IMA Journal of Mathematical Control and Information Page 1 of 16
doi:10.1093/imamci/dnnxxx

Pursuit of a Moving Target with Bounded Speed on a Directed Acyclic Graph Under Partial Information

KRISHNAMOORTHY KALYANAM
InfoSciTex Corporation, a DCS Company,
4027 Colonel Glenn Highway, Suite 210, Dayton, OH 45431, USA.

DAVID CASBEER
Autonomous Control Branch, Air Force Research Laboratory,
2210 8th Street, Wright-Patterson A.F.B., OH 45433, USA.

AND

MEIR PACHTER
Electrical & Computer Engineering Department, Air Force Institute of Technology,
2950 Hobson Way, Wright-Patterson A.F.B., OH 45433, USA.

[Received on 18 February 2005]

The optimal control of a "blind" airborne pursuer searching for an evader moving on a road network with positive bounded speed toward a set of goal vertices is considered. To aid the pursuer and provide feedback information, certain roads in the network have been instrumented with Unattended Ground Sensors (UGSs) that detect evader motion. When the pursuer arrives at an instrumented node, the UGS therein informs the pursuer if and when the evader visited the node. The pursuer is aware of the bounds on the evader's speed. Moreover, the embedded graph with the UGSs as vertices and connecting roads as edges is restricted to be a Directed Acyclic Graph (DAG). At time 0, the evader's entry into the road network is registered at UGS 1, the entry node. The pursuer also arrives at the entry node after some delay $d > 0$ and is thus informed about the presence of the intruder/evader in the network, whereupon the chase is on. Capture entails the pursuer and evader being co-located at an UGS location. However, if the evader reaches an exit node of the graph without being captured, it is deemed to have escaped. We compute the maximum initial delay d and the ensuing pursuit policy, for which capture is guaranteed.

Keywords: Pursuit-Evasion, Partial Information, Dynamic Game.

1. Introduction

We are concerned with capturing a ground target moving on a road network. The operational scenario is as follows. The access road network to a restricted (protected) zone, patrolled by an Unmanned Air Vehicle (UAV), is instrumented with Unattended Ground Sensors (UGSs), placed at critical locations. As the target, referred to hereafter as the "evader", passes by an UGS, the UGS is triggered. A triggered UGS turns, say, from *green* to *red* and records the evader's time of passage. The UGSs are placed on certain edges of the graph. We assume that the layout of the road network and the placement of the UGSs is known to the pursuer/ UAV. The evader's speed is positive and bounded, with the bounds known to the pursuer. When the pursuer arrives at an UGS location, the information stored by the UGS is uploaded to the pursuer, namely, the green/red status of the UGS and, if the UGS is red, the time elapsed (delay) since the evader's passage. The evader can be captured in one of two ways: either the evader and pursuer synchronously arrive at an UGS location, or the pursuer is already loitering/waiting at an UGS location when the evader arrives there. In both cases, the UGS is triggered, instantaneously informs the pursuer, and the evader is captured. The decision problem for the pursuer is to select which UGS to visit next, including possibly staying at the current UGS location (and if so, for how long?) awaiting the arrival of the evader. The decisions are made by the pursuer at discrete time instants, immediately after arriving at and interrogating an UGS. The pursuer, on the other hand, is not restricted to follow the road network. In addition, the pursuer can also wait/loiter at an UGS location for an arbitrary amount of time.

The problem at hand is motivated by a base defense/ asset protection scenario, where a UAV performs the role of a pursuer aided by the Unattended Ground Sensors. The UAV is typically equipped with a gimballed camera, that is used to "capture" visual imagery of the ground target (evader). This imagery is passed on to a human operator, who classifies the target as a potential threat or otherwise and initiates further action. We are dealing with a pursuit evasion problem, with partial information, on a graph. This is a variant of the classic discrete pursuit-evasion game introduced by Parsons *et al.* (1978). Discrete pursuit-evasion games are also known as graph search problems, wherein a pursuer and evader take turns moving from vertex to vertex on a graph. Many variations to this problem have been investigated over the years (see Fomin *et al.* (2008); Chung *et al.* (2011) for a survey). Typical formulations of discrete pursuit-evasion games deal with evaders that are either invisible except when captured or visible to the pursuer at all times (see Cops and Robbers game in Chung *et al.* (2011)). In practical scenarios, a more plausible assumption is that the pursuer has limited sensing capability. Optimal strategies on a square grid, where the pursuer has line of sight visibility is provided in Sugihara *et al.* (1989); Dumitrescu *et al.* (2010). The role of information available to the players, when the pursuer has limited or no visibility of the evader is investigated in Isler *et al.* (2008). Networked robotic systems, where distributed sensors on the ground provide sensing-at-a-distance capability have been considered in Vieira *et al.* (2009); Sinopoli *et al.* (2003). However, the information model considered therein is global i.e., the sensor data from all sensors are instantaneously relayed over a global communication network to all pursuers.

A formulation related to our work, wherein observations of the evader are made by witnesses throughout the graph, is reported in Clarke *et al.* (2009). However, the witness reports are immediately made available to the pursuer, whereas the observation in our work is only available when the pursuer visits an UGS and hence the evader location information is delayed. In Dzyubenko *et al.* (1972), a solution is proposed for the case where the evader's location information is available to the pursuer after a known constant delay. For the problem at hand, the delay seen by the pursuer is a function of his past actions and therefore, not a constant. In Demirbas *et al.* (2003), the authors impose the additional constraint that the sensor data is local; however they allow for the sensors close to the evader to communicate with each other and maintain a "tracking" tree that is rooted at the evader. We note that the distributed sensing and local information pattern sets our work apart from the rest of the literature on pursuit evasion on graphs. Indeed, we are dealing with a deterministic pursuit-evasion game on a directed acyclic finite graph where the evader's strategy is open-loop control and the pursuer has partial information. Such a game was previously considered in Krishnamoorthy *et al.* (2013a,b), where the highly structured graph considered therein, was a Manhattan grid. In the current work, we restrict the road network on which the evader is moving to be a Directed Acyclic Graph (DAG). In Krishnamoorthy *et al.* (2016), we considered a simpler scenario wherein the evader's speed was a known constant. In this work[1], we extend that result by relaxing the known constant speed assumption. Instead, we assume that the evader's speed is bounded with the (positive) bounds known to the pursuer. In Chen *et al.* (2014),

[1]A preliminary version was presented at the IMA Conference on Mathematics of Robotics - see Krishnamoorthy *et al.* (2015b)

the authors have also relaxed the assumption of a directed acyclic graph and provided a sub-optimal solution for the corresponding minimum time capture problem.

Due to the pursuer's information pattern, which is restricted to partial observations of the physical state of the dynamic game, we are running into the difficulties brought about by the *dual control* effect (see Başar *et al.* (2001)), where the current information state determines the pursuer's optimal control while at the same time the information that will become available to the pursuer will be in part determined by his control action. In the remainder of the paper, we set up the model and assumptions in Section 2. This is followed by the performance metric of interest and the corresponding optimization problem one needs to solve in Section 3. We provide a series of Lemmas in Section 3.1 that bring out key properties of the optimal policy; which in turn helps establish an ordered recursive solution methodology in Section 4. In Section 4.1, we illustrate the methodology with a numerical example followed by some concluding remarks in Section 5.

2. Model and Assumptions

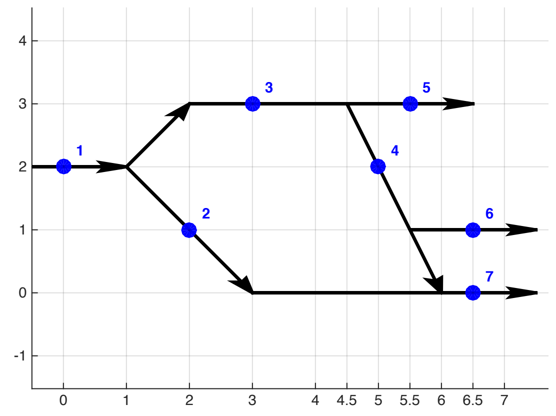

FIG. 1. Example DAG Road Network with UGSs

Fig. 1 depicts an illustrative road network on which the evader or ground target travels. The roads are shown in black (arrows indicate direction of travel) and the numbered UGSs are blue circles. The pursuer makes a decision when it gains new information at an UGS. For this reason, we focus on the embedded graph, $G(\mathscr{U},E)$, where the m UGSs are the vertices, i.e., $\mathscr{U} = \{1,\dots,m\}$ and the connecting roads constitute the edges of the graph. We make the simplifying assumption that G is a Directed Acyclic Graph (DAG). We note that for the application at hand, the road network is restricted and as such, road blocks can be envisioned that prevent the evader from making U-turns or going around in cycles. Moreover, a DAG assumption is consistent with the notion of a goal oriented evader, who is attempting to reach the protected area whilst evading the pursuing UAV. Let $\mathscr{G} \subset \mathscr{U}$ indicate the set of exit/goal nodes that the evader is heading towards. For the example problem, $m = 7$ and $\mathscr{G} = \{5,6,7\}$ - see Fig. 1.

We assume without loss of generality that the entry node into the DAG is UGS 1 and that $1 \notin \mathscr{G}$. Let there be n $(\geqslant 1)$ possible evader paths emanating from node 1 and terminating at an exit node. If node j can be reached from node i by traveling along path k on the network, we let $d_e(i,j;k)$ indicate the distance between the two nodes. Else, we set $d_e(i,j;k) = \infty$. We simply use $d_e(i,j)$; when the path index is self-evident or if every path between i and j is of the same length. The pursuer/UAV is not restricted to travel along the road network and, in general, it may fly direct between any two nodes. We only stipulate that the pursuer's travel distance from node i to node j, $d_p(i,j)$ satisfies the triangle inequality:

$$d_p(i,j) \leqslant d_p(i,s) + d_p(s,j), \tag{2.1}$$

for any $i,j,s \in \mathscr{U}$ and $d_p(j,j) = 0$, $\forall j \in \mathscr{U}$. We assume that the evader travels with positive bounded speed, with the bounds known to the pursuer. Indeed, let v_L and v_U denote the lower and upper bound respectively on the evader's speed, where $0 < v_L \leqslant v_U < \infty$. We assume, without loss of generality, that the pursuer travels at unit speed. We also assume that the pursuer's travel time between any two nodes is strictly less than the evader's travel time between the two, i.e.,

$$d_p(i,j) < \frac{d_e(i,j;k)}{v_U}, \ \forall i,j \in \mathscr{U}, \ \forall k \in \{1,\dots,n\}. \tag{2.2}$$

Let there be n $(\geqslant 1)$ possible evader paths emanating from UGS 1 and terminating at an exit node.

Let $\mathbb{P}_j$, $j = 1,\dots,m$, be the set of paths that go through node j i.e., $\mathbb{P}_j = \{k : d_e(1,j,k) < \infty, \ k = 1,\dots,n\}$. We define the evader state information available to the purser to be $\mathscr{I} = (\mathscr{P};(\underline{\mathrm{n}},\underline{\mathrm{t}}))$. It has two components: the evader path information, $\mathscr{P} \subseteq \mathbb{P}_1 = \{1,\dots,n\}$ and the last known evader (UGS) location $\underline{\mathrm{n}}$ and time of visit, $\underline{\mathrm{t}}$. Let the evader enter the network via UGS 1 at time 0. Since the pursuer also arrives at node 1 after some delay $t_0 > 0$, the initial information available to the pursuer is given by $\mathscr{I} = (\mathscr{P}_0;(1,0))$. The path information, $\mathscr{P}_0 = \{1,\dots,n\}$, indicates that the evader could have taken any one of the n paths emanating from 1. The tacitly assumed information pattern is such that the evader has no situational awareness, which is equivalent to saying that the evader decides on his "strategy", namely, what path it will take and a speed profile, at time 0. The assumption here is that the evader has prior knowledge of the road network but is oblivious to the presence of the UAV. This critical assumption results in a significant simplification of the underlying coupled estimation and control problem. So, in the absence of any feedback information, the evader essentially operates in open loop.

2.1 *Evolution of System State*

Let the pursuer position at decision time t be specified by the UGS index, $p \in \{1,\dots,m\}$. The decision variable, $u \in \{1,\dots,m\}$ indicates the UGS location that the pursuer should visit next. Even though the pursuer and evader motion evolve in continuous time, decisions are made (by the pursuer only) at discrete time steps. The pursuer makes a decision immediately after reaching the UGS location p at time t and obtaining the measurement y therein: $y = -1$ for "green", or $y = \tilde{t} \geqslant 0$ for "red", where $\tilde{t}$ indicates the evader's time of arrival at p. Suppose the evader state information available to the pursuer at time t is $\mathscr{I} = (\mathscr{P};(\underline{\mathrm{n}},\underline{\mathrm{t}}))$, where $\mathscr{P}$ is the path information and $(\underline{\mathrm{n}},\underline{\mathrm{t}})$ is the most recent evader position (UGS location) and time of visit known to the pursuer. The control action u is dependent on the current time, pursuer position and most recent information: $u = \mathscr{F}(t,p,\mathscr{I})$, where the mapping $\mathscr{F}$ is to be determined by an optimality principle - see (3.5) in the sequel. We shall refer to the tuple $(t,p,\mathscr{P},\underline{\mathrm{n}},\underline{\mathrm{t}})$ as the system state. Hereafter, unless otherwise specified, let $(\underline{\mathrm{n}},\underline{\mathrm{t}}) = (1,0)$, i.e., the last known intruder location/time is the entry node at time 0. The pursuer's position and decision time evolve according to:

$$\begin{aligned} p^+ &= u, \\ t^+ &= \begin{cases} t + d_p(p,u), \text{ if } u \neq p, \\ \frac{1}{v_L} \min_{k \in \mathbb{P}_p \cap \mathscr{P}} d_e(1,p;k), \text{ otherwise.} \end{cases} \end{aligned} \tag{2.3}$$

So, if the pursuer decides to stay put at the current location, the next decision epoch is the latest possible time at which new (path) information must be available at the current UGS p. In (2.3), $\mathbb{P}_p \cap \mathscr{P}$ indicates the set of all feasible intruder paths that pass through node p and hence, the time shown indicates the latest possible evader arrival time at p via path $\bar{k} = \arg\min_{k \in \mathbb{P}_p \cap \mathscr{P}} d_e(1,p;k)$. Indeed, the pursuer need stay at the current location only if $\mathbb{P}_p \cap \mathscr{P} \neq \emptyset$ i.e., there is a likelihood of capturing the evader there. While waiting, one of two things will happen: either the evader will show up, leading to capture or the UGS will remain green long enough for the pursuer to determine that the evader has not taken the path $\bar{k}$. The pursuer need never wait at (or revisit) an UGS location after receiving a "red" measurement from it - since the evader never revisits any UGS location (DAG assumption) and so the pursuer gains no new information by doing so.

At the next decision epoch t^+, the information set is updated for the two possible observations as follows:
Red ($y^+ = \tilde{t} \geqslant 0$): This implies that the evader was at node u at time $\tilde{t}$. If this information is more recent, i.e., $\tilde{t} > 0$, the path information at decision epoch t^+ is updated to:

$$\mathscr{P}^+(u,\tilde{t}) = \left\{ k : k \in \mathbb{P}_u \cap \mathscr{P},\ \frac{d_e(1,u;k)}{v_U} \leqslant \tilde{t} \leqslant \frac{d_e(1,u;k)}{v_L} \right\}. \tag{2.4}$$

So we only retain those paths that the evader can take to arrive at u at time $\tilde{t}$. Accordingly, the information state is updated to:

$$\mathscr{I}^+(u,\tilde{t}) = \begin{cases} (\mathscr{P}^+(u,\tilde{t}); (u,\tilde{t})), \text{ if } \tilde{t} > 0, \\ \mathscr{I}, \text{ otherwise.} \end{cases} \tag{2.5}$$

Green ($y^+ = -1$): This implies that the evader has not visited u thus far. Therefore, the path information update is given by:

$$\mathscr{P}^+(u,-1) = \mathscr{P} \backslash Q,\ Q = \left\{ k : k \in \mathbb{P}_u,\ \frac{d_e(1,u;k)}{v_L} \leqslant t^+ \right\}. \tag{2.6}$$

So we remove all the paths that the evader can take to arrive at u no later than t^+. Accordingly, the information state is updated to:

$$\mathscr{I}^+(u,-1) = \begin{cases} (\mathscr{P}^+(u,-1); (1,0)), \text{ if } Q \neq \emptyset, \\ \mathscr{I}, \text{ otherwise.} \end{cases} \tag{2.7}$$

3. Optimization Problem

The evader passes by UGS 1 at time 0. The pursuer arrives for the 1st time at UGS 1 at time $t_0 > 0$ and is tasked with capturing the evader. From Fig. 1, we see that capture is likely possible for small t_0; due to the pursuer's speed advantage (2.2). On the other hand, if t_0 is large, the evader will likely escape, no matter what the pursuer does. We are interested in computing the maximum initial delay t_0 for which capture can be guaranteed. This is valuable information in an operational scenario, for the following reason. As noted earlier, the road network leads to a protected area, that is being guarded against (ground) intrusions by security forces and the pursuer is a UAV. In this case, it would be advantageous to know what is the maximum delay for which a capture guarantee exists. If the actual initial delay seen by the UAV exceeds the maximum, a human operator can be alerted and additional resources allocated to intercept the threat. On the other hand, if the actual delay encountered is no greater than the maximum, then the UAV autonomously pursues the evader, isolates it and transmits the captured image to the human operator for further action.

Let $\mathscr{D}(1|\mathscr{I}_0) > 0$ be the *latest* time that the pursuer can arrive at UGS 1 and still capture the evader with the information, $\mathscr{I}_0 = (\mathscr{P}_0;(1,0))$. Since the evader arrives at node 1 at time 0, $\mathscr{D}(1|\mathscr{I}_0)$ is also the maximal initial delay with a capture guarantee. In a similar fashion, for any UGS, $j = 1,\ldots,m$, we define $\mathscr{D}(j|\mathscr{I})$ to be the latest time the pursuer can arrive at node j and guarantee capture, armed with the information $\mathscr{I}$. If the pursuer arrives at node j at time $t > 0$ and $t \leqslant \mathscr{D}(j|\mathscr{I})$, let $\mu(j|\mathscr{I}) \in \{1,\ldots,m\}$ be the corresponding UGS index to which the pursuer should head towards next, to enable capture.

To mathematically define $\mathscr{D}(1|\mathscr{I}_0)$, we need some additional tools. Let the evader's open loop policy $\pi_e = (\kappa;\mathbf{v}_\kappa)$, where $\kappa \in \mathscr{P}_0$ is the path it chooses to travel on and $\mathbf{v}_\kappa = (v_\kappa(1), v_\kappa(2),\cdots,v_\kappa(m_\kappa - 1))$ be his speed profile. In particular, $v_\kappa(i) \in [v_L, v_U]$ is his average speed between the i^{th} and $(i+1)^{th}$ nodes, and m_κ is the number of nodes, along path κ. Note that the actual evader speed between two nodes is irrelevant since the pursuer can only infer the average speed from the time stamped information available at the nodes. Let the pursuer's feedback policy, $\pi_p = (u_0, u_1, u_2,\cdots,u_{\bar{m}})$ entail a finite sequence of UGSs it visits, where the corresponding decision epochs are given by $t_0, t_1,\cdots,t_{\bar{m}-1}$. The control action: $u_i(\mathscr{I}(t_i))$, $i = 1,\cdots,\bar{m}$, is a function of the information available with the pursuer. Since we are only interested in policies with a capture guarantee, it is necessary that capture occurs at $u_{\bar{m}}$. This implies that the pursuer must know, at decision epoch $t_{\bar{m}-1}$, that the evader is heading towards $u_{\bar{m}}$, so as to be able to capture it there. In other words, the (path) information state at decision epoch $t_{\bar{m}-1}$ must be such that: $\mathscr{P}(t_{\bar{m}-1}) \subseteq \mathbb{P}_{u_{\bar{m}}}$. Let $h(u_{\bar{m}}, \mathscr{I}(t_{\bar{m}-1}))$ indicate the time at which capture occurs at $u_{\bar{m}}$; with the pursuer's information state prior to capture being $\mathscr{I}(t_{\bar{m}-1})$. We have the latest time that the pursuer must leave UGS 1 so as to enable capture given by:

$$J(\pi_p,\pi_e) = h(u_{\bar{m}}, \mathscr{I}(t_{\bar{m}-1})) - \sum_{i=0}^{\bar{m}-1} d_p(u_i,u_{i+1}), \tag{3.1}$$

where we have subtracted the total travel time from the time of arrival at $u_{\bar{m}}$ to compute the time of departure at node 1. Note that the pursuer state and information update equations from one decision epoch to the next are given by (2.3), (2.4) and (2.6). Furthermore, the observations: $y_i(u_i,\pi_e)$, $i = 1,\cdots,\bar{m}$, made by the pursuer are a function of both the pursuer and evader strategies. So, we have the performance metric of interest given by:

$$\mathscr{D}(1|\mathscr{I}_0) = \max_{\pi_p \in \Pi_p(1,\mathscr{I}_0)} \min_{\pi_e} J(\pi_p,\pi_e), \tag{3.2}$$

where $\Pi_p(1,\mathscr{I}_0)$ is the set of all pursuer policies with a capture guarantee, starting from UGS 1 with information $\mathscr{I}_0$. Note that $\Pi_p(1,\mathscr{I}_0)$ is non-empty since it includes the trivial policy wherein the pursuer captures the evader at node 1 itself at time 0. The resulting initial delay is of course 0. Hence, we have the trivial lower bound: $\mathscr{D}(1|\mathscr{I}_0) \geqslant 0$. From (3.2), it immediately follows that:

$$\begin{aligned}
\mathscr{D}(1|\mathscr{I}_0) &= \max_{\pi_p \in \Pi_p(1,\mathscr{I}_0)} \min_{\pi_e} \left[h(u_{\bar{m}}, \mathscr{I}(t_{\bar{m}-1})) - \sum_{i=0}^{\bar{m}-1} d_p(u_i, u_{i+1}) \right] \\
&= \max_{u_1} \left\{ -d_p(1,u_1) + \max_{u_2,u_3,\cdots} \min_{\pi_e} \left[h(u_{\bar{m}}, \mathscr{I}(t_{\bar{m}-1})) - \sum_{i=1}^{\bar{m}-1} d_p(u_i, u_{i+1}) \right] \right\} \qquad (3.3) \\
&= \max_{u_1} \left\{ -d_p(1,u_1) + \min_{y_1} \max_{\pi \in \Pi_p(u_1, \mathscr{I}^+(u_1,y))} \min_{\pi'_e} \left[h(u_{\bar{m}}, \mathscr{I}(t_{\bar{m}-1})) - \sum_{i=1}^{\bar{m}-1} d_p(u_i, u_{i+1}) \right] \right\} \\
\Rightarrow \mathscr{D}(1|\mathscr{I}_0) &= \max_{u \in \mathscr{U}} \left\{ -d_p(1,u) + \min_{y \in Y(u,\mathscr{I}_0)} \mathscr{D}(u|\mathscr{I}^+(u,y)) \right\}, \qquad (3.4)
\end{aligned}$$

where $Y(u_1, \mathscr{I}_0)$ is the set of all possible observations that the pursuer is likely to encounter at node u_1, given that the pursuer chooses to go to u_1 from node 1. In (3.3), we replace the maximization over the policy π_p with incremental optimization via selection of the optimal control variables: $u_i; i = 1, \ldots, \bar{m}$. For the evader, choosing an optimal policy π_e is equivalent to sequentially minimizing $J(.,.)$ over the set of likely observations seen at UGS locations $u_i; i = 1, \ldots, \bar{m}$. Furthermore, the first observation, $y_1 \in Y(u_1, \mathscr{I}_0)$ is not dependent on future pursuer actions, u_2, u_3 and so on. In (3.4), $\mathscr{I}^+(u,y)$ is defined according to (2.4) and (2.6) for the red and green UGS observations respectively. By definition, $\mathscr{D}(u|\mathscr{I}^+(u,y))$ is the latest time at which, armed with the new information $\mathscr{I}^+(u,y)$, the pursuer can leave u and still guarantee capture of the evader. Hence, we have a recursive equation (3.4) for computing the performance metric of interest.

3.1 *Max-Min Optimization*

Generalizing the recursive equation (3.4), we can say the following. If the pursuer is at node j with information $\mathscr{I}$, the latest time it can leave j and still guarantee capture is given by:

$$\mathscr{D}(j|\mathscr{I}) = \max_{u \in \mathscr{U}} \left\{ -d_p(j,u) + \min_{y \in Y(u,\mathscr{I})} \mathscr{D}(u|\mathscr{I}^+(u,y)) \right\}, \qquad (3.5)$$

where $Y(u, \mathscr{I})$ is the set of all possible observations that the pursuer is likely to encounter at node u; given that his information at node j was $\mathscr{I}$. As before, the information update $\mathscr{I}^+(u,y)$ is given by (2.4) and (2.6) for the red and green UGS observations respectively at u. We shall refer to (3.5) simply as the Recursive Equation (RE). We note that, as it stands, RE is not amenable to Dynamic Programming. So, we provide a series of Lemmas that bring out key properties of the optimal control policy, eventually leading to a tractable solution method.

The following result provides a lower bound to the performance metric.

LEMMA 3.1 If the information state $\mathscr{I} = (\mathscr{P}; (1,0))$ satisfies $\mathscr{P} \subseteq \mathbb{P}_u$ for some $u \in \mathscr{U}$, then:

$$\mathscr{D}(u|\mathscr{I}) \geqslant \frac{1}{v_U} \min_{k \in \mathscr{P}} d_e(1,u;k). \qquad (3.6)$$

Proof. The pursuer knows that every possible path that the evader can take goes through u. So, capture is guaranteed at u, as long as the pursuer gets to u at the earliest possible time that the evader can get there, $h(u,\mathscr{I})$. Given that the evader was last seen at node 1 at time 0, we have:

$$h(u,\mathscr{I}) = \frac{1}{v_U}\min_{k\in\mathscr{P}} d_e(1,u;k).$$

$\square$

In particular, since $\mathscr{P}_0 = \mathbb{P}_1$, we have (as noted earlier) the trivial lower bound, $\mathscr{D}(1|\mathscr{I}_0) \geqslant 0$.

LEMMA 3.2 Suppose the optimal control and the optimal (minimizing) observation to RE (3.5) are given by u^* and y^* respectively. Further suppose that: $y^* > 0$ i.e., the pursuer sees a more recent sighting of the evader at u^*. Then, it is optimal for the evader to have traveled at top speed v_U between the nodes 1 and u^*.

Proof. For $y^* > 0$, the information state at u^* is given by:

$$\mathscr{I}^+(u^*,y^*) = (\mathscr{P}^+(u^*,y^*);(u^*,y^*)), \tag{3.7}$$

where (2.4) gives the path information update for a more recent red UGS observation. The recursion (3.5) is therefore given by:

$$\mathscr{D}(j|\mathscr{I}) \quad = \quad -d_p(j,u^*) + \mathscr{D}(u^*|\mathscr{I}^+(u^*,y^*)). \tag{3.8}$$

Let us investigate $\mathscr{D}(u^*|\mathscr{I}^+(u^*,y^*))$. This is the latest time the pursuer can leave u^* and still guarantee capture, with the knowledge that the evader passed through node u^*. This is somewhat equivalent to the initial condition where the evader arrives at node u^* at time 0. Indeed, let $\mathscr{D}(u^*|\bar{\mathscr{I}})$, $\bar{\mathscr{I}} = (\mathscr{P}^+(u^*,y^*);(u^*,0))$ be the corresponding (latest) time that the pursuer can leave u^* and still guarantee capture. By definition,

$$\mathscr{D}(u^*|\mathscr{I}^+(u^*,y^*)) \quad = \quad \mathscr{D}(u^*|\bar{\mathscr{I}}) + y^*, \tag{3.9}$$

where, the optimal (minimizing) evader arrival time at u^* is given by:

$$\begin{aligned} y^* &= \min_{v\in[v_L,v_U]} \min_{k\in\mathbb{P}_{u^*}\cap\mathscr{P}} \frac{1}{v} d_e(1,u^*;k) \\ &= \frac{1}{v_U} d_e(1,u^*;\bar{k}). \end{aligned} \tag{3.10}$$

Here, $\bar{k}$ is the shortest path (among all feasible paths) between 1 and u^*. In other words, the evader will arrive at u^* as early as possible; so as to maximize the delay seen by the pursuer at u^*. In light of (3.10), we also have: $\mathscr{P}^+(u^*,y^*) = \{k : k \in \mathscr{P}\cap\mathbb{P}_{u^*}, d_e(1,u^*;k) = d_e(1,u^*;\bar{k})\}$. $\square$

LEMMA 3.3 The optimal control, u^*, to RE (3.5) satisfies: $Y(u^*,\mathscr{I}) \neq \{-1\}$.

Proof. Suppose $Y(u,\mathscr{I}) = \{-1\}$, i.e., the only possible observation at u is a green UGS. This implies that none of the feasible paths go through u. It follows from the green UGS update (2.7), that the pursuer does not gain any information by visiting u. Rather, it will incur an additional reward of $-d_p(j,u)$. From the triangle inequality (2.1), it follows that the pursuer is better off skipping u. $\square$

LEMMA 3.4 The optimal control, u^*, to RE (3.5) satisfies:

$$\mathscr{D}(u^*|\mathscr{I}^+(u^*,-1)) \geqslant \frac{1}{v_L} \min_{k\in\mathscr{P}\cap\mathbb{P}_{u^*}} d_e(1,u^*;k).$$

Proof. Suppose for control u,

$$\mathscr{D}(u|\mathscr{I}^{+}(u,-1)) < \frac{1}{v_L} \min_{k\in\mathscr{P}\cap\mathbb{P}_u} d_e(1,u;k).$$

So, to guarantee capture, the pursuer must leave u before the earliest time that a path can be eliminated from the feasible set. This implies that: $\mathscr{P}^{+}(u,-1) = \mathscr{P}$ and therefore, it follows from the green UGS update (2.7) that $\mathscr{I}^{+}(u,-1) = \mathscr{I}$. As before, it follows that the pursuer does not gain any information by visiting u. Rather than incur an additional reward of $-d_p(j,u)$, the pursuer is better off skipping u. □

It is important to understand the tradeoff between immediate capture and further reduction in (path) uncertainty embedded in RE (3.5). The pursuer can terminate the search immediately by capturing the evader at a node through which every possible evader path must go through. Otherwise, it can reduce the path uncertainty further in the hope of capturing the evader later and thereby realizing a higher performance metric. The latter option, however, comes with the additional (negative) reward of travel time incurred therein. Indeed, if the pursuer keeps choosing the latter option, there will come a time at which the evader's path is known. At this stage, the RE simplifies considerably as shown below.

LEMMA 3.5 If the evader's path is known to be $\{k\}$ and the information: $\mathscr{I} = (\{k\};(1,0))$,

$$\mathscr{D}(j|\mathscr{I}) \quad = \quad \frac{1}{v_U} d_e(1,x_k) - d_p(j,x_k),$$

where, x_k is the exit node along path k.

Proof. Since the evader's path k is known, the pursuer can capture the evader, in one move, at any node along that path. So, we have:

$$\mathscr{D}(j|\{k\}) \quad = \quad \max_{u;k\in\mathbb{P}_u} \left[\frac{1}{v_U} d_e(1,u) - d_p(j,u)\right].$$

For any intermediate node $u \neq x_k$ along path k, we have:

$$\begin{aligned} d_e(1,x_k) &= d_e(1,u;k) + d_e(u,x_k;k) \\ &\geqslant d_e(1,u;k) + v_U d_p(u,x_k) \\ &\geqslant d_e(1,u;k) + v_U \left(d_p(j,x_k) - d_p(j,u)\right) \\ \Rightarrow \mathscr{D}(j|\mathscr{I}) &= \frac{1}{v_U} d_e(1,x_k) - d_p(j,x_k). \end{aligned} \tag{3.11}$$

The first and second equality above follow from the speed advantage (2.2) and triangle inequality (2.1) respectively. In essence, the pursuer leaves node j just in time to be able to reach the exit node x_k at the earliest possible evader arrival time. We shall refer to (3.11) as the Boundary Condition (BC). □

In light of the triangle inequality, we have shown that Lemmas 3.3 and 3.4 exclude some control actions as being sub-optimal. Indeed, we are now in a position to formally define the optimal restriction to the search space. Let the set:

$$\mathscr{B}(\mathscr{I}) = \left\{ u : \mathscr{P}\cap\mathbb{P}_u \neq \emptyset, \mathscr{D}(u|\mathscr{I}^{+}(u,y)) \geqslant \frac{1}{v(y)} \min_{k\in\mathscr{P}\cap\mathbb{P}_u} d_e(1,u;k), \forall y \in Y(u,\mathscr{I}) \right\}, \tag{3.12}$$

where, the speed:

$$v(y) = \begin{cases} v_L, y = -1, \\ v_U, y \geqslant 0. \end{cases} \tag{3.13}$$

Given Lemmas 3.3 and 3.4, without any loss in optimality, we can rewrite RE as follows:

$$\mathscr{D}(j|\mathscr{I}) = \max_{u \in \mathscr{B}(\mathscr{P})} \left\{ -d_p(j,u) + \min_{y \in Y(u,\mathscr{P})} \mathscr{D}(u|\mathscr{I}^+(u,y)) \right\}. \tag{3.14}$$

One can further refine the restriction as follows. Let the set:

$$\mathscr{B}_1(\mathscr{I}) = \{u : 0 < d_e(1,u;k) < \infty, \forall k \in \mathscr{P}\}. \tag{3.15}$$

In other words, $\mathscr{B}_1(\mathscr{I})$ is the set of all nodes at which capture is guaranteed in one move. It is easy to verify that $\mathscr{B}_1(\mathscr{I}) \subseteq \mathscr{B}(\mathscr{I})$. Let $\mathscr{B}_2(\mathscr{I}) = \mathscr{B}(\mathscr{I}) \backslash \mathscr{B}_1(\mathscr{I})$. By definition, at any $u \in \mathscr{B}_2(\mathscr{I})$, the path uncertainty is always reduced, i.e., $\mathscr{P}^+(u,y) \subset \mathscr{P}$, $\forall y \in Y(u,\mathscr{I})$. As mentioned earlier, this implies that the optimal control either results in a smaller (path) uncertainty or a more recent red UGS (includes the possibility of immediate capture). If neither possibility occurs, then the resulting move is necessarily sub-optimal, given the triangle inequality. This brings us to the most important feature of the revised RE (3.14). The solution to the revised RE for a given node j and information set $\mathscr{I}$ depends *only* on the performance metric for nodes $u \in \mathscr{B}_1(\mathscr{I})$ for which it can be recursively computed and for nodes $u \in \mathscr{B}_2(\mathscr{I})$ wherein the subsequent path information sets are of lesser cardinality. To allay the fears of infinite nested recursions, we note that if $u \in \mathscr{B}_1(\mathscr{I})$, by definition, the corresponding performance metric $\mathscr{D}(u|\mathscr{I}^+(u,y))$ depends only on the sub-graph emanating from u. Since we are dealing with a finite graph, the nested recursions are also finitely numbered. Since the performance metric is readily computed for path information sets of cardinality 1 from the BC (3.11), it follows that an algorithm can be set up such that the performance metrics are computed in the order of increasing cardinality of the path information sets. We establish such an algorithm, i.e., the main result of the paper, in the next section.

4. Ordered Recursive Solution

Let $t_{min} = \frac{1}{v_U} \min_{k \in \mathscr{P} \cap \mathbb{P}_u} d_e(1,u;k)$ and $\mathscr{I}_u = (\{k : d_e(1,u;k) = t_{min} v_U\}; (u,0))$. So, we rewrite RE (3.14) as follows:

$$\mathscr{D}(j|\mathscr{I}) = \max_{u \in \mathscr{B}(\mathscr{I})} -d_p(j,u) + \begin{cases} t_{min} + \mathscr{D}(u|\mathscr{I}_u), \ u \in \mathscr{B}_1(\mathscr{I}), \\ \min_y \begin{cases} t_{min} + \mathscr{D}(u|\mathscr{I}_u), y > 0 \\ \mathscr{D}(u|\mathscr{I}^+(u,-1)), y = -1 \end{cases}, \ u \in \mathscr{B}_2(\mathscr{I}). \end{cases} \tag{4.1}$$

The revised RE (4.1) deals with two cases. In the first case, $u \in \mathscr{B}_1(\mathscr{I})$, and so, the only possible observation at u is a more recent red UGS. Hence, by virtue of Lemma 3.2, the evader must have traveled full speed and taken the shortest path between 1 and u. So, the follow on performance metric, $\mathscr{D}(u|\mathscr{I}^+(u,y))$ is dictated by (3.9). If $u \in \mathscr{B}_2(\mathscr{I})$, we know that the updated path uncertainty is reduced i.e., $\mathscr{P}^+(u,y) \subset \mathscr{P}$. This is covered in the second case, where the two sub-cases deal with red and green observations at u respectively. As before, Lemma 3.2 dictates the solution for the red UGS sub-case.

COROLLARY 4.1 If the optimal control, $u^* \in \mathscr{B}_2(\mathscr{I})$ and is also an exit node, the corresponding optimal observation is a red UGS i.e., $y^* > 0$.

Proof. Let $\bar{k} = \arg\min_{k\in\mathscr{P}\cap\mathbb{P}_{u^*}} d_e(1,u^*;k)$. If u^* is an exit node, $\mathscr{D}(u^*|\mathscr{I}_{u^*}) = 0$. So, we have:

$$\mathscr{D}(j|\mathscr{I}) = -d_p(j,u^*) + \min_{y\in Y(u^*,\mathscr{I})} \begin{cases} \frac{1}{v_U} d_e(1,u^*;\bar{k}), y > 0, \\ \mathscr{D}(u^*|\mathscr{I}^+(u^*,-1)). \end{cases} \tag{4.2}$$

Since $u^* \in \mathscr{B}_2(\mathscr{I}) \subseteq \mathscr{B}(\mathscr{I})$, by definition (3.12), we have:

$$\mathscr{D}(u^*|\mathscr{I}^+(u^*,-1)) \geqslant \frac{1}{v_L} d_e(1,u^*;\bar{k}) \geqslant \frac{1}{v_U} d_e(1,u^*;\bar{k}). \tag{4.3}$$

Hence, the optimal (minimizing) observation satisfies: $y^* > 0$. □

COROLLARY 4.2 If the optimal control, $u^* \in \mathscr{B}_1(\mathscr{I})$ and $\mathscr{D}(u^*|\mathscr{I}_{u^*}) = 0$, then capture occurs at u^*. So, we have:

$$\mathscr{D}(j|\mathscr{I}) = -d_p(j,u^*) + \frac{1}{v_U} d_e(1,u^*;\bar{k}). \tag{4.4}$$

As expected, when u^* is an exit node, (4.4) coincides with the BC (3.11).

Having set up the defining characteristics of the optimal control policy, we shall now establish a procedure to compute it. Let the set of all possible path information sets be $\mathscr{Z} = 2^{\mathscr{P}_0} \setminus \emptyset$. We denote the elements of $\mathscr{Z}$ of cardinality i by $\mathscr{Z}_i^1, \ldots, \mathscr{Z}_i^{o_i}$, where $o_i = \binom{n}{i}$. For instance, $\mathscr{Z}_n^1 = \mathscr{P}_0 = \{1,\ldots,n\}$. At the other extreme, we have $\mathscr{Z}_1^k = \{k\}$, $k = 1,\ldots,n$. To compute the maximal delay at node a with a capture guarantee, we employ the Ordered Recursive Algorithm (*ORA*) prescribed below.

```
Algorithm ORA(a)
1.  for $j \leftarrow 1$ to $m$
2.      for $k \leftarrow 1$ to $|\mathbb{P}_a|$
3.          Compute $\mathscr{D}(j|(\{k\};(a,0))$ as per the BC (3.11)
4.  for $i \leftarrow 2$ to $|\mathbb{P}_a|$
5.      for $q \leftarrow 1$ to $o_i$
6.          Set $\mathscr{I} = (\mathscr{Z}_i^q;(a,0))$
7.          Set $t_{min} = \frac{1}{v_U} \min_{k\in\mathscr{Z}_i^q\cap\mathbb{P}_u} d_e(a,u;k)$
8.          for all $u \in \mathscr{B}_1(\mathscr{I})$
9.              Set $\mathscr{C}(u) = t_{min} + ORA(u)$
10.         for all $u \in \mathscr{B}_2(\mathscr{I})$
11.             Set $\mathscr{C}^+(u) = t_{min} + ORA(u)$
12.             Set $\mathscr{C}^-(u) = \mathscr{D}(u|\mathscr{I}^+(u,-1))$
13.             Set $\mathscr{C}(u) = \min(\mathscr{C}^+(u),\mathscr{C}^-(u))$
14.         for $j \leftarrow 1$ to $m$
15.             Compute $\mathscr{D}(j|\mathscr{I}) = \max_{u\in\mathscr{B}(\mathscr{I})} \{-d_p(j,u) + \mathscr{C}(u)\}$
16. return $\mathscr{D}(a|(\{1,\ldots,|\mathbb{P}_a|\};(a,0)))$
```

So, we have: $\mathscr{D}(1|\mathscr{I}_0) = ORA(1)$.

Steps 1-3 in Algorithm *ORA(a)* compute the performance metric at all nodes for all path information sets of cardinality 1. Steps 4-14 in Algorithm *ORA(a)* compute the performance metric at all nodes for all path information sets of increasing cardinality from 2 to $|\mathbb{P}_a| - 1$. Note that in steps 8 and 9, the recursive call, *ORA*(u) deals only with the sub-graph emanating from u. Also, in step 11, since $\mathscr{P}^+(u,-1) \subset \mathscr{Z}_i^q$,

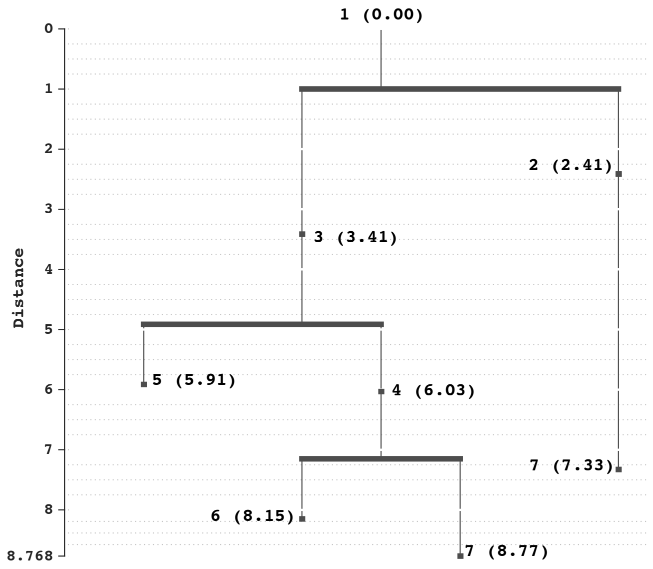

FIG. 2. Evader Paths showing nodes and corresponding evader travel distances in parentheses

we know that $\mathscr{D}(u|\mathscr{I}^+(u,-1))$ has already been computed and is readily available. The optimal control, $\mu(j|\mathscr{I})$ corresponding to the performance metric at step 15 is given by the maximizing control action in step 15.

4.1 *Numerical Example*

For the example problem (see Fig 1), the 4 possible evader paths are:

1) $1 \to 3 \to 5$,
2) $1 \to 3 \to 4 \to 6$,
3) $1 \to 3 \to 4 \to 7$, and
4) $1 \to 2 \to 7$.

In Fig. 2, we show the four different evader paths (ordered from left to right) along with the distances to nodes (in parentheses) from the entry node along each path. Suppose the pursuer knows that the evader has taken either path 2 or 3 i.e., $\mathscr{P} = \{2,3\}$. Also, let the last known evader location and time of visit be $(\underline{\mathrm{n}},0)$. So, we have the information: $\mathscr{I} = (\{2,3\};(\underline{\mathrm{n}},0))$. Since $\mathscr{P} = \mathbb{P}_4 \subset \mathbb{P}_3$, we have:

$$\mathscr{B}_1(\mathscr{I}) = \begin{cases} \{3,4\},\ \underline{\mathrm{n}} = 1, \\ \{4\},\ \underline{\mathrm{n}} = 3, \\ \emptyset,\ \underline{\mathrm{n}} = 4. \end{cases}$$

Furthermore, the only candidates for $\mathscr{B}_2(\mathscr{I})$ are nodes 6 and 7 since these are the only locations at which the path uncertainty can be further reduced. For 6 to be in $\mathscr{B}(\mathscr{I})$, we need from the definition

(3.12):

$$\begin{aligned} \mathscr{D}(6|(\{3\};(\underline{\mathrm{n}},0))) &\geqslant \frac{1}{v_L} d_e(1,6), \\ \Rightarrow \frac{1}{v_U} d_e(\underline{\mathrm{n}},7) - d_p(6,7) &\geqslant \frac{1}{v_L} d_e(\underline{\mathrm{n}},6). \end{aligned} \tag{4.5}$$

We see the interplay between the lower and upper bounds of the evader's speed and the pursuer's (unit) speed in the above relation. On the one hand, the pursuer must wait at node 6 long enough to make sure that the evader does not sneak through - waiting time determined by v_L. On the other hand, the pursuer needs to get to node 7 early enough to intercept the evader there - arrival time determined by v_U! What is even more interesting (and subtle) is the role played by the uncertainty in evader's speed or conversely the knowledge about the evader's speed. This is evident from the fact that (4.5) may be satisfied for $\underline{\mathrm{n}} = 4$, but not necessarily so for $\underline{\mathrm{n}} = 3$ (or 1). It is easy to verify that $7 \notin \mathscr{B}_2(\mathscr{I})$, since $d_e(\underline{\mathrm{n}},7) > d_e(\underline{\mathrm{n}},6)$.

To provide further clarity, let us consider the evader speed bounds: $v_L = 0.498$, $v_U = 0.51$ and focus on the sub-graph emanating from UGS 3. At time 0, the evader passes UGS 3 and proceeds towards the goal nodes. So, the initial information state available to the pursuer is $\mathscr{I}_0 = (\mathscr{P}_0;(3,0))$, where the path information, $\mathscr{P}_0 = \{1,2,3\}$. We employ Algorithm *ORA* to compute the maximal delay $\mathscr{D}(3|\mathscr{I}_0)$ and the corresponding pursuit policy. From Fig. 1, we note that $d_p(6,7) = 1, d_e(3,4) = 1.5 + \sqrt{1.25}, d_e(4,6) = 1 + \sqrt{1.25}$ and $d_e(4,7) = 1 + \sqrt{5}$. It can be easily verified that,

$$\frac{d_e(3,7)}{v_U} - \frac{d_e(3,6)}{v_L} \approx 0.98 < 1 \text{ and } \frac{d_e(4,7)}{v_U} - \frac{d_e(4,6)}{v_L} \approx 1.11 > 1. \tag{4.6}$$

So, (4.5) is satisfied for $\underline{\mathrm{n}} = 4$ but not for $\underline{\mathrm{n}} = 3$. In light of the above inequalities, the optimal pursuit policy takes the form shown in Fig. 3, which shows the decision tree for the pursuer starting with a red UGS at node 3. Fig. 3 shows (color coded) the latest pursuer exit times at future nodes visited by the pursuer, for both red and green observations. The solution dictates that:

$$\mathscr{D}(3,\mathscr{I}_0) = \frac{d_e(3,5)}{v_U} - d_p(3,5) = \frac{2.5}{v_U} - 2.5 \approx 2.4, \tag{4.7}$$

and $\mu(3,\mathscr{I}_0) = 5$. The optimal policy dictates that the pursuer get to node 5 early enough to prevent the evader escaping via that node. If the evader doesn't show up at 5, the pursuer heads to node 4 next. This is in lieu of the inequalities (4.6). Upon determining the evader's average speed between nodes 3 and 4, the pursuer proceeds to node 6. Again, it does so early enough to prevent the evader escaping via that node. The required waiting time at node 6 is diminished due to the intermediate polling of node 4 and the speed information gathered therein. So, if the evader doesn't show up at 6, the pursuer is able to reach node 7 early enough to prevent escape via that node. To conclude, the uncertainty in the evader's speed necessitates the extra move to node 4. One can easily verify that if the evader's speed was a known constant $= v_U$, the resulting maximal delay at 3 would remain the same. However, the policy would differ in that the additional move to 4 is no longer required and the pursuer would directly move from node 5 to node 6. Needless to say, if the evader's speed was a known constant $= v_L$, the resulting maximal delay at 3 would increase to:

$$\frac{d_e(3,5)}{v_L} - d_p(3,5) = \frac{2.5}{v_L} - 2.5 \approx 2.52.$$

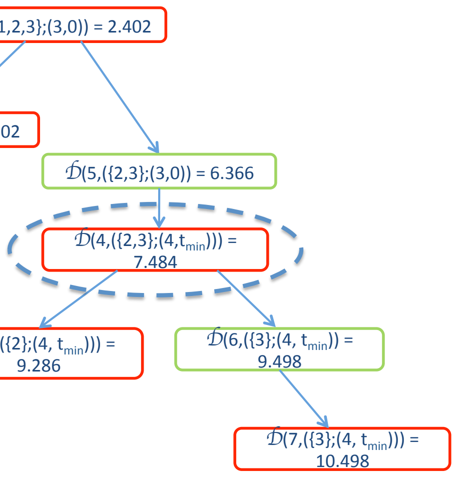
$\acute{\mathcal{D}}(3,(\{1,2,3\};(3,0)) = 2.402$
$\acute{\mathcal{D}}(5,(\{1\};(3,0)) = 4.902$
$\acute{\mathcal{D}}(5,(\{2,3\};(3,0)) = 6.366$
$\acute{\mathcal{D}}(4,(\{2,3\};(4,t_{min}))) = 7.484$
$\acute{\mathcal{D}}(6,(\{2\};(4, t_{min}))) = 9.286$
$\acute{\mathcal{D}}(6,(\{3\};(4, t_{min})) = 9.498$
$\acute{\mathcal{D}}(7,(\{3\};(4, t_{min}))) = 10.498$

FIG. 3. Optimal Decision Tree and Maximal Delay at UGS 3 and All Downstream UGSs

Indeed, when the evader's speed is a known constant, the Algorithm *ORA* simplifies considerably and allows for a non-recursive algorithm that computes the maximal delays for all possible path information sets in the order of increasing cardinality (see Krishnamoorthy *et al.* (2016)).

4.2 *Practical Example*

We tested the proposed algorithm on a real road network testbed located in Camp Atterbury, IN, USA shown in Fig. 4. The road network spread over a 3 km $\times$ 2 km rectangular area constitutes 13 UGSs (shown in red) distributed along roads shown in a Google earth satellite image. The evader is a ground vehicle traveling along the roads at a nominal speed of 20 miles per hour (mph) and the pursuer is a fixed wing UAV traveling at a fixed ground speed of 40 mph. There are 10 different paths that the evader can travel along; starting from the entry node 1. The exit/ goal nodes are 9, 11 and 13. We implemented Algorithm *ORA* for this example on MATLAB running on a MacBook Pro with 2.8 GHz processor (8 cores) and 16 GB memory. Table 1 shows the performance comparison between two cases:

1) known constant evader speed of 20 mph

2) known bounds $[19.8, 20.2]$ on the evader speed in mph.

As expected, the recursive nature of Algorithm *ORA* results in a 40 times increase in computation time for case (2) over case (1). Moreover, the resulting performance i.e., maximal delay at node 1 is also decreased due to the uncertainty in evader speed. As was the case in the simple numerical case discussed earlier, case (2) involves additional pursuer moves that are necessary to reduce the uncertainty in evader speed. This results in a loss in performance. For the sake of brevity, we do not show the entire pursuit policy here. Suffice to say that it is similar in flavor to the decision tree shown earlier (see Fig. 3) for the simple numerical example.

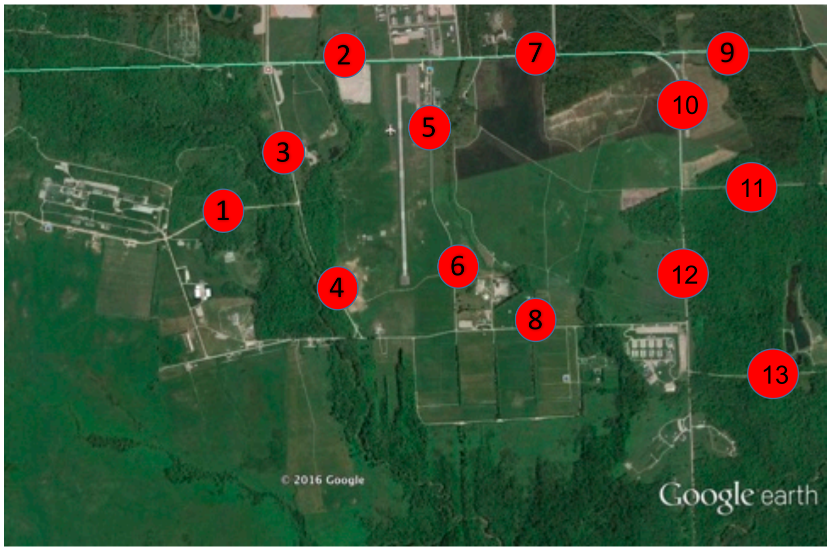

FIG. 4. Road Network with UGSs in Camp Atterbury, IN, USA

5. Conclusion

We investigate a pursuit evasion problem on a directed acyclic graph, where the evader operates in open-loop with bounded speed and the pursuer has partial information on the evader's location gleaned from UGSs embedded in the graph. The pursuer's information state entails the last known evader location and time of passage and an enumeration of feasible evader paths. We establish a recursion to compute the maximal delay at the entry node of the graph for which capture is guaranteed. To ensure tractability, we establish an optimal restriction on the search space. In particular, we show that the recursion depends only on the upper and lower bound evader speeds and therefore the complexity of the problem is greatly reduced. Furthermore, we establish an recursive algorithm that computes the performance metric in the order of increasing cardinality of the path information set.

References

BAŞAR, T. (2001) *Control Theory: Twenty-Five Seminal Papers (1932-1981)*, Wiley-IEEE Press, ch. Dual Control Theory, 181–196.

CHEN, H., KALYANAM, K., ZHANG, W. & CASBEER, D. (2014) Continuous time intruder isolation

TABLE 1 *Performance comparison between known fixed and bounded evader speed*

evader speed (mph)	max. delay @ UGS 1 (min)	CPU time (sec)
20	4.328	0.9
$[19.8, 20.2]$	3.954	42.7

using UGSs on a general graph, *American Control Conference*, Portland, OR, 5270–5275.

CHUNG, T., HOLLINGER, T. & ISLER, V. (2011) Search and pursuit-evasion in mobile robotics, *Autonomous Robots*, **31**, 299–316.

CLARKE, N. (2009) A witness version of the cops and robber game, *Discrete Mathematics*, **309**, 3292–3298.

DEMIRBAS, M., ARORA, A. & GOUDA, M. (2003) Self-Stabilizing Systems, *Lecture Notes in Computer Science*, Springer, ch. A Pursuer-Evader Game for Sensor Networks, **2704**, 1–16.

DUMITRESCU, A., KOK, H., SUZUKI, I. & ZYLINSKI, P. (2010) Vision-based pursuit-evasion in a grid, *SIAM Journal of Discrete Mathematics*, **24**, 1177–1204.

DZYUBENKO, G. & PSHENICHNYI, B. (1972) Discrete differential games with information lag, *Cybernetics and Systems Analysis*, **8**, 947–952.

FOMIN, F. & THILIKOS, D. (2008) An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science*, **399** (3), 236–245.

ISLER, V. & KARNAD, N. (2008) The role of information in cop-robber game, *Theoretical Computer Science*, **399** (3), 179–190.

KRISHNAMOORTHY, K., CASBEER, D. & PACHTER, M. (2016) Pursuit of a moving target with known constant speed on a directed acyclic graph under partial information, *SIAM Journal on Control and Optimization*, **54** (5), 2259–2273.

KRISHNAMOORTHY, K. & PACHTER, M. (2015) Pursuit of a moving ground target on a graph using partial information, *IMA Conference on Mathematics of Robotics*, St Anne's College, Oxford, UK.

KRISHNAMOORTHY, K., DARBHA, S., KHARGONEKAR, P., CASBEER, D., CHANDLER, P. & PACHTER, M. (2013) Optimal minimax pursuit evasion on a Manhattan grid, *American Control Conference*, Washington D. C., 3427–3434.

KRISHNAMOORTHY, K., DARBHA, S., KHARGONEKAR, P., CHANDLER, P. & PACHTER, M. (2013) Optimal cooperative pursuit on a Manhattan grid, *AIAA Guidance, Navigation and Control Conference*, AIAA 2013-4633, Boston, MA.

PARSONS, T.D. (1978) Pursuit-evasion in a graph, *Lecture Notes in Mathematics*, Berlin: Springer Heidelberg, **642**, 426–441.

RASMUSSEN, S. & KINGSTON, S. (2015) Development and flight test of an area monitoring system using unmanned aerial vehicles and unattended ground sensors, *International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, CO, 1215–1224.

SINOPOLI, B., SHARP, C., SCHENATO, L., SCHAFFERT, S. & SASTRY, S. (2003) Distributed control applications within sensor networks, *Proceedings of the IEEE*, **91**, 1235–1246.

SUGIHARA, K. & SUZUKI, I (1989) Optimal algorithms for a pursuit-evasion problem in grids, *SIAM Journal of Discrete Mathematics*, **2**, 126–143.

VIEIRA, M.A.M. GOVINDAN, R. & SUKHATME, G.S. (2009) Scalable and practical pursuit-evasion with networked robots, *Intelligent Service Robotics*, **2**, 247–263.