2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Intruder Capturing Game on a Topological Map Assisted by Information Networks

Jonghoek Kim, Sean Maxon, Magnus Egerstedt, and Fumin Zhang

***Abstract*—Intruder capturing games on a topological map of a workspace with obstacles are investigated. Assuming that a searcher can access the position of any intruder utilizing information networks, we provide theoretical upper bounds for the minimum number of searchers required to capture all intruders on a Voronoi graph. Intruder capturing algorithms are proposed and demonstrated through an online computer game.**

I. Introduction

Monitoring large complex areas, such as an urban environment, is an important application of sensor networks and multi-robot systems. A networked sensing and communication infrastructure, denoted as an *information network*, can be utilized for such a task. To build an information network in a workspace with obstacles, Simultaneous Cooperative Exploration and NeTworking (SCENT) strategies proposed in our previous work [1] can be applied.

In this paper, we study intruder capturing game in a workspace with obstacles, with the free space represented by a Voronoi diagram that has been widely used for topological maps in robotics ([2]–[6]). We simplify the scenario so that a searcher and intruders are restricted to stay on the Voronoi diagram. Obeying the conventions established in the literature on graph searching [7]–[14], an intruder can maneuver at unbounded speed to avoid searchers. Furthermore, an intruder has full knowledge of the environment, positions of the searchers, and the strategies of the searchers. An intruder is *captured* if it is forced to share a node with any searcher.

There are many extensions to the results in the seminal work of Parsons [7]. A closely related work to ours is the helicopter cops and robbers game [15], [16]. In this game, it is assumed that the cops have complete knowledge of any robber's position as if the cops are using helicopters. A robber is captured when a cop lands on the node occupied by the robber and the robber cannot make any move to escape. A *monotone searching strategy* is a search plan which guarantees that if every robber on one edge is captured, then no robber can enter the edge later. It was found in that if we only consider monotone searching strategies, then the minimum number of cops required depends on the number of robbers.

Similar to a cop using a helicopter in [15], [16], we assume that a searcher detects intruders using the information network. However, we require that a searcher moves along edges of a graph continuously, which is distinct from [15], [16] and is closer to autonomous robot applications. In addition, our searching strategy is not monotone, which implies that even if every intruder on an edge is captured, another intruder may enter the edge later. Based on this searching strategy, we derive theoretical upper bound for the minimum number of searchers required to capture all intruders on a general graph, which leads to a result on the Voronoi diagram. Note that this upper bound does not depend on the number of intruders. Our searching strategy is further implemented through an interactive online game [17] to assist humans to determine how to secure a complex graph.

The paper is organized as follows: Section II introduces preliminaries and background information. Section III discusses the intruder capturing problem utilizing an information network. Section IV presents an interactive online game to implement our searching strategy. Section V provides conclusions.

II. Preliminaries and Background Information

A. Graph Theory

We review some general notions in graph theory, e.g., [18]. An undirected graph G is defined by a set $G = (N(G), E(G))$, where $N(G)$ denotes the node set and $E(G)$ is a set of unordered pairs of nodes where multiple edges between node pairs are allowed. A graph G is *connected* if there is a path between every pair of distinct nodes. The *subgraph* of G induced by a set of nodes $S \subset N(G)$ is represented by (S, E_S) where E_s is the edge set. A *cycle* is a closed path, and a *walk* is a path with no self-intersection.

A graph embedded in the plane without edge crossings is called a *plane graph*. The *faces* of a plane graph are the

Jonghoek Kim, Sean Maxon, Magnus Egerstedt, and Fumin Zhang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. Email: {jkim37,sean.maxon}@gatech.edu, {magnus,fumin}@ece.gatech.edu

978-1-61284-799-3/11/$26.00 ©2011 IEEE

maximal regions of the plane that contain no point used in the embedding. We say graph G^* is the *dual graph* of a plane graph G if the nodes of G^* corresponds to the faces of G, and if two faces of G are adjacent, then the two corresponding nodes of G^* are connected by an edge of G^*

An *edge cover* of G, $C(G) \subset E(G)$, is a set of edges such that every node in G is incident to some edge in $C(G)$. The notion $\alpha(G)$ denotes an edge cover of G with the minimum cardinality, i.e., the fewest number of edges. A *matching* in G is a set of edges in G with no shared endpoints. A *perfect matching* in G is a matching such that every node in G is incident to some edge in the matching. Therefore, a perfect matching in G is also an edge cover of G with the minimum cardinality $\frac{|N(G)|}{2}$.

B. The Workspace and Its Voronoi Diagram

Consider a connected and compact workspace $W \subset R^2$ whose boundary, ∂W, is a simple closed curve. In other words, ∂W is continuous and no self-intersection occurs. Let O_1,O_2,...O_{M-1} be $M-1$ disjoint compact obstacles such that $O_i \subset W$. We introduce O_M as a "virtual" obstacle that bounds the workspace, i.e., $\partial W \subset \partial O_M$. We denote the set of obstacles as $S_O = \{O_1, O_2, ...O_M\}$. A Voronoi cell for O_i is $V(O_i)$ with its boundary $\partial V(O_i)$.

The following assumptions, which were also used in our previous papers [1], [19], are made about the workspace: $\bigcup_{O_i \in S_O} \overline{V}(O_i) = W$ where $\overline{V}(O_i) = V(O_i) \bigcup \partial V(O_i)$, and $\partial V(O_i)$ is a simple closed curve for each $O_i \in S_O$, i.e., $\partial V(O_i)$ is continuous and no self-intersection occurs.

The *Voronoi diagram* $V = (N(V), E(V))$ is defined as the union of all cell boundaries. Since $\partial V(O_i)$ is a simple closed curve, each Voronoi cell is also a face of V. A *Voronoi edge* in $E(V)$ is a common boundary edge shared by two Voronoi cells $V(O_i)$ and $V(O_j)$. A *Voronoi vertex* in $N(V)$ is a point where more than two Voronoi edges meet.

Let C be a cycle. $V[C]$ is the subgraph of V enclosed by C. Let $V^*[C]$ denote the dual graph of $V[C]$ with the node corresponding to the unbounded face removed. Then $\partial(V^*[C])$ is the boundary of $V^*[C]$. Let V^* be $V^*[C]$ when we choose $\partial V(O_M)$ as C, i.e., V^* is the dual graph of V with the node representing the unbounded face removed. Each node $n(O_i)$ of V^* corresponds to the Voronoi cell $V(O_i)$. The notion *cycle basis* denotes a cycle in V enclosing a single Voronoi cell. Fig. 1 illustrates $V[C]$, $V^*[C]$, and $\partial(V^*[C])$. Lemma 1 states that $V^*[C]$ is connected, which is straigthtforward.

Lemma 1: Let C be a cycle in a Voronoi diagram V. Then, $V^*[C]$ is connected.

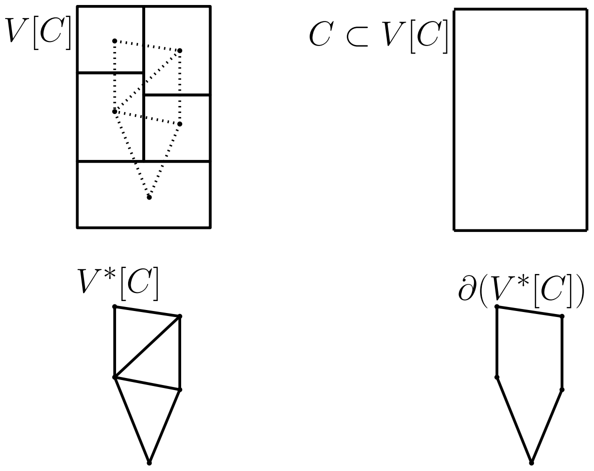

Fig. 1. C, $V[C]$, $V^*[C]$, and $\partial(V^*[C])$.

III. Capturing Intruders

A. Definitions and Assumptions

We consider a graph G. An *intruder* is free to move along the edges. A *guard* can be deployed at a node to prevent an intruder to reach that node. A *searcher* moves along edges of a graph and can access the guarded nodes. The searcher obtains the position of any intruder whenever the searcher visits a node in a graph. An intruder is *captured* if (1) it is on an edge whose one end is guarded while the free searcher moves through the edge from the opposite end, or (2) it is on an edge one end of which has degree one while the free searcher moves through the edge starting from the opposite end.

Following established conventions ([8], [20]–[23]), we assume that an intruder has full knowledge of the environment, searching strategies, and positions of both the searcher and guards. In addition, an intruder can move along edges of G at unbounded speed to avoid both the searcher and guards.

B. Capturing Intruders on a General Graph

Suppose an information network is deployed along the Voronoi edges, and suppose that the searcher can obtain the location of the intruder through the information network. We want to find the minimum number of guards needed to capture all intruders on G with one searcher. Define this minimum number as $g_I(G)$ we derive an upper bound for $g_I(G)$ for a general graph G.

The authors of [16] introduced a searching strategy to capture one intruder on a tree graph T using one free searcher. Suppose n is a node of T. Then, we define a *branch* of T at n as the maximal subtree of T, denoted by T', if n has degree one in T'. Lemma 2 provides the searching strategy [16].

Lemma 2: Suppose that one free searcher and one intruder move along a tree graph T. Then, the searcher can capture the intruder in finite time using the following strategy:

Whenever the searcher meets a node, it obtains the position of the intruder. Then, it chooses the branch containing the intruder and moves through the edge contained in the branch until it meets another node. Iterate this until the intruder is captured.

Since an intruder can move at unbounded speed, it can escape from the free searcher using a cycle in a general graph G. To block the escape of an intruder, guards should be deployed at nodes in G to block all cycles. We say that a cycle C is *blocked* if any node in $N(C)$ is guarded.

We define MinGuard(G) as the minimum number of guarded nodes to block all cycles contained in G. Obviously MinGuard(T)=0 for a tree graph T. For a general graph G, computational method to search for MinGuard(G) corresponds to a set cover optimization problem which is known to be NP-hard. However, approximation algorithms returning near-optimal solutions exist [24].

Suppose MinGuard(G) nodes of G are guarded to block all cycles contained in G. Then no cycle is available to an intruder because the graph is reduced to a tree. Thus, an intruder cannot escape from the free searcher according to Lemma 2. Furthermore, the searcher can capture all intruders by chasing one intruder at a time. Since we need one searcher and MinGuard(G) guards to capture all intruders on G, $g_I(G)$ $\leq$MinGuard(G). Hence, we have the following Lemma:

Lemma 3: $g_I(G)$ $\leq$MinGuard(G).

C. Capturing Intruders on Voronoi Diagrams

In this subsection, we consider the specific structures of a Voronoi diagram $V = (N(V), E(V))$ as introduced in Section II-B to compute $g_I(V)$.

Recall that V^* is the dual graph of V with the node representing the unbounded face removed, and V^* is a connected graph. Depending on the structure of V, V^* can be one node or a connected graph with more than one node. In the case where V^* is one node, V has only one cycle. Thus, $g_I(V) = 1$. This case is trivial.

In the case where V^* is a connected graph with more than one node, we first consider the special case where V^* is a tree graph. In this special case, let $\alpha(V^*)$ denotes an edge cover of V^* with the minimum cardinality, and let $|\alpha(V^*)|$ be the number of edges in $\alpha(V^*)$. We propose Algorithm 1 to guard $|\alpha(V^*)|$ nodes in V. In Theorem 1, we prove that Algorithm 1 blocks all cycles in V.

Algorithm 1 Guarding $|\alpha(V^*)|$ nodes in V (In the case where V^* is a connected graph with more than one node, we replace $\alpha(V^*)$ in this algorithm by $\beta(V^*)$)

```
the edge set $E_C \leftarrow \alpha(V^*)$;
repeat
  select one edge from the edge set $E_C$;
  suppose $n(O_i)n(O_j)^m$ is the selected edge (Note: $m$ is used as an index in order to distinguish among multiple edges that might exist between the two nodes on $V^*$, which are $n(O_i)$ and $n(O_j)$);
  $e^m_{i,j} \subset \partial V(O_i) \bigcap \partial V(O_j) \leftarrow$ the edge in $V$ corresponding to $n(O_i)n(O_j)^m$ in $V^*$;
  if $e^m_{i,j}$ meets an unblocked cycle $C \subset V$, which encloses both $O_i$ and $O_j$ then
    guard the point where $e^m_{i,j}$ meets $C$ (Note: if both ends of $e^m_{i,j}$ meet $C$, then guard any one);
  else
    guard any one of the two end points of $e^m_{i,j}$;
  end if
  mark all cycles containing the guarded node as blocked;
  remove the edge $n(O_i)n(O_j)^m$ from the edge set $E_C$;
until there is no edge left in $E_C$;
```

Theorem 1: If the Voronoi diagram V has its dual graph V^* being a tree graph with more than one node, then $g_I(V) \leq |\alpha(V^*)|$.

Proof: We already know that $g_I(V)$ $\leq$MinGuard(V) according to Lemma 3. If MinGuard(V) $\leq$ $|\alpha(V^*)|$, then $g_I(V)$ $\leq$MinGuard(V) $\leq$ $|\alpha(V^*)|$. This further implies that $g_I(V) \leq |\alpha(V^*)|$.

It remains to show that MinGuard(V) $\leq$ $|\alpha(V^*)|$. We prove by contradiction. Suppose that $|\alpha(V^*)|$ $<$MinGuard(V). Since MinGuard(V) is the minimum number of guarded nodes to block all cycles in V, $|\alpha(V^*)|$ $<$MinGuard(V) implies that if we guard $|\alpha(V^*)|$ nodes in V, then there exists an unblocked cycle, say C, in V. Since $V^*[C] \subset V^*$ is connected and must be a tree graph, then $V^*[C] = \partial V^*[C]$.

We consider two cases depending on the structure of $\partial(V^*[C])$. For these two cases, we will prove that an unblocked cycle, C, cannot exist after we guard $|\alpha(V^*)|$ nodes in V using Algorithm 1. Therefore, MinGuard(V) $\leq |\alpha(V^*)|$.

1) $\partial(V^*[C])$ is a node. In this case, C contains only one cell.

2) $\partial(V^*[C])$ is a tree graph containing at least one edge. In this case, C contains more than one cell.

1. Consider the case where $\partial(V^*[C])$ is a node. Then the node belongs to an edge in the $\alpha(V^*)$. Once a guard corresponding to that edge is deployed using Algorithm 1, then C will be blocked.

2. Consider the case where $\partial(V^*[C])$ is a tree graph containing at least one edge. There exists at least one edge, say $n(Q_1)n(Q_2)$, in $\partial(V^*[C])$, whose one end point has degree one. Suppose $n(Q_1)$ has degree one. Since $\partial(V^*[C])$ is a tree graph, $n(Q_1)n(Q_2) \in E(V^*)$ indicates that there is only one common boundary edge shared by $\partial V(Q_1)$ and $\partial V(Q_2)$. Let $e_{1,2}$ denote the common boundary edge between $\partial V(Q_1)$ and $\partial V(Q_2)$.

In the proof of step 1, we proved that all cycle bases in V are blocked by guarding $|\alpha(V^*)|$ nodes in V. Thus, there is a guarded node in a cycle basis representing $\partial V(Q_1)$. This guarded node can exists on $e_{1,2}$ or outside $e_{1,2}$. If this guarded node is on $e_{1,2}$, then it is at one end point of $e_{1,2}$ using algorithm 1. Otherwise, the guarded node is on C, since $n(Q_1)$ has degree one. In both cases, C is blocked.

Until now, we proved that C is blocked as we guard $|\alpha(V^*)|$ nodes in V using algorithm 1. ■

We derive an upper bound for $g_I(V)$ in the case where V^* is a connected graph with more than one node. We need to introduce a new concept for this purpose. If an edge cover of V^* further satisfies that every cycle in V^* contains at least one edge in this edge cover, then we say the edge cover is *cycle-free*. Let $\beta(V^*) \subset E(V^*)$ denote a cycle-free edge cover of V^* with the minimum cardinality, i.e., the fewest number of edges.

In Fig. 4, The edges in an edge cover of V^* are depicted with dotted lines. The left sub-figure depicts a cycle-free edge cover of V^*. In the right sub-figure, a cycle consisting of four nodes ($n(O_1)$, $n(O_4)$, $n(O_6)$, and $n(O_2)$) does not contain a dotted edge. Thus, the right sub-figure shows an edge cover of V^* that is not cycle-free.

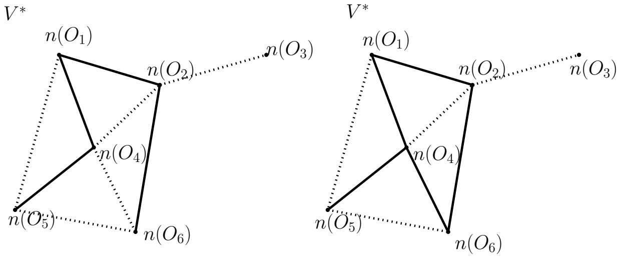

Fig. 2. Illustration of edge covers. Dotted lines in the left sub-figure show a cycle-free edge cover of V^*. Dotted lines in the right sub-figure show an edge cover of V^* that is not cycle-free.

In the case where V^* is a connected graph with more than one node, we replace $\alpha(V^*)$ in algorithm 1 by $\beta(V^*)$ to guard $|\beta(V^*)|$ nodes in V. In the following theorem, we prove that guarding $|\beta(V^*)|$ nodes using algorithm 1 blocks all cycles in V.

Theorem 2: For the Voronoi diagram V such that V^* is a connected graph with more than one node, $g_I(V) \leq |\beta(V^*)|$.

Proof: Since $g_I(V) \leq$MinGuard(V), we will prove that MinGuard$(V) \leq |\beta(V^*)|$ similar to the proof of Theorem 1. By contradiction, suppose that $|\beta(V^*)| <$MinGuard(V). $|\beta(V^*)| <$MinGuard(V) implies that as we guard $|\beta(V^*)|$ nodes in V, there is an unblocked cycle, say C, in V. Since $V^*[C]$ is connected, then $\partial(V^*[C])$ is connected.

Since $\partial(V^*[C])$ is connected, we consider two cases depending on the structure of $\partial(V^*[C])$. For these two cases, we will prove that an unblocked cycle, C, cannot exist as we guard $|\beta(V^*)|$ nodes in V using Algorithm 1. Hence MinGuard$(V) \leq |\beta(V^*)|$.

1) $\partial(V^*[C])$ is a tree graph.
2) $\partial(V^*[C])$ contains a cycle.

1. Since $\partial(V^*[C])$ is a tree graph, $\partial(V^*[C])$ can be a node or a tree graph containing at least one edge. Note that a cycle-free edge cover satisfies the condition for an edge cover. Hence, we replace $\alpha(V^*)$ in the proof of step 1 and 2 in Theorem 1 by $\beta(V^*)$ to obtain the result that C is blocked as we guard $|\beta(V^*)|$ nodes in V.

2. Consider the case where $\partial(V^*[C])$ contains a cycle, say C'. Suppose $N(C') = \{n(Q_1), n(Q_2), ..n(Q_{|N(C')|})\}$ and $E(C') = \bigcup_{i \leq |N(C')|}\{n(Q_i)n(Q_{i+1})\}$. In Fig. 3, $C' \subset \partial(V^*[C])$ is depicted with dashed line segments on $\partial(V^*[C])$.

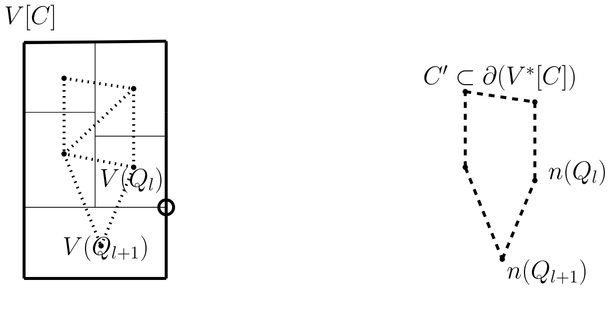

Fig. 3. C, $V[C]$, $V^*[C]$, and $\partial(V^*[C])$ are identical to those in Fig. 1. $C' \subset \partial(V^*[C])$ is depicted with dashed line segments on $\partial(V^*[C])$.

According to the definition of a cycle-free edge cover, $C' \subset V^*$ must contain an edge, say $n(Q_l)n(Q_{l+1})^m$ where m is used to distinguish among multiple edges between $n(Q_l)$ and $n(Q_{l+1})$, in $\beta(V^*) \subset E(V^*)$. Similar to $\alpha(V^*)$, there is one common boundary edge, say

$e_{l,l+1}$, shared by $\partial V(Q_l)$ and $\partial V(Q_{l+1})$ corresponding to $n(Q_l)n(Q_{l+1})^m$. Since both $n(Q_l)$ and $n(Q_{l+1})$ are on the boundary of the unbounded face of $V^*[C]$, $e_{l,l+1}$ meets C. Furthermore, C encloses both Q_l and Q_{l+1}.

Since $n(Q_l)n(Q_{l+1})^m$ is in $\beta(V^*)$, $n(Q_l)n(Q_{l+1})^m$ will be selected while running algorithm 1. Note that C is unblocked and encloses both Q_l and Q_{l+1}. Therefore, we guard the node where $e_{l,l+1}$ meets C according to algorithm 1. In Fig. 3, the circle on C represents the guarded node where $e_{l,l+1}$ meets C. In this way, C is blocked as we guard $|\beta(V^*)|$ nodes in V.

■

From now on, we study the relation between $g_I(V)$ and the number of obstacles in the workspace. Consider the case where there exists a perfect matching in V^* that is also a cycle-free edge cover of V^*. Fig. 4 illustrates this case. In the left sub-figure, the Voronoi diagram V and corresponding V^* are depicted with normal lines and dotted lines respectively. In the right sub-figure, the edges in a perfect matching in V^* are depicted with dashed lines. See that this perfect matching is also a cycle-free edge cover of V^*.

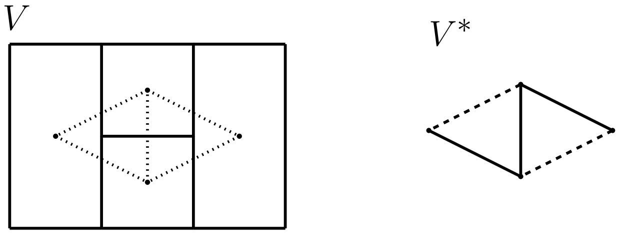

Fig. 4. A perfect matching in V^* that is also a cycle-free edge cover of V^*.

Corollary 1: Consider the Voronoi diagram $V = (N(V), E(V))$. If there exists a perfect matching in V^* that is also a cycle-free edge cover of V^*, then $g_I(V) \leq \frac{M-1}{2}$.

Proof:

We will show that V^* is a connected graph with more than one node. In Lemma 1, $V^* = V^*[C]$ if we choose $\partial V(O_M)$ as C. According to Lemma 1, V^* is a connected graph. Moreover, V^* has more than one node, since there exists a perfect matching in V^*.

Since V^* is a connected graph with more than one node, we have $g_I(V) \leq |\beta(V^*)|$ using Theorem 2. To prove that $g_I(V) \leq \frac{M-1}{2}$, we will show that $|\beta(V^*)| = \frac{M-1}{2}$. Let p denote a perfect matching in V^* that is also a cycle-free edge cover of V^*. Note that a cycle-free edge cover satisfies the condition for an edge cover. Thus, p is a cycle-free edge cover of V^* with the minimum cardinality $\frac{|N(V^*)|}{2}$. This further implies that $|\beta(V^*)| = \frac{|N(V^*)|}{2} = \frac{M-1}{2}$, since there are $M-1$ nodes in V^*.

■

IV. Interactive Online Game

Our intruder capture algorithms are implemented through an interactive online game [17] to assist humans in determining how to secure a complex graph. The game uses a greedy (approximation) algorithm [24] to select which nodes should be guarded so as to block all cycles in a given graph.

Fig. 5 shows a screen capture of the game taken during a running simulation. The user is free to enter any connected graph in the workspace by laying down a series of points and edges. After the graph has been drawn, the user may select edges on the graph where intruders, indicated by blue squares, should be initialized. The game will then deploy guards at selected nodes and indicate this by coloring those nodes red. As each node is chosen by the greedy algorithm, all the associated cycles that will be blocked by placing a guard on that node are momentarily highlighted. This is to assist the user in understanding the algorithm's selection process. When no more cycles remain, the user selects an edge on which the free searcher, depicted as a red square, should be deployed. The simulation then plays out as the searcher sequentially captures each intruder, which for the purposes of the simulation follows a simple evasion strategy of maximizing its distance from the nearest guard or free searcher.

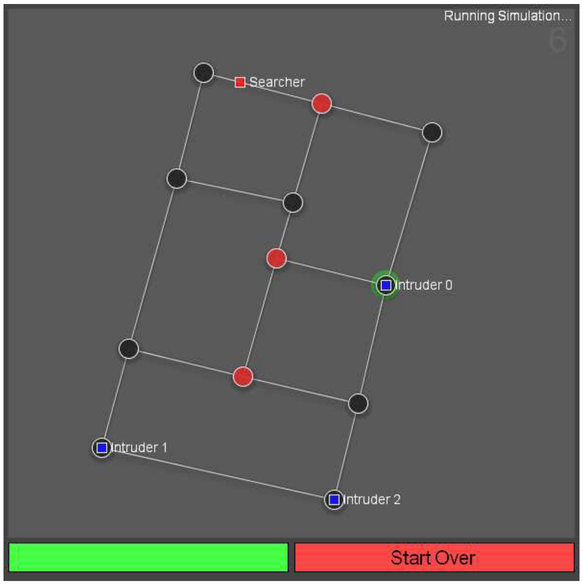

Fig. 5. A screen capture of the game taken during a running simulation.

V. Conclusions

We study intruder capturing game on the topological map of a workspace with obstacles, represented by the Voronoi diagram. Assuming that a searcher can access the position of any intruder utilizing the information network, a graph searching strategy is proposed so that one free searcher and multiple guards capture all intruders on a general graph. Based on this searching strategy, we derive theoretical upper bound for the minimum number of searchers required to capture all intruders on a general graph, which leads to a result on the Voronoi diagram. This strategy is further implemented through an interactive online game to assist humans to determine how to secure a complex graph.

Acknowledgment

The research work is supported by ONR grants N00014-08-1-1007, N00014-09-1-1074, and N00014-10-10712 (YIP), and NSF grants ECCS-0841195 (CAREER), CNS-0931576, and ECCS-1056253.

References

[1] J. Kim, F. Zhang, and M. Egerstedt, "Simultaneous cooperative exploration and networking based on Voronoi diagrams," in *proc. of IFAC Workshop on Networked Robotics*, Colorado, USA, 2009, pp. 1–6.

[2] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 125–137, 2001.

[3] K. Nagatani and H. Choset, "Toward robust sensor based exploration by constructing reduced generalized Voronoi graph," in *proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju, Korea, 1999, pp. 1687–1692.

[4] N. Rao, N. Stoltzfus, and S. Iyengar, "A retraction method for learned navigation in unknown terrains for a circular robot," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 699–707, 1991.

[5] H. Choset, I. Konukseven, and J. Burdick, "Mobile robot navigation: issues in implementating the generalized Voronoi graph in the plane," in *proc. of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington DC, USA, 1996, pp. 241–248.

[6] S. M. Lavalle, *Planning Algorithms*. Cambridge University Press, 2006.

[7] T. D. Parsons, "Pursuit-evasion in a graph," *Theory and Applications of Graphs, Lecture Notes in Mathematics, Springer-Verlag*, vol. 642, pp. 426–441, 1978.

[8] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, "The complexity of searching a graph," *Journal of the ACM*, vol. 35, pp. 18–44, 1988.

[9] A. Kolling and S. Carpin, "The graph-clear problem: definition, theoretical properties and its connections to multirobot aided surveillance," in *proc. of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sandiego, USA, 2009, pp. 1003–1008.

[10] A. Lapaugh, "Recontamination does not help to search a graph," *Journal of the ACM*, vol. 40(2), pp. 224–245, 1993.

[11] F. V. Fomin, P. Fraigniaud, and N. Nisse, "Nondeterministic graph searching: From pathwidth to treewidth," *Algorithmica*, vol. 53(3), pp. 358–373, 2009.

[12] L. Barrire, L. B. Ere, P. Flocchini, P. Fraigniaud, and N. Santoro, "Capture of an intruder by mobile agents," in *proc. of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, Canada, 2002, pp. 200–209.

[13] F. V. Fomin and D. M. Thilikos, "An annotated bibliography on guaranteed graph searching," *Theoretical Computer Science*, vol. 399, pp. 236–245, 2008.

[14] F. Makedon and I. H. Sudborough, "On minimizing width in linear layouts," *Discrete Applied Mathematics*, vol. 23(3), pp. 243–265, 1989.

[15] P. D. Seymour and R. Thomas, "Graph searching and a min-max theorem for tree-width," *J. Combin. Theory Ser. B*, vol. 58, pp. 22–33, 1993.

[16] D. Richerby and D. M. Thilikos, "Graph searching in a crime wave," *SIAM J. discrete mathematics*, vol. 23(1), pp. 349–368, 2009.

[17] J. Kim, S. Maxon, F. Zhang, and M. Egerstedt, "Intruder capture beta 1.2," Website, 2010, http://lamon.gtsav.gatech.edu/∼smaxon3/intruder_applet.html.

[18] B. W. Douglas, *Introduction to Graph Theory*, 2nd ed. Illinois, USA: Prentice Hall, 2001.

[19] J. Kim, F. Zhang, and M. Egerstedt, "A provably complete exploration strategy by constructing Voronoi diagrams," *Autonomous Robots*, vol. 29(3-4), pp. 367–380, 2010.

[20] L. M. Kirousis and C. H. Papadimitriou, "Interval graphs and searching," *Discrete Mathematics*, vol. 55, no. 2, pp. 181–184, 1985.

[21] ——, "Searching and pebbling," *Theoretical Computer Science*, vol. 42, no. 2, pp. 205–218, 1986.

[22] D. Bienstock and P. Seymour, "Monotonicity in graph searching," *Journal of Algorithms*, vol. 12, no. 2, pp. 239–245, 1991.

[23] N. D. Dendris, L. M. Kirousis, and D. M. Thilikos, "Fugutive-search games on graphs and related parameters," *Theoretical Computer Science*, vol. 172, pp. 233–254, 1997.

[24] H. Dutta, "Survey of approximation algorithms for set cover problem," Master's thesis, University of North Texas, 2009.