Intruder Isolation on a General Road Network Under Partial Information

Hua Chen, Krishnamoorthy Kalyanam, *Senior Member, IEEE*, Wei Zhang, *Member, IEEE*, and David Casbeer

***Abstract*— The intruder isolation problem (IIP) involves using an unmanned aerial vehicle (UAV) to isolate an intruder traveling along a road network. The UAV has no visibility of the intruder and can obtain the information only about the intruder's location from unattended ground sensors (UGSs) preinstalled on the road network. An UGS detects a passing intruder and records the time of passage. It can also upload the time-stamped information to the UAV directly overhead. This paper focuses on finding the optimal sequence of UGSs for the UAV to visit and the corresponding waiting time around each UGS to achieve isolation for the two variants of the IIP. This task is challenging due to the continuous movement of the intruder along the road network and the partial information scenario. To address these challenges, we propose an unfolding strategy to transform the road network to a decision tree, which describes all possible routes that the intruder may select to follow. Based on the decision tree, the optimal solution to the isolation problem is computed via dynamic programming (DP). To alleviate the computational complexity of DP, an UGS ranking scheme is proposed. Numerical implementations based on a real road network are presented to demonstrate the effectiveness of the proposed solution approach.**

***Index Terms*— Dynamic programming (DP), game theory, optimal control, pursuit evasion (PE), unmanned aerial vehicle (UAV).**

I. Introduction

THIS paper studies a novel intruder isolation problem (IIP), which involves using an unmanned aerial vehicle (UAV) to isolate a moving intruder traveling on a given road network. In this problem, the UAV is assumed to have no visibility of the intruder. It can only gain information about the intruder's location by visiting different unattended ground sensors (UGSs) and reading their measurements. These UGSs are preinstalled on the road network for detecting a passing intruder and they can upload the time-stamped information to the UAV directly overhead. It is assumed that the intruder has full visibility of the UAV and hence it may act adversarially against the UAV. The intruder tries to exit the network by reaching a prescribed region of the network, and the UAV aims to isolate the intruder and prevent it from exiting the network. Under this scenario, isolation occurs if the UAV knows the intruder's exact location. In this paper, we are interested in finding the optimal sequence of UGSs for the UAV to visit, which minimizes certain performance criteria for isolation. This problem is of crucial importance for realistic surveillance in contested areas, where global communication is not available.

From a game theoretic point of view, this UGS-based intruder-isolation problem can be viewed as a pursuit evasion (PE) game [1] on a graph. Extensive efforts have been devoted into this research field, mainly focusing on the topic of graph searching [2], [3]. Parsons's pioneering work [4] formulated this problem within the graph theory framework and introduced the notion of *search number* of a graph, which is the minimum number of pursuers that can guarantee the capture of the evader. A sufficient and necessary condition for guaranteed capture in a finite planar graph has been derived by Aigner and Fromme in [5] under full visibility assumption. Megiddo *et al.* [6] proved that determining the search number of a general graph is an NP-complete problem for general graphs and can be solved in linear time for trees. Later on, randomized pursuit strategies were developed to analyze this problem [7]–[10]. It has been shown in [7] that an intruder with zero visibility can be captured by a single pursuer on any graph using a random walk strategy with an expected capture time of $O(\mathrm{nm}^2)$, 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 [8]. Isler *et al.* studied the cases where certain constraints are imposed on the visibility of the pursuer or the evader in [9], [10]. In these works, it has been shown that a randomized strategy yields nonzero probability of capture in finite time, yet the expected capture time grows exponentially with the number of vertices of the graph.

Another important class of PE games related to our problem is the one where the intruder is trying not only to avoid the pursuers but also to reach a goal set safely to win the game. These types of problems were studied recently [11], [12], in which solutions based on Voronoi partitions and Hamilton–Jacobi reachability analysis are developed. This variant of classical PE games are closely related to the reachability analysis problems [13]–[15].

Works on closely related UGS-based IIPs have been studied very recently [16]–[20], mostly focusing on the discrete time setting where the UAV and the intruder share a common game domain and each of them occupies one vertex at each time

Manuscript received July 3, 2015; revised January 19, 2016; accepted March 19, 2016. Date of publication April 22, 2016; date of current version December 14, 2016. Manuscript received in final form March 30, 2016. Recommended by Associate Editor C. Canudas-de-Wit.

H. Chen and W. Zhang are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: chen.3824@osu.edu; zhang.491@osu.edu).

K. Kalyanam is with InfoSciTex Corporation (Air Force Research Laboratory contractor), Dayton, OH 45431 USA (e-mail: krishnak@ucla.edu).

D. Casbeer is with the Autonomous Control Branch, U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH 45433 USA (e-mail: david.casbeer@us.af.mil).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2016.2550423

step. Sufficient conditions for guaranteed capture on a simple network with a finite number of nodes in discrete time have been developed [17]. Optimal strategies for pursuers to isolate a slower moving intruder on a Manhattan grid has been studied and given in [16], [18]. Han and McEneaney [21] studied the problem where the UGSs are installed on several intersections of the road network and a ground vehicle is involved to assist in isolating the intruder.

Despite the rich literature of relevant research topics, the IIP studied in this paper has several distinctive challenges that cannot be directly addressed using the existing results. The primary challenge is that the UAV has zero visibility of the intruder and can only gain information about the intruder's location by visiting several auxiliary vertices (UGSs) of the graph. This delayed and incomplete information scenario distinguishes our problem from the classical PE games on graphs. Furthermore, our problem differs from the closely related work studied recently [16]–[18] due to the continuous time setting and the different game domains for the UAV and the intruder. These distinct features render the classical graph search methods and the recently developed solution approaches for simple road networks inapplicable to our problem. In this paper, to address these challenges, we first develop an unfolding strategy that transforms the road network graph to a decision tree, which explicitly characterizes the evolution of the intruder's possible locations. Adopting the decision tree structure, we present a complete solution to the scenario where the intruder's speed is assumed a known constant, which we refer to as the constant speed IIP (CS-IIP) in this paper. We also develop a solution to the scenario where the intruder's speed is assumed to be nonnegative and upper bounded by a known value, which we refer to as the nonconstant speed IIP (NS-IIP), utilizing the same approach. Both problems are reformulated as discrete time optimal control problems and are solved using dynamic programming (DP). To alleviate the computational complexity in solving the optimal control problems via DP, a suboptimal UGS ranking scheme, based on uncertainty reduction, is proposed. Simulation results of the optimal and suboptimal solution algorithms are presented.

The rest of this paper unfolds as follows. In Section II, we present a formulation of general IIPs and specify two particular problems we are interested in. In Section III, we first present the unfolding strategy and then develop the solution approach to CS-IIP based on the decision tree generated by the unfolding strategy. A similar decision-tree-based solution approach to NS-IIP is developed in Section IV. Numerical implementations on a real road network presented in Section V illustrate the effectiveness of our solution approach. Conclusions and possible future research directions are summarized in Section VI.

II. Problem Formulation

A. General Game Setting

Consider a general road network abstracted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, E, \mathcal{S}, \mathcal{F})$, where the following conditions hold.

1) $\mathcal{V} = \{V^i\}_{i=1}^{N_V} \subset \mathbb{R}^2$ is the set of intersections of the road network with labels $I_V = \{1, 2, \ldots, N_V\}$.

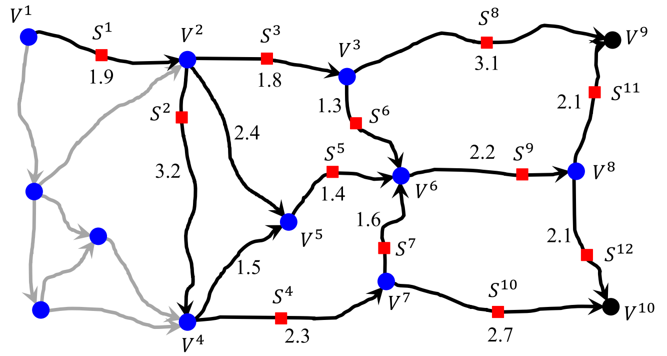


Fig. 1. Road network example and illustration of general game domain $\mathcal{G}$.

2) $\mathcal{E} \subset I_V \times I_V$ is the set of edges where $(i, j) \in \mathcal{E}, i, j \in I_V$ means there is a directed path from vertex V^i to vertex V^j, but not necessarily the opposite unless $(j, i) \in \mathcal{E}$.

3) $E = (e_{i,j}) \in \mathbb{R}_+^{N_V \times N_V}$ is the matrix indicating available paths connecting the intersections and the corresponding lengths, where $e_{i,j}$ is the length of the path if $(i, j) \in \mathcal{E}$ and $e_{i,j} = 0$ otherwise.

4) $\mathcal{S} = \{S^i\}_{i=1}^{N_S} \subset \mathbb{R}^2$ is the set of UGSs with labels $I_S = \{1, 2, \ldots, N_S\}$. We associate a pair $((i_1, i_2), R)$ with each UGS $S^i \in \mathcal{S}$ where $(i_1, i_2) \in \mathcal{E}$ is the edge the UGS lies along and $R \in (0, 1)$ is the distance ratio defined by the ratio of the distance from vertex V^{i_1} to the UGS S^i along the road network divided by the total length of the edge e_{i_1,i_2}.

5) $\mathcal{F} \subset \mathcal{V}$ is the target region of the intruder with the corresponding label set $I_F \subset I_V$.

A road network example is shown in Fig. 1, where the intersections $\mathcal{V}$ are shown as blue and black circles, with the black ones representing the target region $\mathcal{F}$ in particular. The UGSs $\mathcal{S}$ are represented by red squares. The available routes $\mathcal{E}$ and the corresponding matrix E are illustrated by the arrows and the numbers near the arrows.

Suppose an intruder travels along the road network $\mathcal{G}$ and tries to reach the target region $\mathcal{F}$ within a prescribed time horizon $T_h = [0, t_f]$. A UAV is trying to defend the target region by isolating the intruder within the time horizon. It is assumed that the layout of the road network and the deployment of the UGSs are known to the UAV, i.e., $\mathcal{G}$ is known to the UAV.

This game initializes when the UAV receives a positive measurement from a UGS, denoted by S^{i_0}, indicating the presence of an intruder on the network. This measurement is characterized by the *initial delay* in the formulation, denoted by δ and we call the pair (S^{i_0}, δ) a game initialization. One could imagine that prior to the initial measurement, the UAV is patrolling the UGS network looking for potential intruders. Arguably, the optimal patrol strategy would minimize the worst case initial delay. We are not concerned with the patrol strategy for the UAV in this paper, although admittedly it has a bearing on the initial delay.

B. Intruder Isolation Problem Formulation

Let $\mathbf{e}(t) \in \mathbb{R}^2$ be the intruder's real location at time t, let $\mathcal{B}(t) \subset \mathbb{R}^2$ be the set of all possible intruder's location at time t, which is referred to as the *uncertainty set* in the rest of this paper and let $\mathbf{p}(t) \in \mathbb{R}^2$ be the UAV's location at time t. Suppose the UAV flies at a constant speed $v_P > 0$ and can loiter above each UGS to obtain the measurement therein. We are interested in finding the optimal sequence of UGSs for the UAV to visit and the corresponding waiting time around each UGS, which minimizes the performance metric detailed below.

Suppose we consider N decision stages, where N can be chosen such that the total time for UAV to complete the trip is no smaller than t_f. Let $\mathbf{U} = \{U_k = (U_k^1, U_k^2)\}_{k=1}^N$ be the UAV's decision where $U_k^1 \in \mathcal{S} \setminus U_{k-1}^1$ is the next UGS to visit and $U_k^2 \in \mathbb{R}_+$ is the waiting time. Let $l(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k)$ be the cost function at stage k, which has the following feature in general:

$$l(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) \begin{cases} = 0, & \text{if isolation happens} \\ = \infty, & \text{if } \mathcal{B}(t_k) \cap \mathcal{F} \neq \emptyset \\ \in \mathbb{R}_+, & \text{otherwise} \end{cases} \tag{1}$$

where t_k is the time instant when the UAV needs to make the kth decision.

With these settings, the general IIP that we are interested in has the following form:

$$\min_{\mathbf{U}} \sum_{k=1}^{N} l(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) \tag{2}$$

$$\text{s.t. } \mathcal{B}(t_{k+1}) = f_{\mathcal{B}}(t_k, \mathcal{B}(t_k), \mathbf{p}(t_k), U_k, \mathbf{e}((t_k, t_{k+1}])) \tag{3}$$

$$\mathbf{p}(t_{k+1}) = f_{\mathbf{p}}(t_k, \mathbf{p}(t_k), U_k) \tag{4}$$

$$t_{k+1} = f_t(t_k, \mathbf{p}(t_k), U_k) \tag{5}$$

$$t_1 = 0, \quad U_0^1 = S^{i_0}, \quad U_0^2 = 0. \tag{6}$$

In the above problem, the tuple $(t, \mathbf{p}, \mathcal{B})$ can be viewed as the information state, $\mathbf{U}$ is the control decision and $\mathbf{e}$ can be considered as disturbance. Note that the IIP can be formulated to be a discrete time optimal control problem as above, since the UAV can only gain information from the UGSs.

Solving the above problem involves two major challenges: 1) the evolution of $\mathcal{B}$ depends on the intruder's real location $\mathbf{e}(t), t \in T_h$, which is unknown to the UAV in general, and 2) the evolution of $\mathcal{B}$, i.e., $f_{\mathcal{B}}$, is difficult to characterize.

In this paper, we assume worst case intruder actions characterized by the following assumption.

Assumption 1 (Worst Case Assumption): Assume the intruder plays adversarially against the UAV, i.e., the function $\mathbf{e}(t), t \in T_h$ maximizes the cost function.

Note that this assumption addresses the first challenge. In the sequel, we focus on developing solution approach to address the second challenge. In particular, we are interested in the following two IIPs.

C. Constant Speed Intruder Isolation Problem

The CS-IIP studies the problem in which the intruder speed is assumed to be a known constant during the time horizon T_h.

Assumption 2 (Constant Speed Assumption): Assume, the intruder's speed is a constant value $v_I > 0$ during the entire time horizon T_h.

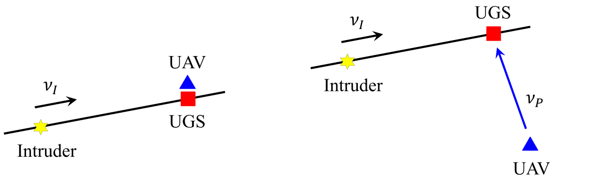


Fig. 2. Two possible scenarios of achieving point isolation condition. (a) Intruder arrives later than UAV. (b) Intruder and UAV arrive simultaneously.

Under this constant speed assumption, the uncertainty set $\mathcal{B}(t)$ consists of several isolated points. In this case, the UAV isolates the intruder if the following condition is achieved.

Definition 1 (Point Isolation Condition): The intruder is said to be isolated in CS-IIP at *isolation time* τ if the intruder's location coincides with the UAV's location at a UGS at time τ, namely, $\{\tau \in T_h | \exists S^j \in \mathcal{S} \text{ s.t. } \mathbf{e}(\tau) = \mathbf{p}(\tau) = S^j\} \neq \emptyset$.

There are two possible scenarios to achieve this isolation condition: 1) the intruder arrives at an UGS S^j where the UAV has arrived earlier and is waiting [see Fig. 2(a)] and 2) the intruder and the UAV arrive at an UGS S^j simultaneously [see Fig. 2(b)].

In CS-IIP, we are interested in finding the optimal sequence of UGSs for the UAV to visit and the corresponding waiting times to minimize the isolation time τ in Definition 1. Therefore, the cost function in this case has the following specific form:

$$l_C(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) = \begin{cases} 0, & \text{if } t_k \text{ is an } \textit{isolation time} \\ \infty, & \text{if } \mathcal{B}(t_k) \cap \mathcal{F} \neq \emptyset \\ t_{k+1} - t_k, & \text{otherwise.} \end{cases} \tag{7}$$

Intuitively, the cost function is the time spent within each decision stage. The CS-IIP is defined formally below:

Definition 2 (CS-IIP): The CS-IIP aims to solve the following optimal control problem:

$$\min_{\mathbf{U}=(U_1,U_2,\ldots,U_N)} \sum_{k=1}^{N} l_C(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) \tag{8}$$

$$\text{s.t. (3)–(6).} \tag{9}$$

D. Nonconstant Speed Intruder Isolation Problem

The NS-IIP studies a more general problem, where the intruder's speed assumption is relaxed as follows:

Assumption 3 (Nonconstant Speed Assumption): Assume, the intruder's speed can vary in the range $[0, v_I]$ with the known upper bound $v_I > 0$.

Under this bounded speed assumption, the uncertainty set $\mathcal{B}(t)$ is no longer several isolated points, but a connected subset of the road network. This is the most significant difference between CS-IIP and NS-IIP, which makes the *point isolation condition* difficult, if not impossible, to achieve. The reason is that under the worst case assumption (Assumption 1) the intruder can stop or reduce its speed to avoid being isolated at the next UGS location. Hence, in NS-IIP, it is essential to adopt a different isolation condition. We start with introducing the following notation, which plays an important role in the new isolation condition.

Definition 3: For any $\mathcal{B}(t)$, the mapping $\mu : \mathcal{B}(t) \mapsto \mu(\mathcal{B}(t))$ returns the total length of the road network covered by $\mathcal{B}(t)$ if $\mathcal{B}(t) \cap \mathcal{F} = \emptyset$ and $\mu(\mathcal{B}(t)) = \infty$ if $\mathcal{B}(t) \cap \mathcal{F} \neq \emptyset$.

In NS-IIP, the intruder is said to be isolated if the following *range isolation condition* is satisfied.

Definition 4 (Range Isolation Condition): The intruder is said to be isolated in NS-IIP with size c, if for a prescribed nonnegative constant $c \in \mathbb{R}_+$ we have $\mu(\mathcal{B}(t_f)) \leq c$, where $\mu(\mathcal{B}(t_f))$ is referred to as the *finite horizon size of uncertainty*.

Note that in NS-IIP, we are interested in minimizing the size of uncertainty set $\mathcal{B}$ at the terminal time t_f since in general, $\mu(\mathcal{B}(t))$ is not monotonic with respect to t. In NS-IIP, the cost function l_N is defined as follows:

$$l_N(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) = \begin{cases} \infty, & \text{if } \mathcal{B}(t_k) \cap \mathcal{F} \neq \emptyset \\ \mu(\mathcal{B}(t_{k+1})) - \mu(\mathcal{B}(t_k)), & \text{otherwise.} \end{cases} \tag{10}$$

The NS-IIP is defined formally below:

Definition 5 (NS-IIP): The NS-IIP aims to solve the following optimal control problem:

$$\min_{\mathbf{U}=(U_1,U_2,\ldots,U_N)} \sum_{k=1}^{N} l_N(\mathcal{B}(t_k), \mathbf{p}(t_k), U_k) \tag{11}$$

$$\text{s.t. } (3) - (6). \tag{12}$$

CS-IIP and NS-IIP are independent of each other but share the common challenge of characterizing the evolution of the uncertainty set $\mathcal{B}(t)$. In the following section, we present an unfolding strategy to transform any given road network to a decision tree which describes all possible routes the intruder may choose to follow and then develop the solution to CS-IIP based on the decision tree.

III. Solution to CS-IIP

A. Unfolding the Graph

In this section, we propose an unfolding strategy which transforms a given road network $\mathcal{G}$ and the game initialization (S^{i_0}, δ) into a decision tree. Intuitively, the decision tree explicitly specifies all possible paths that the intruder may choose to travel from the initial UGS S^{i_0} to the target region $\mathcal{F}$ within the given time horizon T_h. In general, we denote a decision tree by $\mathcal{T} = (\mathcal{N}, \mathcal{A}, \mathcal{U})$, where $\mathcal{N} = \{1, 2, \ldots, N_{\mathcal{N}}\}$ is the set of nodes, $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$ is the set of arcs and $\mathcal{U} = \{1, 2, \ldots, N_{\mathcal{U}}\}$ is the set of UGSs in the decision tree.

To begin with, we first review several important concepts in graph theory related to trees.

Definition 6 (Degree): The degree of a vertex in a graph (tree) is the total number of vertices adjacent to the vertex.

Definition 7 (Tree and Rooted Tree): A tree $\mathcal{T}$ is a connected graph with no cycles and a tree is called a rooted tree if one node has been designated the **root**, which is denoted by $\mathscr{R}(\mathcal{T})$.

Algorithm 1 Unfolding Strategy

```
Given $T_h$, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, E, \mathcal{S}, \mathcal{F})$ and $\mathcal{T} = (\{1,2\}, \{(1,2)\}, \emptyset)$;
 1: Set $i := 2$, $\tilde{\mathcal{T}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{A}}, \emptyset) := \mathcal{T}$,
 2: if $i \in \mathscr{L}(\mathcal{T})$ and $v[i] \notin I_F$ then
 3:     Compute $\mathcal{C}(i) := \{j \in I_V \big| e_{v[i],j} \neq 0$ and $\lambda[i] + e_{v[i],j} \leq T_h\}$ with label $l = 1, 2, \ldots, |\mathcal{C}(i)|$;
 4:     Set $\mathcal{N} := \tilde{\mathcal{N}} \cup \{|\tilde{\mathcal{N}}|+1, |\tilde{\mathcal{N}}|+2, \ldots, |\tilde{\mathcal{N}}|+|\mathcal{C}(i)|\}$ with $v[|\tilde{\mathcal{N}}|+l] = v[j_l]$ and $\lambda[|\tilde{\mathcal{N}}|+l] = \lambda[i] + e_{v[i],j_l}$ for $l = 1, 2, \ldots, |\mathcal{C}(i)|$;
 5:     $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \cup \{(i, |\tilde{\mathcal{N}}|+1), (i, |\tilde{\mathcal{N}}|+2), \ldots, (i, |\tilde{\mathcal{N}}|+|\mathcal{C}(i)|)\}$;
 6:     $i := i + 1$;
 7: end if
 8: if $\tilde{\mathcal{T}} = \mathcal{T}$ then
 9:     End;
10: else if $\tilde{\mathcal{T}} \neq \mathcal{T}$ then
11:     Set $\mathcal{T} := \tilde{\mathcal{T}}$ and go to step 2;
12: end if
```

With the definition of the degree of a tree, the leaf nodes of a rooted tree are defined as follows:

Definition 8 (Leaf Nodes): A leaf in a rooted tree is a vertex of degree 1 that is not the root. The set of all leaf nodes of a rooted tree $\mathcal{T}$ is denoted by $\mathscr{L}(\mathcal{T})$.

Note that, the decision tree we are trying to construct is not a rooted tree but is analogous. In particular, the distances between certain nodes in the decision tree are crucial in our framework and there are several ancillary nodes representing the UGSs that distinguish the decision tree from a classical rooted tree in graph theory.

Given a general road network $\mathcal{G}$, the time horizon T_h and a game initialization (S^{i_0}, δ), the unfolding strategy recursively generates the corresponding decision tree $\mathcal{T}$. We label each node $i \in \mathcal{N}$ with a pair $(v[i], \lambda[i])$ where $v[i] \in I_V$ is the corresponding index of the vertex $V^{v[i]}$ in $\mathcal{V}$ and $\lambda[i]$ is the distance between node i and the root node $\mathscr{R}(\mathcal{T})$ along the path indicated by the decision tree $\mathcal{T}$. The following algorithm summarizes the main steps of the unfolding strategy.

Once the pair $(\mathcal{N}, \mathcal{A})$ is obtained by applying Algorithm 1, the set $\mathcal{U}$ can be easily determined by checking the corresponding UGSs in $\mathcal{S}$ of each arc in $\mathcal{A}$, i.e., for any $(i, j) \in \mathcal{A}$, if there exists an UGS $S^k \in \mathcal{S}$ lying along the edge $(v[i], v[j]) \in \mathcal{E}$, then there is an UGS $l \in \mathcal{U}$ lying along the arc (i, j). Similar to the road network characterization, we associate each UGS $i \in \mathcal{U}$ with a pair $((i_1, i_2), \rho[i])$ where $(i_1, i_2) \in \mathcal{A}$ is the corresponding arc the UGS i lies along and $\rho[i]$ is the distance ratio which equals the value associated with the UGS $((v[i_1], v[i_2]), R)$ in the road network $\mathcal{G}$.

Fig. 3 presents an example using the road network $\mathcal{G}$ in Fig. 1 to illustrate how Algorithm 1 generates a decision tree and the overall decision tree generated is shown in Fig. 4(a).

To facilitate our later discussion on the solution to the CS-IIP, we introduce several notations that will be extensively used in the rest of this paper.

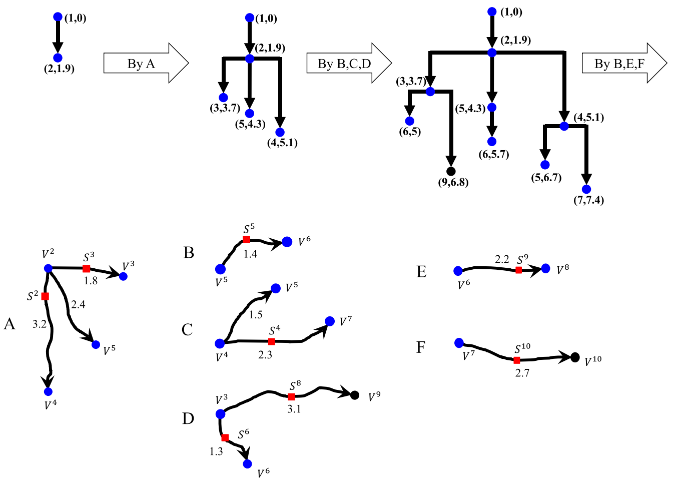
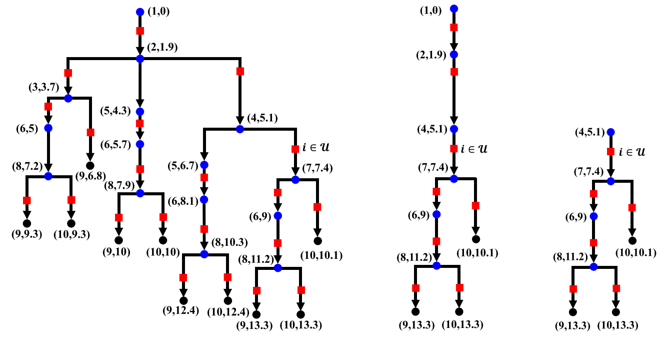


Fig. 3. Illustration of unfolding $\mathcal{G}$ in Fig. 1 assuming the initial UGS $S_{i_0} = S_1$.



(a) (b) (c)

Fig. 4. (a) Overall decision tree obtained given $\mathcal{G}$ in Fig. 1 and the initial UGS $S_{i_0} = S_1$. (b) Illustration of $\mathcal{T}_o$ for $i \in \mathcal{U}$. (c) Illustration of $\mathcal{T}_s$ for $i \in \mathcal{U}$.

Definition 9: Given a decision tree $\mathcal{T} = (\mathcal{N}, \mathcal{A}, \mathcal{U})$ the following holds.

1) $\mathcal{T}_o[i], i \in \mathcal{U}$ is the subtree of $\mathcal{T}$ rooting from $\mathscr{R}(\mathcal{T})$ and passing through UGS i, see Fig. 4(b).
2) $\mathcal{T}_s[i], i \in \mathcal{U}$ is the subtree of $\mathcal{T}$ rooting from node i_1 and passing through UGS i, see Fig. 4(c).
3) $\gamma[i], i \in \mathcal{U}$ is the distance from root $\mathscr{R}(\mathcal{T})$ to UGS $i \in \mathcal{U}$ computed by $\gamma[i] = \lambda[i_1] + \rho[i] e_{v[i_1],v[i_2]}$.
4) $S[i] \in \mathcal{S}, i \in \mathcal{U}$ is the UGS in $\mathcal{G}$ corresponding to UGS $i \in \mathcal{U}$.

Employing the decision tree structure, we present a reformulation of the CS-IIP in the following section and propose a solution algorithm based on DP.

B. Optimal Min–Max Solution

We are interested in finding a sequence of UGSs for the UAV to visit and the corresponding waiting time around each UGS that minimize the isolation time. As discussed in Section III-A, the decision tree explicitly characterizes all possible paths the intruder may follow that originate from the initial UGS $S^{i_0} \in \mathcal{G}$ toward the target region $\mathcal{F}$ within the given time horizon T_h. In CS-IIP, due to the constant speed assumption (Assumption 2), the intruder's possible locations at a given time t' can be easily identified using the decision tree, which are just the intersections of the decision tree $\mathcal{T}$ and the time line at $t = t'$.

Let $x = (\overline{\mathcal{T}}, t, z)$ be the information state in the game, where $\overline{\mathcal{T}} \subseteq \mathcal{T}$ is a generic subtree of the overall decision tree, $t \in T_h$ is the time component, and $z \in \mathcal{U}$ is the UAV's location coinciding with an UGS in $\mathcal{T}$. Note that, both $\overline{\mathcal{T}}$ and z depend on t, but we drop the dependence for notational simplicity. In fact, this information state captures most of the information encoded in the tuple $(t, \mathbf{p}, \mathcal{B})$, which is viewed as the information state in the original problem. Let $\mathcal{X}$ be the set of all possible information states x. The following two important subsets of $\mathcal{X}$ characterize the isolation condition and the escape condition:

Isolation set:

$$\mathcal{X}_I \triangleq \{x \in \mathcal{X} : \mathscr{L}(\overline{\mathcal{T}}) = \emptyset \text{ and } t < t_f\}.$$

Escape set:

$$\mathcal{X}_E \triangleq \{x \in \mathcal{X} : \mathscr{L}(\overline{\mathcal{T}}) \neq \emptyset \text{ and } t \geq \min_{i \in \mathscr{L}(\overline{\mathcal{T}})} \frac{\lambda[i]}{v_I}\}. \tag{13}$$

If the system reaches a state in $\mathcal{X}_E$, it implies that the intruder has reached one of the leaf nodes without being isolated within the time horizon. On the other hand, if the system reaches a state in $\mathcal{X}_I$, then all leaf nodes are cleared within the given time horizon, i.e., the UAV has successfully isolated the intruder. All other information states form the set of nonterminal states, denoted by $\mathcal{X}_C = \mathcal{X} \setminus (\mathcal{X}_E \cup \mathcal{X}_I)$.

Without loss of generality, we consider N decision stages. The control decision is defined as $\xi = (u_0, \ldots, u_{N-1}) \in \prod_{k=0}^{N-1} \mathcal{U}_k$ where $\mathcal{U}_k$ is the UGS set in the decision tree at decision stage k and u_k is the UGS to visit at decision stage k. Due to the constant speed assumption, the waiting time around each UGS can be determined uniquely and precisely. Hence the waiting times do not appear in the decision vector ξ. Moreover, the disturbance vector can be simplified to be just a sequence of 0's and 1's indicating whether the corresponding UGS u_k is triggered or not, ignoring the actual measurement at each UGS. The disturbance vector is denoted by $\eta = (d_0, \ldots, d_{N-1}) \in \mathcal{D}^N$, where $\mathcal{D} = \{0, 1\}$. We use $d_k = 1$ to denote the case where the corresponding UGS is triggered by the intruder, which is called a "confirm," and use $d_k = 0$ to denote the opposite case which is called a "clear."

The dynamics of the information state $x_k = (\overline{\mathcal{T}}_k, t_k, z_k)$ is given by

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, d_k) \\ &= \begin{cases} x_k, & \text{if } x_k \in \mathcal{X}_E \cup \mathcal{X}_I \\ (\overline{\mathcal{T}}_{k+1}, t_{k+1}, z_{k+1}), & \text{otherwise} \end{cases} \end{aligned} \tag{14}$$

where the updated information state $(\overline{\mathcal{T}}_{k+1}, t_{k+1}, z_{k+1})$ is defined element-wise as follows:

1) *Decision Tree:* Update of decision tree is given by

$$\overline{\mathcal{T}}_{k+1} = \begin{cases} \overline{\mathcal{T}}_k \setminus \mathcal{T}_s[u_k], & \text{if } d_k = 0 \\ \mathcal{T}_o[u_k], & \text{if } d_k = 1 \end{cases} \tag{15}$$

where both $\mathcal{T}_o[u_k]$ and $\mathcal{T}_s[u_k]$ are defined in Definition 9.

2) *Decision Time:* $t_{k+1} - t_k$ is the time spent in decision stage k, which is the time for UAV to fly from z_k to UGS u_k plus the waiting time at UGS u_k. Note that given x_k and u_k, the value of $t_{k+1}-t_k$ is the larger value between the time for the UAV to reach u_k and the time for the intruder to reach u_k, which is given by

$$\begin{aligned} t_{k+1} - t_k \\ &= \max\left\{\frac{\|S[u_k] - S[z_k]\|_2}{\nu_P}, \frac{\gamma[u_k]}{\nu_I} - t_k\right\}, \end{aligned} \tag{16}$$

where $\|\cdot\|$ denotes the Euclidean norm.

The first term in the max function in (16) is the time for the UAV to reach UGS u_k and the second term is that for the intruder to reach UGS u_k. Consequently, we have

$$t_{k+1} = d_k\frac{\|S[u_k] - S[z_k]\|_2}{\nu_P} + (1-d_k)\frac{\gamma[u_k]}{\nu_I} + d_k t_k. \tag{17}$$

3) *UAV Location:*

$$z_{k+1} = u_k \tag{18}$$

since z_{k+1} is simply the next UGS location.

Given an initial state $x_0 \in \mathcal{X}_C$, a control decision $\xi \in \prod_{k=0}^{N-1}\mathcal{U}_k$ and a disturbance $\eta \in \mathcal{D}^N$, the running cost l_C defined in (7) is equivalently written as follows:

$$\begin{aligned} &g_k(x_k, u_k) \\ &= \begin{cases} 0, & \text{if } x_k \in \mathcal{X}_I \\ \infty, & \text{if } x_k \in \mathcal{X}_E \\ \max\left\{\dfrac{\|S[u_k]-S[z_k]\|}{\nu_P}, \dfrac{\gamma[u_k]}{\nu_I} - t_k\right\} & \text{otherwise.} \end{cases} \end{aligned} \tag{19}$$

The corresponding finite horizon additive cost function is then given by

$$J_N(x_0, \xi, \eta) \triangleq \sum_{k=0}^{N-1} g_k(x_k, u_k) + \phi(x_N) \tag{20}$$

where $\phi(\cdot)$ is the terminal cost function defined as

$$\phi(x) = \begin{cases} \infty, & \text{if } x \in \mathcal{X}_E \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

The CS-IIP we are interested in is reformulated as the following finite horizon discrete time optimal control problem:

$$\text{CS-IIP}: \begin{cases} \min\limits_{\xi \in \prod_{k=0}^{N-1}\mathcal{U}_k} \max\limits_{\eta \in \mathcal{D}^N} & J(x_0, \xi, \eta) \\ \text{s.t.} & x_{k+1} = f(x_k, u_k, d_k), x_0 \in \mathcal{X}_C. \end{cases} \tag{22}$$

Let $H_k(x)$ be the value function at stage k for the given state x. The Bellman recursion to compute the optimal value function is given by

$$H_k(x) = \begin{cases} \min\limits_{u_k \in \mathcal{U}_k} \max\limits_{d_k \in \mathcal{D}_k} \{g_k(x, u_k) + H_{k+1}(f(x, u_k, d_k))\} \\ \qquad\qquad\qquad\qquad \text{if } 0 \leq k \leq N-2 \\ \phi(f(x, u_k, d_k)), \quad \text{if } k = N-1 \end{cases} \tag{23}$$

where $x \in \mathcal{X}$ is the information state and $\phi(x)$ is the terminal cost.

The optimality condition [22, Th. 17.2] dictates that the optimal solution to (22) and (23) is given by the *saddle-point equilibrium* (ξ^*, η^*) where $\xi^* = (\xi_0^*, \xi_1^*, \ldots, \xi_{N-1}^*)$ and $\eta^* = (\eta_0^*, \eta_1^*, \ldots, \eta_{N-1}^*)$ given below

$$\begin{aligned} \xi_k^*(x) &= \arg\min_{u_k \in \mathcal{U}_k}\left\{\max_{d_k \in \mathcal{D}_k}\{g_k(x, u_k) + H_{k+1}(f(x, u_k, d_k))\}\right\} \\ \eta_k^*(x) &= \arg\max_{d_k \in \mathcal{D}_k}\{H_{k+1}(f(x, \xi_k^*(x_k), d_k))\}. \end{aligned} \tag{24}$$

Remark 1: The optimal solution described in (24) encodes a necessary and sufficient condition for the existence of solution to the CS-IIP in Definition 2, in the sense that (24) returns a solution if and only if CS-IIP is solvable. This is because the new problem utilizing the decision tree information in (22) is equivalent to the original one defined in Definition 2. In addition, the optimal solution obtained by (24) may not be unique. This fact will be illustrated by numerical implementation result in Section V.

Remark 2: It is well-known that closed form solutions to (24) do not exist in general and solving the Bellman recursion (23) suffers from the *curse of dimensionality*. In our problem, the value function $H_k(x)$ needs to be evaluated for every possible state $x \in \mathcal{X}$ to obtain the numerical solution. Unfortunately, time t is the second component of the information state, which makes the set of all information states to be infinite. Efficient discretization of the state space $\mathcal{X}$ is essential to numerically compute the optimal value function H^* and the optimal solution (ξ^*, η^*). In the following section, a monotonicity property of the value function $H_k(x)$ with respect to time t is presented, which helps design an effective discretization of $\mathcal{X}$.

C. Monotonicity Property of the Value Function in CS-IIP

By inspecting the structure of the value function in CS-IIP carefully, we discover an important monotonicity property, which encodes an efficient discretization scheme, allowing for tractable numerical implementation of our algorithm. Consider a single stage decision with a nonterminal information state, $x = (\overline{\mathcal{T}}, t, z) \in \mathcal{X}_C$. Denote by $x^+ = (\overline{\mathcal{T}}^+, t^+, z^+) = f(x, u, d)$ the state obtained by applying control u under disturbance d.

To derive the monotonicity property, we start with introducing two lemmas related to the time component t in information state x and running cost $g(x, u)$.

Lemma 1: Given any nonterminal state $x = (\overline{\mathcal{T}}, t, z) \in \mathcal{X}_C$, any feasible control $u \in \mathcal{U}$ and any disturbance $d \in \{0, 1\}$, value of t^+ is constant for $t \in [\gamma[z], (\gamma[u]/\nu_I) - (\|S[u] - S[z]\|/\nu_P)]$ and is monotonically increasing with rate +1 for $t \in ((\gamma[u]/\nu_I) - (\|S[u] - S[z]\|/\nu_P), +\infty)$.

Proof: From update equation of t in (16), we know that $t^+ = t + \max\{((\|S[u] - S[z]\|/\nu_P), (\gamma[u]/\nu_I) - t\}$. By triangle inequality and the constant speed assumption, we have $(\gamma[u]/\nu_I) \geq t + (\|S[u] - S[z]\|/\nu_P)$. By continuity of t^+ with respect to t, it follows that:

$$t^+ = \begin{cases} \dfrac{\gamma[u]}{\nu_I}, & \text{if } t \in \left[\gamma[z], \dfrac{\gamma[u]}{\nu_I} - \dfrac{\|S[u]-S[z]\|}{\nu_P}\right] \\ t + \dfrac{\|S[u]-S[z]\|}{\nu_P}, & \\ \quad \text{if } t \in \left(\dfrac{\gamma[u]}{\nu_I} - \dfrac{\|S[u]-S[z]\|}{\nu_P}, +\infty\right). \end{cases}$$

As $(\gamma[u]/\nu_I)$ and $(\|S[u]-S[z]\|/\nu_P)$ depend solely on the structure of the decision tree and speed of the intruder ν_I, they are constants here. Hence, we know that t^+ is constant for $t \in [\gamma[z], (\gamma[u]/\nu_I) - (\|S[u]-S[z]\|/\nu_P)]$ and is monotonically increasing with rate 1 for $t \in ((\gamma[u]/\nu_I) - (\|S[u]-S[z]\|/\nu_P), +\infty)$. □

Lemma 1 explores a piecewise linearity property of time component in the information state x.

The following lemma presents a monotonicity property of the running cost $g(x,u)$.

Lemma 2: The running cost function $g(x,u)$ monotonically decreases with rate 1 with respect to time for $t \in [\gamma[z], (\gamma[u]/\nu_I) - (\|S[u]-S[z]\|/\nu_P)]$ and then stays constant.

Proof: Since $g(x,u) = \max\{(\|S[u]-S[z]\|/\nu_P), (\gamma[u]/\nu_I) - t\}$ as given in (19), we know

$$g(x,u) = \begin{cases} \dfrac{\gamma[u]}{\nu_I} - t, & \text{if } t \in \left[\gamma[z], \dfrac{\gamma[u]}{\nu_I} - \dfrac{\|S[u]-S[z]\|}{\nu_P}\right] \\ \dfrac{\|S[u]-S[z]\|}{\nu_P}, & \text{if } t \in \left(\dfrac{\gamma[u]}{\nu_I} - \dfrac{\|S[u]-S[z]\|}{\nu_P}, +\infty\right). \end{cases}$$

Following the same argument as in the proof of Lemma 1, it is clear that the running cost decreases monotonically with rate 1 for $t \in [\gamma[z], (\gamma[u]/\nu_I) - (\|S[u]-S[z]\|/\nu_P]$ then stays constant at $(\|S[u]-S[z]\|/\nu_P)$. □

With the above two lemmas, we are ready to introduce the main theorem.

Theorem 1: The value function $H_k(x)$ is piecewise linear with respect to t with discontinuities appearing when the following conditions are satisfied simultaneously:

$$t = \frac{\gamma[u]}{\nu_I} - \frac{\|S[u]-S[z]\|}{\nu_P} \tag{25}$$

and

$$H_{k+1}(f(x,u^*,0)) < H_{k+1}(f(x,u^*,1)) \tag{26}$$

where, u^* is the optimal control decision given by

$$u^* = \arg\min_{u \in \mathcal{U}} \left\{ g(x,u) + \max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d)) \right\}. \tag{27}$$

Proof: Suppose we know $H_{k+1}(f(x,u,d))$ for every pair (u,d) and the optimal control action y^* is given by (27). Denote by $\hat{t}(u) = (\gamma[u]/\nu_I) - (\|S[u]-S[z]\|/\nu_P)$ the discontinuity for each u due to Lemma 1.

To begin with, we claim that if $t \le \hat{t}(u)$, then $\max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d))$ is constant for all k and $\arg\max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d)) = 0$. We prove this claim by contradiction. Suppose it is not. Then we have $\arg\max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d)) = 1$. Assume time evolves for $\Delta t = (\|S[u]-S[z]\|/\nu_P)$. Since $t \le \hat{t}(u)$, we know that $t + \Delta t \le \hat{t}(u) + \Delta t = (\gamma[u]/\nu_I)$, i.e., UAV reaches u earlier than the intruder. Clearly, $t \le t_f$ is satisfied, if $d = 1$, then $\mathscr{L}(\overline{\mathcal{T}^+}) = \emptyset$, which indicates that capture condition is satisfied, i.e., $H_{k+1}(f(x,u,1)) = 0$. Since $H_{k+1}(f,u,0) \ge 0$, if $H_{k+1}(f,u,0) = 0$, then it follows that both $d = 1$ and $d = 0$ will yield isolation, which is impossible. Therefore, $H_{k+1}(f,u,0) > 0$ and hence $\arg\max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d)) = 0$ is the only possibility.

For $t \le \hat{t}(u)$, $\max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d))$ is constant. From Lemma 2, we know that $g(x,u)$ decreases with constant rate 1 for all $t \le \hat{t}(u)$. Therefore, for each feasible u, we have $g(x,u) + \max_{d \in \mathcal{D}} H_{k+1}(f(x,u,d))$ decreases with constant rate 1 for all $t \le \hat{t}(u)$. Moreover, if $H_{k+1}(f(x,u^*,0)) < H_{k+1}(f(x,u^*,1))$ and $t > \hat{t}(u)$, there is a positive jump at $t = \hat{t}(u)$. Consequently, for the optimal control sequence, $H_k(x)$ has a positive jump at $t = \hat{t}(u^*)$ if $H_{k+1}(f(x,u^*,0)) < H_{k+1}(f(x,u^*,1))$ and is piecewise linear with constant slope -1 for all other t. □

Theorem 1 shows that the value function is piecewise linear with constant slope -1 with respect to time t and jumps positively when certain conditions are satisfied.

Remark 3: The piecewise linearity property encoded in the value function provides an effective way of discretizing the state space. In particular, the value function $H_k(x)$ only needs to be evaluated at the states whose corresponding time components are at the discontinuity points.

However, even with the efficient discretization induced by Theorem 1, the decision making problem is still *NP-hard* and the computational complexity grows exponentially with the size of the problem, in particular with the number of the UGSs in the decision tree, i.e., $|\mathcal{U}|$.

In the following section, we propose a ranking scheme to generate a reduced control space, which results in a tractable suboptimal solution algorithm for our decision making problem.

Remark 4: Numerical solution to (24) still easily becomes intractable even though the state space can be reduced according to Theorem 1. One possible solution is to compute the optimal open-loop min–max solution (ξ_{ol}, η_{ol}) for a given initial state. This approach avoids enumerating all the possible information states, thus reduces the complexity. The solution will be optimal 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 needs to be recomputed. What describes above is the receding horizon approach that computes the optimal sequence and implements the first action on the fly based on the updated information. In Section V, an open-loop min–max solution for a given initial state is presented and the receding horizon control is also implemented for an arbitrarily prescribed intruder's path.

D. Complexity Issue and Suboptimal Solution

Even though we have discussed possible discretization of the state space to alleviate the computational burden, the problem still easily becomes intractable for a large number of UGSs or a large time horizon. To handle this issue, a tractable suboptimal solution algorithm is presented in this section based on a reduction of the control space.

In order to generate a reduced control space for a given information state $x = (\overline{\mathcal{T}}, t, z)$ with $\overline{\mathcal{T}} = (\overline{\mathcal{N}}, \overline{\mathcal{A}}, \overline{\mathcal{U}})$, we propose a scheme to rank the control actions $u \in \overline{\mathcal{U}}$ based on its performance of the min–max uncertainty reduction. Suppose $x^+ = (\overline{\mathcal{T}}^+, t^+, z^+) = f(x, u, d)$ is the state obtained by applying control u under disturbance d. For each $u \in \overline{\mathcal{U}}$ and $d \in \{0, 1\}$, let $\mathscr{L}(u, d) := \mathscr{L}(\overline{\mathcal{T}}^+)$ be the set of leaf nodes associated with the pair (u, d) and associate a number $\Psi(u) = \max\{|\mathscr{L}(u, 1)|, |\mathscr{L}(u, 0)|\}$ with each $u \in \overline{\mathcal{U}}$, which characterizes the min–max uncertainty reduction for each u. We define an order $\succ$ on the set $\overline{\mathcal{U}}$ as follows:

Given any $u_i \in \overline{\mathcal{U}}$, and $u_j \in \overline{\mathcal{U}}$ the following holds.

1) If $\Psi(u_i) > \Psi(u_j)$, then $u_i \prec u_j$.
2) If $\Psi(u_i) < \Psi(u_j)$, then $u_i \succ u_j$.
3) If $\Psi(u_i) = \Psi(u_j)$, then

$$\begin{cases} u_i \prec u_j, & \text{if } \gamma[u_i] > \gamma[u_j] \\ u_i \succ u_j, & \text{if } \gamma[u_i] < \gamma[u_j]. \end{cases}$$

Degree q approximation is then referred to as choosing the largest q control actions to generate a reduced control space $\mathcal{U}(x, q)$ for each information state x. For the suboptimal control strategy, we first prescribe an approximation degree q and compute the reduced control space $\mathcal{U}(x, q)$ for the information state x. The new value iteration is given below in (28)

$$H_k(x) = \begin{cases} \min\limits_{u_k \in \mathcal{U}(x,q)} \max\limits_{d_k \in \mathcal{D}} \{g_k(x, u_k) + H_{k+1}(f(x, u_k, d_k))\} \\ \qquad\qquad \text{if } 0 \le k \le N-2 \\ \phi(f(x, u_k, d_k)), \quad \text{if } k = N-1. \end{cases} \tag{28}$$

Simulation results of this suboptimal algorithm will be presented in Section V.

IV. Solution to NS-IIP

In this section, we develop a solution to NS-IIP that is similar to the solution to CS-IIP presented in the previous section.

For NS-IIP, with a slightly abuse of notations, we redefine the information state in NS-IIP to be $x = (\overline{\mathcal{T}}, t, z, \overline{R})$, where the first three elements have the same meaning as those in CS-IIP, and $\overline{R}(t)$ is the *ratio matrix* introduced to characterize the evolution of the uncertainty set. Let $\overline{R} = (r[i, j]) \in \mathbb{R}^{N_{\mathcal{N}} \times N_{\mathcal{N}}}$ be the *ratio matrix* associated with a decision tree, where $r[i, j]$ is the distance ratio of the length of the segment covered by the uncertainty set with respect to the total length of arc $(i, j) \in \overline{\mathcal{A}}$. We associate each state with a value defined below

$$\overline{\mu}(\overline{R}) = \sum_{(\alpha,\beta)\in\mathcal{E}} e_{\alpha,\beta} \max_{\{(i,j)\in\overline{\mathcal{A}} | v[i]=\alpha, v[j]=\beta\}} r[i, j]. \tag{29}$$

Recall that $v[i], i \in \overline{\mathcal{N}}$ is the vertex corresponding to node i and each vertex may correspond to multiple nodes in $\overline{\mathcal{N}}$, the value $\overline{\mu}(\overline{R})$ computes the length of the road network routes where the intruder may be located along based on the decision tree information, in particular $\overline{R}$. It is obvious that given the information state $x = (\overline{\mathcal{T}}, t, z, \overline{R})$, the value $\overline{\mu}(\overline{R})$ equals the value $\mu(\mathcal{B}(t))$ defined in Definition 3. With an abuse of notation, we let $\overline{\mu}(x) \triangleq \overline{\mu}(\overline{R})$, since the size of the uncertainty set given state x depends solely on $\overline{R}$.

The escape set and the isolation set are defined similarly as those in CS-IIP

Isolation set:

$$\mathcal{X}_I \triangleq \{x \in \mathcal{X} : \mathscr{L}(\overline{\mathcal{T}}) = \emptyset \text{ and } t < t_f\}$$

Escape set:

$$\mathcal{X}_E \triangleq \{x \in \mathcal{X} : \exists j \in \mathscr{L}(\overline{\mathcal{T}}), \exists i \in \overline{\mathcal{N}}, r[i, j] = 1\}. \tag{30}$$

Note that, in CS-IIP, the information state only consists of the first three elements because the uncertainty set $\mathcal{B}$ can be fully characterized by $\overline{\mathcal{T}}$ and t. However, in NS-IIP, given the tuple $(\overline{\mathcal{T}}, t, z)$, the uncertainty set cannot be directly identified. Hence, we introduce the matrix $\overline{R}$ to encode the information about the uncertainty set $\mathcal{B}$.

Moreover, different from the case in CS-IIP, the control decision in NS-IIP involves not only the sequence of UGSs to visit but also a sequence of corresponding waiting times. For simplicity, let $\xi = (u_0, \ldots, u_{N-1}) \in \prod_{k=0}^{N-1} (\mathcal{U}_k \times \mathbb{R}_+)$ be the control decision with $u_k = (u_k^1, u_k^2)$, where $u_k^1 \in \mathcal{U}_k$ is the UGS to visit at stage k and $u_k^2 \in \mathbb{R}_+$ is the corresponding waiting time. Similarly, the disturbance will no longer be a sequence of 0's or 1's, but the UGSs' measurements need to be incorporated as well. The disturbance is given by $\eta = (d_0, \ldots, d_{N-1}) \in (\mathcal{D} \times \mathbb{R}_+)^N$ where $d_k = (d_k^1, d_k^2)$, $d_k^1 \in \mathcal{D} = \{0, 1\}$ and $d_k^2 \in \mathbb{R}_+$ is the measurement taken by UGS u_k^1.

Similar to that in CS-IIP, the dynamics of the information state $x_k = (\overline{\mathcal{T}}_k, t_k, z_k, \overline{R}_k)$ is given by

$$x_{k+1} = (\overline{\mathcal{T}}_{k+1}, t_{k+1}, z_{k+1}) = f(x_k, u_k, d_k) \tag{31}$$

where the updated information state $(\overline{\mathcal{T}}_{k+1}, t_{k+1}, z_{k+1})$ is defined element-wise as follows:

Decision Tree:

$$\overline{\mathcal{T}}_{k+1} = \begin{cases} \mathcal{T}_k^o \setminus \mathcal{T}_s[u_k^1], & \text{if } d_k^1 = 0 \\ \mathcal{T}_o[u_k^1], & \text{if } d_k^1 = 1 \end{cases} \tag{32}$$

and

$$\mathcal{T}_{k+1}^o = \begin{cases} \mathcal{T}_k^o, & \text{if } d_k^1 = 0 \\ \overline{\mathcal{T}}_{k+1}, & \text{if } d_k^1 = 1 \end{cases} \tag{33}$$

with $\mathcal{T}_0^o = \overline{\mathcal{T}_0}$.

Decision Time:

$$t_{k+1} = t_k + \frac{\|S[u_k] - S[z_k]\|}{v_P} + u_k^2. \tag{34}$$

UAV Location:

$$z_{k+1} = u_k^1. \tag{35}$$

Ratio Matrix:

Let $\overline{\mathcal{T}}_{k+1} = (\overline{\mathcal{N}}_{k+1}, \overline{\mathcal{A}}_{k+1}, \overline{\mathcal{U}}_{k+1})$ be the updated decision tree. The update of $\overline{R}_k$ is intricate and requires consideration of many cases, which is detailed in the following.

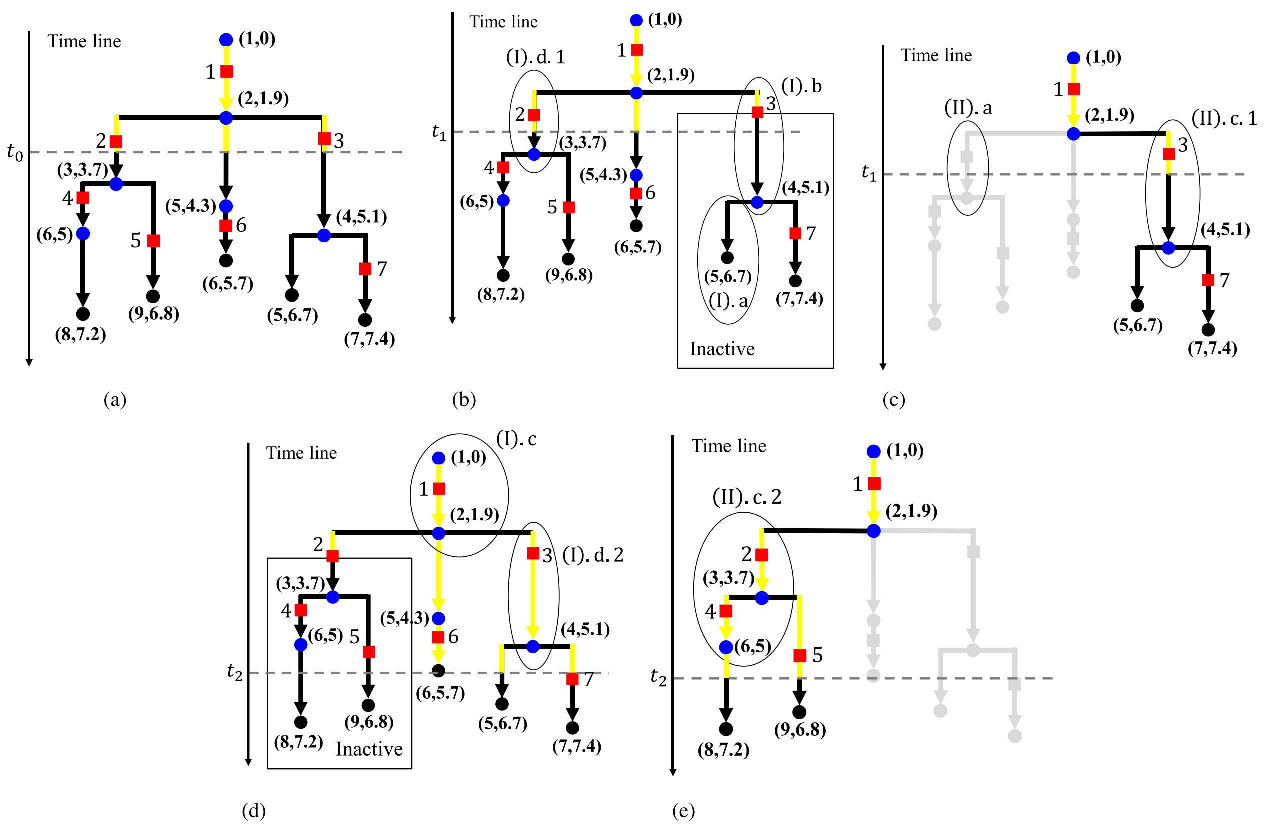


Fig. 5. Example of update of the information state. (a) Initial decision tree. (b) Decision tree with $u = 3$ and $d = 0$. (c) Decision tree with $u = 3$ and $d = 1$. (d) Decision tree with $u = 2$ and $d = 0$. (e) Decision tree with $u = 2$ and $d = 1$.

1) If $d_k^1 = 0$, i.e., the subtree is cleared.

a) If $(i, j) \notin \overline{\mathcal{A}}_{k+1}$, then the branch does not belong to current decision tree, $r_{k+1}[i, j] = 0$;

b) If $i = (u_k^1)_1$, $j = (u_k^1)_2$, i.e., the UGS u_k^1 lies along the arc (i, j), $r_{k+1}[i, j] = \rho[u_k^1]$;

c) If $(i, j) \in \overline{\mathcal{A}}_{k+1}$ and $r_k[i, j] = 1$, then the branch remains active and the ratio remains 1, $r_{k+1}[i, j] = 1$;

d) If $(i, j) \in \overline{\mathcal{A}}_{k+1}$ and $r_k[i, j] \in (0, 1)$, then this case is still divided into two subcases.

i) If $r_k[i, j] + (\nu_I(t_{k+1} - t_k)/(\lambda[j] - \lambda[i]) \leq 1$, then the updated ratio remains less than 1, i.e., the intruder does not pass node j at the end of this decision stage, then $r_{k+1}[i, j] = r_k[i, j] + (\nu_I(t_{k+1} - t_k)/(\lambda[j] - \lambda[i])$.

ii) If $r_k[i, j] + (\nu_I(t_{k+1} - t_k)/(\lambda[j] - \lambda[i]) > 1$, i.e., the intruder already passed node j, then $r_{k+1}[i, j] = 1$ since the ratio of each particular branch has maximum value of 1. Find all l such that $(j, l) \in \mathcal{A}$, set $r_{k+1}[j, l] = (\nu_I(t_{k+1} - t_k) - (\lambda[j] - \lambda[i])/(\lambda[l] - \lambda[j])$. If $r_{k+1}[j, l] > 1$ for some l, then repeat this case.

2) If $d_k^1 = 1$, i.e., the branch is confirmed. The update law of this case is similar to the previous case except that the measurements d_k^2.

a) If $(i, j) \notin \overline{\mathcal{A}}_{k+1}$, then $r_{k+1}[i, j] = 0$;

b) If $(i, j) \in \overline{\mathcal{A}}_{k+1}$ and $j \leq (u_k^1)_1$, i.e., for the branches before the one that the UGS u_k^1 located along, then the branch remains active and the ratio remains unchanged, i.e., $r_{k+1}[i, j] = r_k[i, j]$;

c) If $i = (u_k^1)_1$, $j = (u_k^1)_2$, i.e., the UGS u_k^1 lies along this particular branch, this case is also divided into two cases:

i) If $(d_k^2/(\lambda[j] - \lambda[i])) + \rho[u_k^1] \leq 1$, then the measurement d_k^2 indicates that the intruder is impossible to pass p_j, then $r_{k+1}[i, j] = (d_k^2/(\lambda[j] - \lambda[i])) + \rho[u_k^1]$;

ii) If $(d_k^2/(\lambda[j] - \lambda[i])) + \rho[u_k^1] > 1$, then the measurement indicates that the intruder may have already passed node j, then $r_{k+1}[i, j] = 1$ and $r_{k+1}[j, k] = (d_k^2 - (1 - \rho[u_k^1])(\lambda[j] - \lambda[i]))/(\lambda[l] - \lambda[j])$ for all l such that $(j, l) \in \overline{\mathcal{A}}_{k+1}$. If $r_{k+1}[j, l] > 1$ for some l, then repeat this case.

Fig. 5 shows the above update law of the uncertainty set using a simple example, where all cases described above are highlighted. Note that we just present the update of the uncertainty set without explicitly specifying the values $r[i, j]$ due to the complexity issue. The values $r[i, j]$ can be computed easily once the decision tree and the uncertainty set are known.

The running cost function $g_k(x_k, u_k, d_k)$ in NS-IIP is defined as follows:

$$g_k(x, u_k, d_k) = \begin{cases} \infty, & \text{if } x_k \in \mathcal{X}_E \\ \overline{\mu}(f(x, u_k, d_k)) - \overline{\mu}(x), & \text{otherwise.} \end{cases} \tag{36}$$

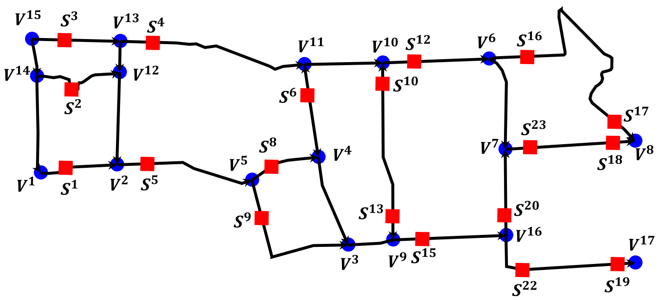
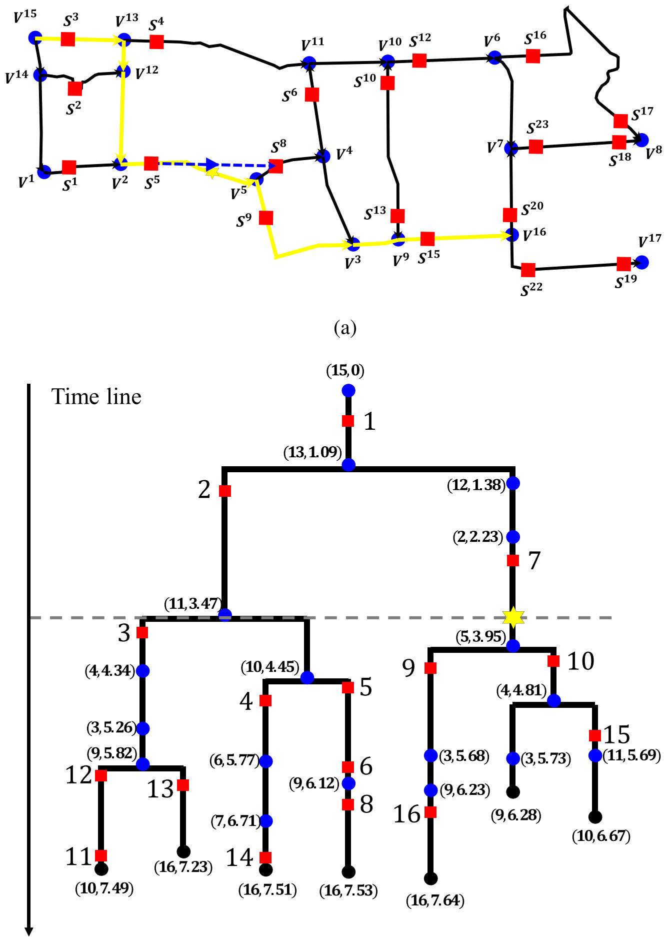


Fig. 6. Real road network at Camp Atterbury, IN.



(a)

(b)

Fig. 7. Simulation snapshot of optimal open-loop min–max solution for CS-IIP. (a) Animation of the game in the road network. (b) Animation of the game in the decision tree.

The corresponding finite horizon cost function is then given by

$$J_N(x_0, \xi, \eta) \triangleq \sum_{k=0}^{N-1} g_k(x_k, u_k). \tag{37}$$

NS-IIP can also be solved using DP. The Bellman recursion for NS-IIP is given as follows:

$$H_k(x) = \min_{u_k \in \mathcal{U}_k \times \mathbb{R}_+} \max_{d_k \in \mathcal{D} \times \mathbb{R}_+} \{g_k(x, u_k) + H_{k+1}(f(x, u_k, d_k))\}. \tag{38}$$

TABLE I

OPTIMAL OPEN-LOOP min–max CONTROL-DISTURBANCE SEQUENCE FOR CS-IIP

Stage k	0	1	2	3	4
Control u_k^*	2 (S^4)	7 (S^5)	10 (S^8)	9 (S^9)	16 (S^{15})
Disturbance d_k^*	0	1	0	1	0
Cost to go H_k	4.690	4.142	3.577	2.378	2.131

TABLE II

EXECUTION TIME AND COST COMPARISON BETWEEN OPTIMAL AND SUBOPTIMAL SOLUTIONS FOR CS-IIP

Degree q	1	2	3	4	5	Optimal
Time (seconds)	-	0.010	0.042	0.247	1.125	957.5
Cost J	∞	4.690	4.690	4.690	4.690	4.690

TABLE III

(u_k^*, d_k^*) SEQUENCE COMPARISON BETWEEN OPTIMAL AND SUBOPTIMAL SOLUTIONS FOR CS-IIP

q \ k		0	1	2	3	4
1	u_k^*	-	-	-	-	-
	d_k^*	-	-	-	-	-
2	u_k^*	-	2 (S^4)	7 (S^5)	9 (S^9)	16 (S^{15})
	d_k^*	-	0	0	1	0
3	u_k^*	-	2 (S^4)	7 (S^5)	9 (S^9)	16 (S^{15})
	d_k^*	-	0	0	1	0
4	u_k^*	-	2 (S^4)	7 (S^5)	9 (S^9)	16 (S^{15})
	d_k^*	-	0	0	1	0
5	u_k^*	2 (S^4)	7 (S^5)	10 (S^8)	9 (S^9)	16 (S^{15})
	d_k^*	0	1	0	1	0
Optimal	u_k^*	2 (S^4)	7 (S^5)	10 (S^8)	9 (S^9)	16 (S^{15})
	d_k^*	0	1	0	1	0

TABLE IV

RECEDING HORIZON CONTROL SOLUTION TO CS-IIP FOR THE PARTICULAR INTRUDE'S PATH

Stage k	0	1	2	3
Control u_k	2 (S^4)	3 (S^6)	4 (S^{12})	6 (S^{13})
Disturbance d_k	1	0	0	1
Cost to go H_k	3.980	3.432	2.130	1.045

The *saddle-point equilibrium* (ξ^*, η^*) is given as follows:

$$\begin{aligned} \xi_k^*(x) &= \arg\min_{u_k \in \mathcal{U}_k \times \mathbb{R}_+} \left\{ \max_{d_k \in \mathcal{D} \times \mathbb{R}_+} \{g_k(x, u_k) + H_{k+1}(f(x, u_k, d_k))\} \right\} \\ \eta_k^*(x) &= \arg\max_{d_k \in \mathcal{D} \times \mathbb{R}_+} \{H_{k+1}(f(x, \xi_k^*(x_k), d_k))\}. \end{aligned} \tag{39}$$

Remark 5: Note that importantly, since the control u_k and disturbance d_k in NS-IIP both involve the time component, the control space $\mathcal{U}_k \times \mathbb{R}_+$ and disturbance $\mathcal{D} \times \mathbb{R}_+$ are both continuous for each k. Due to this fact, computing the saddle-point equilibrium in NS-IIP is a combinatorial problem, which is extremely difficult, if not impossible. To numerically compute the solutions to NS-IIP, we have to discretize the continuous control space. In the simulation presented in Section V, we discretize the continuous space according to the intruder's maximum speed v_I. Similar to the CS-IIP case, a receding horizon control implementation of the solution developed in this section for a prescribed intruder's path is also tested in Section V.

TABLE V

OPTIMAL OPEN-LOOP min–max CONTROL-DISTURBANCE SEQUENCE FOR NS-IIP

Stage	k	0	1	2	3	4	5	6
Control	u_k^1	$2\ (S^4)$	$7\ (S^5)$	$2\ (S^4)$	$7\ (S^5)$	$2\ (S^4)$	$3\ (S^6)$	$5\ (S^{10})$
	u_k^τ	0	0.00975	0	0	0	0	0
Disturbance	d_k	0	0	0	0	1	0	0
Stage	k	7	8	9	10	11	12	
Control	u_k^1	$3\ (S^6)$	$5\ (S^{10})$	$4\ (S^{12})$	$5\ (S^{10})$	$3\ (S^6)$	$3\ (S^6)$	
	u_k^τ	0	0	0	0	0	0.00403	
Disturbance	d_k	0	0	0	0	0	0	

V. SIMULATION ON A REAL ROAD NETWORK

In this section, we implement the solution approach developed in this paper on a real road network at Camp Atterbury (AFRL flight test site) in Indiana (Fig. 6). The road network is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, E, \mathcal{S}, \mathcal{D})$, where $\mathcal{V} = \{V^1, V^2, \ldots, V^{17}\}$, $\mathcal{E}$ and E are shown by arrows and numbers, $\mathcal{S} = \{S^1, S^2, \ldots, S^{23}\}$ and $\mathcal{D} = \{V^8, V^{17}\}$. Initialization of the game is set to be $(S^{i_0}, \delta) = (S^3, 1.5)$, i.e., the UAV receives a positive reading of 1.5 from UGS S^3 and the chase is on. The time horizon is set to be $T_h = 7.65$ for all the simulations below. Moreover, the intruder's speed in CS-IIP (and its maximum speed in NS-IIP) is assumed to be $\nu_I = 1$ and the UAV's maximum speed is assumed to be $\nu_P = 2$.

A. Implementation of CS-IIP Solution

In this section, we first present the optimal and suboptimal min–max solution for the CS-IIP and then present the receding horizon implementation for a fixed intruder's path.

1) Optimal and Suboptimal Min–Max Solution to CS-IIP: Snapshots of animations are shown in the road network and decision tree in Fig. 7. In Fig. 7(a), the highlighted yellow route is the intruder's path corresponding to the optimal control-disturbance pair, the yellow star represents the real intruder's location on the road network and the blue triangle denotes the UAV and the blue dashed arrow indicates the trajectory of the UAV. In Fig. 7(b), the dashed horizontal line represents the time line and the yellow star represents the real intruder's location on the decision tree.

Table I presents the open-loop min–max control-disturbance sequence generated by the optimal solution algorithm. From Table I, it can be seen that the optimal min–max time to capture for the particular initial state is given by 4.690, which is less than T_h, and so the intruder will be isolated.

Note that in Table I and the other tables that follow, the control u_k (or u_k^1) is shown as i (S^j) where $i \in \mathcal{U}$ is the UGS in the decision tree and S^j is the corresponding UGS in the road network.

The suboptimal min–max strategy is also tested on the same road network with the same initial configurations. Table II shows the comparison of the execution time between different degree approximations and the optimal solution, and Table III lists the corresponding open-loop min–max control-disturbance sequences for each approximation. From Table II, we see that degree 1 approximation is not sufficient to guarantee capture, while degrees 2 to 5 approximations can guarantee capture and in fact have the same cost as the optimal solution. The computation time for the suboptimal solution is significantly less than the optimal solution, as can be observed in the table, with no loss in performance. Table III indicates that even though degrees 2 to 5 approximations yield the same cost as the optimal solution, the open-loop min–max control-disturbance sequence may not be the same as the optimal one. This is because the solution to (24) is not necessarily unique.

TABLE VI

EXECUTION TIME AND COST COMPARISON BETWEEN OPTIMAL AND SUBOPTIMAL SOLUTIONS FOR NS-IIP

Degree	q	1	2	3	4	5	Optimal
Time (seconds)		-	14.94	52.67	105.84	232.8	3173
Cost	J	∞	4.987	4.987	4.987	4.987	4.987

2) Receding Horizon Control Implementation of CS-IIP: In order to examine the performance of the receding horizon control, we prescribe an arbitrary intruder's path, which is not necessarily corresponding to the open-loop optimal/suboptimal path, and implement the receding horizon control algorithm. The intruder's path is assumed to be $[V^{15}\ V^{13}\ V^{11}\ V^{10}\ V^{9}\ V^{16}]$ which indicates the vertices the intruder will pass. The receding horizon solution to CS-IIP for this particular intruder's path is given in Table IV, which indicates that the intruder will be isolated with a cost of 3.980.

B. Implementation of NS-IIP Solution

In NS-IIP, even though we can discretize the state space and employ the suboptimal strategy to reduce computational complexity, the control space is still infinite, since the UAV's waiting time around each UGS can be arbitrary. Throughout the implementation part, we assume that the waiting time around each UGS, i.e., u_k^2 in u_k for each k, is either zero or is determined by the intruder's maximum speed, ν_I.

1) Optimal and Suboptimal Min–Max Solution to CS-IIP: We first present the optimal and suboptimal min–max solution obtained through the DP approach, which minimizes the finite horizon size of uncertainty.

Table V shows the optimal open-loop min–max control-disturbance sequence of NS-IIP. The finite horizon size of uncertainty under this particular initial state specification is given by 4.987. The performance comparison between the optimal and suboptimal open-loop min–max solutions is presented in Table VI. We observe that degrees 2 to 5 approximations perform as good as the optimal solution in terms of finite horizon cost, yet the computation time reduces significantly

TABLE VII

RECEDING HORIZON SOLUTION USING ONE UAV WITH FULL HORIZON CONTROL LAW

Stage	k	0	1	2
Control	u_k^1	$2\ (S^4)$	$3\ (S^6)$	$5\ (S^{10})$
	u_k^τ	0	0.7109	0.4039
Disturbance	d_k	1	0	0
Stage	k	3	4	5
Control	u_k^1	$3\ (S^6)$	$5\ (S^{10})$	$6\ (S^{13})$
	u_k^τ	0	0	0
Disturbance	d_k	0	1	0

with respect to the optimal one, which is consistent with the performance of the suboptimal strategy in CS-IIP.

2) Receding Horizon Control for NS-IIP: We apply the receding horizon control approach in NS-IIP trying to achieve the range isolation for particular intruder's paths. Note that in NS-IIP, the intruder's path consists of not only the UGSs that the intruder will pass but also the corresponding passing time due to the nonconstant speed assumption of the intruder's speed. Hence, in our simulation, we pick an intruder's path and randomly assign the passing time of each vertex as long as it does not violate the intruder's speed assumption. The intruder's path is prescribed as

$$\begin{bmatrix} V^3 & V^4 & V^{10} & V^{13} & V^{15} \\ 0.4166 & 1.6 & 5 & 6.3 & 7 \end{bmatrix}$$

where the first row is the order of UGSs that will be visited on the decision tree, and the second row is the corresponding visiting time. The receding horizon solution to NS-IIP for this particular intruder's path is given in Table VII, which indicates that the intruder will be isolated within the segment between UGS S^{10} and S^{13}.

VI. CONCLUSION AND FUTURE WORK

In this paper, we studied two particular IIPs, which involve using one UAV to isolate an intruder moving on a road network. An unfolding strategy is proposed to transform the road network graph into a decision tree incorporating time information. DP is employed to solve both problems. A monotonicity property inherent in the value function for CS-IIP is exploited, which helps us design an efficient discretization of the state space. A suboptimal strategy based on uncertainty reduction is proposed to alleviate the computational complexity in numerically solving DP. Simulations based on a real road network are conducted to demonstrate the effectiveness of all the proposed algorithms.

A scenario involving multiple UAVs in NS-IIP is a promising future research direction. Another future research direction lies in the investigation of other suboptimal ranking schemes. More realistic intruder and UAV dynamics can also be considered in the future research.

REFERENCES

[1] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, vol. 200. Philadelphia, PA, USA: SIAM, 1995.

[2] B. Alspach, "Searching and sweeping graphs: A brief survey," *Le Matematiche*, vol. 59, nos. 1–2, pp. 5–37, 2006.

[3] F. V. Fomin and D. M. Thilikos, "An annotated bibliography on guaranteed graph searching," *Theor. Comput. Sci.*, vol. 399, no. 3, pp. 236–245, 2008.

[4] T. D. Parsons, "Pursuit-evasion in a graph," *Theory and Applications of Graphs*. Berlin, Germany: Springer, 1978, pp. 426–441.

[5] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Appl. Math.*, vol. 8, no. 1, pp. 1–12, 1984.

[6] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, "The complexity of searching a graph," *J. ACM*, vol. 35, no. 1, pp. 18–44, 1988.

[7] R. Aleliunas, R. M. Karp, R. Lipton, L. Lovasz, and C. Rackoff, "Random walks, universal traversal sequences, and the complexity of maze problems," in *Proc. 20th Annu. Symp. Found. Comput. Sci.*, Oct. 1979, pp. 218–223.

[8] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking, "Randomized pursuit-evasion in graphs," *Combinat. Probab. Comput.*, vol. 12, no. 3, pp. 225–244, 2003.

[9] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion with local visibility," *SIAM J. Discrete Math.*, vol. 20, no. 1, pp. 26–41, 2006.

[10] V. Isler and N. Karnad, "The role of information in the cop-robber game," *Theor. Comput. Sci.*, vol. 399, no. 3, pp. 179–190, 2008.

[11] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 1451–1456.

[12] S. Pan, H. Huang, J. Ding, W. Zhang, D. M. Stipanović, and C. J. Tomlin, "Pursuit, evasion and defense in the plane," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 4167–4173.

[13] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proc. 47th IEEE Conf. Decision Control*, Dec. 2008, pp. 3706–3712.

[14] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *Int. J. Robot. Res.*, vol. 30, no. 3, pp. 335–354, 2011.

[15] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[16] K. Kalyanam, S. Darbha, P. Khargonekar, M. Pachter, and P. Chandler, "Optimal cooperative pursuit on a Manhattan grid," in *Proc. AIAA Guid., Navigat., Control Conf. (GNC)*, vol. 4633. 2013.

[17] K. Krishnamoorthy, D. Casbeer, P. Chandler, M. Pachter, and S. Darbha, "UAV search & capture of a moving ground target under delayed information," in *Proc. IEEE 51st Annu. Conf. Decision Control (CDC)*, Dec. 2012, pp. 3092–3097.

[18] K. Krishnamoorthy, S. Darbha, P. P. Khargonekar, D. Casbeer, P. Chandler, and M. Pachter, "Optimal minimax pursuit evasion on a Manhattan grid," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2013, pp. 3421–3428.

[19] K. Krishnamoorthy, D. Casbeer, and M. Pachter, "Pursuit on a graph using partial information," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 4269–4275.

[20] K. Krishnamoorthy, D. Casbeer, and M. Pachter, "Minimum time UAV pursuit of a moving ground target using partial information," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Jun. 2015, pp. 204–208.

[21] S. H. Han and W. M. McEneaney, "Solution of an optimal sensing and interception problem using idempotent methods," in *Recent Advances in Research on Unmanned Aerial Vehicles*. Berlin, Germany: Springer, 2013, pp. 47–67.

[22] J. P. Hespanha, *An Introductory Course in Noncooperative Game Theory*, 2011. [Online]. Available: http://www.ece.ucsb.edu/~hespanha/published/

Hua Chen received the B.E. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2012. He is currently pursuing the Ph.D. degree in electrical and computer engineering with The Ohio State University, Columbus, OH, USA.

His current research interests include control theory and game theory with applications in robotics, automation, and power systems.

Krishnamoorthy Kalyanam (SM'15) received the B.Tech. degree from IIT Madras, Chennai, India, in 2000, and the M.S. and Ph.D. degrees from the University of California at Los Angeles, Los Angeles, CA, USA, in 2003 and 2005, respectively, all in mechanical engineering.

He is currently a Research Scientist with InfoSciTex Corporation, Dayton, OH, USA, and an On-Site Contractor with the Autonomous Control Branch, U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA. His current research interests include autonomous systems, optimization, and cooperative control of unmanned air vehicles.

Wei Zhang (S'07–M'10) received the B.S. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 2003, and the M.S. degree in statistics and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2009.

He was a Post-Doctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, from 2010 to 2011. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA. His current research interests include control and game theory with applications in power systems, robotics, and intelligent transportations.

David Casbeer received the B.S. and Ph.D. degrees from Brigham Young University, Provo, UT, USA, in 2003 and 2009, respectively.

He is currently the Team Lead for the UAV Cooperative and Intelligent Control Team with the Control Science Center of Excellence, Aerospace Systems Directorate, U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA. In this capacity, he leads a team of researchers investigating the cooperative control of autonomous unmanned air vehicles (UAVs) with a particular emphasis on high-level decision making and planning under uncertainty. His current research interests include global positioning system (GPS)-denied cooperative navigation, UAV self-protection, and ground intruder tracking by UAV/unattended ground sensor (UGS).

Dr. Casbeer serves as the Chair-Elect of the AIAA Intelligent Systems Technical Committee and is a Senior Editor of the *Journal of Intelligent and Robotic Systems*.