

MINIMUM LOCAL DISK COVER SETS FOR BROADCASTING IN HETEROGENEOUS MULTIHOP WIRELESS NETWORKS

GUANING CHEN

Department of Information Management, Kainan University, Taoyuan 33857, Taiwan
gchen@mail.knu.edu.tw

CHIH-WEI YI

Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan
yi@cs.nctu.edu.tw

MIN-TE SUN

Department of Computer Science and Information Engineering, National Central University
Taoyuan 32001, Taiwan
msun@csie.ncu.edu.tw

FANG-CHU LIU

Department of Computer Science, National Chiao Tung University
Hsinchu 30010, Taiwan
vict.cs94g@nctu.edu.tw

WEI-CHI LAN

Department of Computer Science and Information Engineering
National Central University, Taoyuan 32001, Taiwan
100582015@cc.ncu.edu.tw

Received 20 April 2010

Accepted 1 February 2012

Communicated by Sartaj Sahni

The concept of forwarding sets is widely adopted in many broadcast protocols for wireless multihop networks to alleviate the broadcast storm problem. In these protocols, after receiving a broadcast message, each node that is requested to relay the message instructs a subset of its 1-hop neighbors, a.k.a. the forwarding set, to further relay it. In this paper, we propose to use the Minimum Local Disk Cover Set (MLDCS) as the forwarding set in heterogeneous multihop wireless networks, where nodes may have different transmission ranges. We show that the minimum local disk cover set of a node in heterogeneous networks is equivalent to its skyline set, and then we propose a divide-and-conquer algorithm with the optimal time complexity to compute the skyline set locally and statelessly. Moreover, unlike other forwarding heuristics, the proposed algorithm requires only 1-hop neighbor information. This helps to reduce the forwarding set formation latency and thus will be more suitable for environments with a frequently changed network topology, such as vehicular ad hoc networks.

Keywords: Minimum disc cover; optimal algorithms; applications in multihop networks.

1. Introduction

A multihop wireless network consists of a collection of wireless devices networked together in a multihop fashion. Due to no need of fixed infrastructure, multihop wireless networks can be flexibly and quickly deployed for various applications, such as personal area networks, smart home appliances, environmental monitoring, battlefield surveillance, and emergency disaster relief. Each device in such networks may change its communications links to other devices dynamically and frequently due to either nodal mobility or extreme environment conditions.

Network-wide broadcasting is one of the fundamental and natural operations in many networks including multihop wireless networks. It is widely and frequently used to disseminate the information, explore the network topology, discover routing paths, and monitor network integrity. The simplest broadcast mechanism is flooding, where each device retransmits a message when it receives a copy of the message for the first time. Despite its simplicity, flooding generates a large amount of unnecessary retransmissions and as a result introduces serious redundancy and transmission collisions. If this phenomenon is not carefully taken care, the result of this broadcast storm can block all useful network traffic and even meltdown a network. In addition to introducing serious redundancy and transmission collisions, the straightforward flooding in a multihop wireless network environment would introduce more channel contention. In case if devices are mobile, such broadcasting operations are expected to be executed more frequently in a multihop wireless network. All this would quickly consume two most precious resources of the multihop wireless networks: energy and frequency bandwidth. The worst broadcast storm problem in a wireless ad hoc network has been studied by Ni *et al.* [13].

To address the broadcast storm problem, various broadcast algorithms [3, 12, 14–16] have been proposed. In these algorithms, when a node receives a broadcast message, instead of triggering all 1-hop neighbors to relay the message, it selects a subset of 1-hop neighbors, referred to as a *forwarding set* or a multipoint relaying set (MPR), to relay the message. To ensure that a broadcast message can reach all nodes in the network, the broadcasting node selects its relay forwarding set to cover all of its 2-hop neighbors. At the same time, to relieve the broadcast storm problem, the forwarding set should be kept as small as possible. Let X_{u_0} be the 2-hop neighbors of the broadcasting node u_0 . For each 1-hop neighbor u of the broadcasting node u_0 , define $C_{u_0}^u$ as the u_0 's 2-hop neighbors that are 1-hop neighbors of u . Obviously, $\mathcal{F} = \{C_{u_0}^u \mid u \text{ is 1-hop neighbors of } u_0\}$ is a family of subsets of X_{u_0} . For every element x of X_{u_0} , there exists a subset $C_{u_0}^u$ in \mathcal{F} such that x belongs to $C_{u_0}^u$. Therefore, if \mathcal{F} and X_{u_0} are given, a minimum forwarding set of the broadcasting node u_0 is exactly corresponding to a minimum cover set of a classical set-covering problem.

Ignoring geometric factors of wireless communications, Qayyum *et al.* pointed out in [14] that the minimum forwarding set problem on general graphs is NP-complete. Therefore, heuristics are used to find the minimum set cover as the forwarding set. In [14], a greedy set cover heuristic is adopted to select MPR, and without too much surprise, the greedy algorithm is with approximation $O(\log \Delta)$. Here Δ is the maximal cardinality of sets. In [1], MPRs are proposed to control flood messages in mesh networks.

Huson and Sen mentioned in their paper [9] that some restricted graphs, such as tree and planar graphs, are unable to model radio networks but arbitrary graphs fail to capture the structural information of the network which may be used to develop better algorithms. They and others [5, 6, 8, 10, 17, 18] proposed geometric disk graphs to model wireless networks for dealing with various problems, such as frequency assignment, broadcast scheduling, network routing, connected dominating sets and disk covering problems.

Călinescu *et al.* [5] used unit disk graphs to model homogenous radio networks when dealing with selecting forwarding neighbors and finding the minimum disk cover problems. They utilized the geometric representation of 1-hop and 2-hop neighbors to propose heuristics that can give six and three approximations respectively in $O(n \log n)$ and $O(n \log^2 n)$ time, where n is the number of 1-hop and 2-hop neighbors. This improved previous known results by Brönnimann and Goodrich [4] $O(1)$ approximation in $O(n^3 \log n)$ time.

To construct forwarding sets, in most previous works each node needs to collect information of its 2-hop neighborhood. However, in multihop wireless networks, network topology could change frequently and dynamically, and the cost of obtaining fresh 2-hop neighbor information is high. It is a good idea not to take the 2-hop neighbor information into consideration when dealing with multihop wireless networks.

Instead of 2-hop information, Sun *et al.* [17] suggested constructing the forwarding set based on the coverage area of 1-hop neighbors and unit disk graph model. The idea behind it is to ensure nodes in the forwarding set of a broadcasting node to cover the same area as all its 1-hop neighbors. The proposed algorithm is localized, distributed, and with the optimal time complexity $O(n \log n)$. However, the algorithm works only when all nodes in the network have the same transmission range. Nodes in the transmission range of a node X , they can correctly receive and decode packets sent from the node X .

All those researchers either assume all nodes of the modeled wireless networks have the same transmission range (homogeneous networks) or have transmission ranges at least greater than some fixed distance (so called quasi unit disk defined by Kuhn [10]). However, in reality, the transmission ranges of nodes in multihop wireless networks are not necessary equal and are not greater than some fixed distance, either. Thai and Du [18] used bidirectional link disk graphs to model multihop wireless networks for connected dominating sets problem.

In this paper, we extend the work in [17] for homogeneous networks to heterogeneous networks in which nodes may have different transmission ranges. A heterogeneous network topology is modeled by bidirectional link disk graph in which each node is associated with a disk centered at the node and the radius be the transmission range of the node. And two nodes have an edge between them if their distance is no larger than any of their transmission ranges. We use the term employed in Sun's paper the coverage area of a node to mean the transmission range of the node. Since the transmission range of a node is modeled as a disk centered at the node, the coverage area of a network is the union of those disks representing their transmission ranges of the network. If the coverage area of a subset of a network has the same coverage area of the network, then such a subset can be chosen as a forwarding set of a broadcasting node in the network. Therefore, we

can study the broadcasting problem through a coverage area approach. Compared to a node coverage approach, our area coverage approach is a conservative one. Although the forwarding set found through an area coverage approach, such as skyline-based algorithm, may be bigger than that found using a node coverage approach, it can be found using only 1-hop neighboring information when an area coverage is employed. We prove that finding the local minimum disk cover set is equivalent to finding the skyline of the coverage area of 1-hop neighbors. In addition, a localized divide-and-conquer algorithm with the optimal time complexity $O(n \log n)$ is proposed to compute our forwarding set.

The remaining of this paper is organized as follows. In Section 2, the forwarding set problem is formulated as the minimum local disk cover set problem. In Section 3, the equivalence between the minimum local disk cover set and the skyline set is built, and a divide-and-conquer algorithm is provided to find the skyline set. In Section 4, the time complexity of proposed algorithm is derived. In Section 5, the simulation results of performance comparison are presented. The conclusion is given in Section 6.

2. Minimum Local Disk Cover Sets

In what follows, we use $B(x, r)$ to denote the closed disk having center at x and radius r . The boundary of a closed subset $S \subset \mathbb{R}^2$ is denoted by ∂S , and thus, $\partial B(x, r)$ is the circle centered at x with radius r . For any two points x and y , \overline{xy} denotes the line segment between x and y , \overrightarrow{xy} denotes the ray (or called a half line) from x to y , and \overleftrightarrow{xy} denotes the line containing the points x and y . $|A|$ is shorthand for the cardinality of a countable set A .

We assume that wireless nodes are distributed on a two-dimensional Cartesian plane. The topology of heterogeneous wireless network is modeled by bidirectional disk graph. In other words, each node u_i is associated with a transmission range r_i , and two nodes u_i and u_j are said to be neighbors each other if and only if their Euclidean distance $\|u_i - u_j\|$ is no larger than $\min(r_i, r_j)$. Instead of saying two nodes u_i and u_j are neighbors each other, we will say that the node u_j is a 1-hop neighboring node of the node u_i or the node u_i is a 1-hop neighboring node of the node u_j . For a node u_i , its (transmission) coverage is modeled as a disk with center at u_i and radius r_i , i.e., $B(u_i, r_i)$. A node u_j is said to be covered by a node u_i if $u_j \in B(u_i, r_i)$. In this case, the node u_j is also said to be a neighboring node of the node u_i .

For a set of nodes V , we say a subset S of V is a (disk) cover set of V if $\bigcup_{u_i \in S} B(u_i, r_i) = \bigcup_{u_i \in V} B(u_i, r_i)$. If there exists a node u_0 in V such that all other nodes in V are neighbors of that node, then V is called a local set and the corresponding disk set $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$ is called a local disk set. A disk from a local disk set is called a local disk. A cover subset S of a local set V is called a local disk cover set. In the following discussion, without loss of generality, we always assume u_0 is a neighbor of all other nodes in V and is called the broadcasting node. We also assume that the point u_0 is located at the origin of the Cartesian plane and the origin is denoted by \mathbf{o} . The problem of a minimum local disk cover set is formally stated as follows:

PROBLEM 1 Minimum Local Disk Cover Set (MLDCS) Problems

Input: Let $V = \{u_0, u_1, \dots, u_n\}$ be a set of disk centers. $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$ is a corresponding disk set such that for all $i, i = 1, 2, \dots, n, \|u_0 - u_i\| \leq \min(r_0, r_i)$, i.e., $\forall i u_i \in B(u_0, r_0)$ and $u_0 \in B(u_i, r_i)$.

Output: A subset S of V such that $\bigcup_{u_i \in S} B(u_i, r_i) = \bigcup_{u_i \in V} B(u_i, r_i)$.

Measure: $|S|$ is minimal.

To alleviate the broadcast storm problem associated with broadcast protocols, the size of the forwarding set needs to be reduced. On the other hand, to ensure a broadcast can reach all nodes in the network, the selection of the forwarding set of a broadcasting node should guarantee that the message will be sent to all its 2-hop neighbors. Based on the idea used in [17], we shall construct a forwarding set of a node to cover the same area as all its 1-hop neighbors. Since the broadcasting node u_0 with its neighbor set forms a local set, we propose to use the *minimum local disk cover set* (MLDCS) as a forwarding set.

We assume that each node can learn the locations and radii of its neighbors through beacon exchanges. In addition, we define the *skyline* for a set of disks as the boundary of the union of disks in the set. Hence, the skyline of a local disk set $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$ is $\partial(\bigcup_{i=0}^n B(u_i, r_i))$. Obviously, a skyline is a closed set and composed of arcs of circles. The collection of centers of disks that contribute arcs (not just a point) to a skyline is called the *skyline set*. In the next section, we shall show that the MLDCS of a local set is the skyline set of the corresponding local disk set, and thus, we can solve the MLDCS problem for a given local disk set by finding the corresponding skyline. In addition, we propose a localized and stateless algorithm to find the skyline set.

3. Skyline Sets

In this section, we give properties of skylines and build the relation between the MLDCS for a given local set and the skyline set for the corresponding local disk set. We then propose a divide-and-conquer algorithm to compute the skyline set.

3.1. Skylines of disk sets

The following geometric lemma and corollary are used to build the relation between MLDCS for a local set $V = \{u_0, u_1, \dots, u_n\}$ and the skyline set for the corresponding local disk set $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$. Note that due to the bi-directional link model, the intersection of coverage of 1-hop neighbors of broadcasting node u_0 ($=\mathbf{o}$) is not empty. It is trivial that for a local disk cover set S the intersection $\bigcap_{u_i \in S} B(u_i, r_i)$ contains the broadcasting node \mathbf{o} . So, for all $u_i, i = 1, 2, \dots, n$, we have $\mathbf{o} \in B(u_i, r_i)$ since $\|\mathbf{o} - u_i\| \leq r_i$.

Lemma 1. Let V be a local set containing u_0 located at the origin \mathbf{o} and $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$ be its local disk set. For any point a on the disk boundary $\partial B(u_i, r_i)$, the line segment $\overline{\mathbf{o}a}$ is contained in $B(u_i, r_i)$, i.e., $\overline{\mathbf{o}a} \subset B(u_i, r_i)$.

Proof. Note that $\mathbf{o} \in B(u_i, r_i)$. Since $B(u_i, r_i)$ is convex and $\mathbf{o}, a \in B(u_i, r_i)$, the line segment $\overline{\mathbf{o}a} \subset B(u_i, r_i)$. □

An example is shown in Fig. 1. Then, we have the following corollary.

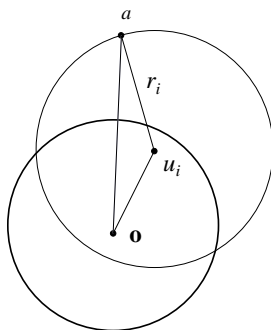


Fig. 1. $\overline{\mathbf{o}a}$ is contained in $B(u_i, r_i)$.

Corollary 2. Let V be a local set containing u_0 located at the origin \mathbf{o} and $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$ be its local disk set. The skyline of this local disk set is composed of a closed sequence of intercepted arcs vertices at the origin.

Proof. First of all, we will show that a ray originated from \mathbf{o} intersects the skyline of the local disk set at exactly one point. Since u_0 , the origin \mathbf{o} , is inside any local disk, any ray R originated from the origin \mathbf{o} intersects the boundary of some local disk at a point a such that the subray R' originated from a is entirely located outside of the skyline except a . Now we claim that the intersection point a must be on the skyline. If not, then a is located in an open set, the interior of the skyline. From the definition of an open set, there are some points other than a are inside the skyline but on the subray $R' - \{a\}$. It is contradictory to the fact that the subray $R' - \{a\}$ is entirely outside the skyline. Next, we only need to show the uniqueness of the intersection point. Assume there were a ray that intersects the skyline at points a and b' . Without loss of generality, we also assume a is farther away from \mathbf{o} than b' . Since a is on the skyline, a is on $\partial B(u_i, r_i)$ for some i . According to Lemma 1, we have $\overline{\mathbf{o}a} \subset B(u_i, r_i)$. This implies b' is inside of $B(u_i, r_i)$ but not on $\partial B(u_i, r_i)$. Therefore, b' cannot be on the skyline. Therefore, we have shown that a ray originated from \mathbf{o} intersects the skyline of the local disk set at exactly one point. Now, if the ray is rotated around the origin one full circle, it will intersect the skyline a closed curve. It is easy to see such curve is formed by intercepted arcs vertices at the origin. Thus the corollary is proved. □

For the purpose of expressing the skyline arc sequence, we split an arc crossing the positive x-axis into two arcs at the intersection point. A skyline arc can be represented by a quadruple $(\alpha_i, u_j, r_j, \alpha_{i+1})$ in which u_j and r_j respectively are the center and radius of the disk contributing this skyline arc, and α_i and α_{i+1} with $\alpha_i < \alpha_{i+1}$ are two polar angles

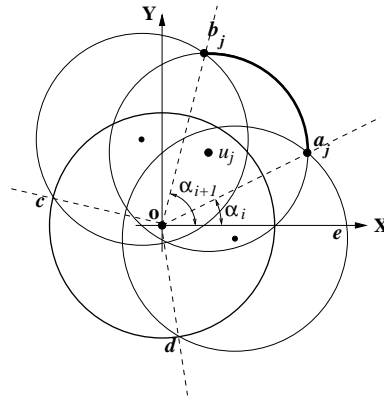


Fig. 2. An arc $a_j b_j$ is represented by 4 parameters $(\alpha_i, u_j, r_j, \alpha_{i+1})$, where u_j and r_j are the center and radius of the disc contributing the arc, and α_i and α_{i+1} are angles corresponding to a_j and b_j observed at o .

corresponding to two endpoints of the skyline arc. An example is given in Fig. 2. Please note that the reference point to measure angles α_i and α_{i+1} is o , not u_i .

A skyline consisting of n arcs are represented as $(\alpha_0, u_{s_0}, r_{s_0}, \alpha_1, u_{s_1}, r_{s_1}, \alpha_2, \dots, \alpha_n)$, where $0 = \alpha_0 < \alpha_1 < \dots < \alpha_n = 2\pi$ and for any $i, i = 1, 2, \dots, n-1, (\alpha_i, u_{s_i}, r_{s_i}, \alpha_{i+1})$ is an arc of the skyline.

Now, the following theorem provides an important relation between the MLDCS of a local set V and the skyline set of the corresponding local disk set.

Theorem 3. For a given local set $V = \{u_0, u_1, \dots, u_n\}$, its MLDCS is the skyline set for the corresponding local disk set $\{B(u_0, r_0), B(u_1, r_1), \dots, B(u_n, r_n)\}$.

Proof. We first prove that the skyline set is a local disk cover set. Assume a skyline is composed of intercepted arcs $a_1 b_1, a_2 b_2, \dots, a_k b_k$ contributed by $B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \dots, B(u_{i_k}, r_{i_k})$, respectively. Let $\triangleleft a_j o b_j$ denote the sector-like area (see Fig. 2) surrounded by line segments $\overline{o a_j}, \overline{o b_j}$, and intercepted arc $a_j b_j$. The covered area $\bigcup_{i=0}^n B(u_i, r_i)$ is equal to the union of sector-like areas $\bigcup_{j=1}^k \triangleleft a_j o b_j$. According to Lemma 1, for each skyline arc $a_j b_j$, we have $\triangleleft a_j o b_j \subseteq B(u_{i_j}, r_{i_j})$. Thus, $\bigcup_{i=0}^n B(u_i, r_i) \subseteq \bigcup_{j=1}^k B(u_{i_j}, r_{i_j})$. This means $\{B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \dots, B(u_{i_k}, r_{i_k})\}$ is a local disk cover set of V .

Next, we prove that this cover set $\{B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \dots, B(u_{i_k}, r_{i_k})\}$ is minimum by claiming that no disks from the local disk cover set can be eliminated to form the new disk cover set of V . Assume that u_i is a center of any disk of the local disk cover set $\{B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \dots, B(u_{i_k}, r_{i_k})\}$. Let a be a point (but not an intersection point) on the skyline arc contributed by the $B(u_i, r_i)$. See Fig. 3.

By the definition of the skyline, a is outside of any disk except $B(u_i, r_i)$. Therefore, for any $j \neq i$, we have $\|u_j - a\| > r_j$. Let $r = \frac{1}{2}(\min_{j \neq i} \|u_j - a\| - r_j)$. For any

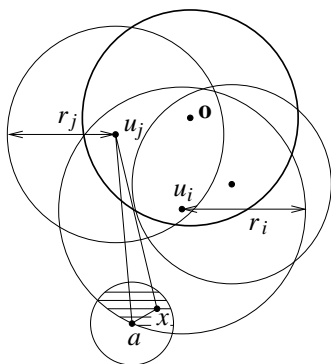


Fig. 3. $B(a, r) \cap B(u_i, r_i)$ is exclusively covered by $B(u_j, r_j)$.

$x \in B(a, r)$ and $j \neq i$,

$$\begin{aligned} \|u_j - x\| &\geq \|u_j - a\| - \|x - a\| \\ &\geq \|u_j - a\| - \frac{1}{2} \left(\min_{j \neq i} \|u_j - a\| - r_j \right) \\ &> \|u_j - a\| - (\|u_j - a\| - r_j) \\ &= r_j. \end{aligned}$$

Thus, for any $j \neq i$, $B(a, r) \cap B(u_j, r_j) = \emptyset$. This implies that $B(a, r) \cap B(u_i, r_i)$ belongs to $B(u_j, r_j)$ but does not belong to any other disk. This means that $\{B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \dots, B(u_{i_k}, r_{i_k})\}$ is a minimum local disk set. So the theorem is proved. \square

3.2. A divide-and-conquer algorithm

According to Theorem 3, computing the MLDCS of a local set is the same as finding the skyline set of the corresponding local disk set. In this subsection, a divide-and-conquer algorithm is proposed to find the skyline set. Recursively, the disk set is divided into two subsets of disks. After skylines of both subsets are found by recursive calls, they are merged to find the skyline of all disks. As stated previously, the reference point to measure angles α_i and α_{i+1} is o . Note that the position u_0 (i.e., o) and the value r_0 are stored as global variables in the *Merge* procedure.

Skyline (DS) is a divide-and-conquer algorithm, and the most of work is done in the procedure *Merge*. There are three steps in *Merge*.

In the first step, two skylines are aligned by splitting arcs such that two skylines have the same angle sequences. For example, assume $SL1 = (\beta_0, u'_1, r'_{u_1}, \beta_1, u'_1, r'_{u_1}, \beta_2, \dots, \beta_k)$ and $SL2 = (\gamma_0, v'_0, r'_{v_0}, \gamma_1, v'_1, r'_{v_1}, \gamma_2, \dots, \gamma_l)$ are two skylines. Let $(\alpha_0, \alpha_1, \dots, \alpha_m)$ be the monotonic sequence of angles such that $\{\alpha_0, \alpha_1, \dots, \alpha_m\} = \{\beta_0, \beta_1, \dots, \beta_k\} \cup \{\gamma_0, \gamma_1, \dots, \gamma_l\}$. Then, $SL1$ and $SL2$ are refined according to angles $\alpha_0, \alpha_1, \dots, \alpha_m$. After that, both lists should have the same angle sequences and numbers of arcs, and we may assume $SL1 = (\alpha_0, u_1, r_{u_1}, \alpha_1, \dots, \alpha_m)$ and $SL2 = (\alpha_0, v_1, r_{v_1}, \alpha_1, \dots, \alpha_m)$.

PROCEDURE 2 *Skyline* ($DS = \{(u_1, r_{u_1}), \dots, (u_n, r_{u_n})\}$)

Require: (u_i, r_{u_i}) represents the center and radius of a disk.

```

if  $|DS| = 1$  then
    return the skyline of  $\{B(u_1, r_{u_1})\}$ 
end if
if  $|DS| > 1$  then
     $DS1 = \left\{ (u_1, r_{u_1}), \dots, \left( u_{\lfloor \frac{n}{2} \rfloor}, r_{u_{\lfloor \frac{n}{2} \rfloor}} \right) \right\}$ 
     $DS2 = \left\{ \left( u_{\lfloor \frac{n}{2} \rfloor + 1}, r_{u_{\lfloor \frac{n}{2} \rfloor + 1}} \right), \dots, (u_n, r_{u_n}) \right\}$ 
     $Skyline1 = Skyline(DS1)$ 
     $Skyline2 = Skyline(DS2)$ 
    return  $Merge(Skyline1, Skyline2)$ 
end if

```

PROCEDURE 3 *Merge* ($SL1, SL2$)

Require: $SL1$ and $SL2$ are skylines.

```

Refine  $SL1$  and  $SL2$  to align arcs in skylines. Then, we may assume  $SL1 = (\alpha_0, u_1, r_{u_1}, \alpha_1, \dots, \alpha_m)$  and  $SL2 = (\alpha_0, v_1, r_{v_1}, \alpha_1, \dots, \alpha_m)$ .
For each  $i, i = 0, 1, \dots, m$ , determine new skyline arcs from  $(\alpha_i, u_i, r_{u_i}, \alpha_{i+1})$  and  $(\alpha_i, v_i, r_{v_i}, \alpha_{i+1})$ .
Combine neighboring skyline arcs that are from the same disk.
return the new skyline

```

In the second step, for each $i, i = 0, 1, \dots, m$, new skyline arcs are determined from $(\alpha_i, u_i, r_{u_i}, \alpha_{i+1})$ and $(\alpha_i, v_i, r_{v_i}, \alpha_{i+1})$ through the following procedure.

Given two arcs (α, u, r_u, β) and (α, v, r_v, β) , we have following three cases to determine the new skyline arc.

- (1) Arcs (α, u, r_u, β) and (α, v, r_v, β) have no intersection. One arc is closer to the origin \mathbf{o} than the other, and the arc closer to the origin \mathbf{o} can not be in the new skyline. For instance, in Fig. 4(c), arc $a'b'$ is the new skyline arc of arcs $a'b'$ and ab between l_1 and l_2 .
- (2) Arcs (α, u, r_u, β) and (α, v, r_v, β) intersect at one point e . Let γ be the angle corresponding to the intersection point. Applying the principle used in case 1, new skyline arcs can be determined from arcs (α, u, r_u, γ) and (α, v, r_v, γ) , and arcs (γ, u, r_u, β) and (γ, v, r_v, β) . For instance, in Fig. 4(c), arcs $b'g$ and ge' are the new skyline arcs of arcs $b'e$ and be' between l_2 and l_5 .
- (3) Arcs (α, u, r_u, β) and (α, v, r_v, β) intersect at two points e, f . Let γ_1 and γ_2 with $\gamma_1 < \gamma_2$ be angles corresponding to intersection points. Applying the principle used in case 1, new skyline arcs can be decided from arcs $(\alpha, u, r_u, \gamma_1)$ and $(\alpha, v, r_v, \gamma_1)$;

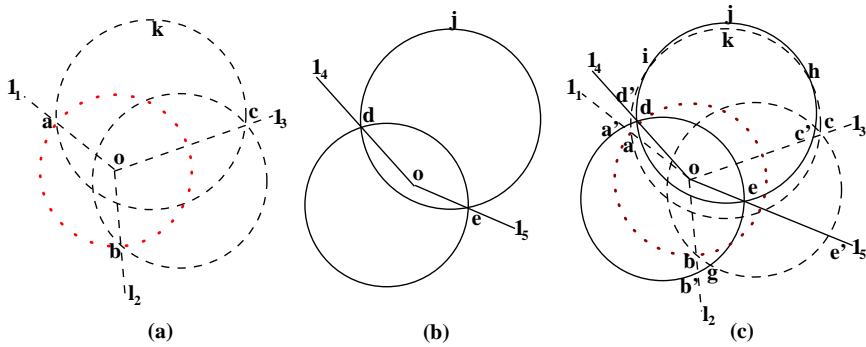


Fig. 4. (a) Skyline one before refined. (b) Skyline two before refined. (c) Merged skyline. In part (c), two arcs (1) between l_1 and l_2 have no intersection, (2) between l_2 and l_5 have one intersection point, (3) between l_3 and l_4 have two intersection points.

arcs $(\gamma_1, u, r_u, \gamma_2)$ and $(\gamma_1, v, r_v, \gamma_2)$; and arcs $(\gamma_2, u, r_u, \beta)$ and $(\gamma_2, v, r_v, \beta)$. For instance, in Fig. 4(c), arcs ch, hji and id' are the new skyline arcs of arcs cd' and $c'd$ between l_3 and l_4 .

In the last step, since one arc may be split into several pieces in the first and/or second steps, we try to combine neighboring skyline arcs if they are from the same disk before returning the new skyline.

4. Time Complexity Analysis

In this section, we show that the time complexity of the proposed algorithm is $\Theta(n \log n)$, where n is the number of disks in a local disk set. The time complexity can be formulated by the following recursive equations:

$$\begin{cases} T(n) = O(1) & \text{if } n = 1, \\ T(n) = 2T(\frac{n}{2}) + f(n) & \text{otherwise.} \end{cases}$$

Here $f(n)$ is the time complexity time of *Merge*. Since $f(n)$ is linear with respect to the number of arcs, and the fact provided in Lemma 11 that the number of arcs of a local disk skyline is at most $2n$, we have $f(n) = O(n)$. Hence, according to the master theorem [7], $T(n) = O(n \log n)$. Leaving the long tedious proof of Lemma 11 at the end of the section, we first state the theorem of time complexity, and then provide related lemmas which support the proof of Lemma 11.

Theorem 4. *The time complexity of Skyline is $\Theta(n \log n)$, where n is the number of disks.*

It has been shown in [17] that the time complexity of the algorithm that computes the minimum local disk cover set for homogeneous networks (i.e., all nodes have the same radius) is $\Omega(n \log n)$. Since homogeneous networks are special cases of heterogeneous networks, the time complexity for the algorithm that computes the minimum local disk cover set for heterogeneous networks is also $\Omega(n \log n)$. Hence, the proposed algorithm is with the optimal time complexity.

4.1. Geometry of local disks

Before we show the main fact that the number of arcs in a skyline of n local disks can not be more than $2n$, we first provide some geometric facts about these disks in the following four lemmas. We will prove the main fact stated as Lemma 11 at the end of the next subsection. From now on, B_i is shorthand for $B(u_i, r_i)$. Let $\{B_0, B_1, \dots, B_n\}$ be a local disk set.

Lemma 5. *Let $U = \{B_1, \dots, B_p\}$ be a subset of a local disk set. If $B_m \in U$ contributes at least three (≥ 3) arcs of the skyline of U , then we can pick three disks B_i, B_j, B_k from $\{B_1, \dots, B_p\} - \{B_m\}$ such that B_m contributes exactly three arcs in the skyline of $\{B_i, B_j, B_k, B_m\}$.*

Proof. Since B_m contributes at least three arcs of the skyline of U , we can choose three skyline arcs from them. Among these three skyline arcs, there are six endpoints which are on the skyline. Now consider the subset $U' (\subseteq U)$ those disks whose boundaries intersect ∂B_m at these six endpoints. The possible number ($|U'|$) of such disks can be three or more since the boundary of each such disk can have at most two intersection points with ∂B_m . If the number of those disks is three, we are done. So, if there are more than three disks, at least one disk intersects ∂B_m exactly at one point on the skyline. If the number of skyline arcs of U' contributed by B_m were decreased when we remove one such disk, the only possibility is that two skyline arcs contributed by B_m are merged and they must intersect the removed disk. This contradicts to that the removed disk intersects ∂B_m exactly at one point on the skyline. Thus, for the skyline formed by the remaining disks, B_m still contributes three skyline arcs, but the number of disks is reduced by one. The skyline arcs contributed by B_m are presented as dashed lines in Figs. 5(a) and 5(b). The figures illustrate a configuration in which B_m contributes three arcs in the skyline of $\{B_1, B_2, B_3, B_4, B_m\}$. The B_m still contributes three arcs in the skyline of $\{B_1, B_2, B_3, B_m\}$ after we remove the disk B_4 whose boundary intersects ∂B_m at exact one point on the skyline. This process

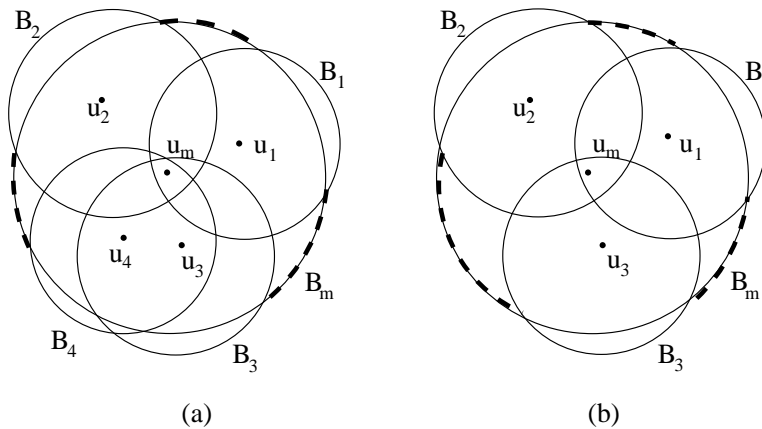


Fig. 5. (a) B_m contributes three arcs to the skyline of $\{B_1, B_2, B_3, B_4, B_m\}$. (b) B_m still contributes three arcs to the skyline of $\{B_1, B_2, B_3, B_m\}$.

can be repeated until the number of the remaining disks is three and B_m contributes exactly three arcs in the skyline of the remaining three disks. Thus the lemma is proved. \square

It is clear that the number of disks in the set U in Lemma 5 should be greater than three. Since if $|U| \leq 3$, the statement “ $B_m \in U$ contributes at least three (≥ 3) arcs of the skyline of U ” can never be true.

Lemma 6. *Let $\{B_1, B_2, B_3\}$ be a subset of a local disk set. The boundaries, ∂B_1 and ∂B_2 , intersect each other at two points a and d . The boundary ∂B_3 containing the points a and d (Here, we mean that points a and d are inside ∂B_3 .) intersects ∂B_1 at points $\{b, e\}$ and intersects ∂B_2 at points $\{c, f\}$. See Fig. 6. A disk B_{k+1} selected from the local disk set is added to the skyline of $\{B_1, B_2, B_3\}$ to form a new skyline of $\{B_1, B_2, B_3, B_{k+1}\}$. If B_{k+1} contributes three arcs to the skyline of $\{B_1, B_2, B_3, B_{k+1}\}$ then the four intersection points $\{b, c, e, f\}$ must be inside the disk ∂B_{k+1} .*

Proof. We prove this lemma by exhaustion on the number of skyline intersection points enclosed by the new skyline arcs. All are claimed to be contradictory to the fact that boundaries of two local disks intersect at most two points if the disk boundary ∂B_{k+1} does not contain the four intersection points $\{b, c, e, f\}$. Figure 6 is provided to aid the proof. The dashed arcs on the figure are some (not all) possible skyline arcs contributed by B_{k+1} .

Fist of all, we assume that the three skyline arcs contributed by B_{k+1} enclose none of the four intersection points $\{b, c, e, f\}$. From the Fig. 6, it is easy to see that these three skyline arcs would be Arcs 1, 3, and 2 or 4. Assume that B_{k+1} contributes Arcs 1, 2, and 3. The non-skyline arc of ∂B_{k+1} between Arcs 1 and 2 would intersect ∂B_2 and ∂B_3 . Therefore, we found that ∂B_{k+1} intersects ∂B_2 at more than two points, two on Arc 1 and one on the above non-skyline arc. The same result can be obtained for the case that B_{k+1} contributes Arcs 1, 3, and 4. This contradicts to the fact that the boundaries of two local disks intersect at two points.

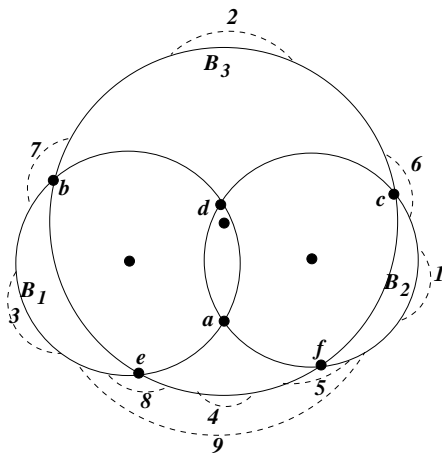


Fig. 6. Possible skyline arcs contributed by B_{k+1} .

Secondly, we assume that the three skyline arcs contributed by B_{k+1} enclose only one of the four intersection points $\{b, c, e, f\}$. Without loss of generality, assume one of the three skyline arcs is Arc 5. So, the remaining two skyline arcs have to come from Arcs 1, 2, 3, and 4. It is easy to see that Arcs 1, 2, or 4 can not be any one of these remaining two skyline arcs since Arc 5 intersects ∂B_2 and ∂B_3 and Arcs 1, 2, and 4 on either ∂B_2 and ∂B_3 . So, the three skyline arcs contributed by B_{k+1} enclose only one of the four intersection points $\{b, c, e, f\}$ is not possible.

Thirdly, we assume that the three skyline arcs contributed by B_{k+1} enclose two of the four intersection points $\{b, c, e, f\}$. There are two situations about the skyline arcs enclosing these points: one situation is that two skyline arcs enclose one intersection point each and the other situation is that one skyline arc, such as Arc 9, encloses two intersection points. It is easy to see that the second situation is not possible. Since the Arc 9 intersecting ∂B_1 and ∂B_2 , Arcs 1 and 3 can not be the skyline arcs. Also since Arcs 2 and 4 intersect the same disk boundary ∂B_3 , both can not be simultaneously chosen as skyline arcs. As for the first situation, the possible skyline arcs choices are $\{3, 5, 6\}$ and $\{1, 7, 8\}$. Using the previous same argument, we derive, in either way, the same contradiction, boundaries of two local disks intersecting more than two points.

Lastly, we assume that the three skyline arcs contributed by B_{k+1} enclose three of the four intersection points $\{b, c, e, f\}$. From the previous argument, no skyline arc can enclose more than one intersection point. Without loss of generality, we assume that the three skyline arcs are Arcs 5, 6, and 7. Using the same argument on Arcs 5 and 7, we derive that ∂B_{k+1} intersects ∂B_3 at three points. Therefore, ∂B_{k+1} must cover the four intersection points $\{b, c, e, f\}$. □

The above lemma can be stated as follows: A four-arc skyline (so it has four intersection points) is formed by three local disks. One of these three local disks contains two intersection points of the other two local disks' boundaries. If a fourth local disk is added and contributes three skyline arcs, then it has to contain the four skyline intersection points.

In addition to forming a four-arc skyline, three local disks can also form a three-arc skyline (so it has three intersection points), where each disk contains one intersection point of the other two disks' boundaries. It can be proved, in the same way, that if adding a fourth local disk which contributes three skyline arcs then it has to contain the three skyline intersection points. Since the proof is very similar to the one of four-arc skyline case, we will only state it as a lemma without proof.

Lemma 7. *Let $\{B_1, B_2, B_3\}$ be a subset of a local disk set. Let a be the intersection point of ∂B_1 and ∂B_2 not in ∂B_3 ; b be the intersection point of ∂B_1 and ∂B_3 not in ∂B_2 ; and c be the intersection point of ∂B_2 and ∂B_3 not in ∂B_1 . See Fig. 7. In order to contribute three arcs, a fourth local disk ∂B_{k+1} must intersect three disks and contain $\{a, b, c\}$.*

Lemma 8. *Let B_1 and B_2 be two disks from a local disk set. Assume their boundaries, ∂B_1 and ∂B_2 , intersect at two points a and d . Let $\overline{ac'}$ and $\overline{ab'}$ be the diameter of B_1 and B_2 , respectively, and let c and b be on the intercepted arc $c'd$ (of inscribed angle $\angle c'ad$) and the intercepted arc $b'd$ (of inscribed angle $\angle b'ad$), respectively. See Fig. 8.*

Int. J. Found. Comput. Sci. 2012.23:1147-1172. Downloaded from www.worldscientific.com by NATIONAL CHIAO TUNG UNIVERSITY on 04/28/14. For personal use only.

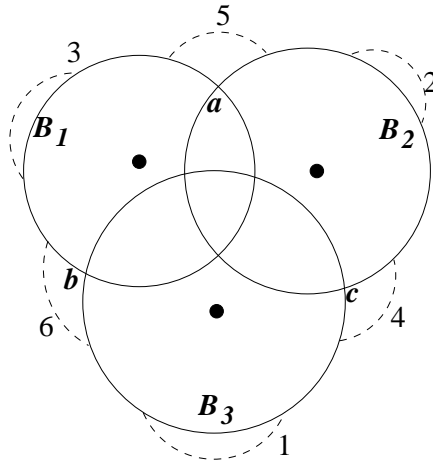


Fig. 7. Possible skyline arcs contributed by B_{k+1} .

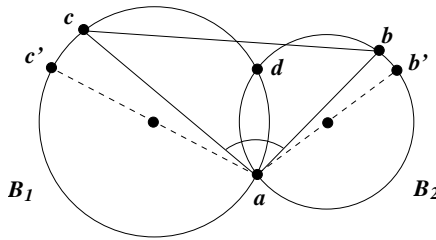


Fig. 8. The structure in Lemma 8.

If the angle $\angle cab$ is obtuse (this implies that two centers of B_1 and B_2 are separated by the line \overleftrightarrow{ad}), we have $\|b - c\| > 2 \min(r_1, r_2)$.

Proof. First, we consider an extreme case in which ∂B_1 and ∂B_2 are tangent at a , i.e., c', a, b' are on a line and d is merged with a and points c' and b' are located on different sides of the tangent line through a . See Fig. 9.

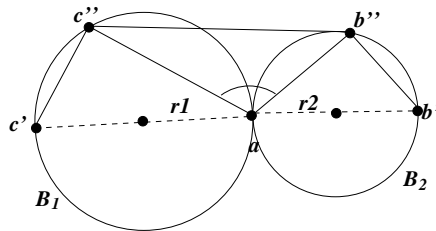


Fig. 9. An extreme case when ∂B_1 and ∂B_2 are tangent.

Since $\angle c'c''a$ and $\angle b'b''a$ are right angles, $\angle c''ab''$ is obtuse, $\angle c''c'a$ greater than $\angle b''ab'$. If $r_1 \leq r_2$, we have

$$\begin{aligned} \|b'' - c''\|^2 &> \|a - c''\|^2 + \|a - b''\|^2 \text{ (law of cosine and } \angle c''ab'' \text{ an obtuse angle)} \\ &\geq \|a - c''\|^2 + \left(\frac{r_2}{r_1} \|c' - c''\|\right)^2 \text{ (} \angle c''c'a \text{ larger than } \angle b''ab' \text{)} \\ &= \|a - c''\|^2 + \|c' - c''\|^2 \\ &\quad + \left(\left(\frac{r_2}{r_1}\right)^2 - 1\right) \|c' - c''\|^2 \\ &= (2r_1)^2 + \left(\left(\frac{r_2}{r_1}\right)^2 - 1\right) \|c' - c''\|^2 \\ &\geq (2r_1)^2 = (2 \min(r_1, r_2))^2. \end{aligned}$$

Similarly, if $r_2 \leq r_1$, we also have

$$\begin{aligned} \|b'' - c''\|^2 &> (2r_2)^2 + \left(\left(\frac{r_1}{r_2}\right)^2 - 1\right) \|b' - b''\|^2 \\ &\geq (2r_2)^2 = (2 \min(r_1, r_2))^2. \end{aligned}$$

Thus, the lemma is correct for this extreme case.

The inequality can be extended for general cases by the following simple observation. Rotate B_1 and/or B_2 of Fig. 9 around a and the positions of points a, b'' and c'' are fixed such that the $\angle c'ab'$ become smaller and closer to the $\angle c''ab''$. But don't let the diameter $\overline{ac'}$ and the diameter $\overline{ab'}$ cross over $\overline{ac''}$ and $\overline{ab''}$, respectively. Let c denote the intersection of the ray $\overrightarrow{ac''}$ and ∂B_1 and b denote the intersection of the ray $\overrightarrow{ab''}$ and ∂B_2 . See Fig. 10.

We have $\|a - b\| \geq \|a - b''\|$ and $\|a - c\| \geq \|a - c''\|$. Thus, $\|b - c\| \geq \|b'' - c''\|$. So, the proof is complete. □

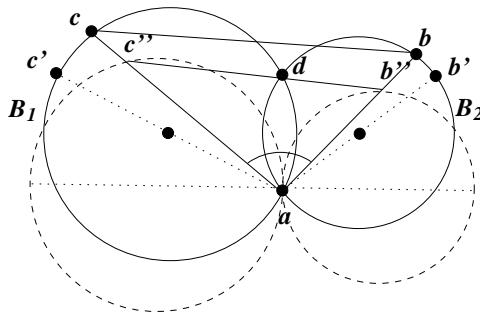


Fig. 10. Rotate B_1 and B_2 around a . c denotes the intersection of the ray $\overrightarrow{ac''}$ and ∂B_1 and b denotes the intersection of the ray $\overrightarrow{ab''}$ and ∂B_2 . Then, we have $\|b - c\| \geq \|b'' - c''\|$.

4.2. Triangle as chords of four circles

Lemma 9. For each edge of an acute or right triangle, a circle is drawn through the two endpoints of the edge (i.e. as a chord of a circle) and its center is outside the triangle and the radius equals to the circumradius of the triangle. Then, the three circles intersect at the orthocenter of the triangle.

Proof. Let Δabc be an acute or right triangle and C_1 be the circumcircle of Δabc . Let \overline{ab} (respectively, \overline{bc} and \overline{ac}) be a chord for C_2 (respectively, C_3 and C_4), a congruent circle of C_1 , with its center outside Δabc . See Fig. 11(a).

To prove this lemma, it is enough to show that the orthocenter of Δabc is on C_2 , C_3 and C_4 . Figure 11(b) illustrates the relation between C_1 and C_2 , and we will prove that the orthocenter is on C_2 . Draw a line from c perpendicular to the segment \overline{ab} intersecting C_2 at d , \overline{ab} at f , C_1 at h and C_2 at e , respectively. Let g be the intersection point of lines \overline{bd} and \overline{ac} . We are going to show \overline{bg} and \overline{ac} are perpendicular, and thus d is the orthocenter of Δabc . Since C_1 and C_2 are congruent and \overline{ab} and \overline{ce} are perpendicular, Δaef and Δacf are congruent. So, $\angle aef = \angle acf$. Since $\angle aed$ and $\angle abd$ are inscribed angles of C_2 and correspond to the same intercepted arc ad , $\angle aed = \angle abd$. Thus, $\angle abd = \angle acf$. In Δdbf and Δdcg , $\angle dbf = \angle dcg$ and $\angle bdf = \angle cdg$, so $\angle bfd = \angle cgd$. Since $\angle bfd = 90^\circ$, $\angle cgd = 90^\circ$. Thus, \overline{bg} and \overline{ac} are perpendicular and \overline{cf} and \overline{ab} are perpendicular, and d is the orthocenter of Δabc . Similarly, we can show that the orthocenter of Δabc is on C_3 and C_4 , too. So, the lemma is proved. \square

Corollary 10. For each edge of an acute or right triangle, draw a circle through the two points of the edge (i.e. as a chord of the circle) and the center is outside the triangle and radius is greater than the circumradius of the triangle. Then, three circles have no intersection.

Now, we are ready to prove Lemma 11. If adding a local disk into a disk set on a plane will only increase the number of skyline arcs by at most two, then the number of skyline

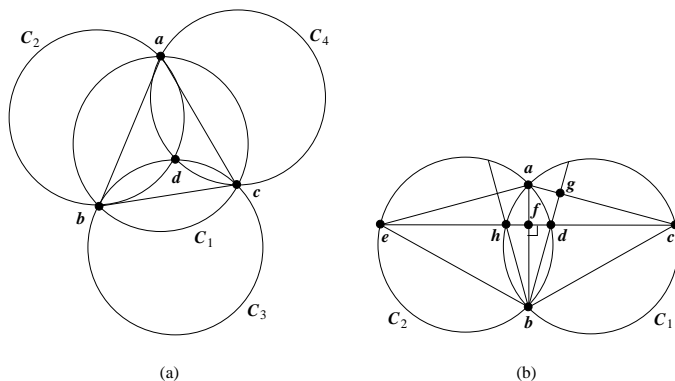


Fig. 11. C_2, C_3, C_4 intersect at the orthocenter of Δabc .

arcs of n disks from a local disk set would be at most $2n$. Since the order of adding disks into the disk set should not change the final skyline, we will prove that lemma by assuming that disks are added into the disk set in a decreasing order of the disk radius.

Lemma 11. *The number of arcs of a skyline of n disks from a local disk set is at most $2n$.*

Proof. We will prove this lemma by mathematical induction on the number of disks n . Without loss of generality, we may assume each disk contributes at least one arc in the skyline and the last added disk is the smallest one.

If $n = 1$, there is only one disk, and thus, the skyline consists of one arc, the boundary of the disk.

If $n = 2$, two boundaries of two disks intersect at most of two points. There are at most two arcs in the skyline. See Fig. 12(a).

If $n = 3$, since the fact that two circles intersect at most two points and assumption that each disk contributes at least one skyline arc, the relationship of three local disks can be categorized into two topologies as shown in Figs. 12(b) and 12(c). In Fig. 12(b), each disk contains one of the intersection points of the other two disks' boundaries, and the skyline is composed of three arcs. In Fig. 12(c), one disk contains two intersection points of the other two disks' boundaries. Note that the case in which three disks have a common intersection point like Fig. 13 is categorized to the first topology.

No matter what, the skyline of three disks is composed of either three or four arcs.

Now, assume that as $n = k (\geq 3)$, the skyline has at most $2k$ arcs. If we can show that after a disk B_{k+1} is added into the set, the number of arcs in the skyline increases at most by two, namely we prove the fact that the number of arcs of the new skyline is no more than $2(k + 1)$ arcs. Assume B_{k+1} contribute at least three arcs. Since B_{k+1} contributes at least three skyline arcs in the skyline of $\{B_1, B_2, \dots, B_{k+1}\}$, by Lemma 5, without loss of generality, we may assume that B_{k+1} contributes three arcs in the skyline of $\{B_1, B_2, B_3, B_{k+1}\}$. From the discussion of $n = 3$, the disks B_1, B_2, B_3 have possible topologies as shown in Fig. 12(b) or 12(c). We will discuss the problem based on these two topologies.

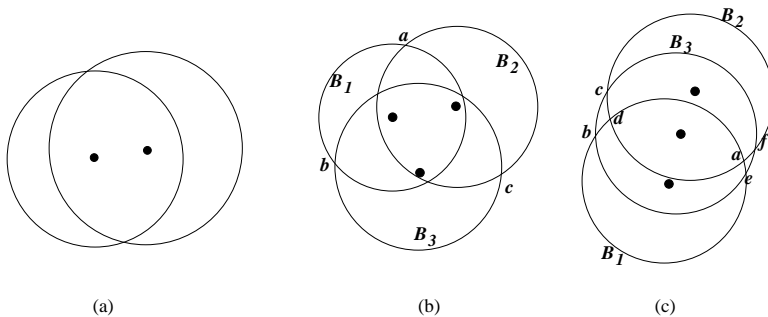


Fig. 12. $n \leq 3$, the skyline contains $2n$ arcs at most. (a) Two disks, (b) and (c) three disks.

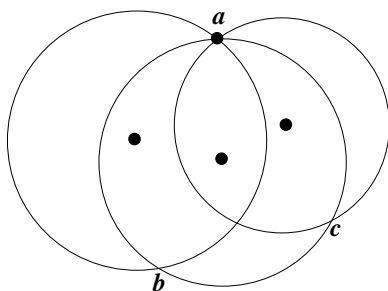


Fig. 13. Three disks have a common intersection point.

Case I: First, we consider the topology like Fig. 12(b). Let a be the intersection point of ∂B_1 and ∂B_2 not in ∂B_3 ; b be the intersection point of ∂B_1 and ∂B_3 not in ∂B_2 ; and c be the intersection point of ∂B_2 and ∂B_3 not in ∂B_1 . In order to contribute three arcs, by Lemma 7, ∂B_{k+1} must intersect three disks' boundaries and contain $\{a, b, c\}$. Now, the problem is discussed according the shape of the triangle Δabc : (1) Δabc is an acute or right triangle; and (2) Δabc is an obtuse triangle.

SubCase I-1: Δabc is an acute or right triangle. Let r_c be the circumradius of Δabc . Since Δabc is an acute or right triangle and B_{k+1} contains a, b, c , we have r_{k+1} is larger than r_c . In addition, since B_{k+1} is the smallest one among B_1, B_2, \dots, B_{k+1} . So, we have $r_c < r_{k+1} \leq r_1, r_2, r_3$. But according to Corollary 10, if r_1, r_2, r_3 are larger than r_c , B_1, B_2, B_3 have no intersection. This is contradictory to the fact that the intersection of B_1, B_2, B_3 is not empty.

SubCase I-2: Δabc is an obtuse triangle. Without loss of generality, we assume $\angle cab$ is obtuse and d is the other intersection point of ∂B_1 and ∂B_2 . Since B_{k+1} must intersect 3 disks' boundaries and contain $\{a, b, c\}$ and $r_{k+1} \leq r_1, r_2, r_3$, degrees of arcs $\{ab, bc, ca\}$ of the skyline of $\{B_1, B_2, B_3\}$ must be larger than π , like Fig. 14(a). If $\overline{ac'}$ is a diameter of B_2 and $\overline{ab'}$ is a diameter of B_1 , b' and c' are on the skyline of $\{B_1, B_2, B_3\}$. c is on the arc $c'd$ and b is on the arc $b'd$. According to Lemma 8, if $\angle cab$ is obtuse, we have $\|b - c\| > 2 \min(r_1, r_2)$. On the other hand, since B_{k+1} contains Δabc , we have $r_{k+1} \geq \frac{1}{2} \|b - c\|$. Thus, we have a contradiction.

Case II: Next, we consider the topology like Fig. 12(c). Without affecting the correctness of following argument, we assume B_3 is the one containing two intersection points of the two boundaries of the other two disks. Let b, e denote intersection points of ∂B_1 and ∂B_3 , and c, f denote intersection points of ∂B_2 and ∂B_3 .

From the description of the relationship of B_1, B_2, B_3 , it is easy to see that the points $\{c, f\}$ and $\{b, e\}$ are located on the opposite of the line \overleftrightarrow{ad} . By Lemma 6, ∂B_{k+1} needs to enclose exactly four intersection points $\{b, c, e, f\}$ in order to contribute three arcs of the skyline of $\{B_1, B_2, B_3, B_{k+1}\}$ like Fig. 14(b). Since B_{k+1} is smaller than B_1, B_2 and has to contain points b, c, e and f , arc be of B_1 outside B_3 and arc cf of B_2 outside B_3 are larger than π . So, the center of B_1 is on the same side of points b, e , and the center of B_2 is on the same side of points c, f . Also, the diameter $\overline{ab'}$ of B_1 and the diameter $\overline{ac'}$ of

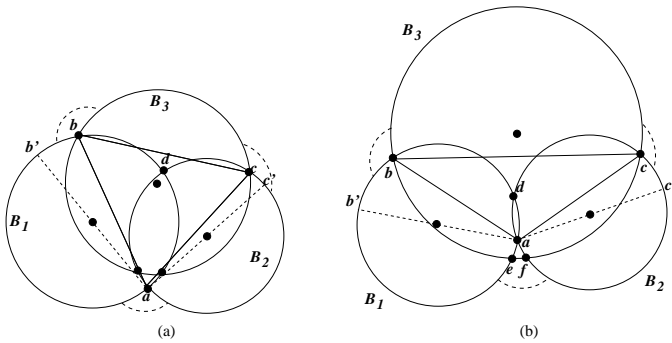


Fig. 14. In this configuration, B_{k+1} does not contribute three arcs. (a) B_{k+1} covers three intersection points and (b) B_{k+1} covers four intersection points.

B_2 are outside of Δabc . Again, since B_{k+1} is smaller than B_3 and has to contain points $\{b, c, e, f\}$, the arc bc of B_3 outside B_1, B_2 is larger than π . So, an inscribed angle $\angle bfc$ (not shown) is greater than $\pi/2$. It is easy to see that the angle $\angle bac$ is obtuse. According to Lemma 8, just like **Case I-2**, we have $2r_{k+1} \geq \|b - c\| > 2 \min(r_1, r_2)$. This is a contradiction.

According to previous discussion, B_{k+1} can not contribute three arcs to the skyline of $\{B_1, \dots, B_{k+1}\}$, and therefore, the number of arcs in the skyline of $\{B_1, \dots, B_{k+1}\}$ is at most $2(k + 1)$. By mathematical induction, we conclude that the number of arcs in the skyline of n disks is upper bounded by $2n$. \square

5. Performance Evaluation

In this section, we compare the performance of the skyline-based algorithm with that of the flooding algorithm and a recently developed greedy algorithm called H2DP[2]. Although the flooding algorithm is considered inefficient use of nodal resources and unnecessary bandwidth consumption, it achieves the highest reachability and requires no network topology information. We use the performance of the flooding algorithm as the lower bound. Dominant Pruning (DP), a well known broadcasting protocol, has been claimed by Lim and Kim [11] as one of the promising approaches that utilizes 2-hop neighboring information to minimize the forwarding sets. H2DP (History-based 2-hop Dominant Pruning) falls in the category of the DP algorithm. H2DP was used in the performance comparison because Agathos and Papapetrou claimed that H2DP is superior to DP in a low mobility (1m/s) environment which is matched to the design goal of the skyline-based algorithm. HDP, an optimization of H2DP, was not used in the performance comparison, because it is designed for highly mobile ($> 20\text{m/s}$) and sparse networks.

In the flooding algorithm, a node will retransmit the broadcast message which is received by the node for the first time. H2DP modifies DP and DP uses 2-hop neighboring information obtained through exchanging “hello” messages. Each rebroadcasting node chooses some of its 1-hop neighbors as forwarding nodes. Only those chosen nodes are allowed to rebroadcast. Nodes inform forwarding neighbors by piggybacking their IDs in

a list in the header of each broadcast message. When a node receives a broadcast message, it checks the header to see if its ID in the list. If so, it uses Greedy Set Cover (GSC) algorithm [7] to determine the forwarding set, given information of which neighbors have already received the broadcast. In addition to the forwarding nodes, H2DP also piggybacks the Packet History in the header of each broadcast message. The Packet History records all the nodes that have received the broadcast message as the message is broadcasted from the source node moving forward toward the current node. Using the Packet History, H2DP is able to eliminate some 2-hop neighboring nodes which have already received the broadcast message. Therefore, each node can build its own forwarding set by GSC approach with less 2-hop neighboring nodes needed to receive the broadcast message. In addition to the method of creating forwarding sets, the skyline-based algorithm also differs from DP in that the nodes in the forwarding set created by the skyline-based algorithm cover the transmission areas of all 1-hop neighbors. Also, the skyline-based algorithm does not need to collect neighbors' 1-hop neighbor information. Furthermore, the skyline-based algorithm does not need the Packet History information used by H2DP.

In order to create forwarding set, each node using either the skyline or H2DP algorithm needs to collect its 1-hop and/or 2-hop neighbor information. Such neighbor information may be stored in beacon (or hello) messages and exchanged periodically. Changing the structure of beacon (or hello) message or adjusting the timing period of sending beacon (or hello) message can affect the simulation results differently. To avoid this artificial effect, we will not consider the creation and sending of beacon (or hello) messages (the effect of MAC layer) during the collection of values of *delay* and *runtime* in the simulation. Therefore, when define *Average delay* and *Average runtime* below, we assume that each node has already received all necessary beacon (or hello) messages from all its 1-hop neighboring nodes. However, the performance impact of those beacon (or hello) messages cannot be ignore, they will be collected separately as another performance metric.

The following performance metrics will be used to evaluate these broadcasting algorithms:

- *Average number of retransmissions*: The number of retransmissions of a broadcast message for an algorithm is defined to be the total number of transmissions of the broadcast message until it is received by all nodes in the network.
- *Average delay*: The delay of a broadcast message for an algorithm is defined as the number of transmissions (or hops) needed for a message broadcasted in the network from the source node to the last node which receives the broadcast for the first time. As mentioned above, this *delay* excludes the time to collect the necessary neighboring information for creating the forwarding set for each node.
- *Average runtime*: The runtime analysis of the skyline-based algorithm has been provided in Section 4. To align with the analysis, we only compare the runtime among algorithms of computing the forwarding set of a given node in its 2-hop neighborhood as did in the analysis. As mentioned above, this *runtime* excludes the time to collect the necessary neighboring information for creating the forwarding set for each node.
- *Average number of beacon messages*: The number of beacon messages is defined to be the total number of beacon (or hello) messages exchanged in a 2-hop neighborhood of a

given node. Those messages are necessary for some algorithms to create the forwarding sets of nodes in a 1-hop or 2-hop neighborhood.

5.1. Simulation setting and assumptions

We use C++ as our simulation tool. The simulated network is generated by randomly placing nodes over an area of square that has side length of 40 units. The area is wrapped both vertically and horizontally to eliminate the edge effect. Each node has a unique ID and is randomly assigned a transmission radius from the range of one and ten. We generate 500 networks for each of the five different node sets, 100, 200, 300, 400, and 500 nodes in the simulated area, respectively. For the case of runtime comparison simulation, we only consider 2-hop neighboring nodes of the broadcasting node in the simulated area. Node 0 is set as the broadcasting node and randomly deployed. We utilize the flooding algorithm to verify whether the generated network is connected or not. One broadcast message was produced and flooded to all nodes in the network from Node 0. If the generated network is not connected, it is discarded and a new one will be generated to ensure the connectivity of the simulated network.

5.2. Simulation results and analysis

5.2.1. Average number of retransmissions

The first performance metric, message broadcasting overhead, is considered in this subsection. The overhead simulation results including average retransmission times with

Table 1. The mean retransmission times, percent changes (Skyline-mean-change/H2DP-mean), and 95% Confidence Intervals (CI) for H2DP and skyline.

Node Density	100	200	300	400	500
H2DP-mean	47.8	75.4	92.6	106.6	119.4
Skyline-mean	64.9	87.1	99.1	110.1	121.6
percent change	36%	16%	7%	3%	2%
H2DP-95% CI	[46.9, 48.6]	[74.7, 76.0]	[92.1, 93.2]	[105.8, 107.3]	[118.7, 120.0]
Skyline-95% CI	[64.5, 65.3]	[86.5, 87.6]	[98.5, 99.7]	[109.5, 110.6]	[120.9, 122.2]

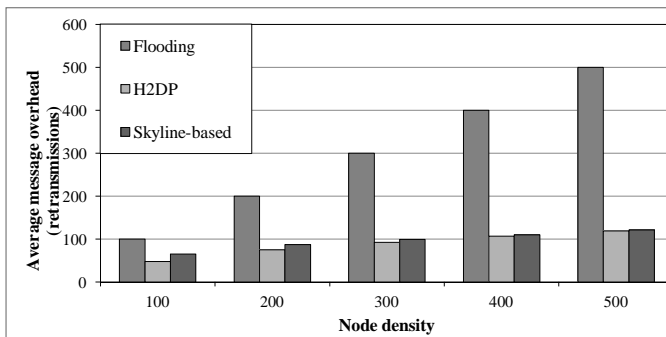


Fig. 15. Overhead performance Comparison of Skyline, H2DP, and flooding.

their percent changes of skyline-based over H2DP and 95% Confidence Intervals (CI) for skyline-based and H2DP algorithms at different node densities are listed in Table 1. The narrower confidence intervals with 95% confidence level imply high precision of our simulation results. It is clear that the number of retransmission times of a broadcast message equals the number of nodes deployed in the network for the flooding case. For a better visualization, the overhead simulation results are also plotted as a bar chart in x-y plane and shown in Fig. 15. The x-axis is the number of nodes deployed in the simulated area. The y-axis is the average number of retransmissions. The fewer retransmissions a broadcast algorithm has, the better performance it has. From the bar chart graph, we see that skyline-based and H2DP algorithms have similar overhead trend and the H2DP algorithm generates slightly less retransmissions than the skyline-based algorithm does. However, from the values of percent changes listed in table, the average numbers of retransmissions of skyline-based algorithm tend to be reduced and towards to that of H2DP. Finally, as expected, both skyline-based and H2DP algorithms have a much lower number of retransmissions compared with the flooding counterpart.

5.2.2. Average delay

In this subsection, we study the delay performance among three algorithms. The delay simulation results including the average delay times and their 95% CI for algorithms at different node densities are collected in Table 2. And its corresponding bar chart graph is shown in Fig. 16. Although the skyline-based algorithm has less delay time than that of the H2DP algorithm, there is not much different between them. However if we add the

Table 2. The mean delay times and 95% CI for all algorithms.

Node Density	100	200	300	400	500
Flooding-mean	6.04	4.78	4.52	4.48	4.39
H2DP-mean	7.68	5.85	5.30	5.09	4.99
Skyline-mean	6.06	4.93	4.70	4.70	4.67
Flooding-95% CI	[5.94, 6.14]	[4.72, 4.84]	[4.46, 4.57]	[4.43, 4.53]	[4.34, 4.44]
H2DP-95% CI	[7.48, 7.87]	[5.75, 5.94]	[5.23, 5.36]	[5.02, 5.16]	[4.93, 5.05]
Skyline-95% CI	[5.96, 6.16]	[4.87, 4.99]	[4.64, 4.75]	[4.64, 4.76]	[4.62, 4.73]

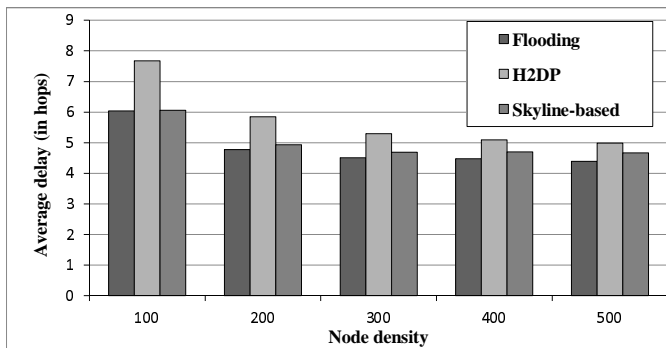


Fig. 16. Delay Performance Comparison of Skyline, H2DP, and flooding.

impact of the collection of 1-hop and 2-hop neighboring information shown in Table 4 of Subsection 5.2.4, the actual delay of the skyline-based algorithm should be much shorter than that of the H2DP algorithm.

5.2.3. Average algorithm runtime

In this subsection, we examine the average runtimes among those algorithms in building a forwarding set for a rebroadcast. For the flooding algorithm, a node will retransmit the broadcast message which is received by the node for the first time, therefore, there is no need to build any forwarding set. The runtime simulation results including the average runtimes with their ratios and the 95% CI for other two algorithms at different node densities are recorded in Table 3. And its corresponding bar chart graph is shown in Fig. 17. The x-axis is the average number of neighbors within a 2-hop neighborhood, and the y-axis is the average runtime in milliseconds. Please note that the recorded values themselves are not so important since they depend on the system processor and I/O used in the simulations. The important is the ratio between the two values. The simulation results ($\approx 0.0780 (n \log n)$) of skyline-based algorithm do match our analysis done in Section 4. The average runtime of the H2DP algorithm is less and about 60% of that of the skyline-based algorithm in building a forwarding set in our simulation environment. This is concluded without considering the impact of the creation and sending of beacon (or hello) messages for calculating forwarding sets. Taking such impact shown in Table 4 of Subsection 5.2.4 into consideration, it is easy to see that the skyline-based algorithm should have a better runtime performance even these two performance metrics (or units) are different.

Table 3. The mean runtimes (ms), Ratios, and 95% CI for H2DP and Skyline.

Node Density	5	10	15	20	25
H2DP-mean	0.181	0.427	0.797	1.282	1.770
Skyline-mean	0.257	0.742	1.352	2.132	2.917
H2DP-mean/Skyline-mean	0.71	0.58	0.59	0.60	0.61
H2DP-95% CI	[0.177, 0.186]	[0.418, 0.435]	[0.785, 0.808]	[1.264, 1.300]	[1.747, 1.792]
Skyline-95% CI	[0.246, 0.268]	[0.727, 0.756]	[1.335, 1.369]	[2.111, 2.154]	[2.900, 2.935]

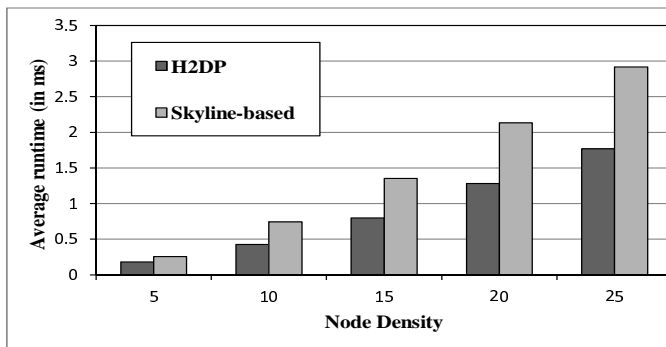


Fig. 17. Runtime Performance Comparison of Skyline and H2DP.

Table 4. The mean number of beacons and 95% CI for H2DP and Skyline.

Node Density	5	10	15	20	25
H2DP-mean	280.7	1516.3	4203.1	8684.2	15394.5
Skyline-mean	70.1	178.3	296.8	431.5	566.3
H2DP-95% CI	[274.1,287.3]	[1482.8,1549.8]	[4117.5,4288.6]	[8512.1,8856.2]	[15109.6,15679.4]
Skyline-95% CI	[69.3,70.8]	[176.5,180.1]	[293.9,299.8]	[427.6,435.4]	[561.0,571.6]

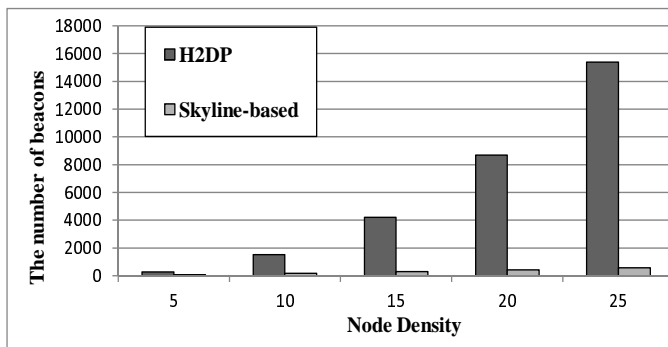


Fig. 18. Beacon Message Performance Comparison of Skyline and H2DP.

5.2.4. Average number of beacon messages

In this subsection, we collect the beacon (or hello) messages under the runtime simulations. Since there is no need to build any forwarding set for the flooding algorithm, we do not have to collect the beacon (or hello) messages. The results including the average number of beacon (or hello) messages and their 95% CI for other two algorithms at different node densities are recorded in Table 4. And its corresponding bar chart graph is shown in Fig. 18. The x-axis is the average number of neighbors within a 2-hop neighborhood, and the y-axis is the average number of beacon (or hello) messages. As can be seen that the skyline-based algorithm has much less average number of beacons needed than that of the H2DP algorithm in building a forwarding set in our simulation environment.

6. Conclusions and Future Direction

The minimum local disk cover set can be used as a forwarding set in a multihop wireless network to alleviate the broadcast storm problem without sacrificing the functionality of the broadcasting. In this paper, we have established the equivalence of the MLDCS for a neighbor set and the corresponding skyline set in heterogeneous multihop networks. We propose a divide-and-conquer algorithm, skyline, to compute the MLDCS, and show that the optimal time complexity of skyline is $O(n \log n)$. Instead of 2-hop neighbor information, MLDCS only need 1-hop neighbor information to select a forwarding set.

Simulation results show that the skyline-based algorithm requires slightly more re-transmissions and higher runtime than that of H2DP algorithm. However, both algorithms have the similar performance in the propagation delay. The above are concluded without

considering the impact of the creation and sending of beacon (or hello) messages for calculating forwarding sets. Taking such impact shown in Table 4 into consideration, it is not hard to see that the skyline-based algorithm should have an overall better performance. Moreover, in an environment with a frequently changed network such as mobile ad hoc networks, those algorithms which require the collection of 2-hop neighbor information may induce other performance overhead, such as the delay latency for information gathering. They will also be more difficult to collect and update 2-hop information. On the other hand, the proposed MLDCS only needs 1-hop neighbor information, therefore, it will be more easier to implement and will perform better than H2DP in mobile ad hoc networks.

Notice that a node may receive the message from another node but not the other way around. Namely, they are not neighbors of each other. Such type of unidirectional links are omitted in our bidirectional link model and therefore the construction of the forwarding set is not discussed in the study. We will investigate how to utilize such type of one-way neighbors to better distribute the power consumption of broadcasting in the future.

References

- [1] S. Abraham *et al.*, “802.11 TGs simple efficient extensible mesh (SEE-Mesh) proposal,” *Tech. Report Document IEEE 802.11-05/0562r2*, November 2007.
- [2] S. Agathos and E. Papapetrou, “Efficient broadcasting using packet history in mobile ad hoc networks,” *The Institution of Engineering and Technology*, vol. 5, pp. 2196–2205, 2011.
- [3] K. Alzoubi, P.-J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless ad hoc networks,” in *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS’02)*, vol. 9, pp. 3849–3855, January 2002.
- [4] H. Brönnimann and M.T. Goodrich, “Almost optimal set covers in finite VC-dimension,” in *Proceedings of the 10th annual ACM Symposium on Computational Geometry*, pp. 293–302, June 1994.
- [5] Gruiă Călinescu, Ion I. Măndoiu, Peng-Jun Wan, and Alexander Z. Zelikovsky, “Selecting forwarding neighbors in wireless ad hoc networks,” *Mobile Networks and Applications*, vol. 9, no. 2, pp. 101–111, 2004.
- [6] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.
- [7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* 2nd edn. (The MIT Press, 2001).
- [8] William K. Hale, “Frequency assignment: Theory and applications,” in *Proceedings of the IEEE*, vol. 68, no. 12, pp. 1497–1514, December 1980.
- [9] Mark L. Huson and Arunabha Sen, “Broadcast scheduling algorithms for radio networks,” in *Proceedings of the IEEE Military Communications Conference (MILCOM’05)*, vol. 2, pp. 647–651, November 1995.
- [10] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger, “Ad hoc networks beyond unit disk graphs,” *Wireless Networks*, vol. 14, no. 5, pp. 715–729, 2008.
- [11] H. Lim and C. Kim, “Flooding in wireless ad hoc networks,” *Computer Communications*, vol. 24, no. 23-4, pp. 353–33, 2001.
- [12] W. Lou and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–122, 2002.
- [13] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The Broadcast Storm problem in a mobile ad hoc network,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 151–162, August 1999.

- [14] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks," in *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02)*, vol. 9, pp. 3866–3875, January 2002.
- [15] I. Stojmenovic, S. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.
- [16] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," Rutgers University, Tech. Rep. CAIP Technical Report 248, September 2000.
- [17] Min-Te Sun, Chih-Wei Yi, Chuan-Kai Yang, and Ten H. Lai, "An optimal algorithm for the minimum disc cover problem," *Algorithmica*, vol. 50, no. 1, pp. 58–71, 2008.
- [18] My T. Thai and Ding-Zhu Du, "Connected dominating sets in disk graphs with bidirectional links," *IEEE Communications Letters*, vol. 10, no. 3, pp. 138–140, 2006.