The *minimum disc cover* can be used to construct a dominating set on the fly for energy-efficient communications in mobile ad hoc networks, but the approach used to compute the minimum disc cover proposed in previous studies is computationally relatively expensive. In this paper, we show that the disc cover problem is in fact a special case of the general α-hull problem. In spite of being a special case, the disc cover problem is not easier than the general α-hull problem. In addition to applying the existing α-hull algorithm to solve the disc cover problem, we present a simple, yet optimal divide-and-conquer algorithm that constructs the minimum disc cover for arbitrary cases, including those degenerate cases where the α-hull approach would fail.

Keywords Minimum disc cover · α-hull · Optimal algorithms · Applications in ad hoc networks

M.-T. Sun (✉)
Department of Computer Science and Software Engineering, Auburn University,
Auburn, AL 36849, USA
e-mail: sunmint@eng.auburn.edu

C.-W. Yi
Department of Computer Science, National Chiao Tung University, Hsinchu City 30010, Taiwan
e-mail: yi@cs.nctu.edu.tw

C.-K. Yang
Department of Information Management, National Taiwan University of Science and Technology,
Taipei 106, Taiwan
e-mail: ckyang@cs.ntust.edu.tw

T.-H. Lai
Department of Computer Science and Engineering, Ohio State University,
Columbus, OH 43210, USA
e-mail: lai@cse.ohio-state.edu

1 Introduction

Finding a minimum cover is an interesting combinatorial problem. Many well known problems (e.g., Vertex Cover, Dominating Set, Set Cover, and Covering by Cliques [5]) can all be viewed as such problems. In this paper, we discuss a specific minimum cover problem, which was first introduced in [12].

Let $\Delta = \{D_0, D_1, D_2, \dots, D_n\}$ be a set of discs of radius R with all their origins (centers) located inside D_0. Given Δ, the *minimum disc cover problem* seeks to identify a minimum subset of Δ, say Δ', such that the union of the discs in Δ' is equal to the union of the discs in Δ. Without loss of generality, we may assume $R = 1$.

The disc cover problem has applications in inter-vehicle communications [9], location-based mobile ad hoc networks [6–8, 11, 13], and multicast medium access control [10]. For instance, assume that all mobile stations transmit data with the same transmission radius. It happens frequently that a mobile station s needs to broadcast a message to the entire ad hoc network. If the set of s's immediate neighbors is denoted as $N(s)$, then instead of asking every station in $N(s)$ to rebroadcast s's message, it is more efficient to ask only the stations in a smaller set $N'(s) \subset N(s)$ to do it if $N'(s) \cup \{s\}$ cover the same area as $N(s) \cup \{s\}$ do. This simple broadcast protocol has been shown in [11] to significantly reduce the number of retransmissions and therefore effectively alleviate the broadcast storm problem defined in [13].

In contrast to the other minimum cover combinatorial problems mentioned earlier, the disc cover problem *can* be solved in polynomial time. In [12], an algorithm similar to the Graham-scan algorithm for convex hull [2] is proposed that constructs a minimum disc cover in $\mathcal{O}(n^{4/3})$ time, where n is the number of discs. As most existing applications of the minimum disc cover problem are for mobile ad hoc networks, where the computing power and battery life of each node are relatively limited, more efficient algorithms are always desirable unless the existing ones are already optimal.

In this paper, we propose two new methods for the minimum disc cover problem. The first one reduces the covering problem to the *α-hull problem* [4], which can then be solved using an existing algorithm. However, this method assumes that no more than three disc origins fall on the circumference of a disc. Our second method solves the minimum disc cover problem directly using a *simple* divide-and-conquer strategy. Although both methods have the same time complexity, $\mathcal{O}(n \log n)$, the second method enjoys two strengths: (1) it is conceptually simple; and (2) it requires no aforementioned assumption (as entailed by the first method). As the final contribution of this paper, we will prove that any algorithm that solves the minimum disc cover problem needs $\Omega(n \log n)$ time in the worst case, thereby establishing the optimality of both of our algorithms.

2 The Minimum Disc Cover and α-Hull Problems

Minimum Disc Cover was originally formulated as a problem in mobile ad hoc networks, where each node (i.e., mobile station) is assumed to have the same transmission range. If the transmission range is normalized to 1, the coverage area of a node i can be modelled as unit disc D_i centered at node i's location. We will denote node

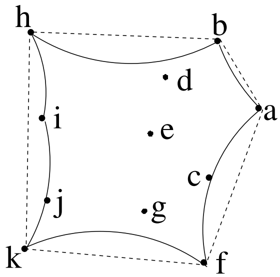

Fig. 1 The α-hull and convex hull (*the dashed lines*) for a given set of points

i's location as O_i, or simply i if there is no ambiguity, and denote its coverage disc as $D(O_i)$, or simply D_i. The Euclidean distance between nodes i and j is denoted as $\mathrm{dist}(O_i, O_j)$, or $\mathrm{dist}(i, j)$. The minimum disc cover problem is formally defined as follows.

Definition 1 Minimum Disc Cover Problem

Instance: A set of unit discs, $\Delta = \{D_0, D_1, D_2, \ldots, D_n\}$, such that $\mathrm{dist}(D_0, D_i) < 1$[1] for all i, $1 \le i \le n$.
Question: Find a subset $\Delta' \subseteq \Delta$ such that $|\Delta'|$ is minimum and

$$\bigcup_{D_i \in \Delta'} D_i = \bigcup_{D_j \in \Delta} D_j.$$

(Note that no two discs share the same origin. Also, D_0 may be included in Δ'.)

We will relate the minimum disc cover problem to another problem, called α-hull. For any negative number α and any set of points S on a two-dimensional plane, the *α-hull for S* is defined as the intersection of all closed complements of discs (with radius $-1/\alpha$) that contain all the points in S [4]. (Note that it is the discs' complements, not the discs themselves, that contain the points in S.)

Since we are only interested in unit discs, we will assume $\alpha = -1$ unless otherwise indicated. With such an assumption, the shape of an α-hull resembles an inward curved convex hull, where each edge of the α-hull is an arc of a unit circle. A convenient way to visualize the shape of an α-hull is to envision the points in S as nails fixed on the plane and a steel wheel of radius 1 is rolled around the nails to obtain the outer contour of S. The contour is the α-hull for S. Figure 1 illustrates a set of points' α-hull as well as its convex hull.

The following theorem shows a simple connection between the minimum disc cover and α-hull problems.

Theorem 1 *Let $\Delta = \{D_0, D_1, \ldots, D_n\}$ be an instance of the minimum disc cover problem, and let $S = \{O_0, O_1, \ldots, O_n\}$ be the corresponding set of origins. Then the set of all vertices of S's α-hull, with $\alpha = -1$, corresponds to a minimum cover for Δ.*

[1] In this paper, we adopt the open disk model. However, the same results can be had for the close disk model, where $\mathrm{dist}(D_0, D_i) < 1$ is replaced by $\mathrm{dist}(D_0, D_i) \le 1$ in the problem definition.

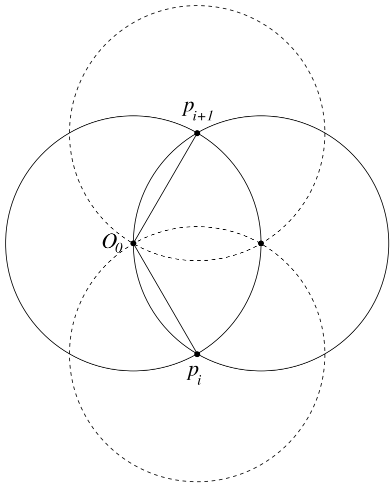

Fig. 2 Largest possible arc not including O_0 is 120°; and $D(p_i)$ and $D(p_{i+1})$ (represented by *dotted circles*) together cover the corresponding sector of $D(O_0)$

Proof Let $S = \{O_0, O_1, O_2, \ldots, O_n\}$ be the set of disc origins for a given disc set Δ such that $\forall i$ $\mathrm{dist}(O_0, O_i) < 1$. Let $\alpha = -1$. As mentioned earlier, imagining the nodes in S as nails, the shape of the α-hull is exactly the outer contour obtained by rolling a steel unit disc around the nails. Let H be the α-hull for S, and let $P = \{p_1, \ldots, p_m\} \subseteq S$ be the set of all vertices of H in the same order they were rolled over by the rolling steel unit disc in the counterclockwise direction. We will show that $\Delta' = \{D(p_1), \ldots, D(p_m)\}$ forms a minimum disc cover for Δ. Let

$$U'_D = \bigcup_{1 \le i \le m} D(p_i).$$

We first show $D(O_0) \subseteq U'_D$. Assume $O_0 \notin P$ (the case where $O_0 \in P$ is trivial). For each consecutive pair p_i and p_{i+1} in P (note that p_m and p_1 are regarded consecutive), the rolling steel unit disc passing through p_i and p_{i+1} does not enclose O_0 in it. (O_0 would have been included in P if it were inside the rolling unit disc; see Fig. 2 for illustration.) It is easy to see that the union of $D(p_i)$ and $D(p_{i+1})$ covers the sector of $D(O_0)$ from $\overrightarrow{O_0 p_i}$ to $\overrightarrow{O_0 p_{i+1}}$ in the counterclockwise direction. Since the same argument applies to all consecutive pairs in P, it follows that $D(O_0) \subseteq U'_D$.

Next, we show $D(O_i) \subseteq U'_D$ for every origin $O_i \in S$. It suffices to consider the case where $O_i \notin P$. For convenience, write O_i as a. Assume there exists a point x in $D(a)$ such that $x \notin U'_D$. We will show that $d(x, a) \ge 1$. Given that all points in H are inside $D(O_0) \subseteq U'_D$, x must be outside the disc $D(O_0)$ (i.e., $d(x, O_0) > 1$). The line segment $\overline{ax}$ must intersect some arc of H, say $\widehat{pq}$, at some point t (see Fig. 3). Let the origin of the rolling steel unit disc passing through p and q be O. Because O_0 is in S, the angle $\angle p O_0 q$ cannot be more than 120°, as shown in Fig. 2. The point x, which appears on the concave side of $\widehat{pq}$ and is not covered by $D(p) \cup D(q)$, must be at least 1 unit (radius of discs) away from $\widehat{pq}$. (In Fig. 3, point x is on the north of O.) Thus, $d(x, a) = d(x, t) + d(t, a) \ge d(x, t) + d(a, \widehat{pq}) \ge d(x, t) + 1 \ge 1$. This proves $D(O_i) \subseteq U'_D$ for every disc $D(O_i)$ in Δ. Therefore, Δ' is a disc cover for Δ.

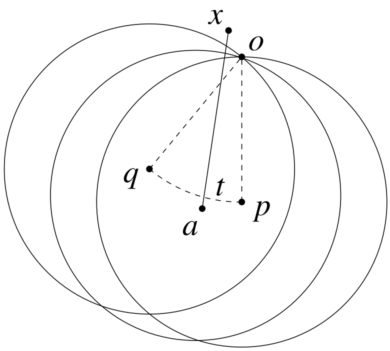

Fig. 3 The distance between a and x is at least 1 if x is not covered by $D(p)$ and $D(q)$

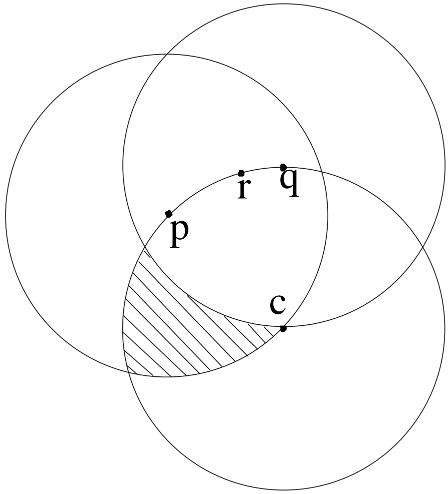

Fig. 4 The disc centered at a vertex of the α-hull covers some area not covered by other nodes in S

Now, we establish the optimality of Δ' by showing that any disc cover for Δ must contain Δ' as a subset. Let C be the set of origins of any disc cover for Δ, and let p be any point in P. We need to show $p \in C$.

If the steel unit disc rolls around a point $p \in P$ without touching another one, all other points in S have to be at least 2 units away from p. The only possibility of this scenario is when $p = O_0$ and $S = \{O_0\}$.

Now suppose there are at least two points in S. Then, a point $p \in P$ must be an endpoint of some arc $\widehat{pq}$ on the α-hull's boundary. $\widehat{pq}$ is part of the circumference of the rolling steel unit disc. Let us denote the origin of the rolling steel unit disc passing through p and q as c. The disc $D(c)$ must contain no point in S inside it. Note that points in S could appear on the arc $\widehat{pq}$, for example the point r in Fig. 4. Since S contains only a finite number of points, it is easy to see from Fig. 4 that there is always an area close to c (i.e., the part of the shaded region close to c in Fig. 4) that is only covered by disc $D(p)$ but not by any other disc in Δ. Therefore, p must be in C. □

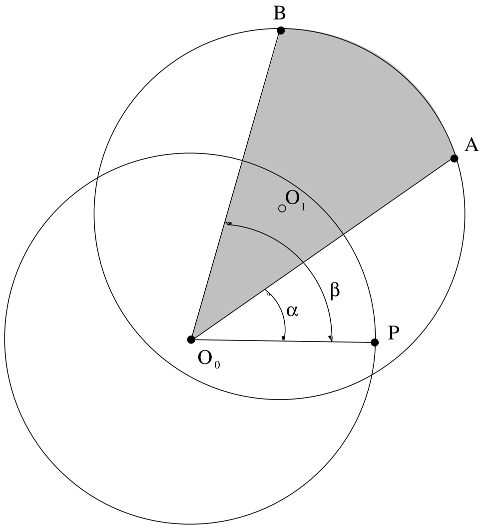

Fig. 5 The *shaded area* is covered by $D(O_1)$

Theorem 1 establishes the connection between the minimum disc cover and α-hull problems. In [4], there is an optimal algorithm that constructs the α-hull of a set of n points in $\mathcal{O}(n \log n)$ time. Using this algorithm, one can solve the minimum disc cover problem in $\mathcal{O}(n \log n)$ time, which is an improvement over the $\mathcal{O}(n^{4/3})$ algorithm in [12]. This approach, however, has a weakness—the α-hull algorithm in [4] needs to compute the Delaunay Triangulations and will fail if there are more than *three* disc origins falling on the circumference of a circle.

In the next section we will present a conceptually simpler divide-and-conquer algorithm that works for all cases, including degenerate ones.

3 The Divide-and-Conquer Algorithm

3.1 Theoretical Basis

We state the following simple, yet useful fact as a lemma for ease of reference. Figure 5 illustrates the situation.

Lemma 1 *Let O_0 and O_1 be a pair of points with* $\mathrm{dist}(O_0, O_1) < 1$, *and let $\widehat{AB}$ be an arc on the boundary of the unit disc $D(O_1)$ which is outside the area of the unit disc $D(O_0)$. The area surrounded by $\widehat{AB}$, $\overline{AO_0}$, and $\overline{BO_0}$ is completely covered by $D(O_1)$.*

Now, suppose we are given $S = \{O_0, O_1, \ldots, O_n\}$, and the set of corresponding unit discs $\Delta = \{D(O_0), D(O_1), \ldots, D(O_n)\}$. Consider the union of discs $U = \bigcup_{O_i \in S} D(O_i)$. As illustrated in Fig. 6, the area of U is bounded by a series of arcs, say $(\widehat{A_1A_2}, \widehat{A_2A_3}, \ldots, \widehat{A_{k-1}A_k}, \widehat{A_kA_{k+1}})$, where $A_1 = A_{k+1}$. If the region bounded by $\widehat{PQ}$, $\overline{PO}$, and $\overline{QO}$ is denoting by Sector($\widehat{PQ}$, $\overline{PO}$, $\overline{QO}$) the area U can be

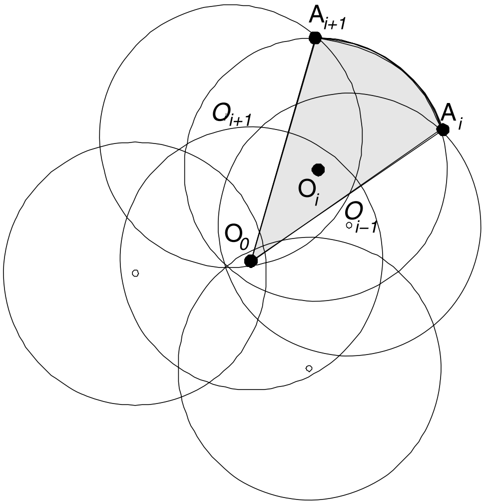

Fig. 6 The boundary of $\bigcup_{O_i \in S} D(O_i)$ is formed by a series of arcs $\langle \widehat{A_i A_{i+1}} \rangle$

viewed as the union of Sector($\widehat{A_i A_{i+1}}$, $\overline{A_i O_0}$, $\overline{A_{i+1} O_0}$), $1 \le i \le k$. We show in the following lemma that the unit discs that contribute arcs $\widehat{A_i A_{i+1}}$ $(1 \le i \le k)$ form a minimum disc cover of Δ.

Lemma 2 *Let* $\widehat{A_1 A_2}, \widehat{A_2 A_3}, \ldots, \widehat{A_{k-1} A_k}, \widehat{A_k A_{k+1}}$ *be the arcs surrounding* $U = \bigcup_{O_i \in S} D(O_i)$, *where* $A_1 = A_{k+1}$; *and for* $1 \le i \le k$, *let* $D(O_{t_i})$ *be the unit disc that contributes* $\widehat{A_i A_{i+1}}$. *The set of unit discs*, $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$, *is a minimum disc cover for* Δ.

Proof We first claim that any unit disc $D(O_{t_i})$ that contributes an arc to the boundary of U must be included in any disc cover of Δ. To see this, observe that if $\widehat{A_i A_{i+1}}$ is contributed by O_{t_i}, then the area close to $\widehat{A_i A_{i+1}}$ is not covered by any disc other than $D(O_{t_i})$.

Now, the area U can be thought of as a collection of disjoint pieces, each surrounded by $\widehat{A_i A_{i+1}}$, $\overline{A_i O}$, and $\overline{A_{i+1} O}$, where i ranges from 1 to k. By Lemma 1, each such piece is covered by a unit disc $D(O_{t_i})$. Thus, $\{D(O_{t_1}), \ldots, D(O_{t_k})\}$ is a disc cover for Δ. Applying the above claim, the cover is minimum. □

Notice that a unit disc $D(O_{t_i})$ may contribute more than one arc to the boundary of U. This implies that the same unit disc $D(O_{t_i})$ may appear more than once in the disc cover set $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$. In fact, a unit disc in Δ may contribute zero, one, or two arcs to the boundary of U. The possibility of contributing more than one arc is the main reason why the minimum disc cover problem is nontrivial. This will be further discussed in Sect. 3.5.

3.2 Abstract of Algorithm

By Lemma 2, to find the minimum disc cover for Δ, it suffices to find the arcs that make up the boundary of U. This can be done in a divide-and-conquer fashion as follows.

Algorithm Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; i, j$)
 if $i = j$
  return the boundary of $D(O_0) \cup D(O_i)$
 else
  $m := \lfloor (i + j)/2 \rfloor$
  ArcList$_1$:= Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; i, m$)
  ArcList$_2$:= Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; m + 1, j$)
  return ArcMerge(ArcList$_1$, ArcList$_2$)

Boundary ($\{O_0, O_1, O_2, \ldots, O_n\}; i, j$) returns the boundary of $U_D(i, j)$, where $U_D(i, j)$, referred to as a *super disk*, is defined as

$$U_D(i, j) = D(O_0) \cup D(O_i) \cup D(O_{i+1}) \cup \cdots \cup D(O_j).$$

It is important to note that $D(O_0)$ is always present in $U_D(i, j)$ and $1 \le i \le j \le n$. This is a typical divide-and-conquer algorithm. In order to find the boundary of $U_D(i, j)$, we find the boundary of $U_D(i, m)$ and that of $U_D(m + 1, j)$, and then merge the two boundaries to obtain that of $U_D(i, j)$. Here, each boundary is represented as a list of arcs, and the hard part of the algorithm is how to merge two arc lists. We will discuss these issues in the next two subsections.

3.3 Representation of Arc Lists

An arc can be characterized by three parameters, (O_t, α, β), where O_t is the origin of the disc that contributes the arc, and α and β are the degrees of two angles, as depicted in Fig. 5. It is not hard to see that

$$\alpha = \begin{cases} \cos^{-1}\left(\frac{\overrightarrow{O_0A}\cdot(1,0)}{\|\overrightarrow{O_0A}\|}\right), & \text{if } (1,0) \times \overrightarrow{O_0A} \ge 0, \\ \cos^{-1}\left(\frac{\overrightarrow{O_0A}\cdot(1,0)}{\|\overrightarrow{O_0A}\|}\right) + \pi, & \text{otherwise} \end{cases}$$

and β can be similarly computed.

Notice that the angles are computed using O_0 as the apex, rather than the origin of the disc contributing the arc. In Fig. 5, $\overline{O_0P}$ is parallel to the x-axis.

The boundary of a super disc is thus a list of Arc$(O_{s_i}, \alpha_{s_i}, \beta_{s_i})$, $0 \le i \le \max$. For ease of discussion, if an arc Arc(O_k, α_k, β_k) crosses 360° (i.e., $\alpha_k < 360°$ and $\beta_k > 360°$), we split it into two arcs Arc$(O_k, 0°, \beta_k)$ and Arc$(O_k, \alpha_k, 360°)$. Thus, without loss of generality, we may assume that no arc crosses 360°. Furthermore,

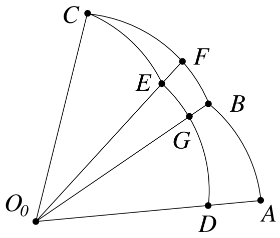

Fig. 7 Splitting $\widehat{BC}$ into $\widehat{BF}$ and $\widehat{FC}$, and splitting $\widehat{DE}$ into $\widehat{DG}$ and $\widehat{GE}$

we sort each list based on the value of the first angle (i.e., α_i) in ascending order, so that $\beta_{s_i} = \alpha_{s_{i+1}}$, $0 \le i \le \max$, with the understanding that $i + 1$ is computed modulo max. Thus, the boundary of a super disc can be represented simply as $(O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_{\max}}, \alpha_{s_{\max}})$, where $0° = \alpha_{s_0} < \cdots < \alpha_{s_{\max}} < 360°$.

3.4 Merging Two Arc Lists

Now, given two sorted arc lists

$$\begin{cases} \mathrm{ArcList}_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_{\max}}, \alpha_{s_{\max}}), \\ \mathrm{ArcList}_2 = (O_{t_0}, \alpha_{t_0}, O_{t_1}, \alpha_{t_1}, \ldots, O_{t_{\max}}, \alpha_{t_{\max}}) \end{cases}$$

representing the boundaries of super discs $U_D(i, m)$ and $U_D(m+1, j)$, respectively, we want to *merge* them so that the resulting list represents the boundary of $U_D(i, j) = U_D(i, m) \cup U_D(m+1, j)$.

The first step is to *split* the arcs in each list into smaller arcs, if necessary, so that the two refined arc lists share the same sequence of angles (i.e., α's). For instance, as illustrated in Fig. 7, $\widehat{BC}$ is split into $\widehat{BF}$ and $\widehat{FC}$, and $\widehat{DE}$ is split into $\widehat{DG}$ and $\widehat{GE}$. With such splits, arc lists $\{\widehat{AB}, \widehat{BC}\}$ and $\{\widehat{DE}, \widehat{EC}\}$ become $\{\widehat{AB}, \widehat{BF}, \widehat{FC}\}$ and $\{\widehat{DG}, \widehat{GE}, \widehat{EC}\}$, respectively, which share the same sequence of angles with O_0 at the apex. As another example, if

$$\begin{cases} \mathrm{ArcList}_1 = (O_{s_0}, 0°, O_{s_1}, 30°, O_{s_2}, 140°, O_{s_3}, 200°), \\ \mathrm{ArcList}_2 = (O_{t_0}, 0°, O_{t_1}, 120°, O_{t_2}, 240°), \end{cases}$$

they will be refined to

$$\begin{cases} \mathrm{ArcList}_1' = (O_{s_0}, 0°, O_{s_1}, 30°, O_{s_1}, 120°, O_{s_2}, 140°, O_{s_3}, 200°, O_{s_3}, 240°), \\ \mathrm{ArcList}_2' = (O_{t_0}, 0°, O_{t_0}, 30°, O_{t_1}, 120°, O_{t_1}, 140°, O_{t_1}, 200°, O_{t_2}, 240°). \end{cases}$$

Now, let us consider the general case. After splitting, the given lists will contain the same number of arcs:

$$\begin{cases} \mathrm{ArcList}_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_k}, \alpha_{s_k}), \\ \mathrm{ArcList}_2 = (O_{t_0}, \alpha_{s_0}, O_{t_1}, \alpha_{s_1}, \ldots, O_{t_k}, \alpha_{s_k}) \end{cases}$$

for some k. To *merge* the two lists, we simply *merge* each individual pair of corresponding arcs, $\mathrm{Arc}(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $\mathrm{Arc}(O_{t_i}, \alpha_i, \alpha_{i+1})$ as follows. (See Fig. 8 for illustration.)

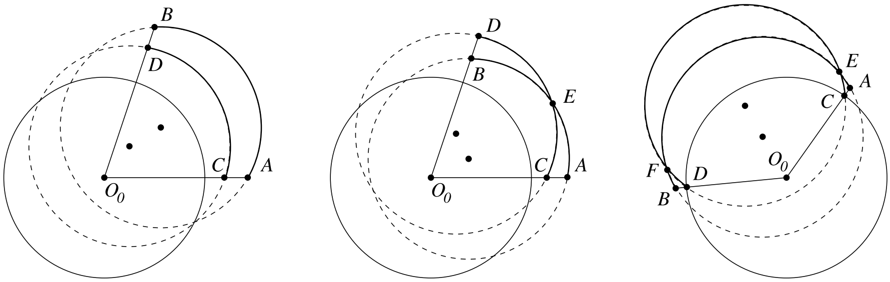

Fig. 8 Cases 1, 2 and 3, where two arcs with the same angle span are merged

Case 1: $\mathrm{Arc}(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $\mathrm{Arc}(O_{t_i}, \alpha_i, \alpha_{i+1})$ have no intersection. One arc is closer to O_0 than the other. The merger will drop the arc closer to O_0 and keep only the farther one. For instance, in the left diagram of Fig. 8, the merger will result in arc $\widehat{AB}$.

Case 2: $\mathrm{Arc}(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $\mathrm{Arc}(O_{t_i}, \alpha_i, \alpha_{i+1})$ intersect at one point. In this case, we drop the two inner sub-arcs and keep the two external ones. Thus, in the middle example of Fig. 8, merging arcs $\widehat{AB}$ and $\widehat{CD}$ will yield arcs $\widehat{AE}$ and $\widehat{ED}$.

Case 3: $\mathrm{Arc}(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $\mathrm{Arc}(O_{t_i}, \alpha_i, \alpha_{i+1})$ intersect at two points (E and F in Fig. 8, right). In this case, we drop the three inner sub-arcs and keep the three external ones. Thus, merging arcs $\widehat{AB}$ and $\widehat{CD}$ will result in arcs $\widehat{AE}$, $\widehat{EF}$ and $\widehat{FB}$.

3.5 Time Complexity of the Proposed Algorithm

We will show that the above proposed divide-and-conquer algorithm has time complexity $\mathcal{O}(n \log n)$. Our analysis will hinge on the following lemma.

Lemma 3 *A unit disc $D(O_i)$ can contribute at most two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$.*

Proof Recall that the reference point O_0 is fixed and all the other unit disc origins are inside the unit disc $D(O_0)$. Hence, if we consider the boundary of the union of the unit disc $D(O_0)$ and any other disc, say $D(O_1)$, the arc contributed from $D(O_1)$ must have an angle (with respect to O_1) of less than 240°. Let the arc be $\widehat{AB}$. (See Fig. 9 for illustration.) Now, if we place a third unit disc so that it "breaks" arc $\widehat{AB}$ into two pieces, the origin of that disc must be more than 1 unit away from each of A and B and less than 1 unit away from some part of $\widehat{AB}$ (illustrated as the shaded area in Fig. 9). With the third unit disc being added, $D(O_1)$ now contributes $\widehat{AP}$ and $\widehat{QB}$ (rather than $\widehat{AB}$) to the boundary. (In other words, arc $\widehat{AB}$ contributed by $D(O_1)$ is now broken into $\widehat{AP}$ and $\widehat{QB}$.) It is not hard to see that arcs $\widehat{AP}$ and $\widehat{QB}$ must be at least 120° apart *with respect to* O_1. In general, each time an arc is cut into two pieces by another unit disc, the resulting smaller pieces (that still contribute to the boundary) must be at least 120° apart with respect to the origin of the contributing

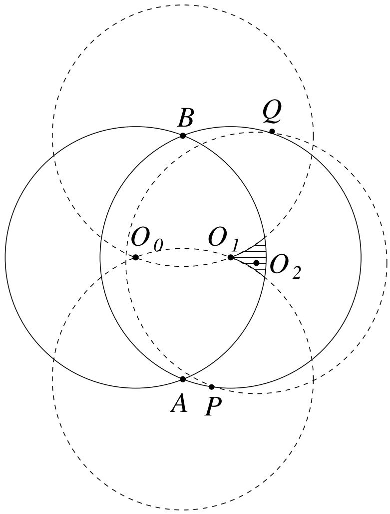

Fig. 9 The possible area for the origin of a third disc that breaks $\widehat{AB}$ into two arcs $\widehat{AP}$ and $\widehat{QB}$

disc. Now, if a disc is able to contribute three arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$, the three arcs must be pairwisely separated by at least 120°. Adding up the degrees of the angles for the gaps and the degrees of the angles for the arcs themselves, the total would exceed 360°, which is not possible. Therefore, a unit disc can only contribute at most two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$. □

Now we are ready to show that our algorithm has time complexity $\mathcal{O}(n \log n)$.

Theorem 2 *Algorithm Boundary has time complexity* $\mathcal{O}(n \log n)$, *where n is the number of unit discs in* Δ.

Proof The running time, $T(n)$, of **Algorithm** Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; i, j$) satisfies the recurrence

$$T(n) = \begin{cases} T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + T(\text{ArcMerge}) & \text{if } n > 1, \\ O(1) & \text{if } |n| = 1, \end{cases}$$

where $n = j - i + 1$. Since each disc can only contribute at most two arcs (by Lemma 3), ArcList_1 and ArcList_2 each contain at most $\mathcal{O}(n)$ "unsplit" arcs. After splitting, each list still contains at most $\mathcal{O}(n)$ arcs. Since merging two arcs with the same angle span (as described at the end of Sect. 3.4) takes $O(1)$ time, merging the two (refined) arc lists takes $\mathcal{O}(n)$ time. So, the time complexity of ArcMerge is $\mathcal{O}(n)$. Solving the above recurrence then yields $T(n) = \mathcal{O}(n \log n)$. □

4 Optimality of the Time Complexity

In an earlier section we established the connection between the minimum disc cover and α-hull problems; and it has been shown in [4] that any algorithm that constructs

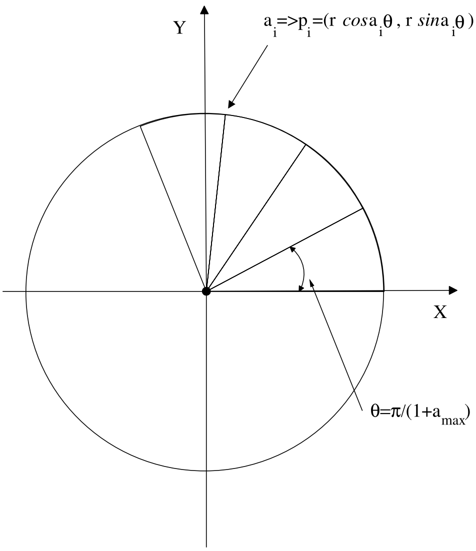

Fig. 10 Mapping a_i in S to $p_i = (r\cos(\frac{a_i \cdot \pi}{a_{\max}+1}), r\sin(\frac{a_i \cdot \pi}{a_{\max}+1}))$

the α-hull for an *arbitrary* set of points S needs at least $\Omega(n\log n)$ time in the worst case, where $n = |S|$. However, since our minimum disc cover problem confines the input unit disc origins to be inside the unit disc $D(O_0)$, it is, therefore, questionable whether the minimum disc cover problem still has the same time complexity lower bound as the α-hull problem. In the following theorem, we answer this question in the affirmative by reducing the element uniqueness problem (which is known to be $\Omega(n\log n)$ in time complexity [3]) in linear time to our disc cover problem.

Definition 2 Element Uniqueness Problem

Instance: A multiset of non-negative integers $S = \{a_1, a_2, \ldots, a_n\}$.
Question: Are there elements a_i and a_j in S, with $i \neq j$, such that $a_i = a_j$?

Theorem 3 *The minimum disc cover problem has a time complexity of* $\Omega(n\log n)$.

Proof It suffices to reduce the element uniqueness problem to the minimum disc cover problem in $\mathcal{O}(n)$ time. Given a multiset of non-negative integers $S = \{a_1, a_2, \ldots, a_n\}$, we first scan to find the largest element in S, say $a_{\max}$. Let $\theta = \frac{\pi}{a_{\max}+1}$, and r be a positive constant less than 1. We then map each element a_i in S to $p_i = (r\cos(a_i \cdot \theta), r\sin(a_i \cdot \theta))$, which is a point on the circumference of a disc of radius r centered at origin O. (Figure 10 illustrates such a mapping.) The scan and mapping evidently take $O(n)$ amount of time. Let $\Delta = \{D(O), D(p_1), D(p_2), \ldots, D(p_n)\}$ be an instance of the minimum disc cover problem, where $D(O)$ and $D(p_i)$ are unit discs. Since all points p_i are on the upper semi-circumference of the disc of radius r centered at O, every distinct element in $\{O, p_1, p_2, \ldots, p_n\}$ must contribute to the minimum disc cover for Δ. Thus, S has no duplicates if and only if Δ's minimum disc cover has cardinality $n + 1$. $\square$

5 Conclusion and Future Work

We showed the minimum disc cover problem to be a special case of the α-hull problem. Then we showed that being a special case does not make it simpler than the general α-hull problem in terms of time complexity—both problems need at least $\Omega(n \log n)$ time. We proposed two optimal algorithms for constructing minimum disc covers. The first applies an existing algorithm for the α-hull. However, this approach needs to construct Delaunay Triangulations and would fail should more than three disc origins fall on the circumference of a circle. Our second method is a divide-and-conquer algorithm, which is conceptually simpler and works well even for degenerate cases.

Although the proposed algorithms are optimal in terms of time complexity, there may still be room for improvement when the algorithm is used in applications such as message broadcast or media access control in mobile ad hoc networks. One possibility is for each node to keep track of its neighbors' movements and "predict" when the members of its minimum disc cover are likely to change, and only then re-compute the new minimum cover. This will reduce the number of times a node computes its minimum disc cover, thereby reducing the power consumed by such computations. This is known as the *Kinetic Collision Detection* model [1]. This model will be explored in our future work.

References

1. Basch, J., Erickson, J., Guibas, L.J., Hershberger, J., Zhang, L.: Kinetic collision detection for two simple polygons. In: Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 102–111, 1999
2. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry. Springer, Berlin (2000)
3. Dobkin, D.P., Lipton, R.J.: On the complexity of computations under varying sets of primitives. J. Comput. Syst. Sci. **18**, 86–91 (1979)
4. Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R.: On the shape of a set of points in the plane. IEEE Trans. Inform. Theory **IT-29**, 551–559 (1983)
5. Garey, M.L., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
6. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 56–67, 2000
7. Ko, Y.-B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. In: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 66–75, 1998
8. Peng, W., Lu, X.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: Proceedings of the First ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc), pp. 129–130, 2000
9. Sun, M., Feng, W., Lai, T.-H., Yamada, K., Okada, H., Fujimura, K.: GPS-based message broadcast for adaptive inter-vehicle communications. In: Proceedings of IEEE International Conference on Parallel Processing, pp. 279–286, 2000
10. Sun, M., Huang, L., Wang, S., Arora, A., Lai, T.-H.: Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks, Special Issue of Wiley Wireless Communications and Mobile Computing on Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing, 2003
11. Sun, M., Lai, T.-H.: Location aided broadcast in wireless ad hoc network systems. In: Proceedings of IEEE Wireless Communications and Networking Conference, pp. 597–602, 2002

12. Sun, M., Lai, T.-H.: Computing optimal local cover set for broadcast in ad hoc networks. In: Proceedings of IEEE International Conference on Communications, pp. 3291–3295, 2002
13. Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network. Wirel. Netw. **8**(2/3), 153–167 (2002)