Pursuit on a Graph Using Partial Information

K. Krishnamoorthy, D. Casbeer, P. Chandler, and M. Pachter

***Abstract*— The optimal control of a "blind" pursuer searching for an evader moving on a road network and heading at a known speed toward a set of goal vertices is considered. To aid the "blind" pursuer, certain roads in the network have been instrumented with Unattended Ground Sensors (UGSs) that detect the evader's passage. When the pursuer arrives at an instrumented node, the UGS therein informs the pursuer if and when the evader visited the node. The pursuer's motion is not restricted to the road network. In addition, the pursuer can choose to wait/loiter for an arbitrary time at any UGS location/node. At time 0, the evader passes by an entry node on his way towards one of the exit nodes. The pursuer also arrives at this entry node after some delay and is thus informed about the presence of the intruder/evader in the network, whereupon the chase is on - the pursuer is tasked with capturing the evader. Because the pursuer is "blind", capture entails the pursuer and evader being collocated at an UGS location. If this happens, the UGS is triggered and this information is instantaneously relayed to the pursuer, thereby enabling capture. On the other hand, if the evader reaches one of the exit nodes without being captured, he is deemed to have escaped. We provide an algorithm that computes the maximum initial delay at the entry node for which capture is guaranteed. The algorithm also returns the corresponding optimal pursuit policy.**

I. 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 is instrumented with Unattended Ground Sensors (UGSs), placed at critical locations. As the target, referred to 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 speed of the evader, the layout of the road network and the placement of the UGSs is 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. Without loss of generality, we assume the evader is traveling on the road network at unit speed. The pursuer, on the other hand, is not restricted to be on the road network, although only upon visiting an UGS can he update the information state. In addition, the pursuer can also wait/loiter at an UGS location for an arbitrary amount of time.

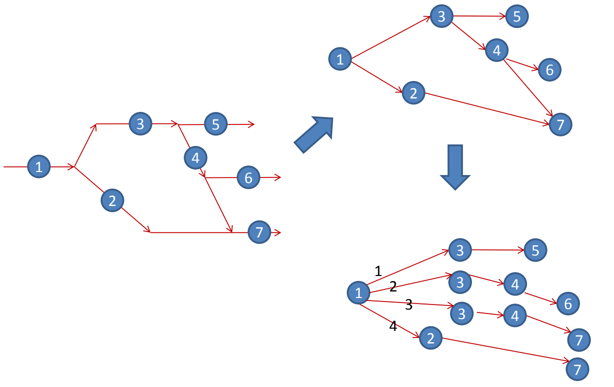


Fig. 1: Road Network, UGSs Graph, and Four Possible Evader Paths

In Fig. 1 an illustrative road network is shown. The roads are shown in red (arrows indicate direction of travel) and the numbered UGSs are blue circles. Let there be m UGSs on the network, indexed by $j = 1, \ldots, m$. Since information is only available (and capture only possible) at the UGS locations, we focus on the embedded graph, $G(\mathcal{U}, E)$, that has the UGS locations as vertices, i.e., $\mathcal{U} = \{1, \ldots, m\}$. We make the critical assumption that G is a directed acyclic graph. To visualize the setup, see Fig. 1, where the corresponding graph, G, is shown in the top right. Here, node 1 is the entry node into the network. A directed edge, $e \in E$ between two nodes on the graph has a weight that equals the distance along the road network between the nodes. For each $j \in \mathcal{U}$, let $C(j) \subset \mathcal{U}$ indicate the set of child nodes that the evader can get to from j. Let $\mathcal{G} = \{j : j \in \mathcal{U} \text{ and } C(j) = \emptyset\}$ indicate the set of exit/goal nodes that the evader is heading

Corresponding author: K. Krishnamoorthy krishnak@ucla.edu

This work is approved for public release, distribution unlimited: 88ABW-2014-4329

K. Krishnamoorthy is with the InfoSciTex corporation, Dayton, OH 45431

D. Casbeer is with the Autonomous Control Branch, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

P. Chandler (Retd.) was with the Autonomous Control Branch, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

M. Pachter is with the Department of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433

towards. In Fig. 1, nodes 5, 6 and 7 are the exit nodes. Furthermore, for each $c \in C(j)$, let the distance along the road network between the parent and child node be indicated by $T(j,c)$. Since the evader travels at unit speed, this is also the time taken by the evader to go from node j to its child node c. The pursuer's travel time from node i to node j is given by a scaled distance metric, $d_V(i,j)$. For example, it could represent the Euclidean distance between the nodes divided by the pursuer's speed. Here, we allow the metric to be more general, so long as it satisfies the triangle inequality, i.e.,

$$d_V(i,j) \le d_V(i,s) + d_V(s,j), \tag{1}$$

for any $i,j,s \in \mathcal{U}$ and $d_V(j,j) = 0,\ \forall j \in \mathcal{U}$. The above generalization allows us to model different scenarios, e.g., the pursuer could be an Unmanned Air Vehicle (UAV). We assume that the pursuer is faster than the evader, that is, the pursuer's travel time between any UGS and its child node is strictly less than the evader's travel time between the two nodes, i.e.,

$$d_V(j,c) < T(j,c),\ \forall c \in C(j),\ \forall j \in \mathcal{U}. \tag{2}$$

Without loss of generality, we assume that the evader (upon entering the network) first visits node 1 at time 0 and $1 \notin \mathcal{G}$. Let there be $n\ (\ge 1)$ possible evader paths denoted by $P_1, \dots, P_n$ emanating from node 1 and terminating at an exit node. For the example problem, see the enumeration of the 4 possible evader paths shown on the bottom right of Fig. 1. We represent an evader path P_k, $1 \le k \le n$, by the following notation: $P_k = (1 \to s_k^2 \to \dots \to s_k^{\ell_k})$, where s_k^i is the i^{th} UGS along path k and $s_k^{\ell_k} \in \mathcal{G}$. Here, ℓ_k is the number of UGSs along path k. For example, in Fig. 1 $P_1 = (1 \to 3 \to 5)$, so $s_1^2 = 3$, $s_1^3 = 5$ and $\ell_1 = 3$.

A. Properties of the Evader's Path

Let $\mathcal{T}_k(j)$ be the time of arrival of the evader to the j^{th} UGS along path k.

$$\mathcal{T}_k(j) = T(1, s_k^2) + \sum_{r=2}^{j-1} T(s_k^r, s_k^{r+1}),\ j = 2, \dots, \ell_k. \tag{3}$$

So, the length of each path is given by $|P_k| = \mathcal{T}_k(\ell_k)$. If the evader were to pick the shortest path to an exit node, then he would choose, $\bar{k} = \arg\min_{k=1}^{n} |P_k|$. Since G is a directed acyclic graph, the evader cannot visit any particular UGS more than once. However, it is possible that the evader can reach an UGS, $U \in \mathcal{U}$ via different paths. So, for UGS/node U_j in the graph, $j = 1, \dots, m$, we associate the set, $\mathcal{L}_j = \{\mathcal{L}_j(1), \dots, \mathcal{L}_j(n)\}$, where $\mathcal{L}_j(k)$ is the time at which the evader would visit node j while traveling along path k. Here, time is measured relative to time 0, when the evader visits node 1. If node j does not appear in some path $k \in \{1, \dots, n\}$, then we set the corresponding time, $\mathcal{L}_j(k) = \infty$. We assume without loss of generality, that $\forall j, \exists k$ such that $\mathcal{L}_j(k) < \infty$. This condition implies that every UGS appears, at least, in one of the paths. Clearly, if this were not the case, such an UGS can be removed from consideration. By definition, we have $\mathcal{L}_1 = \{0, \dots, 0\}$, since node 1 is visited by the evader at time 0 and along every possible evader path emanating from UGS 1. We also define the set, $\mathcal{P}_j$, $j = 1, \dots, m$, to be the set of paths that contain node j. By definition, $\mathcal{P}_j = \{k : \mathcal{L}_j(k) < \infty,\ k = 1, \dots, n\}$ and $\mathcal{P}_j \ne \emptyset,\ \forall j$. We define the initial uncertainty in evader path information available to the purser to be $\mathcal{I}_0$. Since the evader could have taken any one of n paths, $\mathcal{I}_0 = \{1, \dots, n\}$. Note that this definition of evader position uncertainty appears to be unusual in that for small initial delays, the pursuer will know where the evader is on the road that contains node 1 (e.g., see left plot in Fig. 1) and so, there is no uncertainty in his position/state; but we still say his path is uncertain in that $\mathcal{I}_0 = \{1, 2, 3, 4\}$ - see bottom right plot in Fig. 1. This is an important point: because the tacitly assumed information pattern is s.t. the evader has no situational awareness, one could argue that the evader might as well decide on his "strategy", namely, what path he will take, at $t = 0$ - in other words, the evader operates in *open-loop*. So we stipulate that at each point in time, and based on the evidence collected so far, the information of the pursuer is the currently feasible set of possible paths, one of which the evader, having made his choice at time 0, is currently traveling on. This definition of path uncertainty, meaning, the uncertainty about which of the n paths the evader is actually traveling on, results in a significant simplification of the underlying coupled estimation and control problem. Hereafter, we shall use the words uncertainty and information interchangeably with reference to the set of complete paths that the evader is possibly traveling on.

B. Evolution of System State

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 these decisions immediately after reaching an UGS location at time t and obtaining the measurement y therein: $y = -1$ for "green", or $y = d$ for "red" + delay d. Let the pursuer position at decision time t be specified by the UGS index, $p \in \{1, \dots, m\}$. The decision variable, u indicates the UGS location $u \in \{1, \dots, m\}$ that the pursuer should visit next.

The control action u is dependent on the current time, pursuer position and most recent information state: $u = \mathcal{F}(t, \mathcal{I}, p)$, where the mapping $\mathcal{F}$ is to be determined by an optimality principle - see (10) in the sequel. So, the pursuer's position and pursuer decision time evolve according to:

$$\begin{aligned} p^+ &= u, \\ t^+ &= \begin{cases} t + d_V(p,u),\ u \ne p \\ \min_{k \in \mathcal{I}} \mathcal{L}_p(k),\ u = p \end{cases} \end{aligned} \tag{4}$$

So, if the pursuer decides to stay put at the current location, the next decision epoch is the earliest possible time at which new information becomes available at the current UGS p. We denote by y the measurement the pursuer made at node p. The observation could either be a red UGS p with delay $d \ge 0$ i.e., $y = d$, or a green UGS p; whereupon the observation is denoted by $y = -1$. Note that the pursuer may choose $u = p$ only if the observation $y = -1$. If the pursuer observes a red

UGS, it confirms that the evader did pass through UGS p and there is no value in the pursuer staying at p any longer. Indeed, it would be detrimental to the search effort (in terms of time to capture).

Suppose the evader path uncertainty information available to the pursuer at p is $\mathcal{I}$. We calculate the information/path uncertainty set at time t^+ for the two possible observations at u as follows:

Red ($y^+ = d \geq 0$): The pursuer will observe a red UGS with delay $d \geq 0$ where $d \in \{s | s = t^+ - \mathcal{L}_u(k), s \geq 0, k \in \mathcal{P}_u \cap \mathcal{I}\}$. This implies that the evader was at the location of UGS u at time $t^+ - d$. Therefore, the information at time t^+ will be:

$$\mathcal{I}^+(u, d) = \{k : k \in \mathcal{I}, \mathcal{L}_u(k) = t^+ - d\}. \tag{5}$$

So we only retain those paths from $\mathcal{I}$ that are consistent with the evader passing through u at time $t^+ - d$.

Green ($y^+ = -1$): The pursuer will observe a green UGS at time t^+. This implies that the evader has not visited u thus far. Therefore, the information update is given by:

$$\mathcal{I}^+(u, -1) = \{k : k \in \mathcal{I}, \mathcal{L}_u(k) > t^+\}. \tag{6}$$

So we only retain those paths from $\mathcal{I}$ that are consistent with the evader passing through u at a time greater than t^+.

The game will terminate at UGS p^+ if at time t^+ the new observation is $y = 0$. It is also possible that having periodically updated the path uncertainty set $\mathcal{I}$ and reapplied (4), the pursuer stayed put at UGS p until time $\max_{k \in \mathcal{I}} \mathcal{L}_p(k)$ whereupon if the last observation $y = 0$ the evader is captured. If this observation is $y = -1$ instead, implying that the evader did not take any of the paths that pass through p, the control $u \neq p$ is applied and the pursuer finally moves on. The crucial point here is that although "to wait or not" is a decision to be made by the pursuer, the waiting time itself is *purely* determined by the evader arrival times and pursuer observations. This comes about because of the assumptions: 1) constant evader speed and 2) acyclic graph.

II. Optimization Problem Statement

The evader passes by node 1 at time 0. The pursuer arrives for the 1^{st} time at node 1 at time $t_0 > 0$ and is tasked with capturing the evader. Obviously, (see Fig. 1) when t_0 is small capture is possible, given the pursuer's speed advantage (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 a capture guarantee exists. This is valuable information in an operational scenario, for the following reason. The road network could lead to a protected area, that is being guarded against (ground) intrusions by security forces and the pursuer could be an 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 measured by the UAV exceeds the maximum, a security alert "close the gates!" could be issued 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 can autonomously pursue the evader, isolate it and transmit the captured image to a human operator for further action.

To pose this as an optimization problem, we introduce the following concept. Let $\mathcal{D}(1|\mathcal{I}_0) > 0$ be the *latest* time that the pursuer can arrive at/leave node 1 and still capture the evader, knowing that the evader could have taken any one of n paths, $P_1, \ldots, P_n$. Again, time is measured relative to time 0 which is the time the evader passes node 1. The evader path information available to the pursuer at node 1 is given by $\mathcal{I}_0 = \{1, \ldots, n\}$. In a similar fashion, for any UGS, $j = 1, \ldots, m$, we define $\mathcal{D}(j|\mathcal{I})$ to be the latest time the pursuer can arrive at/leave node j and guarantee capture, armed with the path information $\mathcal{I}$. Note that the arrival time to an UGS = the departure time, also in the case where the UGS is "green" and the pursuer decides to stay put. If the pursuer arrives at node j at time $t > 0$ and $t \leq \mathcal{D}(j|\mathcal{I})$, let $\mu(j|\mathcal{I}) \in \{1, \ldots, m\}$ be the corresponding UGS index to which the pursuer should head towards next, to enable capture.

Recall that each path P_k, $k = 1, ..., n$, contains an exit node and the exit node of path k is $s_k^{\ell_k}$. For the exit node $s_k^{\ell_k}$, the latest time that the pursuer can arrive there and still guarantee capture, knowing that the evader has taken path P_k is clearly $|P_k|$, the time at which the evader reaches the said node. Thus,

$$\mathcal{D}(s_k^{\ell_k} | \{k\}) = |P_k|. \tag{7}$$

Concerning the pursuer's strategy μ: if $t < |P_k|$, $\mu(s_k^{\ell_k} | \{k\}) = s_k^{\ell_k}$ i.e., the pursuer stays put at the exit node. In general, if the path information is the singleton $\{k\}$, the corresponding latest pursuer arrival time for node j, $j = 1, \ldots, m$, is given by:

$$\begin{aligned} \mathcal{D}(j|\{k\}) &= \max_{i=1}^{\ell_k} \left[\mathcal{T}_k(i) - d_V(j, s_k^i)\right], \\ &= |P_k| - d_V(j, s_k^{\ell_k}), \ \forall k. \end{aligned} \tag{8}$$

The second equality above follows from the triangle inequality (1) and speed advantage (2) assumptions. In essence, the pursuer reaches the exit node $s_k^{\ell_k}$ of path k from node j, just in time to capture the evader. So, the corresponding "go to" UGS is given by, $u = \mu(j|\{k\}) = s_k^{\ell_k}$.

Lemma 1: If the path uncertainty set $\mathcal{I}$ satisfies $\mathcal{I} \subseteq \mathcal{P}_j$ for some $j \in \{1, ..., m\}$, then:

$$\mathcal{D}(j|\mathcal{I}) \geq \min_{k \in \mathcal{I}} \mathcal{L}_j(k). \tag{9}$$

Proof: Since $\mathcal{I} \subseteq \mathcal{P}_j$, all the paths in the uncertainty set $\mathcal{I}$ go through node j. So, the pursuer can guarantee capture by arriving at node j at time $t = \min_{k \in \mathcal{I}} \mathcal{L}_j(k)$, which is the earliest time that the evader can pass through node j by taking any path, $k \in \mathcal{I}$. ■

A. Max-Min Optimization

Suppose the pursuer is at UGS index p with path information $\mathcal{I}$ and decides to visit u next. Upon reaching u, the information will change to: $\mathcal{I}^+(u, y)$, where y is the observation that the pursuer will make at u. Recall that $\mathcal{I}^+(u,y)$ is updated according to (5) and (6) for the red and green UGS observations respectively. By definition, $\mathcal{D}(u|\mathcal{I}^+(u,y))$ is the latest time at which, armed with the new information $\mathcal{I}^+(u,y)$, the pursuer can arrive at/leave u and still guarantee capture of the evader. So, the latest time that the pursuer can leave p and still capture the evader should satisfy the Recursive Equation (RE):

$$\mathcal{D}(p|\mathcal{I}) = \max_{u \in \mathcal{U}} \left[\min_{y \geq -1} \mathcal{D}(u|\mathcal{I}^+(u,y)) - d_V(p,u) \right]. \quad (10)$$

This is so, because before visiting u the pursuer cannot know whether the observation will be a red or green UGS. Hence, to guarantee capture, it has to assume the worst-case scenario that will result in the smaller of two possible pursuer exit times at u. To compute the latest pursuer exit time from p, we subtract the travel time from p to u. Finally, we take the max over all possible nodes to get the latest possible exit time from p with a capture guarantee. Per our convention, the corresponding optimal control, $\mu(p|\mathcal{I}) = u^*$, where u^* is the maximizing control in (10). We will use RE (10) to compute $\mathcal{D}(1|\mathcal{I}_0)$. Before doing so, we introduce a control constraint, $u \in \mathcal{B}(\mathcal{I}) \subset \mathcal{U}$ in (10) that will enable us to compute $\mathcal{D}(1|\mathcal{I}_0)$ in an orderly recursive fashion. In the next section, we will show that this constraint does not result in any loss in optimality.

III. Ordered Recursive Solution

Consider the simplest possible scenario: $n = 1$ i.e., there is only one path from the start node 1 to some exit node, $s_1^{\ell_1} \in \mathcal{G}$. To guarantee capture, it is sufficient for the pursuer to get to $s_1^{\ell_1}$ no later than the time that the evader gets there. So, the maximum delay at node 1 with a capture guarantee is given by,

$$\mathcal{D}(1|\{1\}) = |P_1| - d_V(1, s_1^{\ell_1}) > 0, \quad (11)$$

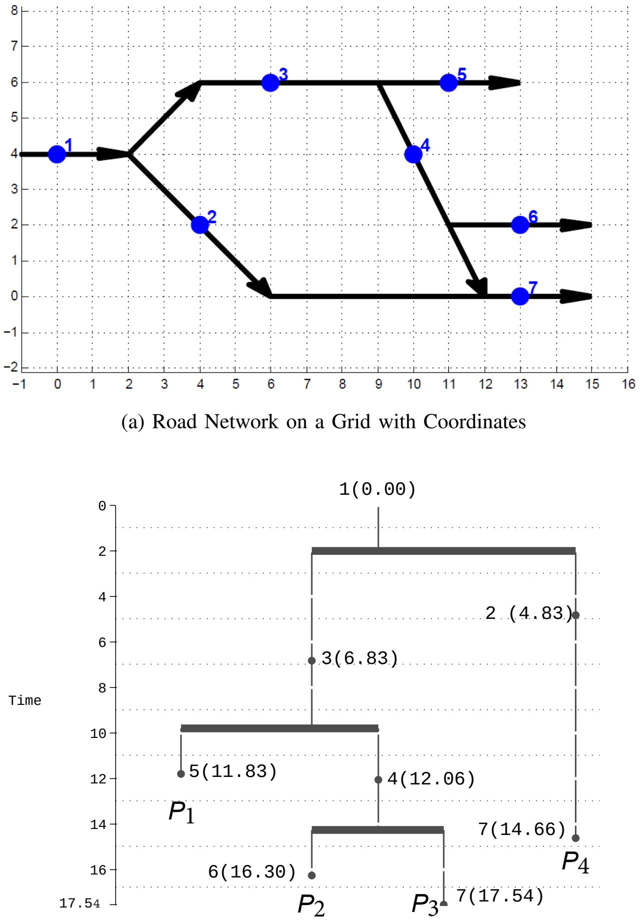
where the inequality follows from the pursuer speed advantage assumption (2). The optimal policy dictates, $\mu(1) = s_1^{\ell_1}$. For $n = 1$, there is no uncertainty in the evader's path and so, the pursuer heads straight to the exit node $s_1^{\ell_1}$ and "captures" the evader. This scenario is also reflected in (8), where the evader's path k is known to the pursuer. Since we are interested in the case where there is uncertainty in the path, our only recourse is to the RE (10), as applied to node 1 under information, $\mathcal{I}_0 = \{1, \ldots, n\}$, and so,

$$\mathcal{D}(1|\mathcal{I}_0) = \max_{u} \left[\min_{y \geq -1} \mathcal{D}(u|\mathcal{I}^+(u,y)) - d_V(1,u) \right] \quad (12)$$

As mentioned earlier, the above equation is recursive in nature. The only exception is the case where the uncertainty set's cardinality is 1, whereupon (8) provides us the values of:

$$\mathcal{D}(j|\{k\}),\ j = 1, \ldots, m,\ k = 1, \ldots, n. \quad (13)$$

A natural question that arises is the following: could we compute the exit times corresponding to uncertainty sets of cardinality 2 given (13)? The answer is yes and we illustrate the simple case of the uncertainty set $\{2,3\}$ and then generalize the method to sets of higher cardinality.



(a) Road Network on a Grid with Coordinates

(b) Evader Paths showing nodes and time

Fig. 2: Example Road Network: a) Grid and b) 4 Possible Evader Paths

Towards this end, we re-draw the example road network in Fig. 2a, with a grid in the background, to highlight the (x,y) coordinates of nodes and the distances along edges. In Fig. 2b, we show the four different evader paths (ordered from left to right) along with the evader's time of arrival $\mathcal{L}_j(k)$ (in parentheses) at nodes along each path. Indeed, $P_1 = (1 \to 3 \to 5)$, $P_2 = (1 \to 3 \to 4 \to 6)$, $P_3 = (1 \to 3 \to 4 \to 7)$ and $P_4 = (1 \to 2 \to 7)$.

Suppose we wish to compute $\mathcal{D}(6|\{2,3\})$ having already computed the values, $\mathcal{D}(6|\{2\}) = |P_2|$ and $\mathcal{D}(6|\{3\}) = |P_3| - d_V(6,7)$ from (8). Here, $|P_2| < |P_3|$ as shown in Fig. 2b. To guarantee capture, the pursuer waits at node 6 until time $|P_2|$. If the evader does not show up at 6, then the pursuer knows that the evader has taken path 3 instead. So, the pursuer proceeds to node 7 and intercepts the evader at time $|P_3|$. But this is possible only if the following condition is met:

$$|P_2| + d_V(6,7) < |P_3|. \quad (14)$$

Let us suppose that condition (14) holds. So, $\mathcal{D}(6|\{2,3\}) = |P_2|$. For all nodes $j \in \{1, \ldots, m\}$ and the uncertainty set

$\mathcal{I} = \{2,3\}$, we have from the RE (10):

$$\mathcal{D}(j|\mathcal{I}) = \max_{u\in\mathcal{B}} \left[\min\left\{\mathcal{D}(u|\mathcal{I}^r(u)), \mathcal{D}(u|\mathcal{I}^g(u))\right\} - d_V(j,u)\right], \tag{15}$$

where $\mathcal{I}^r(u) = \mathcal{I} \cap \mathcal{P}_u$ and $\mathcal{I}^g(u) = \mathcal{I} \backslash \mathcal{I}^r(u)$ are the updated uncertainty sets for the red and green observations respectively at u. We define the restriction $\mathcal{B} \subset \mathcal{U}$ as follows. The node $u \in \mathcal{B}$ if the following conditions are met:
1) $\mathcal{I}^r(u)$ and $\mathcal{I}^g(u)$ are both singleton sets, and
2) $\mathcal{D}(u|\mathcal{I}^g) \geq \min_{k\in\mathcal{I}^r(u)} \mathcal{L}_u(k)$.
Condition 1) implies that u is a special UGS in that, by going to it, the pursuer can reduce the uncertainty set $\{2,3\}$ to either $\{2\}$ or $\{3\}$ depending on whether it observes a red or green UGS (or vice-versa) at u. Condition 2) requires that the latest exit time from u for the uncertainty set $\mathcal{I}^g(u)$ (green observation) must be greater than the earliest possible evader visit time at u. Note that this requirement is already satisfied for the red observation (see Lemma 1). In other words, the pursuer will visit u only if there is a possibility that information will become available on whether or not the evader took a path through u. Else, there is no value in visiting u and one may as well ignore it. Furthermore, capture is guaranteed from u for either observation, with the corresponding pursuer exit times given by $\mathcal{D}(u|\mathcal{I}^r(u))$ and $\mathcal{D}(u|\mathcal{I}^g(u))$. In the example (see Fig. 2), $\mathcal{B} = \{6\}$ $\forall j \in \mathcal{U}$, since visiting node 6 at time $|P_2|$ can reduce the uncertainty from $\{2,3\}$ to either $\{2\}$ or $\{3\}$ and capture is guaranteed thereafter for either observation. Note that $7 \notin \mathcal{B}$ since it fails condition 2) in that:

$$\mathcal{D}(7|\{2\}) = |P_2| - d_V(7,6) < |P_2| < |P_3|.$$

So, for the uncertainty set $\{2,3\}$, it makes no sense for the pursuer to go to 7; since it has to leave 7 (before time $|P_3|$) with the uncertainty unchanged. Given the triangle inequality constraint (1), it is therefore sub-optimal for the pursuer to visit node 7. So, we have justified the restriction $\mathcal{B}$ in (15), which enables us to compute all the nodes' exit times for the uncertainty set $\{2,3\}$ of cardinality 2 from the exit times for uncertainty sets of lower cardinality.

In general, a similar restriction allows us to compute the pursuer exit times for uncertainty sets in an orderly fashion (in the increasing order of cardinality). Let the set of all possible uncertainty sets be $\mathcal{Z} = 2^{\mathcal{I}_0} \setminus \emptyset$. We denote the elements of $\mathcal{Z}$ of cardinality i by $\mathcal{I}_i^1, \dots, \mathcal{I}_i^{o_i}$, where $o_i = \binom{n}{i}$. For instance, $\mathcal{I}_n^1 = \mathcal{I}_0$. At the other extreme, we have $\mathcal{I}_1^k = \{k\}$, $k = 1, \dots, n$. Suppose $\mathcal{D}(j|\mathcal{I}_i^q)$ has been computed for $q = 1, \dots, o_i$, $\forall j$ and $i = 1, \dots, r$ for some $r \geq 1$. Then, $\forall q \in \{1, \dots, o_{r+1}\}, \forall j \in \mathcal{U}$,

$$\mathcal{D}(j|\mathcal{I}_{r+1}^q) = \max_{u\in\mathcal{B}(\mathcal{I}_{r+1}^q)} \left[\min_{y\geq -1} \mathcal{D}(u|\mathcal{I}^+(u,y)) - d_V(j,u)\right]. \tag{16}$$

Let $\mathcal{I}^r(u) = \mathcal{I}_{r+1}^q \cap \mathcal{P}_u$ and $\mathcal{I}^g(u) = \mathcal{I}_{r+1}^q \backslash \mathcal{I}^r(u)$. The three distinct possibilities at u are:

1) $\mathcal{I}^r(u) = \mathcal{I}_{r+1}^q$ which implies that the evader must pass through UGS u.
2) $\mathcal{I}^r(u) \subset \mathcal{I}_{r+1}^q$ which implies that the uncertainty is reduced at u for both red and green observations.
3) $\mathcal{I}^r(u) = \emptyset$ which implies that a green UGS is the only possible observation at u.

We define the restriction $\mathcal{B}(\mathcal{I}_{r+1}^q) \subset \mathcal{U}$ as follows. A node $u \in \mathcal{B}(\mathcal{I}_{r+1}^q)$ if $\mathcal{I}^r(u) \subseteq \mathcal{I}_{r+1}^q$ and the following condition is satisfied:

$$\mathcal{D}(u|\mathcal{I}^g(u)) \geq \min_{k\in\mathcal{I}^r(u)} \mathcal{L}_u(k) \text{ if } \mathcal{I}^r(u) \subset \mathcal{I}_{r+1}^q. \tag{17}$$

Note that the above result already holds, if the observation is a red UGS (see Lemma 1).

The restriction above implies the following: the pursuer will visit u only if one of two things happen. Either capture if possible at u or the uncertainty is reduced at u with capture guaranteed for either observation (red or green). The third possibility 3) implies that the only possible observation at u is a green UGS with no reduction in the uncertainty! Clearly, in this case, there is no information to be gained by visiting u and hence it can be removed from consideration.

Furthermore, from the triangle inequality constraint (1), it follows that the only reason to visit u under possibility 1) is to immediately capture the evader at u. As before, there is no value in visiting u otherwise, since there is no additional information available at u. So, we have the following result.

Lemma 2: If the optimal control u to (16) is such that $\mathcal{I}^r(u) = \mathcal{I}_{r+1}^q$, then capture occurs at u. So, we have $\mathcal{D}(u|\mathcal{I}^r(u)) = \min_{k\in\mathcal{I}^r(u)} \mathcal{L}_u(k)$.

In conclusion, we note that the uncertainty is either reduced or capture occurs in the next decision epoch. Since the pursuer exit times are already available for uncertainty sets of lower cardinality (former) and it is provided by Lemma 2 for the latter case, we can compute $\mathcal{D}(j|\mathcal{I}_{r+1}^q)$. Finally, to compute $\mathcal{D}(1|\mathcal{I}_0)$, we employ the following Ordered Recursive Algorithm (ORA).

Algorithm *ORA*

```
for $j \leftarrow 1$ to $m$
    for $k \leftarrow 1$ to $n$
        $\mathcal{D}(j|\{k\}) = |P_k| - d_v(j, s_k^{\ell_k})$
for $i \leftarrow 2$ to $n-1$
    for $q \leftarrow 1$ to $o_i$
        for $j \leftarrow 1$ to $m$
            Compute $\mathcal{D}(j|\mathcal{I}_i^q)$ using (16)
Compute $\mathcal{D}(1|\mathcal{I}_0)$ using (16)
return $\mathcal{D}(1|\mathcal{I}_0)$
```

Note that the optimal pursuit strategy is constrained to enforce a reduction in entropy! Indeed the entropy i.e., the cardinality of the uncertainty set will reduce, at least by 1, for every move (including waiting) made by the pursuer. As a result, the game will terminate in no more than n steps/moves! The Algorithm *ORA* has a time complexity of $\mathcal{O}(2^n m \log m)$. This is due to the number of all possible uncertainty sets: $2^n - 1$, the number of nodes for which the exit time is computed: m and the time complexity of the max operation: $\log m$.

A. Pursuer Decision Tree

To evaluate the iterative algorithm prescribed earlier, we implement it on the example problem shown in Fig. 2. We assume that the pursuer travels between any two nodes at a constant speed, V. We choose the speed such that (14) is satisfied, i.e.,

$$\begin{aligned} |P_3| - |P_2| &> d_V(6,7) = \frac{2}{V} \\ \Rightarrow V &> \frac{2}{\sqrt{5}-1} \approx 1.618, \end{aligned} \tag{18}$$

where the distance between nodes 6 and 7 equals 2 (see Fig. 2a). So, we choose $V = 1.62$ and implement Algorithm *ORA*. Fig. 3 shows the decision tree for the pursuer starting with a red UGS at node 1. The solution dictates that $\mathcal{D}(1,\{1,2,3,4\}) \approx 4.84$ and $\mu(1,\{1,2,3,4\}) = 3$. Fig. 3 also shows (color coded) the latest pursuer exit times at future nodes visited by the pursuer, for both red and green observations. Eventually, capture of the evader occurs at one of the exit nodes, 5, 6 or 7. Interestingly, the optimal evader path that contributes to the least pursuer exit time at node 1 is $P_1 = (1 \to 3 \to 5)$, which is also the shortest path, i.e., $1 = \arg\min_k |P_k|$!

If we pick $V = 1.61$ instead, we get the decision tree shown in Fig. 4. In this case, the maximum delay at node 1 with a capture guarantee reduces to ≈ 2.9. This is so because the slower moving pursuer has to capture the evader at node 3 itself, if the evader picks any path other than P_4. In other words, node 3 acts like an exit node under the reduced speed. If one were to reduce the pursuer speed even further, below some critical speed, $\underline{V}$, the algorithm will return $\mathcal{D}(1,\{1,2,3,4\}) = 0$, indicating that no initial delay can be tolerated at node 1 for any speed $V < \underline{V}$. At the other extreme, one can easily confirm that if the pursuer is able to travel at infinite speed, the corresponding $\mathcal{D}(1,\{1,2,3,4\}) = |P_1|$, the earliest evader exit time.

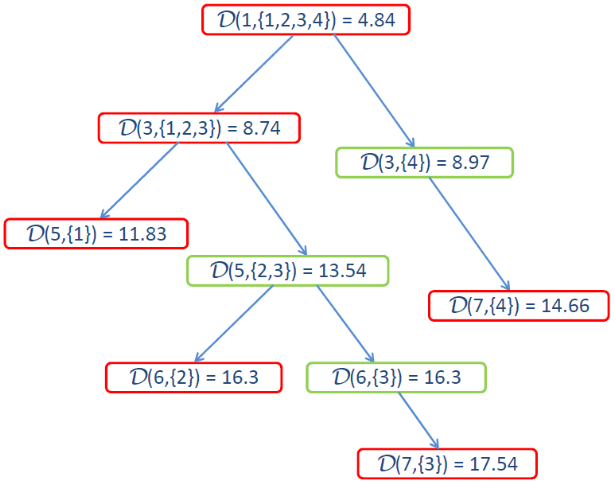


Fig. 3: Decision Tree and Latest Exit Times for $V = 1.62$

B. Reducing the Computational Burden

Since the algorithm scales exponentially with the number of possible evader paths, we explore avenues that reduce the computation time. We note that for a given graph, $G(\mathcal{U}, E)$, certain uncertainty sets will never be encountered by the pursuer if it employs a "guaranteed capture" policy. For instance, in the example problem (see Fig. 2b), the pursuer will never encounter the uncertainty set $\{1,4\}$. The reasoning behind this goes as follows. Initially the pursuer is at node 1 armed with the uncertainty set $\{1,2,3,4\}$. Now the only way the pursuer can reduce the uncertainty set to $\{1,4\}$ is by investigating node 4 and confirming that paths 2 and 3 were indeed not taken. To do so, the pursuer has to (possibly) wait at node 4 until time $\mathcal{T}_2(3) \approx 12.06$. But, $\mathcal{T}_2(3) > |P_1|$ and so, by waiting, the pursuer will necessarily allow the evader to escape via path 1! Indeed, it is possible to enumerate all the *realizable* uncertainty sets, that the pursuer will encounter in its search.

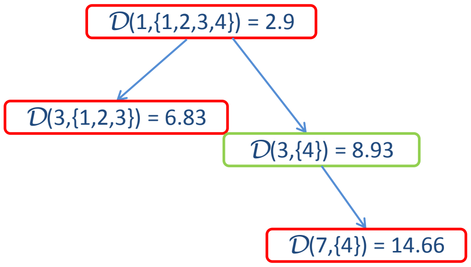


Fig. 4: Decision Tree and Latest Exit Times for $V = 1.61$

For the example problem, the realizable sets listed in Table I, are computed in the following manner. At time 0, the only information available at UGS 1 is $\{1,2,3,4\}$. At time $\mathcal{T}_4(2) \approx 4.83$, information is available at UGS 2 that can reduce the uncertainty to either $\{4\}$ or $\{1,2,3\}$ depending on whether it is red or green. At time $|\mathcal{P}_1| \approx 11.83$, information is available at UGS 5 about whether or not the evader took path 1. Hence the following additional uncertainty sets can be realized: $\{1\}$, $\{2,3,4\}$ and $\{2,3\}$. Note that for any time greater than $|\mathcal{P}_1|$, 1 can no longer appear in a uncertainty set, since it would imply that the evader has escaped. This is reflected in the table (see entries after row 4). We continue the aforementioned procedure to enumerate the sets, until the last UGS/time combination, i.e., $(7, |\mathcal{P}_3|)$. Upon completing the table, we collect all the sets that appear in column 2 of Table I. This gives us the set of all realizable sets: $\mathcal{Y} = \{\{1\},\{2\},\{3\},\{4\},\{2,3\},\{1,2,3\},\{2,3,4\},\{1,2,3,4\}\}$.

TABLE I: Realizable Uncertainty Sets at different UGSs in Chronological Order

(UGS, Time)	Realizable Sets
$(1, 0.00)$	$\{1,2,3,4\}$
$(2, 4.83)$	$\{1,2,3,4\}, \{4\}, \{1,2,3\}$
$(3, 6.83)$	$\{1,2,3,4\}, \{4\}, \{1,2,3\}$
$(5, 11.83)$	$\{1,2,3,4\}, \{4\}, \{1,2,3\}, \{1\}, \{2,3,4\}, \{2,3\}$
$(4, 12.06)$	$\{4\}, \{2,3,4\}, \{2,3\}$
$(7, 14.66)$	$\{4\}, \{2,3,4\}, \{2,3\}$
$(6, 16.30)$	$\{2,3\}, \{2\}, \{3\}$
$(7, 17.54)$	$\{3\}$

So, we only deal with 8 sets, as opposed to the $2^4 -$

$1 = 15$ possible combinations. We can now selectively apply Algorithm *ORA*, so that only $\mathcal{D}(j|\mathcal{I}),\ \forall\mathcal{I} \in \mathcal{Y}$ are computed. Note that there is no loss in optimality, by skipping the non-realizable sets. For a general graph, the reduction in number of sets depends on the structure of the graph. Nonetheless, for large n, any reduction from $2^n - 1$ could lead to substantial savings in computation time.

C. Partial Information, Dynamic Game, and Dual Control

We are calculating the maximal allowable delay at UGS 1 s.t. a pursuit strategy exists which guarantees the evader's capture before the latter reaches one of the the goal nodes, $j \in \mathcal{G}$. This is 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 [1], [2], where the highly structured graph considered therein, was a Manhattan grid. 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 [3], 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 current control. Things are not made easier by the "minimum time" control flavor of the optimization problem at hand and these difficulties are particularly exacerbated in the context of our dynamic game setting. A solution exists because the optimization problem is discrete and finite but the computational complexity of the algorithm is high.

IV. Conclusions

The optimal control of a pursuer with limited sensing capability tasked with intercepting a blind evader on a road network instrumented with UGS is considered. The pursuer is interrogating the UGS, some of which were triggered by the passing evader, and as such has access to partial observations only of the physical system's state. Specifically, the maximal allowable delay at an UGS s.t. a pursuit strategy exists which guarantees the evader's capture before the latter reaches his goal $\mathcal{G}$ is calculated and the attendant pursuit strategy is obtained. Thus, a deterministic pursuit-evasion game on a directed acyclic finite graph where the blind evader's strategy is open-loop control and the pursuer has partial information, is solved. Due to the pursuer's information pattern, which is restricted to partial observations of the physical state of the deterministic game at hand, the difficulties brought about by partial information in a dynamic game setting and the attendant $dual\ control$ effect, could not be avoided; whence the computational complexity of the solution algorithm. However, in the process of establishing the maximal delay at UGS 1 s.t. capture of the evader is possible, the maximal delays for guaranteed capture at all the UGS that are on the n paths emanating from UGS 1 are also calculated. Finally, the scenario where there are no goal vertices but the directed acyclic graph is infinite is also of interest.

References

[1] K. Krishnamoorthy, S. Darbha, P. Khargonekar, D. W. Casbeer, P. Chandler, and M. Pachter, "Optimal minimax pursuit evasion on a Manhattan grid," in *American Control Conference*, Wasington D.C., 2013, pp. 3427–3434.

[2] K. Krishnamoorthy, S. Darbha, P. Khargonekar, P. Chandler, and M. Pachter, "Optimal cooperative pursuit on a Manhattan grid," in *AIAA Guidance, Navigation and Control Conference*, no. AIAA 2013-4633, Boston, MA, 2013.

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