# Minimum Disc Cover Set Construction in Mobile Ad Hoc Networks

Min-Te Sun*, Xiaoli Ma*, Chih-Wei Yi†, Chuan-Kai Yang‡, Ten H. Lai§
*Wireless Engineering Research and Education Center, Auburn University
Email: {sunmint, maxiaol}@eng.auburn.edu
†Department of Computer Science, Illinois Institute of Technology
Email: yichihw@iit.edu
‡Department of Information Management, National Taiwan University of Science and Technology
Email: ckyang@cs.ntust.edu.tw
§Department of Computer Science and Engineering, Ohio State University
Email: lai@cse.ohio-state.edu

***Abstract*— The *minimum disc cover set* can be used to construct the dominating set on the fly for energy-efficient communications in mobile ad hoc networks. The approach used to compute the minimum disc cover set proposed in previous study has been considered too expensive. In this paper, we show that the disc cover set problem is in fact a special case of the general $\alpha$-hull problem. In addition, we prove that the disc cover set problem is not any easier than the $\alpha$-hull problem by linearly reducing the element uniqueness problem to the disc cover set problem. In addition to the known $\alpha$-hull approach, we provide an optimal divide-and-conquer algorithm that constructs the minimum disc cover set for arbitrary cases, including the degenerate ones where the traditional $\alpha$-hull algorithm incapable of handling.**

## I. Introduction

Finding a minimum cover is an interesting combinatorial problem. Many well-known problems, such as *Vertex Cover*, *Dominating Set*, *Covering by Cliques*, and *Set Minimum Cover* [4], can be viewed as such a problem. In this paper, we are interested in a specific minimum cover problem which has been formulated [13].

Let $\Delta = \{D_0, D_1, D_2, \ldots, D_n\}$ be a set of discs of radius $R$ with all their origins located inside $D_0$. Given $\Delta$, the *disc cover set problem* is to find a minimum subset of $\Delta$, say $\Delta'$, such that the union of the discs in $\Delta'$ is equal to the union of the discs in $\Delta$. The minimum disc cover set problem has applications in inter-vehicle communications [11], broadcast in location-based mobile ad hoc networks [6] [7] [8] [12], and multicast medium access control [14]. For instance, assume all mobile stations transmit data wirelessly at the same distance. If we denote a mobile station as $s$ and its immediate neighbors as $N[s]$ with $s$ itself included, the effect of having all neighbors broadcast a message is theoretically the same (i.e., cover the same area) as having only the members in the minimum cover set of the coverage areas (discs) associated with $N[s]$ broadcast the message. If the minimum cover set can be constructed efficiently, a node can simply instruct only neighbors in the minimum disc cover set retransmit the broadcast message. This simple broadcast protocol effectively reduces the number of retransmissions and alleviates the channel contention and message collisions caused by massive broadcast retransmissions.

Contrary to the other minimum cover combinatorial problems mentioned above, the disc cover set problem *can* be solved in polynomial time. In [13], a Graham-scan based algorithm is provided that constructs the minimum disc cover set in $\mathcal{O}(n^{4/3})$ time complexity, where $n$ is the number of discs (or participating neighbors). As mentioned earlier, most of the known applications of the minimum disc cover set problem lie in the area of mobile ad hoc networks, where the computing power and battery of each node are limited. Hence, the $\mathcal{O}(n^{4/3})$ time complexity is still considered expensive and inefficient.

In this paper, we propose two methods for the minimum disc cover set problem. The first method involves the reduction of our covering problem into another problem — the *$\alpha$-hull problem* [5] — and then solves the latter using an existing algorithm. This method needs to assume that no more than three disc origins fall at the circumference of a disc. The second method solves the minimum disc cover set problem directly using a simple divide-and-conquer strategy. It works conveniently without the need of the previous assumption. Both methods have a time complexity of $\mathcal{O}(n \log n)$. As the last contribution of this paper, we show that any algorithm that solves the minimum disc cover set problem needs at least $\mathcal{O}(n \log n)$ time in the worst case, thereby establishing the optimality of both our methods.

## II. Definition of Minimum disc Cover Set and its Connection to $\alpha$-hull

The disc cover set problem is originated from mobile ad hoc networks. In such networks, each node (i.e., mobile station) is usually assumed to have the same transmission power, thus assumed the same transmission radius. If the transmission radius is denoted as $R$, the coverage area of a node $i$ can be modeled as a disc $D_i$ with radius $R$ centered at the location of the node. If the location of node $i$ is known as $p_i$, the corresponding disc $D_i$ can also be referred as $D(p_i)$. If the locations of node $i$ and $j$ are $p_i$ and $p_j$, the Euclidean distance of $i$ and $j$ is denoted as either $dist(i, j)$ or $dist(p_i, p_j)$. Using the above denotations, the minimum disc cover problem can be formally defined as follows.

*Definition 1:* **Minimum Disc Cover Set Problem**
**INSTANCE:** A set of disc $\Delta = \{D_0, D_1, D_2, \ldots, D_n\}$ such that $\forall i \ dist(0, i) < R$.
**QUESTION:** Find a subset $\Delta' \subseteq \Delta$ such that $\|\Delta'\|$ is

minimum and

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

Notice that $D_0$ may also be included in the minimum disc cover set.

On the other hand, the $\alpha$-hull (for negative $\alpha$ value) for an arbitrary set of points $S$ on the two-dimensional plane is defined as the intersection of all closed complements of discs (with radius $-1/\alpha$) that contain all the points in $S$. The shape of the $\alpha$-hull for $S$ is like an inward curved convex hull, where each edge of the hull is an arc (of a circle with radius $-1/\alpha$). A good way to visualize the shape of such a $\alpha$-hull is to imagine the points in $S$ as nails fixed on the two-dimensional plane and a steel circle with radius $-1/\alpha$ rolled around the nails to obtain the outer contour of $S$. In Figure 1, the shape of the $\alpha$-hulls for points in the figure is denoted by the dash line.

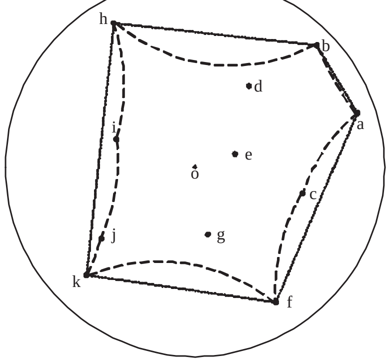

Fig. 1. $\alpha$-hull (in dash line) for a given set of points

The following theorem shows a simple connection between the minimum disc cover set and the $\alpha$-hull.

*Theorem 1:* Let $S$ be the set of disc origins for a given disc set problem, the collection of vertices in the $\alpha$-hull for $S$, where $\alpha = -1/R$, forms the minimum disc cover set.

*Proof:*

Let $S = \{O_0, O_1, O_2, \ldots, O_n\}$ be the set of disc origins for a given disc set $\Delta$ such that $\forall i \; dist(O_0, O_i) < R$. As discussed earlier, imaging nodes in $S$ as nails, the shape of a $\alpha$-hull (of negative $\alpha$) is exactly the outer contour obtained by rolling a steel circle of radius $R$ around the nails. Without loss of generosity, let $H$ be the $\alpha$-hull for $S$, $P = \{p_1, \ldots, p_m\} \subseteq S$ be the set of all vertices of $H$ in the order they were hit by the steel circle when it is rolled around the nails counterclockwisely, and

$$U_D = \bigcup_{1 \le i \le m} D_{p_i}.$$

To prove that the discs associated with nodes in $P$ form a disc cover, we first show $D(O_0) \subseteq U_D$. Assume $O_0 \notin P$ (the case where $O_0 \in P$ is obvious). For each consecutive pair $p_i$ and $p_{i+1}$ in $P$ where $1 \le i \le m$ (note that the subscription $i+1$ is computed module $m$), they must both fall in $D(O_0)$ and the rolling steel circle passing through $p_i$ and $p_{i+1}$ should not contain $O_0$ inside it. ($O_0$ would have been included in the $\alpha$-hull if it is inside of the rolling steel circle. See Figure 2 for illustration.) It is easy to see that the combination of $D(p_i)$ and $D(p_{i+1})$ covers the sector of $D(O_0)$ from $\overrightarrow{O_0 p_i}$ to $\overrightarrow{O_0 p_{i+1}}$ counterclockwisely. Since the same argument applies to all consecutive pairs in $P$, the collection of discs $D(p_i)$, $1 \le i \le m$, should cover the whole disc $D(O_0)$.

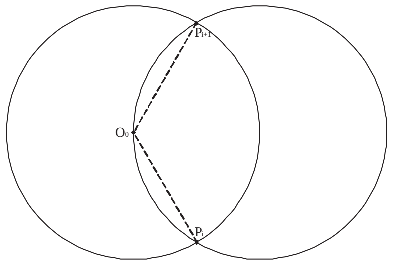

Fig. 2. Largest possible arc not including $O_0$ is 120° and $D(p_i)$ and $D(p_{i+1}$ together cover the corresponding sector of $D(O_0)$)

Next, we show $D(a) \subset U_D$ for every point $a \in S$. It suffices to consider the case where $a \notin P$.

Assume there exists a point $x$ in $D(a)$ such that $x \notin U_D$. We will show $d(x,a) \ge R$. Given points in $H$ are all inside of $D(O_0) \subseteq U_D$, as was shown above, $x$ must be outside the disc $D(O_0)$ (i.e., $d(x, O_0) > R$). The line segment $\overline{ax}$ must intersect an arc of $H$. Let that arc be $\overset{\frown}{pq}$ , where both $p$ and $q$ are in $P$ and the intersection be $t$. In addition, let the origin of the steel rolling circle passing through $p$ and $q$ be $O$. Because $O_0$ is included in $S$, the angle $\angle pO_0q$ has 120° at most as illustrated again in Figure 2. The nearest point $y$ on the concave side of $\overset{\frown}{pq}$ that is not covered by $D(p) \cup D(q)$ has a distance at least $R$ from $\overset{\frown}{pq}$ . (See Figure 3 for illustration.) Thus, $d(x,a) = d(x,t) + d(t,x) \ge d(x,t) + d(y, \overset{\frown}{pq}) \ge d(x,t) + R \ge R$.

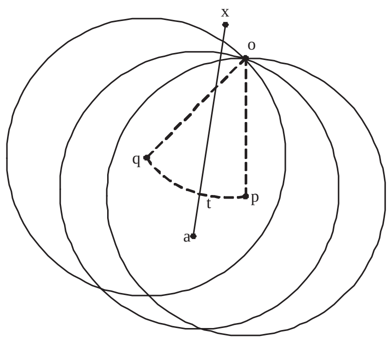

Fig. 3. Trajectory of center forms arcs in the $\alpha$-hull

We have shown that $U_D$ covers any disc with origin in $S$. In other words, the discs associated with vertices of the $\alpha$-hull is a cover set. Now, we show any cover set must contain the discs associated with vertices of the $\alpha$-hull.

Let $C$ be the origins of any cover set for discs with origins in $S$, and let $p$ be any point in $P$. If $p$ is an isolated point in

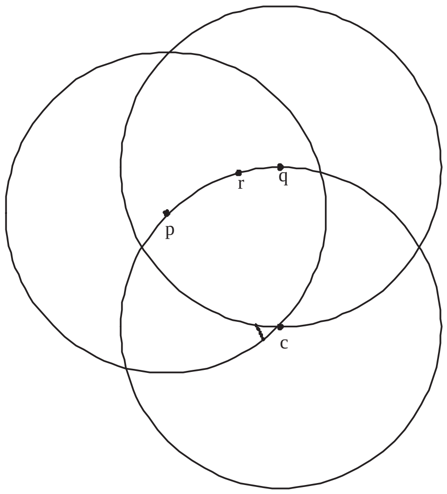

Fig. 4. A vertex of $\alpha$-hull

the $\alpha$-hull, it obviously must be included in $C$. Now suppose $p$ is not an isolated point. Then $p$ is an endpoint of some arc on the $\alpha$-hull's boundary — let it be $\overset{\frown}{pq}$, where $p$ and $q$ are the endpoints of the arc. $\overset{\frown}{pq}$ is part of the circumference of the rolling steel circle. Let us denote the origin of the rolling steel circle passing through $p$ and $q$ be $c$. The disc $D(c)$ should contain no point in $S$ inside it. Note that points in $S$ could appear on the arc $\overset{\frown}{pq}$, such as $r$ in Figure 4. Since $S$ contains only finite number of points, it is easy to see from Figure 4 that there is always an area close to $c$ (i.e., the small triangular region close to $c$ in Figure 4) that is only covered by disc $D(p)$ but not by any other points of $S$. Since a cover set needs to cover all the area, $D(p)$ has to be included in any cover set. Q.E.D. ■

## III. $\alpha$-HULL ALGORITHM FOR THE MINIMUM DISC COVER SET

In Section II, we have established the connection between the minimum disc cover problem and the $\alpha$-hull problem. In [5], the following algorithm is provided to construct the $\alpha$-hull in optimal time complexity $\mathcal{O}(n \log n)$.

```
Algorithm α-hull($S = \{O_0, O_1, O_2, \ldots, O_n\}$)
    Construct Delaunay Triangulations
    Determine the $\alpha$-extreme points of S
    Determine the $\alpha$-neighbors of S
    Output the $\alpha$-hull
```

Because the Delaunay Triangulations contain all the information necessary for the second and third statements, this algorithm efficiently computes the vertices of the $\alpha$-hull and at the same time the minimum disc cover set is obtained. The correctness and analysis of this algorithm can be found in [5]. However, there are two pieces of information missing here. First, since the very first step of the above algorithm is to compute the Delaunay Triangulations, it fails on cases when there are more than 3 disc origins falling on the circumference of a circle. Second, while this algorithm has been shown to be optimal, it does not tell us if the algorithm is still optimal when all points in $S$ fall in one single disc (i.e., $D(O_0)$ in our case). In other words, we do not know if there is a better $\alpha$-hull algorithm with time complexity lower than $\mathcal{O}(n \log n)$ when our extra constraint is present. In Section IV, we will first show the relationship between the minimum disc cover and the boundary of disc union. This relationship can then be used to design an intuitive divide-and-conquer algorithm that works for all cases, including the degenerate ones. In Section V, we will show that there is no algorithm capable of computing minimum disc cover set in time complexity lower than $\mathcal{O}(n \log n)$.

## IV. ALTERNATIVE ALGORITHM FOR THE MINIMUM DISC COVER SET

### A. Theoretical Basis of Our Algorithm

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

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

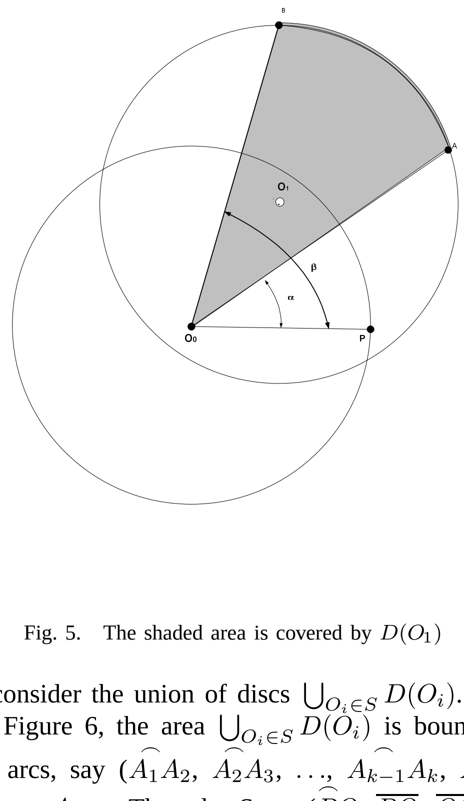

Fig. 5. The shaded area is covered by $D(O_1)$

Now, consider the union of discs $\bigcup_{O_i \in S} D(O_i)$. As illustrated in Figure 6, the area $\bigcup_{O_i \in S} D(O_i)$ is bounded by a series of arcs, say ($\overset{\frown}{A_1 A_2}$, $\overset{\frown}{A_2 A_3}$, ..., $\overset{\frown}{A_{k-1} A_k}$, $\overset{\frown}{A_k A_{k+1}}$), where $A_1 = A_{k+1}$. Then, let Sector($\overset{\frown}{PQ}$, $\overline{PO}$, $\overline{QO}$) be the region bounded by $\overset{\frown}{PQ}$, $\overline{PO}$, and $\overline{QO}$, $\bigcup_{O_i \in S} D(O_i)$ can be viewed as the union of Sector($\overset{\frown}{A_i A_{i+1}}$, $\overline{A_i O_0}$, $\overline{A_{i+1} O_0}$),

$1 \leq i \leq k$. In the following lemma, we show that the discs that contribute arcs $\widehat{A_i A_{i+1}}$ ($1 \leq i \leq k$) form a minimum disc cover set.

*Lemma 2:* Let $\widehat{A_1 A_2}$, $\widehat{A_2 A_3}$, ..., $\widehat{A_{k-1} A_k}$, $\widehat{A_k A_{k+1}}$ be the arcs surrounding $\bigcup_{O_i \in S} D(O_i)$, where $A_1 = A_{k+1}$; and for $1 \leq i \leq k$, let $D(O_{t_i})$ be the disc that contributes $\widehat{A_i A_{i+1}}$. The discs, $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$, form the minimum disc cover set of discs with origins in $S = \{O_0, O_1, O_2, \ldots, O_n\}$.

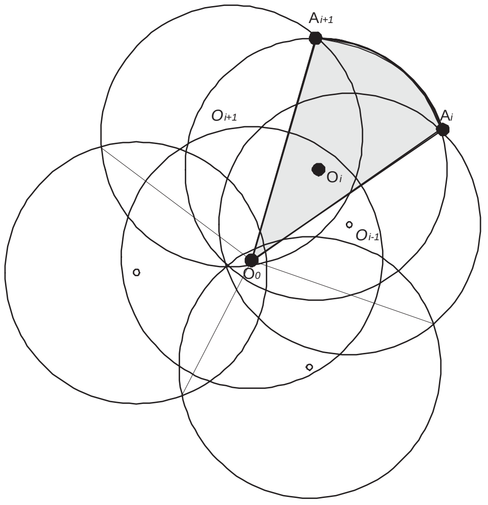

Fig. 6. The boundary of $\bigcup_{O_i \in S} D(O_i)$ is formed by a series of arcs $<\widehat{A_i A_{i+1}}>$

*Proof:* First, we claim that any disc $D(O_{t_i})$ that contributes an arc to the boundary of $\bigcup_{O_i \in S} D(O_i)$ must be included in any disc cover set. For, otherwise, if $\widehat{A_i A_{i+1}}$ is part of the boundary of $\bigcup_{O_i \in S} D(O_i)$ contributed by $O_{t_i}$, the area close to $\widehat{A_i A_{i+1}}$ will not be covered by any disc other than $D(O_{t_i})$.

Now if we can show that $\{O_{t_1}, O_{t_2}, \ldots, O_{t_k}\}$ forms a geometric cover set for $S$, we know that it must be the minimum disc cover set. The area $\bigcup_{O_i \in S} D(O_i)$ can be thought of as a collection of the disjoint small 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 $k$. According to Lemma 1, each of these small pieces is well covered by the disc $D(O_{t_i})$. In other words, the union of $D(O_{t_i})$, where $i$ ranged from 1 to $k$, covers the whole area $\bigcup_{O_i \in S} D(O_i)$. Thus, $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$ is a disc cover set, and also the minimum one. Q.E.D. ■

Notice that a disc $D(O_{t_i})$ may contribute more than one arc to the boundary of $\bigcup_{O_i \in S} D(O_i)$. That implies that the same 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 disc may contribute either zero, one, or two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$.) The possibility of such multiple appearances is the main reason why the minimum disc cover set is nontrivial. This will be further discussed in Section IV-E.

## B. Abstract of Algorithm

According to Lemma 2, finding the minimum disc cover set for discs with origins in $S = \{O_0, O_1, \ldots, O_n\}$ is equivalent to finding the arcs that make up the boundary of $\bigcup_{O_i \in S} D(O_i)$. We develop an efficient algorithm that identifies these arcs.

Since the boundary of the union of a collection of discs is formed by arcs, we represent the boundary by a list of arcs. Given two boundaries (i.e., two arc lists), if we are able to efficiently find the boundary of the union, which is again a list of arcs, we can come up with a divide-and-conquer algorithm for the computation of the minima disc cover set. The abstract of the algorithm is 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($ArcList_1$, $ArcList_2$)
```

Basically **Boundary(**$\{O_0, O_1, O_2, \ldots, O_n\}; i, j$**)** returns the boundary (represented by a list of arcs) of $D(O_0) \cup \bigcup_{i \leq t \leq j} D(O_t)$. Note that $D(O_0)$ is always present and $1 \leq i \leq j \leq n$. In the rest of the paper, we refer $D(O_0) \cup \bigcup_{i \leq t \leq j} D(O_t)$ as super disc $U_D(i, j)$. To compute the minimum disc cover set for all discs with origins in $S$, the algorithm is invoked by **Boundary(**$\{O_0, O_1, O_2, \ldots, O_n\}; 1, n$**)** (i.e., to compute super disc $U_D(1, n)$).

It is a divide-and-conquer algorithm. In order to find the boundary of $Superdisc(i, j) = D(O_0) \cup D(O_i) \cup D(O_{i+1}) \cup \cdots \cup D(O_j)$, we find the boundary of super disc $U_D(i, m) = D(O_0) \cup D(O_i) \cup D(O_{i+1}) \cup \cdots \cup D(O_m)$ and that of super disc $U_D(m+1, j) = D(O_0) \cup D(O_{m+1}) \cup D(O_{m+2}) \cup \cdots \cup D(O_j)$, and then combine the two boundaries, where each boundary is a list of arcs. Evidently, the core of the algorithm is the merger of two arc lists.

## C. Representation of Arc Lists

An arc can be characterized by three parameters, ($O_t$, $\alpha$, $\beta$), where $O_t$ is the origin of the disc that contributes the arc, and $\alpha$ and $\beta$ are the degrees of two angles, which are depicted in Figure 5) and formally defined as follows:

$$\alpha = \begin{cases} cos^{-1}\left(\frac{\overrightarrow{O_0 A} \cdot (1,0)}{\|\overrightarrow{O_0 A}\|}\right), \text{ if } (1,0) \times \overrightarrow{O_0 A} \geq 0 \\ cos^{-1}\left(\frac{\overrightarrow{O_0 A} \cdot (1,0)}{\|\overrightarrow{O_0 A}\|}\right) + \pi, \text{ otherwise} \end{cases}$$

and

$$\beta = \begin{cases} cos^{-1}\left(\frac{\overrightarrow{O_0 B} \cdot (1,0)}{\|\overrightarrow{O_0 B}\|}\right), \text{ if } (1,0) \times \overrightarrow{O_0 B} \geq 0 \\ cos^{-1}\left(\frac{\overrightarrow{O_0 B} \cdot (1,0)}{\|\overrightarrow{O_0 B}\|}\right) + \pi, \text{ otherwise} \end{cases}$$

Notice that the angles of the arc are obtained by using $O_0$ as a reference point, instead of the origin of the disc contributing the arc. In Figure 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 there is an arc $Arc(O_k, \alpha_k, \beta_k)$ in the list that spans across $360°$ (i.e., $\alpha_k < 360°$ and $\beta_k > 360°$), we split it into two arcs $Arc(O_k, 0°, \beta_k)$ and $Arc(O_k, \alpha_k, 360°)$. If the list is sorted based on the value of the first angle (i.e., $\alpha_i$) in ascending order, then $\beta_{s_i} = \alpha_{s_{i+1}}$, $0 \le i \le \max$, with the understanding that $i+1$ is computed module $\max$. 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 $0° = \alpha_{s_0} < \cdots < \alpha_{s_{\max}} < 360°$.

## D. Merging Two Arc Lists

Now suppose we are given two sorted arc lists

$$\begin{cases} ArcList_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_{\max}}, \alpha_{s_{\max}}) \\ ArcList_2 = (O_{t_0}, \alpha_{t_0}, O_{t_1}, \alpha_{t_1}, \ldots, O_{t_{\max}}, \alpha_{t_{\max}}), \end{cases}$$

which represent the boundaries of super disc $U_D(i,m)$ and $U_D(m+1,j)$, respectively. We want to *merge* the two arc lists so that the resulting list represents the boundary of $U_D(i,j) = U_D(i,m) \cup U_D(m+1,j)$.

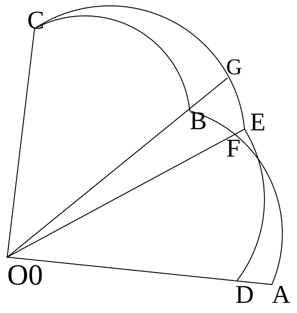

Fig. 7. Splitting $\overset{\frown}{AB}$ into $\overset{\frown}{AF}$ and $\overset{\frown}{FB}$, and splitting $\overset{\frown}{EC}$ into $\overset{\frown}{EG}$ and $\overset{\frown}{GC}$

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., $\alpha$'s). As illustrated in Figure 7, $\overset{\frown}{AB}$ is split into $\overset{\frown}{AF}$ and $\overset{\frown}{FB}$, and $\overset{\frown}{EC}$ is split into $\overset{\frown}{EG}$ and $\overset{\frown}{GC}$. After these two splits, the corresponding pair arcs of arc list $\{\overset{\frown}{AF}, \overset{\frown}{FB}, \overset{\frown}{BC}\}$ and arc list $\{\overset{\frown}{DE}, \overset{\frown}{EG}, \overset{\frown}{GC}\}$ share the same angle span with respect to the reference point $O_0$. Additionally, for instance, if

$$\begin{cases} ArcList_1 = (O_{s_0}, 0°, O_{s_1}, 30°, O_{s_2}, 140°, O_{s_3}, 240°) \\ ArcList_2 = (O_{t_0}, 0°, O_{t_1}, 120°, O_{t_2}, 240°), \end{cases}$$

they will be refined to

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

Now, returning to the general case, suppose we are given two arc lists. After splitting, both lists should contain the same number of arcs:

$$\begin{cases} ArcList_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_k}, \alpha_{s_k}) \\ ArcList_2 = (O_{t_0}, \alpha_{s_0}, O_{t_1}, \alpha_{s_1}, \ldots, O_{t_k}, \alpha_{s_k}), \end{cases}$$

By *merging* the two lists, we simply *merge* each individual pair of corresponding arcs, $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ as follows.

*Case 1:* $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ have no intersection. As illustrated in the left side of Figure 8, one arc is closer to $O_0$ than the other. By merging the two arcs, we means dropping the arc close to $O_0$, keeping only the farther one. In Figure 8, merging the two arcs in the left diagram will result in arc $\overset{\frown}{AB}$.

*case 2:* $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ intersect at a point, say $E$. (See Figure 8 for illustration.) In this case, we drop the two inner sub-arcs and keep the two external ones. Thus, using the example of the right side of Figure 8, merging arc $\overset{\frown}{AB}$ and $\overset{\frown}{CD}$ will result in arc $\overset{\frown}{CE}$ and $\overset{\frown}{EB}$.

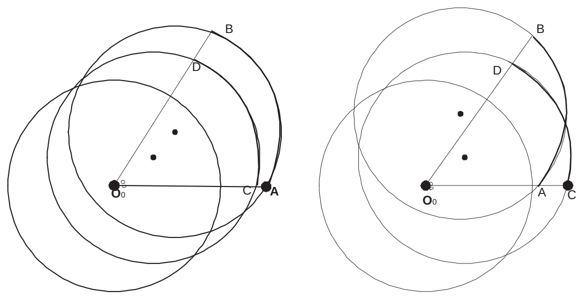

Fig. 8. Case 1 and case 2 when two arcs with the same angle span are merged

## E. Time Complexity of the Proposed Algorithm

In the algorithm, the base case is to compute the arcs for two intersected discs. Since the location of the disc origins and radius $R$ are known, the computation and arc sorting can be accomplished in constant time. In addition, when merging two arcs with the same angle span, any case mentioned in Section IV-D takes only constant time.

We claim that the divide-and-conquer algorithm we proposed is of time complexity $\mathcal{O}\mathcal{O}(n \log n)$. Before we prove it, we first introduce the following lemma.

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

*Proof:* First, we know that the reference point $O_0$ is fixed and all the other disc origins are less than $R$ away from $O_0$. Hence, if we consider the boundary of the union of disc $D(O_0)$ and any other disc, say $D(O_1)$, the arc contributed from $D(O_1)$ must have central angle (with respect to $O_1$) less than $240°$. Let the arc be $\overset{\frown}{AB}$. Now if we want to place a third disc so that it "breaks" the arc $\overset{\frown}{AB}$ into two pieces, the origin of that third disc must be 1) away from both $A$ and $B$

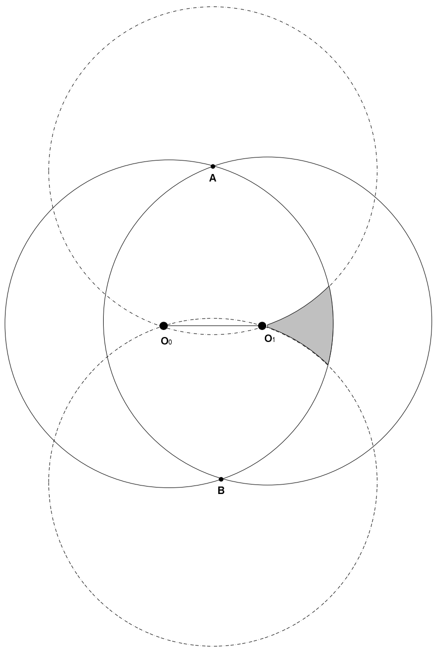

Fig. 9. The possible area for the origin of a third disc that will break $\widehat{AB}$ into two

by more than $R$ and be less than $R$ away from some part of $\widehat{AB}$. It is not hard to see that after adding a third disc with origin inside the shaded area, the remains of the arc $\widehat{AB}$, say $\widehat{AP}$ and $\widehat{AQ}$, must be more than $120^o$ central angle away from each other (with respect to $O_1$). In other words, each time an arc is cut into two pieces by another disc, the resulting smaller pieces must be more than $120^\circ$ central angle (with respect to the origin of the disc contributing the arc) away from each other.

If a disc is able to contribute three arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$, its circumference is cut into arcs by more than three other discs. Based on the above argument, each pair of these three arcs is at least $120^o$ central angle away from each other. By adding up the degree of central angle for the gaps and the degree of the central angle for the arcs themselves, the total would exceed $360^o$, which is not possible. Therefore, a disc can only contribute at most two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$. Q.E.D.

■

Figure 9 shows the arc $\widehat{AB}$ contributed from disc $D(O_1)$. If a third disc is added to cut $\widehat{AB}$ into two arcs, the origin of that third disc must appear inside the shaded region.

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

*Theorem 2:* The divide-and-conquer algorithm described in Section IV-B computes minimum disc cover set has time complexity $\mathcal{O}(n \log n)$, where $n$ is the number of discs in $\Delta$.

*Proof:*

A simple observation of the algorithm skeleton gives us the following formula for the time complexity of our algorithm.

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

Now if we are able to show that the time complexity of procedure $ArcMerge$ is $\mathcal{O}(n)$, $T(n)$ would be $\mathcal{O}(n \log n)$. It is clear that the time complexity to merge two arcs with the same angle span (with respect to $O_0$) is $O(1)$. Since each disc can only contribute at most two arcs, the total number of arcs is bounded by $O(n)$. This means that the time complexity of the procedure *ArcMerge* is bounded by a constant factor of $\mathcal{O}(n)$, which remains to be $\mathcal{O}(n)$, where $n$ is the number of discs in $\Delta$. Q.E.D.

■

## V. Optimal Time Complexity of Minimum disc Cover Set Computation

In [5], it has been shown that the optimal algorithm that constructs the $\alpha$-hull for an arbitrary set of points $S$ has time complexity $\mathcal{O}(n \log n)$, where $n = \|S\|$. However, since in the minimum disc cover set problem, the points in $S$ (i.e., disc origins) are confined in a circle of radius $R$, it is unknown whether or not there exists an algorithm capable of constructing the minimum disc cover set in time complexity lower than $\mathcal{O}(n \log n)$. In this section, we prove that there is no such algorithm. The idea is to show that the element uniqueness problem, which has a lower bound of $\mathcal{O}(n \log n)$ [3], can be reduced in linear time to the disc cover set 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 $S$, where $i \neq j$, such that $a_i = a_j$?

*Theorem 3:* The solution to the minimum disc cover set problem has lower bound $\mathcal{O}(n \log n)$.

*Proof:* We show that the element uniqueness problem can be reduced to the minimum disc cover set problem in $\mathcal{O}(n)$ time. Given a set of non-negative integers $S = \{a_1, a_2, \ldots, a_n\}$, we first do one pass of scanning to find the largest element in $S$, say it is $a_{max}$. Let $\theta = \frac{2\pi}{a_{max}+1}$, we then map each element $a_i$ in $S$ to $p_i = (\cos(a_i \cdot \theta), \sin(a_i \cdot \theta)$, which is a point on the circumference of the unit disc centered at origin $O$. The scan and the mapping take $O(n)$. It is clear that all different points on the circumference must be included in the minimum disc cover set for $\{O, p_1, p_2, \ldots, p_n\}$. Let the size of the resulting minimum disc cover set be $m$. We know that $m \geq n$ means all elements in $S$ are unique. Otherwise, there must be duplicates in $S$.

The reduction from the element uniqueness problem to the minimum disc cover set problem takes $\mathcal{O}(n)$, lower than $\mathcal{O}(n \log n)$. If there exists an algorithm capable of constructing the minimum disc cover in time complexity lower than $\mathcal{O}(n \log n)$, we would have an algorithm to solve the element uniqueness problem in less than $\mathcal{O}(n \log n)$. Hence, we conclude that the lower bound of the minimum disc cover set problem is at least $\mathcal{O}(n \log n)$. Q.E.D.

■

The conversion of an element $a_i$ in $S$ to a point on circumference of the unit disc is illustrated in Figure 10.

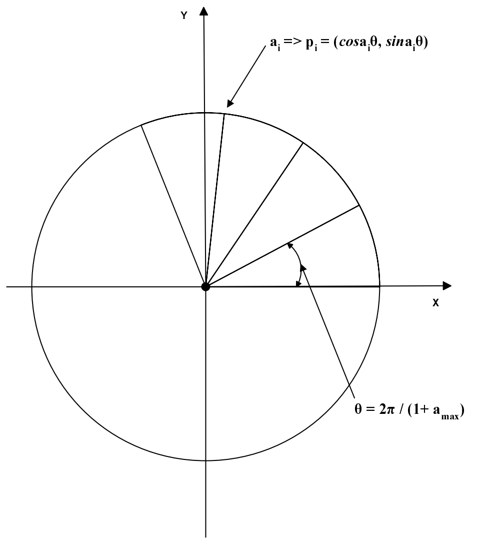
Y
$a_i \Rightarrow p_i = (\cos a_i\theta, \sin a_i\theta)$
X
$\theta = 2\pi / (1+ a_{max})$

Fig. 10. Mapping $a_i$ in $S$ to $P = (\cos(\frac{a_i \cdot 2\pi}{a_{max}+1}), \sin(\frac{a_i \cdot 2\pi}{a_{max}+1}))$

## VI. Conclusion and Future Work

In this paper, we have shown the minimum disc cover set problem as a special case of the $\alpha$-hull problem. Furthermore, we demonstrate that the extra restriction imposed by our minimum disc cover set problem (i.e., all disc origins are inside a circle with the same radius) does not make the problem any easier by proving that no algorithm with lower time complexity than $\mathcal{O}(n \log n)$ can compute the minimum disc cover set. In addition, we provide two different optimal algorithms for the minimum disc cover set construction. The first one is based on the concept of the $\alpha$-hull. However, since the first step of this algorithm is to construct Delaunay Triangulations, it fails on the degenerate cases when more than three disc origins fall on the circumference of a circle. On the other hand, the second divide-and-conquer algorithm we proposed is capable of finding the minimum disc cover efficiently for all circumstances.

Although the proposed algorithms are optimal in terms of time complexity, there is still room for improvement when the algorithm is used for message broadcast or media access control in mobile ad hoc networks. That is, if a node keeps track of the movement and speed of its neighbors, it is able to "predict" exactly when the members of its minimum disc cover set will change and do the computation only when necessary. This further reduces the number of times for a node to compute its minimum disc cover set and thus greatly reduces the energy consumed for such computation. This is known as *Kinetic Collision Detection* model [2]. This option will be explored in our future work.

## References

[1] T. H. Cormen, et al, "Introduction to Algorithms," Cambridge, MA: The MIT Press, 1990.

[2] J. Basch, et al, "Kinetic Collision Detection for Two Simple Polygons," Proc. 10th Symp. Discrete Algorithms, ACM and SIAM, pp. 102-111, Jan. 1999.

[3] D. P. Dobkin and R. J. Lipton, "On the Complexity of Computations Under Varying Sets of Primitives," Journal of Computer and System Sciences, vol. 18 , pp. 86-91, 1979.

[4] M. L. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," San Francisco: W. H. Freeman, 1979.

[5] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the Shape of a Set of Points in the Plane," IEEE Transactions on Information Theory, vol. IT-29, pp. 551-559, 1983.

[6] C. Intanagonwiwat, et al, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," ACM MOBICOM, pp. 56-67, Aug. 2000.

[7] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," ACM MOBICOM, pp. 66-75, Aug. 1998.

[8] S. Y. Ni et al, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," ACM MOBICOM, pp. 151-162, Aug. 1999.

[9] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," Proc. MobiHoc, pp.129-130, Aug. 2000.

[10] F. P. Preparata and M. I. Shamos, "Computational Geometry: An Introduction," Springer-Verlag, 1991

[11] M. Sun, et al, "GPS-Based Message Broadcasting for Inter-vehicle Communications," Proc. IEEE ICPP, pp. 279-286, Aug. 2000.

[12] M. Sun and T. H. Lai, "Location Aided Broadcast in Wireless Ad Hoc Network Systems," Proc. IEEE WCNC, pp. 597-602, Mar. 2002.

[13] M. Sun and T. H. Lai, "Computing Optimal Local Cover Set for Broadcast in Ad Hoc Networks," Proc. IEEE ICC pp. 3291-3295, Apr. 2002.

[14] M. Sun, et al, "Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks," Proc. IEEE ICPP, pp. 527-536, Aug. 2002.

[15] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," Proc. ACM DIAL M, pp. 7-14, Aug. 1999.