Optimal Jammer Placement in the Real Plane to Partition a Wireless Network

Jixin Feng, Warren E. Dixon, Tan F. Wong, and John M. Shea
University of Florida, Gainesville, FL,
Email:{fengjixin@ufl.edu, wdixon@ufl.edu, twong@ece.ufl.edu, jshea@ece.ufl.edu}

***Abstract*—We consider the problem of jammer placement to partition a wireless network, where the network nodes and jammers are located in the real plane. In previous research, we found optimal and suboptimal jammer placements by reducing the search space for the jammers to the locations of the network nodes. In this paper, we develop techniques to find optimal jammer placements over all possible jammer placements in the real plane. Our approach finds a set of candidate jammer locations (CJLs) such that a jammer-placement solution using the CJLs achieves the minimum possible cardinality among all possible jammer placements in the real plane. The CJLs can be used directly with the optimal and fast, suboptimal algorithms for jammer placement from our previous work.**

I. Introduction

Jamming attacks for wireless networks have been an active area of research for many years [1]–[6]. The majority of this work focuses on either jamming single communication links or considers a flat network structure. Recent work has considered jammer placement problems that take into account the topology of a wireless network. Ref. [7] places jammers to disrupt network traffic flows, while [8] considers node mobility techniques for jamming of flows and anti-jamming.

In a mesh or ad hoc network, jamming attacks that target individual links may fail because these networks often have a self-healing property because ad hoc routing can be used to reroute traffic around jammed links. Thus, in our previous research [9]–[12], we have focused on jamming attacks that can partition a wireless network into multiple disconnected subnetworks. The problem of jammer placement to partition a wireless is complicated by several facts. First, connectivity is a global property of a wireless network that depends on the topology of the network. Second, the set of wireless links that will be blocked by a jammer is highly sensitive to the particular location of that jammer in Euclidean space. Third, the search space for jammer locations is the entire Euclidean space being considered – for our work, we have constrained the search to the real plane. In light of these issues, in our previous work we have reduced the complexity of finding optimal jammer placements by constraining the potential jammer locations to the locations of nodes in the network being jammed. However, this may make the jammer placement suboptimal and is also not realistic, as in many real scenarios, the jammer locations must be kept secret from the communicators.

This research was supported by the National Science Foundation under grants 1217908, 1320086, and 1642973.

In this paper, we develop a technique to find a set of jammer locations when the jammers can be placed at any locations in the real plane. Our approach is based on the observation that it is not necessary to consider many possible jammer locations because they are either:

- suboptimal, because there are better locations that jam a proper superset of the jammed nodes if a jammer were placed at one of those locations, or
- redundant, because there are other jammer locations that jam the same set of nodes.

The main contributions of this work are: 1) We present two algorithms that find a set of candidate jammer locations (CJLs) in the real plane based on finding minimum covering disks in a depth-first search through the network. 2) We prove that an optimal jammer placement in the CJLs is also optimal among all jammer placements in the real plane, in the sense that no jammer placement in the real plane can achieve a lower cardinality for the objective of partitioning the network to some specified degree.

The network model, jammer model, and jammer placement objective are describe in Section II. In Section III we present two algorithms for finding the CJLs and prove the optimality of the CJLs for the jamming problem we consider. In Section IV, we review algorithms to find optimal or suboptimal jammer placements from a discrete set of locations. The performance of these algorithms is assessed via simulation results in Section V. The paper is concluded in Section VI.

II. System Model and Problem Formulation

A. Network and Jammer Model

We consider communication networks in the real plane, where the network can be modeled as a simple Euclidean graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with the radios as the vertices $\mathcal{V}$ and the communication link as the edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each $v_i \in \mathcal{V}$ has an associated location $(x_i, y_i) \in \mathbb{R}^2$, and nodes v_i and v_j share an edge if radios i and j can communicate directly. For convenience, we let the Euclidean distance between two vertices v_i and v_j with locations (x_i, y_i) and (x_j, y_j), respectively, be denoted by $d_E(v_i, v_j)$. This model can be used for a variety of wireless nonfading and fading channel models. For the purposes of this paper, we will consider a protocol model, where $(v_i, v_j) \in \mathcal{E}$ if $d_E(v_i, v_j) \leq r_c$.

We consider the effect of the placement of jammers on the topology of the communication network. Following our

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

previous work [9], [10], [12], we also use a protocol model for the effect of the jammer, such that there is a jamming radius r_j such that any radio at a distance less than or equal to r_j of a jammer will be unable to communicate with its neighbors.

B. Jammer Placement Problem Formulation

The jammer placement problem can be formulated as follows. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the network before jammer placement and let

$$\mathcal{J} = \left\{ J_1, J_2, \ldots, J_{N_J} \mid J_i \in \mathbb{R}^2,\ i = 1, 2, \ldots, N_J \right\}$$

be a set of jammer placements when there are N_J jammers. Let $\mathcal{H} = \mathcal{H}(\mathcal{V}', \mathcal{E}'; \mathcal{G}, \mathcal{J})$ be the residual network that is left after jammers at positions in $\mathcal{J}$ disrupt the communications in the original network $\mathcal{G}$. Then $\mathcal{H}$ is specified by its remaining vertices $\mathcal{V}'$ and edges $\mathcal{E}'$, which can be determined from

$$\begin{aligned} \mathcal{V}' &= \mathcal{V} \backslash \{ v \in \mathcal{V} \mid d_E(u, v) \leq r_j, u \in \mathcal{J} \} \\ \mathcal{E}' &= \mathcal{E} \backslash \{ (u, v) \mid u \in \mathcal{V} \backslash \mathcal{V}' \text{ or } v \in \mathcal{V} \backslash \mathcal{V}' \}. \end{aligned}$$

For the remaining nodes in $\mathcal{V}'$, we aim to find a partition

$$\begin{aligned} &\Gamma_K(\mathcal{H}) = \left\{ \mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_K \subset \mathcal{V}' \right\} \\ &\text{s.t. } 0 < |\mathcal{V}_i| \leq b_i && i = 1, \ldots, K \\ &\quad \mathcal{V}_i \cap \mathcal{V}_j = \emptyset && i \neq j \\ &\quad \{ (u, v) \mid u \in \mathcal{V}_i, v \in \mathcal{V}_j, i \neq j \} \cap \mathcal{E}' = \emptyset && (u, v) \in \mathcal{E}. \end{aligned}$$

Here, K is the minimum number of disconnected clusters and b_i bounds the number of nodes in cluster i. To simplify exposition, we bound the residual cluster cardinalities by

$$b_K = \frac{|\mathcal{V}|}{K}$$

.

Let $\mathcal{J}_O$ be the jammer placement set with minimum cardinality that achieves the specified partitioning goal. The generalized jammer placement problem is defined on network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and number of cluster K as finding:

$$\begin{aligned} \mathcal{J}_O &= \arg \min_{\mathcal{J} \in (\mathbb{R}^2)^{N_J}} |\mathcal{J}| \\ \text{s.t. } &\exists \Gamma_K(\mathcal{H}(\mathcal{V}', \mathcal{E}'; \mathcal{G}, \mathcal{J})) \\ &b_i = b_K = \frac{|\mathcal{V}|}{K}. \end{aligned}$$

An example illustrating jammer placement to partition a network of 100 nodes into two residual subnetworks with fewer than 50 nodes each is shown in Fig. 1. The example network is created as a random geometric graph, with an edge between any two nodes within a fixed communication distance. A node is jammed if it is within the same communication distance of a jammer. Two jammers, located at the center of the large shaded circles, are sufficient to partition this network. The jammers disrupt communication at all of the red/circle nodes, and result in two residual partitions, indicated by blue/square and green/triangle nodes, respectively.

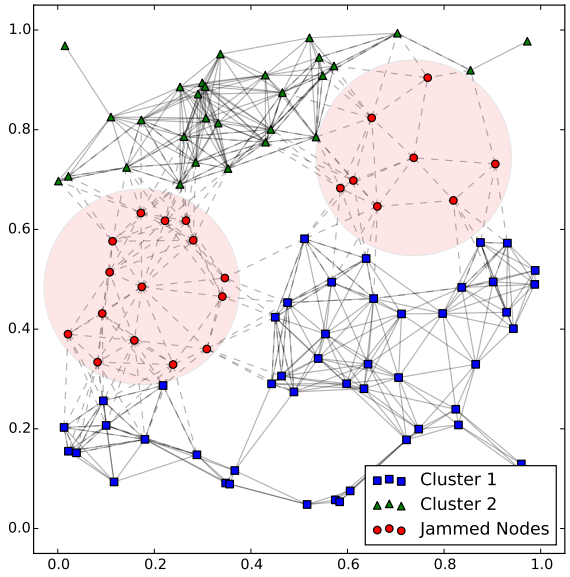

Fig. 1. Jammer placement to partition 100-node network into two residual networks with fewer than 50 nodes each. Two jammers (at centers of red circles) are sufficient. Red nodes are jammed. Solid (dashed) lines indicate communications link that are not jammed (jammed).

For non-trivial networks, problems related to searching for edge/vertex separator are usually $\mathcal{NP}$-Complete or $\mathcal{NP}$-Hard [13], [14] even with strict constraints on network complexity [15]. We also observed such high computational demand in our previous research [10]. In the situation where searching for optimal jammer placement solution is not feasible, we developed a suboptimal searching technique [12] which offers orders of magnitudes speed improvement with close-to-optimal solutions.

III. Search for Candidate Jammer Locations

As previously described in Section I, not every location in the plane needs to be considered for jammer placement. This motivates us to find a set of candidate jammer locations (CJLs), such that a minimum-cardinality jammer placement over the CJLs will have the same cardinality as a minimum-cardinality jammer placement over the entire real plane.

We introduce some notation used in our algorithm. We define an augmented disk $\mathbf{d_i}$ as a tuple of values $\mathbf{d_i} = ((x_i, y_i, r_i), \mathbf{v_i})$, where (x_i, y_i) represents the center of a closed disk in the plane, r_i represents the radius of the disk, and $\mathbf{v_i} \subseteq \mathcal{V}$ denotes the set of vertices in $\mathcal{G}$ that lie within the specified disk. Let $\mathcal{D} = \{\mathbf{d_0}, \mathbf{d_1}, \ldots\}$ denote a set of such augmented disks. We define a partial order on the $\mathbf{d_i}$ by

$$\begin{aligned} \mathbf{d_i} = \mathbf{d_j} &\iff \mathbf{v_i} = \mathbf{v_j}, \text{ and} \\ \mathbf{d_i} < \mathbf{d_j} &\iff \mathbf{v_i} \subset \mathbf{v_j}. \end{aligned}$$

Extending the notation of [16], let $\mathrm{md}(\mathbf{v})$ be the (augmented) closed disk of smallest radius that contains all $v_i \in \mathbf{v}$;

Data: $\mathcal{G}(\mathcal{V},\mathcal{E})$, r_j, γ
Result: $\mathcal{D} = \{\mathbf{d_i} = ((x_i, y_i, r_i), \mathbf{v_i})\}$
create empty stack S and solution set $\mathcal{D}$;
foreach $v_i \in \mathcal{V}$ **do**
  Push $\mathcal{V}$ to S;
  /* depth-first search */
  while S *is* ***not*** *empty* **do**
    $\mathbf{v} = $ S.pop();
    if $\exists d_i = ((x_i, y_i, r_i), \mathbf{v_i}) \in \mathcal{D} \ni \mathbf{v} = \mathbf{v_i}$ **then**
      continue
    end
    calculate $(x, y, r) = \mathrm{md}(\mathbf{v})$;
    if $r > \gamma$ **then**
      find $\{b_1, b_2, \ldots\} = B((x, y, r), \mathbf{v})$;
      foreach $b_i \in \{b_1, b_2, \ldots\}$ **do**
        push $\mathbf{v} \backslash b_i$ into S;
      end
    else
      add $((x, y, r), \mathbf{v})$ to $\mathcal{D}$
    end
  end
end
/* prune redundant solutions */
foreach $\mathbf{d_i} \in \mathcal{D}$ **do**
  foreach $\mathbf{d_j} \in \mathcal{D}$ **do**
    if $i \neq j$ ***and*** $\mathbf{d_j} \leq \mathbf{d_i}$ **then**
      remove $\mathbf{d_j}$ from $\mathcal{D}$
    end
  end
end

Algorithm 1: Find sufficient set of minimum disks $\mathbf{d_i}$ via DFS on complete set of vertices $\mathcal{V}$.

Data: $\mathcal{G}(\mathcal{V},\mathcal{E})$, r_j, γ
Result: $\mathcal{D} = \{\mathbf{d_i} = ((x_i, y_i, r_i), \mathbf{v_i})\}$
create empty stack S and solution set $\mathcal{D}$;
foreach $v_i \in \mathcal{V}$ **do**
  find $\mathcal{N}_{2r_j}(v_i)$ and push to S;
  /* depth-first search */
  while S *is* ***not*** *empty* **do**
    $\mathbf{v} = $ S.pop();
    if $\exists d_i = ((x_i, y_i, r_i), \mathbf{v_i}) \in \mathcal{D} \ni \mathbf{v} = \mathbf{v_i}$ **then**
      continue
    end
    calculate $(x, y, r) = \mathrm{md}(\mathbf{v})$;
    if $r > \gamma$ **then**
      find $\{b_1, b_2, \ldots\} = B((x, y, r), \mathbf{v})$;
      foreach $b_i \in \{b_1, b_2, \ldots\}$ **do**
        push $\mathbf{v} \backslash b_i$ into S;
      end
    else
      add $((x, y, r), \mathbf{v})$ to $\mathcal{D}$
    end
  end
end
/* prune redundant solutions */
foreach $\mathbf{d_i} \in \mathcal{D}$ **do**
  foreach $\mathbf{d_j} \in \mathcal{D}$ **do**
    if $i \neq j$ ***and*** $\mathbf{d_j} \leq \mathbf{d_i}$ **then**
      remove $\mathbf{d_j}$ from $\mathcal{D}$
    end
  end
end

Algorithm 2: Find sufficient set of minimum disks $\mathbf{d_i}$ via DFS from expanded neighborhoods of vertices.

i.e., $\mathrm{md}(\mathbf{v}) = ((x, y, r), \overline{\mathbf{v}})$, where (x, y, r) satisfy

$$\begin{aligned} \min_{x\in\mathbb{R}, y\in\mathbb{R}, r\in\mathbb{R}^+} \quad & r \\ \text{s.t.} \quad & d_E(v_i, (x, y)) \leq r, \qquad \forall v_i \in \mathbf{v} \end{aligned} \tag{1}$$

and $\overline{\mathbf{v}} = \{w \in \mathcal{V} \mid d_E(w, (x, y)) \leq r\}$.

In the following definitions, let $\mathrm{md}(\mathbf{v}) = ((x, y, r), \overline{\mathbf{v}})$. The radius of $\mathrm{md}(\mathbf{v})$ is denoted by $|\mathrm{md}(\mathbf{v})|$ and satisfies $|\mathrm{md}(\mathbf{v})| = r$. From the construction of $\mathrm{md}(\mathbf{v})$, we see that if $\mathbf{v} \subset \mathbf{w}$, then $|\mathrm{md}(\mathbf{v})| \leq |\mathrm{md}(\mathbf{w})|$. For a vertex $u \in \mathcal{V}$, we say $u \in \mathrm{md}(\mathbf{v})$ if $u \in \overline{\mathbf{v}}$. The boundary of a set of vertices $\mathbf{v}$ is $B(\mathbf{v}) = \{w \in \mathcal{V} \mid d_E(w, (x, y)) = r\}$. The open minimum disk contain $\mathbf{v}$ is $\underline{\mathrm{md}}(\mathbf{v}) = ((x, y, r),\ w \backslash B(\mathbf{v}))$. Define the neighborhood vertex set $\mathcal{N}_r(v) = \{w \in \mathcal{V} \mid d_E(v, w) < r\}$.

Algorithm 1 uses a depth-first search (DFS) starting from the entire network to find a set of CJLs by finding consecutively smaller subsets of the network. Algorithm 2 also finds a set of CJLs but reduces complexity compared to Algorithm 1 by using a sequence of depth-first searches starting from neighborhoods of the network nodes. In the following, we prove that Algorithm 1 and Algorithm 2 with stopping radius $\gamma = r_j$ find a set of CJLs that are sufficient to find an optimal solution to the problem defined in Section II-B. First, we introduce some preliminary results. Because of space constraints, proofs are omitted for the results that are simple to prove.

Proposition 1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple Euclidean graph, and $\mathbf{v} \subseteq \mathcal{V}$ and $u \in \mathcal{V}$, $u \notin \mathbf{v}$ be given. Let $\mathbf{w} = \mathbf{v} \cup \{u\}$. If $|\mathrm{md}(\mathbf{w})| > |\mathrm{md}(\mathbf{v})|$, then $u \in \mathrm{B}(\mathbf{w})$.*

Proof. Suppose that $\exists u \in \mathcal{V}, u \notin \mathbf{v}$ such that for $\mathbf{w} = \mathbf{v} \cup \{u\}$, $|\mathrm{md}(\mathbf{w})| > |\mathrm{md}(\mathbf{v})|$ but $u \notin \mathrm{B}(\mathbf{w})$. Then $\mathrm{B}(\mathbf{w}) = \mathrm{B}(\mathbf{v})$. But the minimum disc containing a set of vertices is determined solely by the vertices on the boundary of the disc [16]. Thus, $\mathrm{md}(\mathbf{w}) = \mathrm{md}(\mathrm{B}(\mathbf{w})) = \mathrm{md}(\mathrm{B}(\mathbf{v})) = \mathrm{md}(\mathbf{v})$, which contradicts $|\mathrm{md}(\mathbf{w})| > |\mathrm{md}(\mathbf{v})|$. □

Proposition 2. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple Euclidean graph, and let $\mathbf{v} \subseteq \mathcal{V}$ and $u \in \mathcal{V}$, $u \notin \underline{\mathrm{md}}(\mathbf{v})$ be given. Then $|\mathrm{md}(\mathbf{v} \cup \{u\})| = |\mathrm{md}(\mathbf{v})|$ if and only if $u \in \mathrm{B}(\mathbf{v})$.*

Corollary 1. *If $u \notin \mathrm{md}(\mathbf{v})$, then $|\mathrm{md}(\mathbf{v} \cup \{u\})| > |\mathrm{md}(\mathbf{v})|$.*

Next we show that if we take any vertex set and create an augmented vertex set by adding the vertex to it that results in the smallest increase in the radius of the enclosing minimum disk, then any other vertex not in the augmented vertex set will

not be in the open minimum disk containing the augmented vertex set.

Lemma 1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple Euclidean graph, and $\mathbf{v} \subseteq \mathcal{V}$ be given. Let*

$$u = \arg \min_{u \in \mathcal{V} \setminus \overline{\mathbf{v}}} |\mathrm{md}(\mathbf{v} \cup \{u\})|. \quad (2)$$

Then $\forall z \in \mathcal{V}$ such that $z \notin \overline{\mathbf{v}} \cup \{u\}$, $z \notin \underline{\mathrm{md}}(\mathbf{v} \cup \{u\})$.

Proof. Let u satisfy (2), and suppose there is $z \notin \overline{\mathbf{v}} \cup \{u\}$ such that $z \in \underline{\mathrm{md}}(\mathbf{v} \cup \{u\})$.

Consider the vertex set $\mathbf{w} = \overline{\mathbf{v}} \cup \{z\}$.

By construction, $z \notin \overline{\mathbf{v}} \cup u \Rightarrow z \notin \mathrm{md}(\mathbf{v})$.

As previously noted, $|\mathrm{md}(\mathbf{v} \cup z)| \geq |\mathrm{md}(\mathbf{v})|$. If $|\mathrm{md}(\mathbf{v} \cup z)| > |\mathrm{md}(\mathbf{v})|$, then by Proposition 1 $z \in \mathrm{B}(\mathbf{w})$. If $|\mathrm{md}(\mathbf{v} \cup z)| = |\mathrm{md}(\mathbf{v})|$, then by Proposition 2 $z \in \mathrm{B}(\mathbf{v}) = \mathrm{B}(\mathbf{w})$.

Since $z \in \underline{\mathrm{md}}(\mathbf{v} \cup u)$, $|\mathrm{md}(\mathbf{v} \cup z)| \leq |\mathrm{md}(\mathbf{v} \cup u)|$, but by (2), $|\mathrm{md}(\mathbf{v} \cup z)| \geq |\mathrm{md}(\mathbf{v} \cup u)|$. Thus, $|\mathrm{md}(\mathbf{v} \cup z)| = |\mathrm{md}(\mathbf{v} \cup u)|$. Now consider the intersection of $\mathrm{md}(\mathbf{v} \cup z)$ and $\mathrm{md}(\mathbf{v} \cup u)$. If the two disks are the same, then by Lemma 1, $z \in B(\mathbf{v} \cup u)$, which contradicts $z \in \underline{\mathrm{md}}(\mathbf{v} \cup \{u\})$. If the two disks are not the same, then the intersection of the two disks can be contained in a smaller disk [16]. Since by assumption $z \in \underline{\mathrm{md}}(\mathbf{v} \cup \{u\})$, z must lie in the intersection of the two disks, which means that u was not the solution to (2).

Thus, we conclude that it must be that $z \notin \underline{\mathrm{md}}(\mathbf{v} \cup \{u\})$. □

Lemma 2. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple graph, and $\mathbf{v} \subseteq \mathcal{V}$ be given. Then $\overline{\mathbf{v}}$ will be found at one of the vertices of the DFS tree in Algorithm 1 with stopping radius $\gamma \leq \mathrm{md}(\mathbf{v})$.*

Proof. From the set $\mathbf{v}$, find $\overline{\mathbf{v}}$. By definition $\overline{\mathbf{v}}$ is within radius $\mathrm{md}(\mathbf{v})$.

First we note that if $\overline{\mathbf{v}}$ is on the DFS search tree with stopping radius $\gamma = 0$, then $\overline{\mathbf{v}}$ is on the DFS search tree with stopping radius $\gamma \leq \mathrm{md}(\mathbf{v})$, since the specified stopping radius will only remove vertex sets $\mathbf{w}$ that satisfies $\mathrm{md}(\mathbf{w}) < \gamma$.

Let $\mathbf{w}_0 = \overline{\mathbf{v}}$. If $\mathbf{w}_0 \neq \mathcal{V}$, then let $i = 1$ and:

1) Step 1: Find a node u satisfying (2), and let $\mathbf{w}_i = \mathbf{w}_{i-1} \cup \{u\}$. By Lemma 1, at each step $u \in \mathrm{B}(w)$.
2) Step 2: If $\mathbf{w}_i = \mathcal{V}$, stop. Otherwise, let $i = i + 1$ and return to step 1.

Let n denote the value of i at the end of the iterations. Note that $\mathbf{w}_i$ and $\mathbf{w}_{i-1}$ differ by one vertex that is on the boundary of $\mathbf{w}_i$. Thus $\mathbf{w}_n, \mathbf{w}_{n-1}, \ldots, \mathbf{w}_0$ is a sequence of vertices on the DFS tree from the root at $\mathbf{w}_n = \mathcal{V}$ to $\mathbf{w}_0 = \overline{\mathbf{v}}$. □

Theorem 1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple Euclidean graph, and $\mathbf{v} \subseteq \mathcal{V}$ be given. If $\mathbf{v}$ can be covered by a jammer with jamming radius r_j, then the DFS tree found by Algorithm 1 with $\gamma = r_j$ contains at least one disk $\mathbf{d} = ((x, y, r), \mathbf{w})$ such that $\mathbf{v} \subseteq \mathbf{w}$. Thus, the nodes in $\mathbf{v}$ can be jammed by placing at jammer at (x, y), the center of the disk $\mathbf{d}$ that is on the DFS tree found by Algorithm 1.*

Proof. By Lemma 2, $\overline{\mathbf{v}}$ is on the DFS tree with stopping radius $\gamma \leq \mathrm{md}(\mathbf{v})$. As in Lemma 2, label the vertex sets from $\overline{\mathbf{v}}$ to the root of the DFS tree as $\mathbf{w}_0 = \overline{\mathbf{v}}, \mathbf{w}_1, \ldots, \mathbf{w}_n = \mathcal{V}$.

If $|\mathrm{md}(\mathcal{V})| < r_j$, then $\mathrm{md}(\mathcal{V})$ satisfies the theorem. Otherwise, $|\mathbf{w}_0| \leq |\mathbf{w}_1| \leq \ldots \leq |\mathbf{w}_n|$. Thus there must be some set $\mathbf{w}_j$ such that $|\mathbf{w}_j| \leq r_j$ but $|\mathbf{w}_{j+1}| > r_j$. By Lemma 2, this $\mathbf{w}_{j+1}$ will be found by the DFS search with $\gamma = r_j$, and $\mathbf{w}_j$ will also be found by the DFS since the DFS search continues to the first level where $|\mathbf{w}_i| \leq \gamma$. This $\mathbf{w}_j$ satisfies the theorem. □

Lemma 3. *If $\mathbf{v} \subseteq \mathcal{V} \in \mathcal{G}$ can be enclosed by a circle with radius r and centered at (x, y), then for $\forall v \in \mathbf{v}$, $\mathbf{v} \subseteq \mathcal{N}_v^{2r}$*

Proof. Let (x, y) denote the center of the circle with radius r that contains $\mathbf{v}$. Then $\forall v \in \mathbf{v}$, $d_E(v_i, (x, y)) \leq r$. By the triangle inequality, $d_E(v_i, v_j) \leq d_E(v_i, (x, y)) + d((x, y), v_j) \leq 2r$ for $\forall v_i, v_j \in \mathbf{v}$.

Then by definition, $\mathbf{v} \subseteq \mathcal{N}_v^{2r}$ for $\forall v \in \mathbf{v}$. □

Theorem 2. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple graph, and $\mathbf{v} \subseteq \mathcal{V}$ be given. If $\mathbf{v}$ can be covered by a jammer with jamming radius r_j, then $\exists d = ((x, y, r), \mathbf{w})$ with $\mathbf{v} \subseteq \mathbf{w}$ such that d is on the DFS tree found by Algorithm 2 with $\gamma = r_j$. Thus, the nodes in $\mathbf{v}$ can be jammed by placing at jammer at (x, y).*

Proof. By applying Lemma 3, $\mathbf{v} \subseteq \mathcal{N}_v^{2r_j}$ for all $v \in \mathbf{v}$. Algorithm 2 is equivalent to applying Algorithm 1 for each $\mathcal{N}_v^{2r_j}$, $v \in \mathbf{v}$. Apply Theorem 1 with the subgraph of $\mathcal{G}$ that has vertex set $\mathcal{N}_v^{2r_j}$. Then for each $\mathcal{N}_v^{2r_j}$, $\exists \mathbf{d} = ((x, y, r), \mathbf{w})$ with $\mathbf{v} \subseteq \mathbf{w}$ such that d is on the DFS tree found by Algorithm 2 with $\gamma = r_j$. □

Theorem 3. *Let $\mathbf{j}$ be a set of jamming locations in the real plane, $j_i \in \mathbb{R}^2$. Then for each location, $j_i \in \mathbf{j}$, there is at least one alternative location k_i corresponding to one of the augmented disks found by either Algorithm 1 or Algorithm 2, such that the jammer placed at k_i will jam a superset of the nodes jammed by a jammer placed at j_i. Thus, the centers of the augmented disks found by Algorithm 1 or Algorithm 2 are sufficient to find an jamming strategy of minimum cardinality.*

Proof. For each jamming location j_i, find the set of vertices $\mathbf{v}_i$ that would be jammed. By applying Algorithm 1 or Algorithm 2, Theorem 1 and Theorem 2 guarantee that the DFS trees created will contain at least one vertex set that is a superset of $\mathbf{v}_i$. Chose any such vertex set $\mathbf{w}$ and call the minimum disk containing that set $\mathrm{md}(\mathbf{w}) = ((x_w, y_w, r_w), \mathbf{w})$. Thus, the jammer at j_i can be replaced with a jammer at (x_w, y_w) with no loss of optimality. Since this can be done for every $j_i \in \mathbf{j}$, any solution $\mathbf{j}$ with $j_i \in \mathbb{R}^2$ can be replaced with a solution of equal cardinality where the locations correspond to the centers of minimum disks on a DFS search tree created by either Algorithm 1 or Algorithm 2. Thus these algorithms produce a candidate set of points from which can be found a solution of minimum cardinality that jams a superset of the nodes of the solution $\mathbf{j}$. □

IV. Searching for Jammer Placement Locations

In our previous work [10], [12], we have developed optimal and suboptimal approaches to find a minimum cardinality jammer placement over a constrained set. These formulations

naturally extend to using the centers of the $\mathcal{D}$ found by Algorithm 1 instead of the locations of the network nodes. We briefly review those algorithms here.

For network $\mathcal{G}(\mathcal{V},\mathcal{E})$ and $\mathcal{D}$, we introduce the jamming graph $\mathcal{G}_J(\mathcal{V}_J,\mathcal{E}_J)$, in which $\mathcal{V}_J = \mathcal{V}\cup\mathcal{V}_\mathcal{D}$ and $\mathcal{E}_J = \{(i,k)\in \mathcal{V}_J\times\mathcal{V}_J \mid d_E(i,k)\le r_j\}$. Note that $\mathcal{V}_\mathcal{D}$ is the set of the central locations of all CJLs in $\mathcal{D}$, and $\mathcal{E}_J$ is the edge set in which an edge between nodes i and k indicates that a jammer placed at location of node i will block all reception at node placed at location k, and vice versa. Let $\mathbf{A}^{(J)}$ denote the adjacency matrix for $\mathcal{G}_J$, with $A^{(J)}_{i,k}=1$ if there is an edge connecting nodes i and k and $A^{(J)}_{i,k}=0$, otherwise. We say that an edge (link) is jammed if at least one of the nodes connected to the link is jammed. Let $N=|\mathcal{V}|$ and $N_\mathcal{D}=|\mathcal{V}_\mathcal{D}|$.Note that, in jamming graph $\mathcal{G}_J$, nodes in $\mathcal{V}$ are labeled from 1 to N and nodes in $\mathcal{V}_\mathcal{D}$ are labeled from $N+1$ to $N+N_\mathcal{D}$.

As we have shown in Theorem 3, placing jammers solely on locations included in $\mathcal{V}_\mathcal{D}$ is sufficient for finding the jammer placement solution with minimum cardinality.

A. Optimal Approach

We first introduce some notation that is used in the optimal approach. Jammers are restricted to be placed at the locations of one of the nodes or CJLs. Let $\mathbf{x}$ be the binary vector with length $|\mathcal{V}_\mathcal{D}|$, which indicates whether a jammer should be placed at location of CJL. And let $\mathbf{y}^{(k)}$ be the binary vector with length N, where $k\in\{0,\dots,K\}$ with K is the number of clusters. Let $\mathbf{y}^{(0)}$ denote a special cluster that includes all nodes being jammed by at least one jammer in $\mathcal{J}$. Hence $y^{(0)}_i=1 \iff \exists J_m\in\mathcal{J}\ni d_E(v_i,J_m)\le r_j$, and $y^{(k)}_i=1 \iff v_i\in$ cluster k, $y^{(k)}_i=0$ otherwise for $k\in 1,\dots,K$.

Then in [10], the optimal formulation of jammer placement problem with CJL is formulated as the binary integer linear program (ILP):

$$\min \quad \sum \mathbf{x}$$

$$\text{s.t.} \quad \sum_{k=0}^{K} y_i^{(k)} = 1 \qquad i=1,\dots,N \tag{3}$$

$$\sum \mathbf{x} \ge 1 \tag{4}$$

$$\sum \mathbf{y}^{(k)} \ge 1 \qquad k=1,\dots,K \tag{5}$$

$$\sum \mathbf{y}^{(1)} \le b_K \tag{6}$$

$$\sum \mathbf{y}^{(k)} - \sum \mathbf{y}^{(k-1)} \le 0 \qquad k=2,\dots,K \tag{7}$$

$$y_i^{(0)} - \sum_{m=N+1}^{N_J} x_{m-N} A^{(J)}_{m,i} \le 0 \qquad i=1,\dots,N \tag{8}$$

$$y_i^{(k)} - y_j^{(k)} - y_i^{(0)} - y_j^{(0)} \le 0 \qquad (i,j)\in\mathcal{E},\ k=1,\dots,K \tag{9}$$

$$x_i, y_i^{(k)} \in \{0,1\} \qquad i=1,\dots,N,\ k=0,\dots,K \tag{10}$$

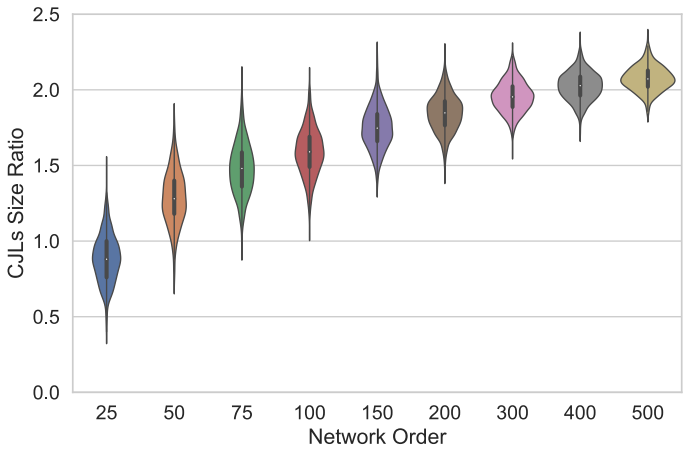

Fig. 2. Distributions of ratio between the size of augmented disk set and network order

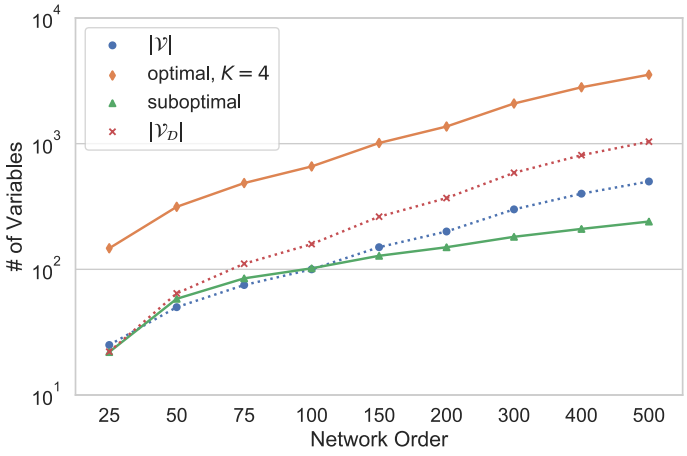

Fig. 3. Average size of integer programming under both formulations comparing to network order and order of $\mathcal{G}_J$

The size of our optimal ILP formulations is $(K+1)\times N+|\mathcal{V}_\mathcal{D}|$. The distribution of the ratio between $|\mathcal{V}_\mathcal{D}|$ and $|\mathcal{V}|$ for a set of 1000 random network topologies (see Section V for details) is shown in Fig. 2 as a violin plot. For most networks with order up to 300, the number of CJLs doesn't go beyond $2N$, which doesn't affect the size of ILP as significantly as K when $K\ge 2$.

B. Suboptimal Approach

Solving a large integer programming problem can be computationally intensive. So, we also present results for a suboptimal jammer placement algorithm developed in [12] that

- finds an edge separator $\mathcal{E}_S$ that partitions $\mathcal{G}$ into K clusters via multi-resolution graph cut, and
- finds the optimal jamming set (OJS), which is the minimum cardinality jammer placement that jams at least one vertex of each edge in the edge separator, via an ILP of size much smaller than that considered in Section IV-A.

Readers are referred to [12] for details of this approach.

The average sizes (in terms of the number of variables) for the integer programs used in both the optimal and suboptimal formulations are compared in Fig. 3 for 1000 random network topologies (again, see Section V). For reference, the cardinalities of the original network and the CJLs are shown. The average ILP size for the suboptimal approach is much smaller than for the optimal approach and has a much lower rate of growth. The average ILP size for the suboptimal approach is even smaller than the network order for networks with over 100 nodes.

V. Simulation Results

In this section, we compare the performance and running time of jammer placement algorithms using the CJLs with that of jammer placement at only the network nodes' locations. All simulations are programmed in Python with the network analysis module NetworkX and the Gurobi integer linear programming solver. The simulations were executed on computers with a 3.1GHz Intel Core i7 CPU with 8MB cache and 16GB RAM running Ubuntu 16.04 LTS. All figures show the performance averaged over 1000 different network topologies.

The simulation results are all for graphs generated using the Random Geometric Graph (RGG):

$$\mathcal{G}(\mathcal{V},\mathcal{E}) = \mathrm{RGG}(n, r_c)$$
$$\mathcal{V} = \{v_i(x_i, y_i) \mid x_i, y_i \sim U[0, \sqrt{\tfrac{n}{100}})\}$$
$$\mathcal{E} = \{(u, v) \mid u, v \in \mathcal{V}, d_E(u, v) \leq r_c\} \subset \mathcal{V} \times \mathcal{V}.$$

Each simulation topology is generated by randomly placing nodes in the region $[0, \sqrt{\frac{n}{100}}) \times [0, \sqrt{\frac{n}{100}})$ and then creating edges for all pairs of vertices u and v if they are within the communication radius, $d_E(u, v) \leq r_c$. This graph models a scenario in which:

- Radios are homogeneous with identical spatial density, equal transmit power, omnidirectional antennas, and equal noise figure.
- The channel obeys an exponential path-loss model.
- The jamming radius is equal to the communication radius ($r_j = r_c = 0.15$).

For all the simulations, the objective is to partition the network into 4 disconnected subnetworks, with the cardinality of each subnetwork constrained to be no more than one-fourth the original network order.

We first consider the effectiveness of using the CJLs for jammer placement instead of using the locations of the network nodes, $\mathcal{V}$. As shown in Fig. 4, by using the CJLs as the set of possible jammer locations, the average number of jammers required to achieve the network partition objective is reduced for both optimal and suboptimal search algorithms for finding a minimum cardinality jammer placement. More details of the relative performance are also shown in Table I; the performance of the optimal solver is not listed for networks of over 100 nodes because of the extremely long times required to find such solutions. The results in Table I (top of next page) show that the average number of jammers required is reduced by up to 21%, with typical improvements around 12% to 14% for the suboptimal solver. The performance of the suboptimal solver with the CJLs is only slightly worse than the performance of the optimal solver without the CJLs.

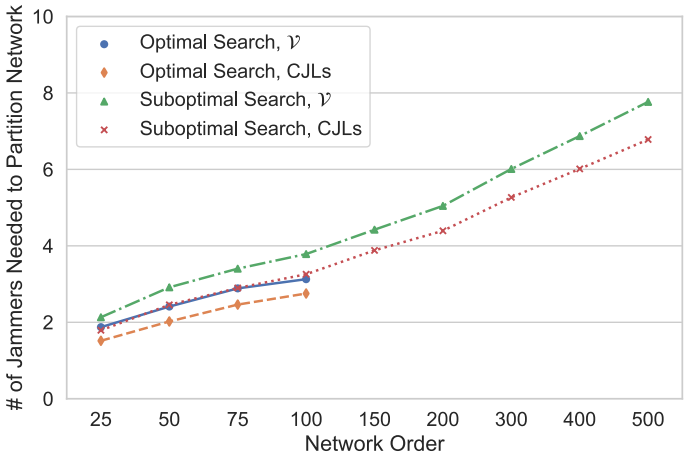

Fig. 4. Value of jammer placement objective function (number of jammers needed to meet the partitioning objective) for networks with different orders under optimal and suboptimal formulation, with or without CJL

TABLE II
CJL Search Time

Order	Time(s)
25	0.434
50	5.540
75	15.571
100	29.977
150	74.581
200	141.406
300	340.570
400	637.464
500	1028.107

The average time needed to find the CJLs for networks with different sizes is shown in Table II. The average time to solve for the CJLs is less than 30 seconds for networks with up to 100 nodes, but an average of approximately 17 minutes is needed for networks with 500 nodes. The total time to solve the jammer placement problem will depend on the time to find the CJLs plus the time to determine the optimal jammer placements over the CJLs, which is larger than the original problem of placing the jammers at the locations of the communicators.

The suboptimal approaches to solving the jammer placement problem are capable of reducing the execution time by orders of magnitudes [12]. This reduction in execution time still holds with the CJLs. Table III shows that for networks with up to 500 nodes, a jammer placement solution can still be found within a few milliseconds, with or without CJLs, which is almost negligible compared to the time needed to find the CJLs or solve the ILP optimally.

Table IV compares the average execution times of solving the original (without the CJLs) ILP optimally with that of searching for the CJLs. The results show that as the network size grows, the time needed to optimally solve the ILP grows

TABLE I
AVERAGE NUMBER OF JAMMERS REQUIRED.

	Optimal solver			Suboptimal solver		
Order	no CJL	w/ CJL	Improvement (%)	no CJL	w/ CJL	Improvement (%)
25	1.9	1.5	21%	2.1	1.8	14%
50	2.4	2.0	17%	2.9	2.5	14%
75	2.9	2.5	14%	3.4	2.9	15%
100	3.1	2.8	10%	3.8	3.3	13%
150	—	—	—	4.4	3.9	11%
200	—	—	—	5.0	4.4	12%
300	—	—	—	6.0	5.3	12%
400	—	—	—	6.9	6.0	13%
500	—	—	—	7.8	6.8	13%

TABLE III
SUBOPTIMAL FORMULATION ILP TIME COMPLEXITY

Ord	no CJL(ms)	w/ CJL(ms)	Δ
25	3.026	2.799	-7.49%
50	2.787	2.713	-2.66%
75	2.156	2.844	31.93%
100	2.202	2.686	21.95%
150	2.123	3.028	42.60%
200	2.398	3.596	49.94%
300	2.668	4.616	72.97%
400	3.313	5.400	62.99%
500	3.607	6.364	76.47%

TABLE IV
OPTIMAL FORMULATION ILP TIME COMPLEXITY AND CJL OVERHEAD

Order	ILP w/out CJL(s)	Search CJL(s)	Ratio
25	3.619	0.434	11.98%
50	21.864	5.540	25.34%
75	103.573	15.571	15.03%
100	530.792	29.977	5.65%

much faster than the time to find the CJLs. In combination with the performance results above, these results suggest that finding the CJLs and then using the suboptimal solver for the larger jammer placement problem may offer a good tradeoff between performance and complexity.

VI. CONCLUSION

In this paper, we developed an approach to finding a minimum cardinality jammer placement to partition a wireless network when jammers can be placed anywhere in the real plane. Our approach is based on finding a set of candidate jammer locations (CJLs) that are a sufficient search space for a minimum cardinality jammer placement. We develop two depth-first search algorithms to find the CJLs, and we evaluate the effects on performance and running time of using the CJLs in comparison to our previous approach of constraining jammers to be placed at the locations of the network nodes. The results show that by extending the search space to the CJLs, the average number of jammers required to partition a wireless network is decreased by approximately 10% to 20%, at the expense of an increase in complexity that is primarily attributable to the time to find the CJLs.

REFERENCES

[1] M. B. Pursley and W. E. Stark, "Performance of Reed-Solomon coded frequency-hop spread-spectrum communications in partial-band interference," *IEEE Trans. Commun.*, vol. 33, no. 8, pp. 767–774, Aug. 1985.

[2] M. B. Pursley and H. B. Russell, "Routing in frequency-hop packet radio networks with partial-band jamming," *IEEE Trans. Commun.*, vol. 41, pp. 1117–1124, July 1993.

[3] R. Negi and A. Perrig, "Jamming analysis of MAC protocols," Carnegie Mellon University, Tech. Rep., Feb. 2003.

[4] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM Int. Symp. Mobile Ad-Hoc Net. and Comput. (MobiHoc*. ACM, 2005, pp. 46–57.

[5] R. Chinta, T. F. Wong, and J. M. Shea, "Energy-efficient jamming attack in IEEE 802.11 MAC," in *Proc. 2009 IEEE Military Commun. Conf. (MILCOM)*, Boston, Oct. 2009.

[6] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[7] P. Tague, D. Slater, G. Noubir, and R. Poovendran, "Quantifying the impact of efficient cross-layer jamming attacks via network traffic flows," Network Security Lab (NSL), University of Washington, Tech. Rep., 2009.

[8] P. Tague, "Improving anti-jamming capability and increasing jamming impact with mobility control," in *Proc. IEEE Conf. on Mobile Adhoc and Sensor Syst. (MASS)*, Nov. 2010, pp. 501–506.

[9] J. Feng, X. Li, E. L. Pasiliao, Jr., and J. M. Shea, "Jammer placement to partition wireless network," in *Proc. IEEE Global Commun. Conf. Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles*, Austin, TX, Dec. 2014, pp. 1487–1492.

[10] J. Feng, E. L. Pasiliao, Jr., W. E. Dixon, and J. M. Shea, "An optimal jamming strategy to partition a wireless network," in *Proc. IEEE Military Commun. Conf.*, Tampa, FL, Oct. 2015, pp. 978–984.

[11] J. Feng, W. E. Dixon, and J. M. Shea, "Positioning helper nodes to improve rebustness of wireless mesh networks to jamming attacks," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[12] ——, "Fast algorithms for jammer placement to partition a wireless network," in *Proc. 2017 IEEE Int. Conf. Commun.*, Paris, France, May, pp. 1–6.

[13] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Comp. Sci.*, vol. 1, no. 3, pp. 237–267, 1976.

[14] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, "The complexity of multiway cuts," in *Proc. 24th Annual ACM Symp. Theory of Computing*. ACM, 1992, pp. 241–251.

[15] T. N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard," *Inform. Proc. Let.*, vol. 42, no. 3, pp. 153–159, 1992.

[16] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *New results and new trends in computer science*. Springer, 1991, pp. 359–370.