Continuous-Time Intruder Isolation Using Unattended Ground Sensors on a General Graph

Hua Chen, Krishna Kalyanam, Wei Zhang and David Casbeer

Abstract— This paper studies the continuous-time intruder isolation problem on a general road network graph under delayed information scenario. Several Unattended Ground Sensors (UGSs) are pre-installed along certain edges of the graph for detecting intruder motion and recording the detection time. Measurements of UGSs' can only be obtained by a UAV when the UAV is within a communication range of a UGS. The goal of this paper is to find the optimal path for the UAV to follow in order to capture the intruder within the shortest time, based on the delayed information from the visited UGSs. We propose an unfolding strategy to transform the road network graph to a decision tree incorporating delayed measurement information. Based on the decision tree, both optimal and sub-optimal min-max solutions are developed. Several interesting properties of the corresponding optimal value function are also derived. Numerical simulations based on a real road network are presented to demonstrate the effectiveness of the proposed strategies.

I. INTRODUCTION

This paper studies the intruder isolation problem, which involves using an Unmanned Aerial Vehicle (UAV) to isolate an intruder based on measurements taken locally by several Unattended Ground Sensors (UGSs). The intruder is assumed to travel on a road network at a constant speed, aiming at reaching a target territory of the network. The adversarial UAV flies at a faster speed and its motion needs not to follow the road network. Each UGS is able to detect the intruder motion and to send the measurement to the UAV whenever the UAV is inside the UGS's communication range. Isolation is achieved when the intruder arrives at a UGS above which the UAV is loitering. The goal of this paper is to find the best UGS sequence for the UAV to visit so as to minimize the isolation time under the worst possible intruder actions. This problem is of crucial importance for realistic surveillance in contested areas, where global communication is not available.

This UGS based intruder isolation problem can be thought of as an adversarial pursuit-evasion (PE) game under delayed information in a graph. Extensive efforts have been devoted into the field of PE games in graphs. Parsons's pioneering work [1] formulates the problem within the framework of graph theory. Considerable amount of research has been conducted on structure, capture time and search number of a graph [2]. It has been shown that an evader with zero visibility can be captured by a single pursuer on any graph using a random walk strategy with an expect capture time of $O(nm^2)$, where n is the number of vertices and m is the number of edges of the graph. Adler et al. improved the result to $O(n \log n)$ and proved the optimality of this bound in [3]. Isler et al. studied the cases in which certain constraints are imposed on the visibility of the pursuer or evader in [4], [5] and showed that a randomized strategy yields non-zero capture probability within finite time, yet the capture time grows exponentially with the number of vertices. Sufficient and necessary condition for guaranteed capture in a finite planar graph has been derived by Aigner and Fromme in [6] under full visibility assumption. Scenarios involving target domain of the evader have also been considered [7], [8] in general domains but not graphs and solutions employing Hamilton-Jacobi reachability approach are proposed.

Previous works mainly address the problem of finding the search number and the capture time of a PE game in graphs. However, the distinctive challenges which prevent us from directly applying the existing results in the literature are threefold: 1) information to the pursuer is delayed and only available at certain auxiliary nodes (UGSs) of the graph; 2) game domains for the intruder and the UAV are different; and 3) a target region of the intruder is considered.

Endeavors on closely related UGS-based intruder isolation problems are carried out in [9], [10], [11], yet limited to some special cases, where simple networks, such as finite grids and a special Manhattan grid road network, are considered in discrete time. Sufficient conditions for guaranteed capture on a simple network with a finite number of nodes in discrete time have been developed [9]. Optimal control of pursuers isolating a slower moving evader on a Manhattan grid road network has been studied and the corresponding strategies have been given [10], [11].

In this paper, we focus on the general continuous-time intruder isolation problem in which the intruder's movement is constrained within a general network graph and the UAV makes decisions under delayed information. We present an unfolding strategy to transform the road network graph to a decision tree which fully characterizes the uncertainties in the intruder's position. The decision-making problem is then interpreted as confirming or clearing certain branches of the decision tree. An optimal 'min-max' solution is developed using dynamic programming [12]. A sub-optimal 'min-max' solution is also presented to help reduce the computational

Hua Chen and Wei Zhang are with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210. Email: chenhu@ece.osu.edu, zhang@ece.osu.edu

Krishna Kalyanam is with InfoSciTex corporation, Dayton, OH 45431. Email: krishnak@ucla.edu

David Casbeer is with the Autonomous Control Branch, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433. Email: david.casbeer@us.af.mil
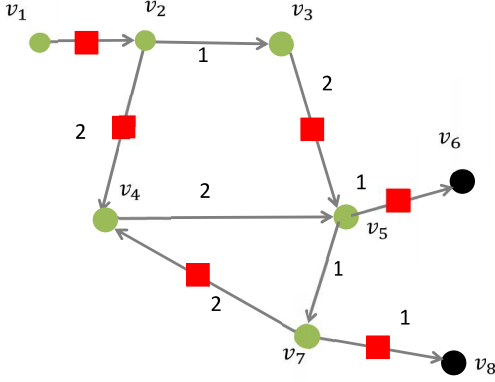
Fig. 1: Example of a simple road network with UGSs

complexity.

This paper unfolds as follows. A concise but rigorous formulation of the problem is provided in Section II. We present the optimal 'min-max' solution to the intruder isolation problem in Section III. Section IV presents a sub-optimal strategy which deals with the complexity issue that arises in computing the optimal solution. Numerical studies demonstrating the feasibility of the proposed solution methods are provided in Section V and concluding remarks are provided in Section VI.

II. PROBLEM FORMULATION

We consider a general road network described by a directed graph $\mathcal{G}_R = (\mathcal{V}_R, \mathcal{E}_R)$, where $\mathcal{V}_R$ is the set of vertices which represents the intersections of the road network, and $\mathcal{E}_R \subset \mathcal{V}_R \times \mathcal{V}_R$ is the set of edges which represents the available paths connecting the vertices. Suppose there are n_V vertices in $\mathcal{G}_R$, let $E_R = \{e_{i,k}\}_{i,k=1}^{n_V}$ be a weighted incidence matrix associated with the graph, where $e_{i,k}$ is the length of path connecting vertex i and vertex j if $(i,k) \in \mathcal{E}_R$ and $e_{i,k}$ equals infinity otherwise. Assume that n_U UGSs (Unattended Ground Sensors) are pre-installed along certain edges of the road network. Let $\mathbb{U} = \{1, \cdots, n_U\}$ be the index set of all UGSs and let $(v_1(j), v_2(j))$, $j \in \mathbb{U}$ be the edge of the graph associated with UGS j. A road network example is given in Fig. 1, where the circles represent the intersections, the arrows represent the directed road paths, the numbers represent the length of paths, and the squares represent the UGSs.

Suppose an intruder is traveling on $\mathcal{G}_R$ at a constant speed s_I while a UAV tasked with capturing the intruder flies at a faster speed s_U. The intruder's objective is to reach a set of target vertices $D \subset \mathcal{V}_R$ on the graph, represented by black circles in Fig. 1, without being captured by the UAV within a prescribed time horizon $T_h = [0, t_f]$. The intruder can be detected whenever it passes a UGS and the corresponding UGS can store the information about whether and when the intruder passes by. Communication range between the UAV and each UGS is very small as compared with the size of the road network. Hence, we assume that the UAV can obtain

measurements from a UGS only when it is directly above the corresponding UGS. It is also assumed that the layout of the road network and the deployment of the UGSs are known to the UAV *a prior*. Capture condition is defined as the intruder arrives at a UGS above which the UAV is loitering or the UAV and the intruder simultaneously arrive at the same UGS.

This game starts when the UAV receives a positive reading $\hat{t}_0$ for the first time from a UGS j_0. We are interested in finding the optimal sequence of UGSs for the UAV to visit and the associated waiting time around each UGS within the time horizon T_h, which minimizes the capture time based on the capture condition given above.

III. OPTIMAL MIN-MAX SOLUTION

In this section, we firstly propose an unfolding strategy which transforms the road network graph to a decision tree to incorporate the delayed information. Within the decision tree framework, the optimal min-max solution is developed using dynamic programming (DP). An important property of the value function associated with DP is discovered to help design valid discretization of the state space, which makes numerical solution possible to be achieved.

A. Unfolding the Graph

The first key step to solve the intruder isolation problem is to characterize the uncertainty set of the intruder's locations. We propose a scheme which unfolds the graph, to obtain a spatial-temporal tree $\tilde{\mathcal{G}} = (\mathcal{V}_{\tilde{\mathcal{G}}}, \mathcal{E}_{\tilde{\mathcal{G}}})$ characterizing the uncertainty set. Here, $\mathcal{V}_{\tilde{\mathcal{G}}}$ is the set of nodes in the decision tree and $\mathcal{E}_{\tilde{\mathcal{G}}} \subset \mathcal{V}_{\tilde{\mathcal{G}}} \times \mathcal{V}_{\tilde{\mathcal{G}}}$ is the set of arcs in the decision tree. Denote by $g = (\lambda_v(g), \lambda_t(g)) \in \mathcal{V}_{\tilde{\mathcal{G}}}$ a node in the decision tree, where $\lambda_v \in \mathcal{V}_R$ is the corresponding vertex in the road network graph and λ_t is the time for the intruder to arrive at this node for the first time. Set of children nodes $\mathcal{C}(g)$ of g is implicitly defined as follows

$$\mathcal{C}(g) = \{m \in \mathcal{V}_{\tilde{\mathcal{G}}} : e_{\lambda_v(g), \lambda_v(m)} \neq 0 \text{ and } \lambda_t(m) \leq t_f\}, \quad (1)$$

which means that if m satisfies: 1) there is an available path connecting the corresponding vertices of g and m, and 2) time for the intruder to reach m does not exceed t_f, then m is a children node of g. We denote by $\mathcal{R}$ the root of the decision tree and $\mathcal{L}$ the set of leaf nodes of a generic decision tree. Then the unfolding strategy can be described in an iterative way. Let $\Phi_1 = \mathcal{R}$ and let $\Phi_{l+1} = \bigcup_{g \in \Phi_l} \mathcal{C}(g)$ be the set of the children nodes of all the nodes in Φ_l. The unfolding strategy generating $\mathcal{V}_{\tilde{\mathcal{G}}}$ involves the following steps:

(1) For $l = 1$, initialize the iteration with $\Phi_1 = \mathcal{R}$;
(2) For each l, generate $\Phi_{l+1} = \mathcal{C}(\Phi_l)$;
(3) Repeat step (2) until $\Phi_{l+1} = \emptyset$;
(4) $\mathcal{V}_{\tilde{\mathcal{G}}} = \bigcup_l \Phi_l$.

Let $E_G = \{\epsilon_{i,j}\}_{i,j=1}^{|\mathcal{V}_G|}$ be the associated incidence matrix with $\epsilon_{i,j} = e_{\lambda_v(i), \lambda_v(j)}$ if $j \in \mathcal{C}(g)$ and $\epsilon = 0$ otherwise. Throughout this paper, $|\cdot|$ denotes the cardinality of a set.

The decision tree associated with the road network graph in Fig. 1 rooting from v_1 with time horizon $T_h = [0, 10]$ is shown in Fig. 2. Black circles represent the leaf nodes
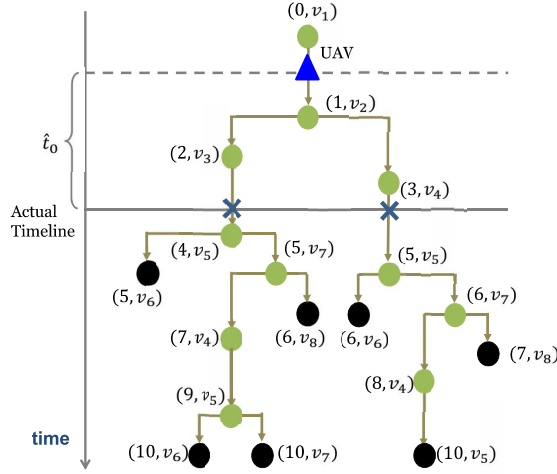
Fig. 2: Decision tree corresponding to road network graph in Fig. 1 with time horizon $T_h = [0, 10]$

while the green ones represent the others. The initial UAV position is represented by a blue triangle which coincides with the UGS along arc (v_1, v_2). The intersections of the actual timeline and the decision tree represent the possible locations of the intruder which are represented as the crosses in the figure.

B. Solution Algorithm Based on Dynamic Programming

Let $\tilde{\mathcal{G}}$ be the overall decision tree associated with the problem. Denote by $\mathcal{T}$ a generic subtree of $\tilde{\mathcal{G}}$ and $Leaf[\mathcal{T}]$ the set of leaf nodes in $\mathcal{T}$. Denote by $\mathcal{U}_{\tilde{\mathcal{G}}}$ the set of the UGSs in the decision tree. For each UGS $z \in \mathcal{U}_{\tilde{\mathcal{G}}}$, let $z = (\tilde{p}(z), \hat{p}(z), \gamma(z))$, where $\tilde{p}(z)$ denotes the first tree node of the arc where z is located, $\hat{p}(z) \in \mathbb{R}^2$ denotes the physical UGS location in real road network, and $\gamma(z)$ denotes the time instant when the intruder first reaches this UGS. Throughout this paper, $\|\cdot\|$ represents the classical Euclidean norm in $\mathbb{R}^2$.

Although time evolves continuously, decision makings only take place at discrete time instants when the UAV needs to determine which UGS to visit next. Control decision is a sequence of UGSs, which is the order of UGSs UAV needs to visit. We consider N decision stages, where N can be chosen such that the total time for the UAV to complete the trip is no smaller than t_f. Let $\mathcal{U} \subset \mathcal{U}_{\tilde{\mathcal{G}}}^N$ be the control space consisting of all the sequences $\mu = (u_0, \ldots, u_{N-1}) \in \mathcal{U}_{\tilde{\mathcal{G}}}^N$ with distinct entries, i.e., $u_i \neq u_j$ for all $i \neq j$. Denote by $\xi = (d_0, \ldots, d_{N-1})$ a disturbance sequence, which represents the intruder's actions. Each control action u_k may lead to two outcomes depending on the corresponding disturbance d_k. We use $d_k = 1$ to represent the case in which the corresponding UGS has been visited by the intruder and $d_k = 0$ to represent the opposite case.

Let $x = (\mathcal{T}, t, z) \in \mathcal{X}$ be a generic information state, where $\mathcal{T} \subseteq \tilde{\mathcal{G}}$ is a decision tree, t is current time value, z is the current UAV position which coincides with a UGS

position, and $\mathcal{X}$ is the set of all possible information states. There are two important subsets of $\mathcal{X}$, which are known as escape set and capture set. Escape set is defined as the set of information states for which the intruder reaches a leaf node without being captured, given by:

$$\mathcal{X}_e \triangleq \{(\mathcal{T}, t, z) : Leaf[\mathcal{T}] \neq \emptyset \text{ and } t \geq \min_{m \in Leaf[\mathcal{T}]} \gamma(m)\}. \tag{2}$$

Capture set is defined as the set of information states for which all leaf nodes of the current decision tree $\mathcal{T}$ are cleared within the time horizon, given by:

$$\mathcal{X}_c \triangleq \{(\mathcal{T}, t, z) : Leaf[\mathcal{T}] = \emptyset \text{ and } t < t_f\}. \tag{3}$$

The set of non-terminal states is represented as $\mathcal{X}_I \triangleq \mathcal{X}_c^c \cap \mathcal{X}_e^c$.

Dynamics of information state depends on both control and disturbance. Consider a nonterminal state $x = (\mathcal{T}, t, z) \in \mathcal{X}_I$ with control $u = z^+ \neq z, z^+ \in \mathcal{U}$, where $z^+ = (\tilde{p}^+, \hat{p}^+, \gamma^+)$ is a UGS different from $z = (\tilde{p}, \hat{p}, \gamma)$. Denote by $x^+ = (\mathcal{T}^+, t^+, z^+)$ the updated information state obtained by applying u. State update law is presented in the following, with respect to each component of x. Overall state update equation for the information state is given by:

$$x^+ = \begin{cases} x, & \text{if } x \in \mathcal{X}_e \cup \mathcal{X}_c, \\ f(x, u, d), & \text{otherwise.} \end{cases} \tag{4}$$

where the function f is given as follows:

$$f(x, u, d) = \begin{bmatrix} \Psi(\mathcal{T}, u, d) \\ t^+(z, u, t) \\ z^+ \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{T}^+ \\ t + \max\left\{\frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \gamma^+ - t\right\} \\ z^+ \end{bmatrix}. \tag{5}$$

Ψ is the state update equation for decision tree, given by:

$$\mathcal{T}^+ = \Psi(\mathcal{T}, u, d) = \begin{cases} \mathcal{T} \setminus \mathcal{T}', & \text{if } d = 0, \\ \mathcal{T}', & \text{if } d = 1. \end{cases}$$

We proceed to the definition of cost function after determining the structure of state update law. Running cost $l(x, u)$ is defined as the time UAV takes to confirm or clear the branch where u is located given current state x, which is the maximum of the time for the UAV to reach UGS u and the time for the intruder to reach UGS u given current state x. If the UAV reaches UGS u earlier than the intruder, it has to wait until when the intruder may reach this UGS to determine whether the intruder chooses this path. If the UAV reaches u later than the intruder, it can directly determine whether UGS u is triggered or not. Formulation of $l(x, u)$ is given by:

$$l(x, u) = \begin{cases} 0, & \text{if } x \in \mathcal{X}_c, \\ \infty, & \text{if } x \in \mathcal{X}_e, \\ \max\left\{\frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \gamma^+ - t\right\}, & \text{otherwise.} \end{cases} \tag{6}$$

Terminal cost function ϕ is defined as:

$$\phi(x) = \begin{cases} \infty, & \text{if } x \in \mathcal{X}_e, \\ 0, & \text{otherwise.} \end{cases}$$

For a given control-disturbance sequence (μ, ξ), where $\mu = (u_0, \ldots, u_{N-1})$ and $\xi = (d_0, \ldots, d_{N-1})$, the finite horizon additive cost function is given by:

$$J_N(x_0, \mu, \xi) = \sum_{k=0}^{N-1} l(x_k, u_k) + \phi(x_N).$$

The effect of the disturbance on the above cost function is through the state dynamics x_k.

The decision-making problem is formulated as finding the optimal state-feedback strategy $\mu(x)$ to minimize the closed-loop cost. The Bellman recursion to compute the optimal value function in this case is given by:

$$V_k(x) =$$
$$\begin{cases} \min_{u \in \mathcal{U}} \left\{ l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) \right\}, & \text{if } k \leq N-2 \\ \phi(x), & \text{if } k = N-1. \end{cases}$$

The optimal min-max feedback control and disturbance strategies are given by:

$$\begin{aligned} u_k^*(x) &= \arg\min_{u \in \mathcal{U}} \left\{ l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) \right\}, \\ d_k^*(x) &= \arg\max_{d \in \mathcal{D}} V_{k+1}(f(x, u_k^*(x), d)). \end{aligned} \tag{7}$$

To obtain the optimal (u^*, d^*) pair described in (7) using numerical backward induction, the value function needs to be evaluated for every information state $x \in \mathcal{X}$. Since time is a component of x, $\mathcal{X}$ is an uncountably infinite set. Therefore, we need to discretize time and use interpolation method to perform the numerical computation. Given a large time horizon, size of the set of discretized information states could be prohibitively large, which leads to a significant computational burden (curse of dimensionality). To tackle this complexity issue, we exploit a structure property in the optimal value function .

C. Monotonicity Property of the Value Function

This subsection presents an important monotonicity property of the value function which helps design efficient discretization of the state space contributing to the reduction of computational complexity.

To derive the monotonicity property, we firstly introduce two lemmas about the time component in the information state x and the running cost $l(x, u)$.

Lemma 1: For any non-terminal state $x = (\mathcal{T}, t, z) \in \mathcal{X}_I$ and for any $u = z^+ \in \mathcal{U}, z^+ \neq z$, the function $t^+(z, u, t)$ is constant with respect to t in $[\gamma, \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}]$ and is monotonically increasing with rate 1 for $t > \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}$.

Proof: From the state update function f in (5), we know that $t^+(z, u, t) = t + \max\left\{ \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \gamma^+ - t \right\}$. By simple geometric property and the constant speed assumption, we

have $\gamma^+ > \frac{\|\hat{p} - \hat{p}^+\|}{s_U}$. By continuity of $t^+(z, u, t)$ with respect to t, we know that:

$$\begin{cases} t^+(z, u, t) = \gamma^+, & \text{if } \gamma \leq t \leq \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \\ t^+(z, u, t) = t + \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, & \text{if } t > \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}. \end{cases}$$

Clearly, we know that $t^+(z, u, t)$ is constant with respect to t in $[\gamma, \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}]$ and is monotonically increasing with rate 1 for $t > \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}$. ■

Lemma 2: The running cost $l(x, u)$ monotonically decreases with rate -1 and then stays costant.

Proof: Since $l(x, u) = \max\left\{ \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \gamma^+ - t \right\}$, as analyzed above, we know:

$$\begin{cases} l(x, u) = \gamma^+ - t, & \text{if } \gamma \leq t \leq \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \\ l(x, u) = \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, & \text{if } t > \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}. \end{cases}$$

which is clear that the running cost decreases monotonically with rate -1 in $[\gamma, \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}]$, then stays constant. ■

Theorem 1: Given any information state x, the value function $V_k(x)$ is piecewise linear with respect to t and the discontinuities appear when the following conditions are satisfied:

$$\begin{cases} t = \gamma^+ - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}, \\ \text{and } V_{k+1}(f(x, u^*, 0)) < V_{k+1}(f(x, u^*, 1)). \end{cases}$$

where, u^* is the optimal strategy obtained as: $u^* = \arg\min_{u \in \mathcal{U}} \left\{ l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) \right\}$.

Proof: Suppose we know $V_{k+1}(f(x, u, d))$ for every pair (u, d) and the optimal control is given by: $u^* = \arg\min_{u \in \mathcal{U}} \left\{ l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) \right\}$. Denote by $t^c(u) = \gamma^+(u) - \frac{\|\hat{p} - \hat{p}^+\|}{s_U}$ the discontinuity for each u due to Lemma 1. Firstly, we claim that for $t \leq t^c(u)$, $\max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d))$ is constant and $\arg\max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) = 0$. Suppose not, i.e., $\arg\max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d)) = 1$ and assume time evolves for $t' = \frac{\|\hat{p} - \hat{p}^+\|}{s_U}$. Since $t \leq t^c(u)$, we know that $t + t' \leq t^c(u) + t' = \gamma^+(u)$, i.e., UAV reaches u earlier than the intruder. Clearly, $t \leq t_f$ is satisfied, if $d = 1$, then $Leaf[\mathcal{T}_{k+1}] = \emptyset$, capture condition is satisfied. However, $V_{k+1}(f(x, u, 0)) > 0$, therefore, $d = 0$ in this case. Hence, for $t \leq t^c(u)$, $\max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d))$ is constant. From Lemma 2, we know that $l(x, u)$ decreases with constant rate -1 for $t \leq t^c(u)$. Therefore, for each u, $l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d))$ decreases with constant rate -1 for $t \leq t^c(u)$. Additionally, if $V_{k+1}(f(x, u^*, 0)) < V_{k+1}(f(x, u^*, 1))$ and $t > t^c(u)$, there is a positive jump at $t = t^c(u)$. Hence, for the optimal control sequence, $V_k(x)$ has a positive jump at $t = t^c(u^*)$ if $V_{k+1}(f(x, u^*, 0)) < V_{k+1}(f(x, u^*, 1))$. ■

Remark 1: Theorem 1 shows that the value function is piecewise linear with constant rate -1 with respect to time t. Given the value functions of the states with time components at all discontinuities, value functions of other information

states can be obtained analytically due to this theorem. Therefore, we only need to compute the value function for a reduced set of information states with time component at all possible discontinuities for each u, which will alleviate some of the computational burden.

IV. SUB-OPTIMAL MIN-MAX SOLUTION

Even if we can reduce the set of information states remarkably using the property presented in the previous section, given a complex road network and a large time horizon, the size of the corresponding control space of the set of information states could also be very large. Therefore, efficient sub-optimal solutions are needed to tackle this complexity issue. In this subsection, we propose a sub-optimal strategy which involves using a reduced control space in computing the value function.

In order to generate the reduced control space for a given information state $x = (\mathcal{T}, t, z)$, we propose a scheme to rank each $\hat{u} \in \mathcal{U}$ based on its performance of min-max uncertainty reduction. For each $\hat{u} \in \mathcal{U}$, let $L(\hat{u}, \hat{d})$ be the number of leaf nodes given the control-disturbance pair $(\hat{u}, \hat{d})$. Hence, $L(\hat{u}, 1) = |Leaf[\mathcal{T}^+(\mathcal{T}, \hat{u}, 1)]|$ and $L(\hat{u}, 0) = |Leaf[\mathcal{T}^+(\mathcal{T}, \hat{u}, 0)]|$. Assign $\Omega(\hat{u}) = \max\{L(\hat{u}, 1), L(\hat{u}, 0)\}$ as the characterization of uncertainty reduction based on the worst intruder action for each $\hat{u} \in \mathcal{U}$. We define an order on $\mathcal{U}$ as: $\hat{u}_i \succ \hat{u}_j$ if $\Omega(\hat{u}_i) > \Omega(\hat{u}_j)$. Degree l approximation is then defined as choosing the first l smallest control actions to generate a reduced control space $\hat{\mathcal{U}}_{l,x}$ for the current information state x. For the sub-optimal control strategy, we firstly choose some l as the approximation degree and compute the reduced control space $\hat{\mathcal{U}}_{l,x}$ for the corresponding information state x. The value iteration is shown below in (8). Simulation results and performance analysis will be presented in Section V.

$$V_k(x) = \tag{8}$$

$$\begin{cases} \min_{u \in \hat{\mathcal{U}}_{l,x}} \{l(x, u) + \max_{d \in \mathcal{D}} V_{k+1}(f(x, u, d))\}, & \text{if } k \leq N - 2 \\ \phi(x), & \text{if } k = N - 1. \end{cases} \tag{9}$$

Remark 2: The optimal min-max strategy discussed in the previous section is obtained through backward Dynamic Programming (DP) which suffers from the curse of dimensionality due to the large information state space. To compute a solution, we need to compute the value function for all possible information states. An alternative approach is to compute the optimal (or sub-optimal) open-loop min-max solution (μ, ξ) for a given initial state. This approach avoids enumerating all the possible information states, thus reducing the complexity. The min-max sequence will be the optimal one if the intruder also adopts the min-max strategy. Once the initial state changes or the intruder acts differently from the min-max sequence, optimal min-max sequence need to be recomputed. In Section V, we apply the second open-loop approach to numerically compute the value functions for a given initial condition.

V. SIMULATION RESULTS
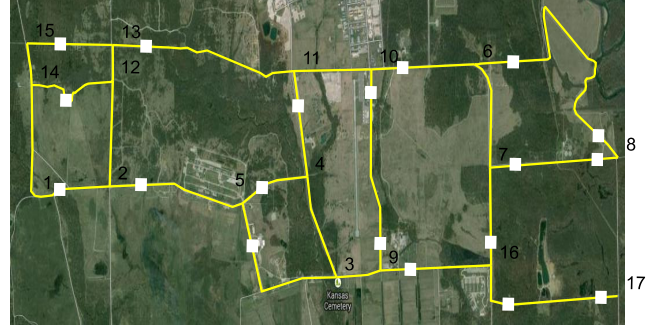
(a) Road network test bed at Camp Atterbury, IN
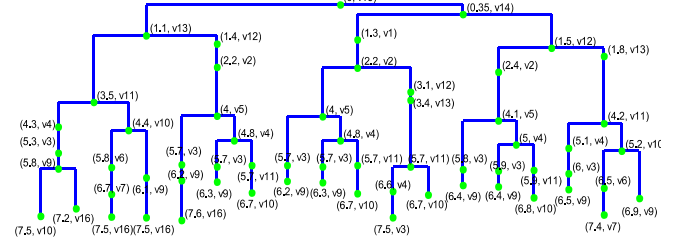
(b) Decision tree of the real road network with time horizon T_h starting from node v_{15}

Fig. 3: Real road network graph and corresponding decision tree

In this section, we use the real road network in Fig. 3(a) to test the algorithms proposed in previous sections. Vertices of the road network are labeled with numbers and UGSs along the edges are shown as brown squares. Vertex v_{15} is set to be the initial node and v_8 and v_{17} are referred to as the target vertices of the intruder. In this section, time horizon is set to $[0, 7.65]$. The corresponding decision tree is shown in Fig. 3(b). Time delay $\hat{t}_0$ for the UAV is set to be 1.5 and the UAV speed is configured at 2 times the intruder speed which is set to be 1.

A. Optimal Control Strategy

Firstly, we test the optimal min-max control strategy on the above configuration. A snapshot of the animation is shown in Fig. 4. Fig. 4 (a) shows the UAV-intruder movement on the road network. The actual intruder location is represented as the red star while the gray stars denote the other possible intruder (ghost) locations, based on the UAV's information. The intruder's path is highlighted in bold red lines. UAV position is denoted by the blue triangle and the dashed green line indicates the UAV's movement. Fig. 4 (b) shows the confirm/clear behavior of the decision tree where the gray

TABLE I: Optimal control solution

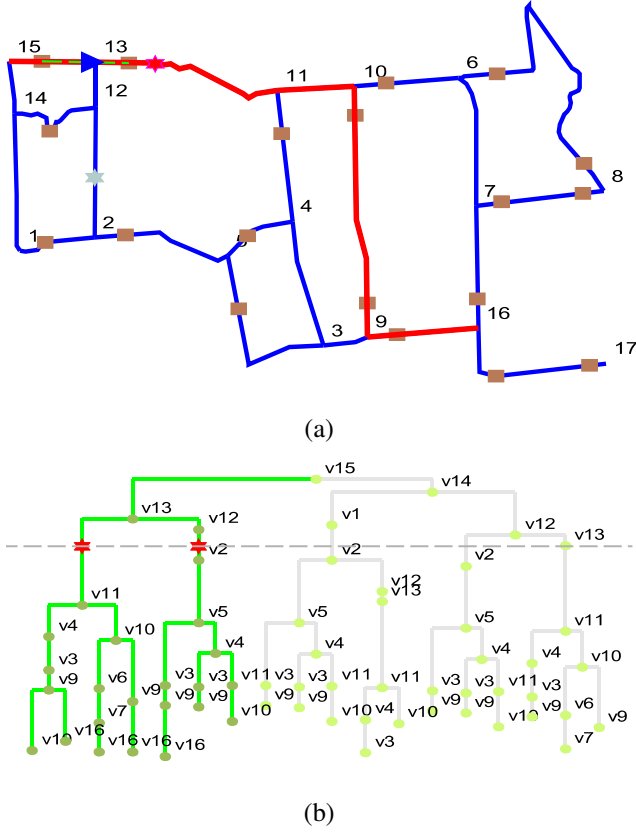| μ^*(node index)	2	4	8	15	16
d^*	1	1	0	0	0
cost	4.3968	3.8485	2.1304	1.0445	0

Fig. 4: (a) Simulation snapshot of optimal min-max solution in real road network; (b) The corresponding decision tree

part of the decision tree denotes the cleared subtree while the green part denotes the remaining part. The dashed gray line is the current time line. In Fig. 4, the intruder is moving from v_{13} to v_{11} and the UAV is moving from UGS (v_{15}, v_{13}) to (v_{13}, v_{11}) and will confirm the branch. The optimal min-max (u^*, d^*) pairs and the corresponding remaining costs are given in Table I. From Table I, we observe that capture happens under the worst case scenario. Therefore, we can conclude that the intruder can be isolated under the worst case scenario before it reaches any target vertex.

B. Sub-optimal Control Strategy

TABLE II: Running time and costs comparison between optimal and sub-optimal solutions

Degree(l)	1	2	3	4	5	Optimal
Time(sec)	0.0359	0.0102	0.0419	0.2466	1.1246	2289.3
Cost	∞	4.3968	4.3968	4.3968	4.3968	4.3968

Comparisons between optimal solution and sub-optimal solution are presented in Table II. We test the sub-optimal strategy with degrees 1 to 5. From Table II, we see that degree 1 approximation is not able to guarantee capture, while degrees 2 to 5 approximations can guarantee capture and have the same cost as the optimal solution. Indeed, the computation time for the sub-optimal solution is significantly

smaller than the optimal solution with no loss in performance.

VI. Conclusions and Future Work

In this paper, we develop an optimal control strategy to solve the continuous time intruder isolation problem on general graphs with an intruder traveling at a constant speed and a UAV having access only to delayed information. However, the complexity issue that naturally arises in backward dynamic programming prevents us from solving complex road networks with large time horizons. A monotonicity property inherent in the optimal value function helps us alleviate this problem. A sub-optimal strategy based on uncertainty reduction is proposed to help further reduce the computational complexity due to the size of the control space and it performs well compared to the optimal strategy.

In the future, we plan to investigate other sub-optimal ranking schemes incorporating real time uncertainty set of intruder position. In addition, the authors intend to extend this work to more complex and realistic scenarios, where the intruder moves at non-constant speeds and multiple UAVs are involved.

References

[1] T. D. Parsons, "Pursuit-evasion in a graph," *Theory and applications of graphs*, pp. 426–441, 1978.
[2] B. Alspach, "Searching and sweeping graphs: a brief survey," *Le matematiche*, vol. 59, no. 1, 2, pp. 5–37, 2006.
[3] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking, "Randomized pursuit-evasion in graphs," *Combinatorics Probability and Computing*, vol. 12, no. 3, pp. 225–244, 2003.
[4] V. Isler and N. Karnad, "The role of information in the cop-robber game," *Theoretical Computer Science*, vol. 399, no. 3, pp. 179–190, 2008.
[5] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion with local visibility," *SIAM Journal on Discrete Mathematics*, vol. 20, no. 1, pp. 26–41, 2006.
[6] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 1–12, 1984.
[7] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A casestudy on capture-the-flag," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
[8] S. Pan, H. Huang, W. Zhang, J. Ding, D. M. Stipanovic, and C. J. Tomlin, "Pursuit, evasion, and defense in the plane," in *Proceedings of the American Control Conference*, 2012.
[9] K. Krishnamoorthy, D. Casbeer, P. Chandler, M. Pachter, and S. Darbha, "UAV search & capture of a moving ground target under delayed information," in *Decision and Control (CDC), 51st Annual Conference on*, 2012, pp. 3092–3097.
[10] K. Kalyanam, S. Darbha, P. Khargonekar, M. Pachter, and P. Chandler, "Optimal cooperative pursuit on a manhattan grid," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, no. AIAA 2013-4633, 2013.
[11] K. Krishnamoorthy, S. Darbha, P. Khargonekar, D. Casbeer, P. Chandler, and M. Pachter, "Optimal minimax pursuit evasion on a manhattan grid," in *American Control Conference (ACC)*, 2013, pp. 3421–3428.
[12] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*, ser. Athena Scientific optimization and computation series. Athena Scientific, 2005, no. v. 1.