SIAM J. CONTROL OPTIM.
Vol. 54, No. 5, pp. 2259–2273

PURSUIT OF A MOVING TARGET WITH KNOWN CONSTANT SPEED ON A DIRECTED ACYCLIC GRAPH UNDER PARTIAL INFORMATION*

KRISHNAMOORTHY KALYANAM†, DAVID W. CASBEER‡, AND MEIR PACHTER§

Abstract. We consider the optimal control of a "blind" pursuer searching for an evader moving on a road network with fixed speed toward a set of goal locations. To aid the pursuer and provide feedback information, certain roads in the network have been instrumented with unattended ground sensors (UGSs) that detect the evader's motion. When the pursuer arrives at an instrumented node, the UGS therein informs the pursuer whether and when the evader visited that node. The pursuer is also made aware of the evader's speed. Moreover, the embedded graph comprised of the UGSs as vertices and connecting roads as edges is restricted to being a directed acyclic graph (DAG). 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's entry into the road network is registered at UGS 1, the entry node to the graph. The pursuer also arrives at the entry node after some delay d 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. Capture entails the pursuer and evader being co-located 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 of the graph without being captured, he is deemed to have escaped. We provide an algorithm that computes the maximum initial delay d for which capture is guaranteed. The algorithm also returns the corresponding optimal pursuit policy.

Key words. pursuit-evasion, partial information, dynamic game, dual control, directed acyclic graph

AMS subject classifications. 49K35, 49N75

DOI. 10.1137/140994216

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. 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 UGSs. The UAV is typically equipped with a gimballed camera, which 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 any further action. As the ground target, referred to hereafter as the evader, passes by a UGS, the UGS is triggered. A triggered UGS turns, say, from *green* to *red* and records the evader's time of passage. We assume that the layout of the road network and the placement of the UGSs is known to the pursuer/UAV. We also assume that the evader's speed is a constant and is known to the pursuer. When the pursuer arrives at an UGS

*Received by the editors December 8, 2014; accepted for publication (in revised form) June 6, 2016; published electronically September 1, 2016.
http://www.siam.org/journals/sicon/54-5/99421.html
†InfoSciTex Corporation, a DCS Company, Dayton, OH 45431 (krishnak@ucla.edu, http://sites.google.com/site/krishnakalyanam).
‡Autonomous Control Branch, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH 45433 (david.casbeer@us.af.mil).
§Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45433 (meir.pachter@afit.edu).

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 on camera. The evader's goal is to reach certain restricted locations on the road network without being captured on camera by the UAV. Note that for waiting/loitering, the UAV does not have to hover over the UGS; commercial UGSs broadcast data over WiFi, and data transfer requires only that the UAV be within communication range. In our field tests, the fixed wing UAV typically orbits above an UGS so as to collect data from it. For more details on flight testing and real world implementation of the problem considered herein, the reader is referred to [14].

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 [13]. 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 of this problem have been investigated over the years (see [8, 3] 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 the Cops and Robbers game in [3]). 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, are provided in [16, 6]. The role of information available to the players, when the pursuer has limited or no visibility of the evader, is investigated in [9]. Networked robotic systems, where distributed sensors on the ground provide sensing-at-a-distance capability, have been considered in [17, 15]. 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 [4]. However, the witness reports are immediately made available to the pursuer, whereas the observation in our work becomes available only when the pursuer visits a UGS, and hence the evader location information is delayed. Interestingly, a solution is readily available when the evader's location information is available to the pursuer after a known constant delay [7]. For the problem at hand, the delay seen by the pursuer is a function of his past actions and therefore not a constant. In a related work [5], the authors impose the 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 model 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 graph, where the evader's strategy is open-loop control and the pursuer has partial information. Such a game was previously considered in [11, 12], wherein the highly structured graph considered was a Manhattan grid. In the current work, we require that the embedded graph that comprises the UGSs as vertices and connecting roads as edges be a directed acyclic graph (DAG). In a related work [2], the authors have also relaxed the assumption of a directed acyclic graph and provided a suboptimal solution for the corresponding minimum time capture problem. This paper extends the preliminary work presented in [10] and incorporates the following significant improvements:

(1) A rigorous mathematical statement of the underlying optimization problem.
(2) A reorganization of the main result via a series of lemmas characterizing the optimal solution that ultimately lead to the ordered recursive solution method.
(3) A characterization of the role played by the pursuer's speed in the performance metric of interest.

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 difficulties brought about by the *dual control* effect [1], 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.

The remainder of the paper is organized as follows. We set up the model and assumptions in section 2. This is followed by the performance metric of interest and the corresponding optimization problem that one needs to solve, in section 3. The main focus of the paper, i.e., an optimal restriction to the search space that results in a tractable solution method, is provided in subsection 3.1. The resulting ordered recursive algorithm for solving the max-min optimization problem is provided in section 4. In subsection 4.1, we validate the optimal restriction on an example problem. This is followed by numerical results and an investigation of the role of pursuer speed in section 5. Finally, we provide some concluding remarks and directions for future research in section 6.

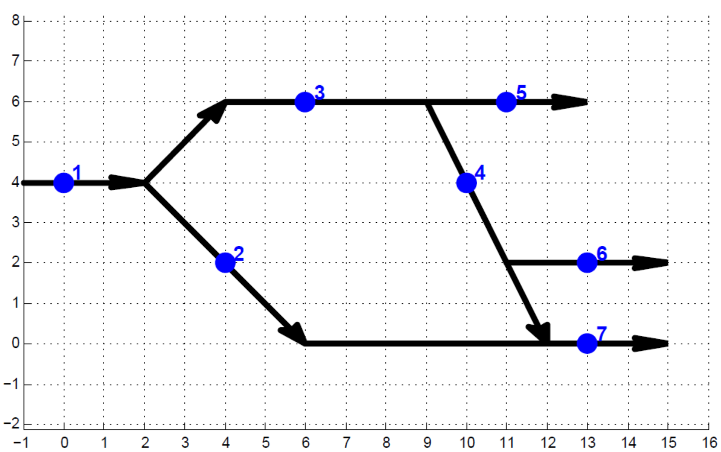

Fig. 1. *Example road network with UGSs.*

2. Model and assumptions. Figure 1 depicts an illustrative road network. The roads are shown in black (arrows indicate direction of travel), and the numbered UGSs are shown as solid blue circles. The pursuer is able to make decisions only after it gains new information at an UGS. For this reason, it is prudent to focus on the embedded graph, $G(\mathcal{U}, E)$, where the m UGSs are the vertices, i.e., $\mathcal{U} = \{1, \ldots, m\}$, and the connecting roads constitute the edges of the graph. We make the simplifying assumption that G is a DAG. The authors are aware that this assumption is restrictive, but it is critical in making the solution tractable and is also a good theoretical first step towards a solution for more complicated graphs. Let $\mathcal{G} \subset U$ indicate the set of exit/goal nodes that the evader is heading towards. For the example problem, $m = 7$

and $\mathcal{G} = \{5, 6, 7\}$—see Figure 1.

We denote the entry node into the DAG by 1 and assume that the evader enters the network via node 1 at time 0 and that, furthermore, $1 \notin \mathcal{G}$. Let there be n (≥ 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 all paths between i and j are of the same length. We assume, without loss of generality, that the evader travels at unit speed. So, $d_e(1, j; k)$ also indicates the evader's time of arrival at node j if he chooses to travel along path k. The pursuer's travel time from node i to node j is given by a metric, $d_p(i, j)$ that satisfies

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

for any $i, j, s \in \mathcal{U}$ and $d_p(j, j) = 0$ $\forall j \in \mathcal{U}$. 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.,

$$d_p(i, j) < d_e(i, j; k), \qquad \forall i, j \in \mathcal{U},\ \forall k \in \{1, \ldots, n\}. \tag{2.2}$$

For the example problem (see Figure 1), the four 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. $1 \to 2 \to 7$.

We define the set $\mathbb{P}_j$, $j = 1, \ldots, m$, to be the set of paths that go through node j; i.e., $\mathbb{P}_j = \{k : d_e(1, j; k) < \infty,\ k = 1, \ldots, n\}$. For instance, $\mathbb{P}_4 = \{2, 3\}$ in the example problem. We define the evader path information available to the purser to be $\mathcal{P} \subseteq \mathbb{P}_1 = \{1, \ldots, n\}$. Recall that the evader enters the network via node 1 at time 0 and the pursuer also arrives at node 1 after some delay $t_0 > 0$. So, the initial path information available to the pursuer is given by $\mathcal{P}_0 = \{1, \ldots, n\}$; i.e., the evader could be on any one of the n paths emanating from 1. Note that, since the evader's speed is a known constant, the path information $\mathcal{P}$ along with the current time, t, is sufficient to compute the set of all possible locations of the evader at time t. We use the path information set as the information state since it results in a significant simplification of the underlying coupled estimation and control problem.

The tacitly assumed information pattern is that the evader has no situational awareness, which is equivalent to saying that the evader decides on his strategy, namely, what path he will take, at time 0. The simplifying assumption here is that the evader has prior knowledge of the road network but is unaware of the UGS locations and also oblivious to the presence of the UAV. This critical assumption helps avoid complications arising out of adversarial evader actions and the resulting game theoretic formulation. The assumption is also supported by our flight tests [14], wherein the UGSs are typically buried or otherwise hidden from plain sight. Moreover, the pursuer is typically a small (wingspan of 5 ft) electric motor-driven UAV flying at a high altitude (500 ft), and hence, is very difficult to sight from the ground. So, in the absence of any reliable feedback information, the evader employs an open-loop control strategy.

2.1. Evolution of the system state. Let the pursuer position at decision epoch t be specified by the UGS index, $p \in \{1, \ldots, m\}$. The decision variable,

$u \in \{1, \dots, m\}$, indicates the UGS location that the pursuer should visit next. Even though the pursuer and evader trajectories 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 $y = d$ for "red" with delay $d \geq 0$. Suppose that the evader path information available to the pursuer at time t is $\mathcal{P}$. The control action u is dependent on the current time, pursuer position, and most recent information: $u = \mathcal{F}(t, p, \mathcal{P})$, where the mapping $\mathcal{F}$ is to be determined by an optimality principle—see (3.6) in what follows. We shall refer to the tuple $(t, p, \mathcal{P})$ as the system state. The pursuer's position and decision time evolve according to

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

Thus, 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 node p. Indeed, the pursuer need stay at the current location only if $\mathbb{P}_p \cap \mathcal{P} \neq \emptyset$, 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 taken a path $k \notin \mathbb{P}_p$. 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 nothing new by doing so. Upon moving to u, the path information set at time t^+ is updated for the two possible observations at u as follows:

Red ($y^+ = d \geq 0$): This implies that the evader was at node u at time $t^+ - d$. So, the path information at time t^+ is updated to

$$\mathcal{P}^+(u, d) = \left\{k : k \in \mathbb{P}_u \cap \mathcal{P},\ d_e(1, u; k) = t^+ - d\right\}. \tag{2.4}$$

So, we retain only those paths that the evader can take to arrive at u at time $t^+ - d$.

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

$$\mathcal{P}^+(u, -1) = \mathcal{P} \backslash Q, \qquad Q = \left\{k : k \in \mathbb{P}_u,\ d_e(1, u; k) \leq t^+\right\}. \tag{2.5}$$

So we remove all the paths that the evader can take to arrive at u no later than t^+.

3. Optimization problem.

The evader passes by UGS 1 at time 0. The pursuer arrives for the first time at UGS 1 at time $d > 0$ and is tasked with capturing the evader. From Figure 1, we see that capture is possible for small d, given the pursuer's speed advantage (2.2). On the other hand, if d is large, the evader will likely escape, no matter what the pursuer does. We are interested in computing the maximum initial delay d for which a capture guarantee exists. This is valuable information in an operational scenario for the following reason. As noted earlier, the road network leads to a protected area, which is being guarded against (ground) intrusions by 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 $\mathcal{D}(1|\mathcal{P}_0) > 0$ be the *latest* time that the pursuer can arrive at UGS 1 and still capture the evader with the path information, $\mathcal{P}_0$. Since the evader arrives at node 1 at time 0, $\mathcal{D}(1|\mathcal{P}_0)$ is also the maximal initial delay with a capture guarantee. In a similar fashion, for any UGS, $j = 1, \ldots, m$, we define $\mathcal{D}(j|\mathcal{P})$ to be the latest time the pursuer can arrive at node j and guarantee capture, armed with the information $\mathcal{P}$. If the pursuer arrives at node j at time $t > 0$ and $t \leq \mathcal{D}(j|\mathcal{P})$, let $\mu(j|\mathcal{P}) \in \{1, \ldots, m\}$ be the corresponding UGS index, which the pursuer should head towards next to enable capture.

To mathematically define $\mathcal{D}(1|\mathcal{P}_0)$, we proceed in the following fashion. Let the evader's open-loop policy be κ, where $\kappa \in \{1, \ldots, n\}$ is the path he chooses to travel. Let the pursuer's feedback policy, $\pi = (u_0 = 1, u_1, u_2, \ldots, u_{\bar{m}})$, entail a finite sequence of UGSs that it visits, where the corresponding decision epochs are given by $t_0, t_1, \ldots, t_{\bar{m}-1}$. The control action, $u_i(\mathcal{P}(t_{i-1})),\ i = 1, \ldots, \bar{m}$, is a function of the path information available with the pursuer at decision epoch t_{i-1}. Furthermore, the observations, $y_i(\pi, \kappa),\ i = 1, \ldots, \bar{m}$, made by the pursuer are a function of both the pursuer and evader strategies. Since we are only interested in policies with a capture guarantee, it is necessary that capture occurs at $u_{\bar{m}}$, i.e., $y_{\bar{m}} = 0$. 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 him there. In other words, the path information at decision epoch $t_{\bar{m}-1}$ must be such that $\kappa \in \mathcal{P}(t_{\bar{m}-1}) \subseteq \mathbb{P}_{u_{\bar{m}}}$; i.e., all possible evader paths go through node $u_{\bar{m}}$.

Let $\Pi(1, \mathcal{P}_0)$ be the set of all pursuer policies with a capture guarantee, starting from UGS 1 with information $\mathcal{P}_0$. The latest time that the pursuer can leave UGS 1 and guarantee capture is given by

$$J(\pi, \kappa) = d_e(1, u_{\bar{m}}; \kappa) - \sum_{i=0}^{\bar{m}-1} d_p(u_i, u_{i+1}), \tag{3.1}$$

where we have subtracted the total (pursuer) travel time from the evader's time of arrival at $u_{\bar{m}}$, to compute the required time of departure at node 1. Note that the state update equations from one decision epoch to the next are given by (2.3), (2.4), and (2.5). So, we have the performance metric of interest given by

$$\mathcal{D}(1|\mathcal{P}_0) = \max_{\pi \in \Pi(1, \mathcal{P}_0)} \min_{\kappa \in \mathcal{P}_0} J(\pi, \kappa). \tag{3.2}$$

Note that $\Pi(1, \mathcal{P}_0)$ is nonempty since it includes the trivial policy wherein the pursuer stays put and captures the evader at node 1 at time 0. The resulting initial delay is, of course, 0. Hence, we have the trivial lower bound $\mathcal{D}(1|\mathcal{P}_0) \geq 0$.

From (3.2), we have

$$\mathcal{D}(1|\mathcal{P}_0) = \max_{\pi\in\Pi(1,\mathcal{P}_0)} \min_{\kappa\in\mathcal{P}_0} \left[d_e(1,u_{\bar{m}};\kappa) - \sum_{i=0}^{\bar{m}-1} d_p(u_i,u_{i+1}) \right]$$

$$= \max_{u_1} \left\{ -d_p(1,u_1) + \max_{u_2,u_3,\dots} \min_{\kappa\in\mathcal{P}_0} \left[d_e(1,u_{\bar{m}};\kappa) - \sum_{i=1}^{\bar{m}-1} d_p(u_i,u_{i+1}) \right] \right\} \tag{3.3}$$

$$= \max_{u_1} \Bigg\{ - d_p(1,u_1) + \min_{y_1} \max_{\pi\in\Pi(u_1,\mathcal{P}^+(u_1,y_1))} \min_{\kappa\in\mathcal{P}^+(u_1,y_1)} \left[d_e(1,u_{\bar{m}};\kappa) - \sum_{i=1}^{\bar{m}-1} d_p(u_i,u_{i+1}) \right] \Bigg\} \tag{3.4}$$

$$\Rightarrow \mathcal{D}(1|\mathcal{P}_0) = \max_{u\in\mathcal{U}} \left\{ -d_p(1,u) + \min_{y\in Y(u,\mathcal{P}_0)} \mathcal{D}(u|\mathcal{P}^+(u,y)) \right\}, \tag{3.5}$$

where $Y(u_1,\mathcal{P}_0)$ is the set of all possible observations that the pursuer is likely to encounter at node u_1. In (3.3), we replace the maximization over the policy π with incremental optimization via selection of the optimal control variables: u_i, $i = 1,\dots,\bar{m}$. For the evader, choosing an optimal path κ is equivalent to sequentially minimizing $J(.,.)$ over the set of likely observations seen at UGS locations u_i, $i = 1,\dots,\bar{m}$. Furthermore, the first observation, $y_1 \in Y(u_1,\mathcal{P}_0)$, is not dependent on future pursuer actions, u_2, u_3, etc. This fact is reflected in (3.4). In (3.5), the updated path information, $\mathcal{P}^+(u_1,y_1)$, is given by (2.4) and (2.5) for the red and green UGS observations, respectively. By definition, $\mathcal{D}(u|\mathcal{P}^+(u_1,y_1))$ is the latest time at which, armed with the updated path information, $\mathcal{P}^+(u_1,y_1)$, the pursuer can leave u_1 and still guarantee capture of the evader. Hence, we have a recursive equation (3.5) for computing the performance metric of interest.

3.1. Max-min optimization. Generalizing the recursive equation (3.5), we can say the following. If the pursuer is at node j with path information $\mathcal{P}$, the latest time it can leave j and still guarantee capture is given by

$$\mathcal{D}(j|\mathcal{P}) = \max_{u\in\mathcal{U}} \left\{ -d_p(1,u) + \min_{y\in Y(u,\mathcal{P})} \mathcal{D}(u|\mathcal{P}^+(u,y)) \right\}, \tag{3.6}$$

where $Y(u,\mathcal{P})$ 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 $\mathcal{P}$. As before, the path update $\mathcal{P}^+(u,y)$ is given by (2.4) and (2.5) for the red and green UGS observations, respectively, at u. We shall refer to (3.6) 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 path information $\mathcal{P} \subseteq \mathbb{P}_u$ for some $u \in \mathcal{U}$, then*

$$\mathcal{D}(u|\mathcal{P}) \geq \min_{k\in\mathcal{P}} d_e(1,u;k). \tag{3.7}$$

Proof. Since $\mathcal{P} \subseteq \mathbb{P}_u$, the pursuer knows that every possible path that the evader can take goes through u. So, capture is guaranteed at u, so long as the pursuer gets to u at the earliest possible time that the evader can get there: $\min_{k\in\mathcal{P}} d_e(1,u;k)$. ∎

It is important to understand the trade-off between immediate capture and further reduction in (path) uncertainty embedded in RE. The pursuer can terminate the search immediately by capturing the evader at a node through which every possible evader path must pass. On the other hand, 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. 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, RE simplifies considerably, as shown below.

LEMMA 3.2. *If the path information is the singleton* $\{k\}$,

$$\mathcal{D}(j|\{k\}) = d_e(1, x_k) - d_p(j, x_k),$$

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

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

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

$$\begin{aligned} d_e(1, x_k) &= d_e(1, u; k) + d_e(u, x_k; k) \\ &\geq d_e(1, u; k) + d_p(u, x_k) \\ &\geq d_e(1, u; k) + d_p(j, x_k) - d_p(j, u) \\ \Rightarrow \mathcal{D}(j|\{k\}) &= d_e(1, x_k) - d_p(j, x_k). \end{aligned} \tag{3.8}$$

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 when the evader arrives there. ∎

From the above narrative, it appears that the optimal control u^* must result in either immediate capture or a reduction in path uncertainty. Indeed, any other control action is suboptimal, as shown below.

LEMMA 3.3. *The optimal control,* u^**, to RE* (3.6) *satisfies* $Y(u^*, \mathcal{P}) \neq \{-1\}$.

Proof. Suppose $Y(u, \mathcal{P}) = \{-1\}$; i.e., the only possible observation at u is a green UGS. This implies that none of the paths in the uncertainty set go through u. It follows 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. ∎

LEMMA 3.4. *The optimal control,* u^**, to RE* (3.6) *satisfies*

$$\mathcal{D}(u^*|\mathcal{P}^+(u^*, y)) \geq \min_{k\in\mathcal{P}\cap\mathbb{P}_{u^*}} d_e(1, u^*; k) \qquad \forall y \in Y(u^*, \mathcal{P}).$$

Proof. Suppose for control u there

$$\exists y \in Y(u, \mathcal{P}) \quad \text{s.t.} \quad \mathcal{D}(u|\mathcal{P}^+(u, y)) < \min_{k\in\mathcal{P}\cap\mathbb{P}_u} d_e(1, u; k).$$

The optimal (minimizing) observation y^* satisfies $\mathcal{D}(u|\mathcal{P}^+(u, y^*)) \leq \mathcal{D}(u|\mathcal{P}^+(u, y))$, and so, to guarantee capture, the pursuer must leave u before the earliest time that new information can be obtained therein. This implies that $\mathcal{P}^+(u, y^*) = \mathcal{P}$. 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. ◻

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

$$\mathcal{B}(\mathcal{P}) = \left\{ u : \mathcal{P} \cap \mathbb{P}_u \neq \emptyset, \mathcal{D}(u|\mathcal{P}^+(u, y)) \geq \min_{k \in \mathcal{P} \cap \mathbb{P}_u} d_e(1, u; k) \ \forall y \in Y(u, \mathcal{P}) \right\}. \tag{3.9}$$

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

$$\mathcal{D}(j|\mathcal{P}) = \max_{u \in \mathcal{B}(\mathcal{P})} \left\{ -d_p(j, u) + \min_{y \in Y(u, \mathcal{P})} \mathcal{D}(u|\mathcal{P}^+(u, y)) \right\}. \tag{3.10}$$

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

$$\mathcal{B}_1(\mathcal{P}) = \{u : \mathcal{P} \subseteq \mathbb{P}_u, d_e(1, u; k_1) = d_e(1, u; k_2) \ \forall k_1, k_2 \in \mathcal{P}\}. \tag{3.11}$$

In other words, $\mathcal{B}_1(\mathcal{P})$ is the set of all nodes at which capture is guaranteed in one move and all paths connecting them to the entry node are of identical length. It is easy to verify that $\mathcal{B}_1(\mathcal{P}) \subseteq \mathcal{B}(\mathcal{P})$. Indeed, for any $u \in \mathcal{B}_1(\mathcal{P})$, $\mathcal{P} \subseteq \mathbb{P}_u \Rightarrow \mathcal{P}^+(u, y) = \mathcal{P} \ \forall y \in Y(u, \mathcal{P})$. By virtue of Lemma 3.1, we have $\mathcal{D}(u|\mathcal{P}) \geq \min_{k \in \mathcal{P}} d_e(1, u; k)$. We show in the following result that if the optimal control $u^* \in \mathcal{B}_1(\mathcal{P})$, capture must occur at u^*.

LEMMA 3.5. *If the optimal control $u^* \in \mathcal{B}_1(\mathcal{P})$, then*

$$\mathcal{D}(u^*|\mathcal{P}^+(u^*, y^*)) = C(u^*, \mathcal{P}), \tag{3.12}$$

where $C(u^*, \mathcal{P}) = d_e(1, u^*; k)$ for any $k \in \mathcal{P}$.

Proof. For any $u \in \mathcal{B}_1(\mathcal{P})$, Lemma 3.1 tells us that $\mathcal{D}(u|\mathcal{P}^+(u, y)) \geq C(u, \mathcal{P})$. Suppose $\mathcal{D}(u|\mathcal{P}^+(u, y)) > C(u, \mathcal{P})$. This implies that the pursuer will see a red UGS observation, and the path information remains unchanged. Hence, $\mathcal{P}^+(u, y) = \mathcal{P} \ \forall y \in Y(u, \mathcal{P})$. So, the pursuer has to continue the pursuit with the same information as before, albeit having incurred an additional reward of $-d_p(j, u)$. Hence, as before, it would be better off skipping u altogether. ◻

So, we conclude that if the optimal control $u^* \in \mathcal{B}_1(\mathcal{P})$, then the search must terminate at u^*. Let $\mathcal{B}_2(\mathcal{P}) = \mathcal{B}(\mathcal{P}) \backslash \mathcal{B}_1(\mathcal{P})$. By definition, at any $u \in \mathcal{B}_2(\mathcal{P})$ the path uncertainty is always reduced; i.e., $\mathcal{P}^+(u, y) \subset \mathcal{P} \ \forall y \in Y(u, \mathcal{P})$. As mentioned earlier, this implies that the optimal control results in either a smaller (path) uncertainty or immediate capture. This brings us to the most important feature of the revised RE (3.10). The solution to the revised RE for a given node j and information set $\mathcal{P}$ depends *only* on the performance metric for nodes $u \in \mathcal{B}_1$ for which it is readily available (see Lemma 3.5) and for nodes $u \in \mathcal{B}_2$ wherein the subsequent path information sets are of lesser cardinality. Since the performance metric is readily computed for path information sets of cardinality 1 from (3.8), it follows that an algorithm can be set up such that the performance metrics are computed in the order of increasing cardinality of the information sets. We establish such an algorithm, i.e., the main result of the paper, in the next section.

4. Ordered recursive solution. For clarity, we rewrite RE (3.10) as follows:

$$\mathcal{D}(j|\mathcal{P}) = \max_{u\in\mathcal{B}(\mathcal{P})} \begin{cases} -d_p(j,u) + C(u,\mathcal{P}), \quad u \in \mathcal{B}_1(\mathcal{P}), \\ -d_p(j,u) + \min_{y\in Y(u,\mathcal{P})} \mathcal{D}(u|\mathcal{P}^+(u,y)), \quad u \in \mathcal{B}_2(\mathcal{P}), \end{cases} \tag{4.1}$$

where $C(u,\mathcal{P}) = d_e(1,u;k)$ for any $k \in \mathcal{P}$. The revised RE (4.1) deals with two cases. In the first case, $u \in \mathcal{B}_1(\mathcal{P})$, the information set $\mathcal{P}$ is such that capture is feasible by going to u (all possible evader paths through u are of the same length). In this case, the delay is given by the evader's arrival time minus the purser travel time. If $u \in \mathcal{B}_2(\mathcal{P})$, then by Lemmas 3.3 and 3.4, we know that the updated path uncertainty is reduced; i.e., $\mathcal{P}^+(u,y) \subset \mathcal{P}$. This is covered in the second case.

Let the set of all possible information sets be $\mathcal{Z} = 2^{\mathcal{P}_0} \setminus \emptyset$. We denote the elements of $\mathcal{Z}$ of cardinality i by $\mathcal{Z}_i^1, \dots, \mathcal{Z}_i^{o_i}$, where $o_i = \binom{n}{i}$. For instance, $\mathcal{Z}_n^1 = \mathcal{P}_0 = \{1,\dots,n\}$. At the other extreme, we have $\mathcal{Z}_1^k = \{k\}$, $k = 1,\dots,n$. To compute $\mathcal{D}(1|\mathcal{P}_0)$, we employ the following ordered recursive algorithm (ORA).

Algorithm 1 ORA.

1: **for** $j = 1 : m$ **do**
2:   **for** $k = 1 : n$ **do**
3:     Compute $\mathcal{D}(j|\{k\})$ as per (3.8).
4:   **end for**
5: **end for**
6: **for** $i = 2 : n-1$ **do**
7:   **for** $q = 1 : o_i$ **do**
8:     **for** $j = 1 : m$ **do**
9:       Compute $\mathcal{D}(j|\mathcal{Z}_i^q)$ using (4.1).
10:     **end for**
11:   **end for**
12: **end for**
13: Compute $\mathcal{D}(1|\mathcal{P}_0)$ using (4.1).

Steps 1–5 in Algorithm 1 compute the performance metric at all nodes for all path information sets of cardinality 1. Steps 6–12 in Algorithm 1 compute the performance metric at all nodes for all path information sets of increasing cardinality from 2 to $n-1$. Lastly, step 13 computes the maximal delay at the entry node 1, i.e., performance metric at node 1 for the initial path information set of cardinality n. Algorithm 1 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 latest exit time is computed, m; and the time complexity of the max operation, $\log m$.

4.1. Example problem. We shall illustrate the rationale behind the optimal restriction on the example road network (see Figure 1). For clarity, we show the four different evader paths (ordered from left to right) along with the time of arrival at nodes (in parentheses) in Figure 2. Let $\ell_k = d_e(1,x_k)$ indicate the length of path k. Given the evader's unit speed, ℓ_k is also the time of arrival of the evader at the exit node x_k along path k. Suppose the pursuer is at node j and knows that the evader has taken either path 2 or 3; i.e., the information set $\mathcal{P} = \{2,3\}$. We wish to compute $\mathcal{D}(j|\mathcal{P})$. From (3.8), we have $\mathcal{D}(j|\{k\}) = \ell_k - d_p(j,x_k)$ for $k = 2,3$. From the definition of the immediate capture set (3.11), we have $\mathcal{B}_1(\mathcal{P}) = \{1,3,4\}$. But, from the triangle inequality (2.1) and speed advantage (2.2) assumptions, one can

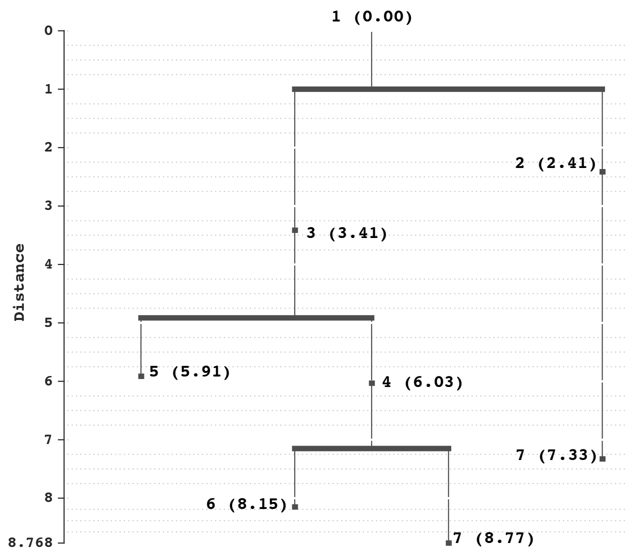

FIG. 2. *Evader paths showing nodes and corresponding evader travel distances in parentheses.*

show that the pursuer need only consider the *latest possible* capture node: UGS 4. To obtain the set $\mathcal{B}_2$, we investigate nodes 6 and 7, i.e., the only two nodes at which the information set $\mathcal{P}$ can be further reduced. It is immediately apparent that $7 \notin \mathcal{B}_2$ by virtue of $\ell_2 < \ell_3$, which results in $\mathcal{D}(7|\{2\}) < \ell_3$, thereby violating the requirement in the definition of the optimal restriction (3.9). Furthermore, $6 \in \mathcal{B}_2$ iff

$$\begin{aligned} \mathcal{D}(6|\{3\}) &= \ell_3 - d_p(6,7) \geq \ell_2 \\ &\Rightarrow \ell_2 + d_p(6,7) \leq \ell_3. \end{aligned} \tag{4.2}$$

In other words, having left UGS 6 at time ℓ_2, the pursuer must have sufficient speed to be able to arrive at 7 before the evader gets there. Hence, there are two possibilities. If (4.2) is satisfied, the optimal control $u^* = 6$ and

$$\mathcal{D}(j|\mathcal{P}) = -d_p(j,6) + \mathcal{D}(6|\{2\}) = \ell_2 - d_p(j,6).$$

If (4.2) is not satisfied, the optimal control $u^* = 4$ and $\mathcal{D}(j|\mathcal{P}) = -d_p(j,4) + d_e(1,4)$. In the latter case, the pursuer, lacking in speed, must capture the evader at UGS 4 itself. If the evader were to move beyond UGS 4, the game would be lost, and so UGS 4 acts like an exit node for the information set $\{2,3\}$.

4.2. Reducing the computational burden. Since Algorithm 1 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 information sets will never be encountered by the pursuer if it employs a policy with guaranteed capture. For instance, in the example problem (see Figure 2), the pursuer will never encounter the information set $\{1,4\}$. The reasoning behind this goes as follows. Initially the pursuer is at UGS 1, armed with the information set $\{1,2,3,4\}$. Now the only way the pursuer can reduce the information set to $\{1,4\}$ is by investigating UGS 4 and confirming that paths 2 and 3 were indeed not taken. To get this information, the pursuer must arrive at UGS 4 no earlier than time $d_e(1,4)$. However, $d_e(1,4) > \ell_1$, and so the evader will have necessarily escaped (in the worst case) via path 1! Hence, capture can no longer be guaranteed, and so the information set $\{1, 4\}$ cannot be realized in the course of pursuing a *guaranteed capture* policy. In a similar fashion, it is possible to enumerate, a priori, all the *realizable* information sets that the pursuer will likely encounter.

Table 1
Realizable uncertainty sets at different UGSs, in chronological order.

Time	Realizable sets
0	$\{1, 2, 3, 4\}$
$d_e(1, 2)$	$\{1, 2, 3, 4\}, \{4\}, \{1, 2, 3\}$
$d_e(1, 3)$	$\{1, 2, 3, 4\}, \{4\}, \{1, 2, 3\}$
ℓ_1	$\{1, 2, 3, 4\}, \{4\}, \{1, 2, 3\}, \{1\}, \{2, 3, 4\}, \{2, 3\}$
$d_e(1, 4)$	$\{4\}, \{2, 3, 4\}, \{2, 3\}$
ℓ_4	$\{4\}, \{2, 3, 4\}, \{2, 3\}$
ℓ_2	$\{2, 3\}, \{2\}, \{3\}$
ℓ_3	$\{3\}$

Indeed, for the example problem, one can enumerate all the realizable sets (see Table 1). The list in the table is compiled in the following manner. At time 0, the only information available at UGS 1 is $\{1, 2, 3, 4\}$. At time $d_e(1, 2)$, information is available at UGS 2 that can further reduce the uncertainty to either $\{4\}$ or $\{1, 2, 3\}$, depending on whether it is red or green. At time ℓ_1, information is available at UGS 5 about whether or not the evader took path 1. Hence the following additional information sets can be realized: $\{1\}$, $\{2, 3, 4\}$, and $\{2, 3\}$. Note that, for any time greater than ℓ_1, path 1 can no longer appear in an information set, since that would imply that the evader has escaped (in the worst case)! This is reflected in the table (see entries after row 4). We continue the aforementioned procedure until time ℓ_3, when the evader reaches the last possible exit node, 7. Upon completing the table, we collect all the sets that appear in column 2 of Table 1. This gives us the set of all realizable sets: $\mathcal{S} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$. So, we deal with only eight sets, as opposed to the $2^4 - 1 (= 15)$ possible combinations. We can now selectively apply Algorithm 1, so that only $\mathcal{D}(j|\mathcal{P})\ \forall \mathcal{P} \in \mathcal{S}$ are computed and the rest are skipped. Note that there is no loss in optimality by skipping the nonrealizable sets. For a general graph, the reduction in the 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.

5. Numerical results and role of pursuer speed. Intuitively, it is clear that the performance should be better at higher pursuer speeds. To better understand the relationship between pursuer speed and the performance metric of interest, let us make the following simplifying assumption. The pursuer travels between any two nodes at a constant speed V. Suppose we choose V such that (4.2) is satisfied, i.e.,

$$
\begin{aligned}
& d_p(6, 7) = \frac{2}{V} \leq \ell_3 - \ell_2 \\
(5.1) \qquad & \Rightarrow V \geq \frac{2}{\sqrt{5} - 1} \approx 1.618,
\end{aligned}
$$

where the physical distance between nodes 6 and 7 equals 2 (see Figure 1). So, we choose $V = 1.62$ and implement Algorithm 1. Figure 3 shows the decision tree for the pursuer starting at UGS 1. The solution dictates that the corresponding maximal delay at UGS 1, $\mathcal{D}(1, \{1, 2, 3, 4\}) \approx 4.84$, and the optimal control, $\mu(1, \{1, 2, 3, 4\}) = 3$.

Figure 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 UGS 1 is $(1 \to 3 \to 5)$, which is the shortest path: 1.

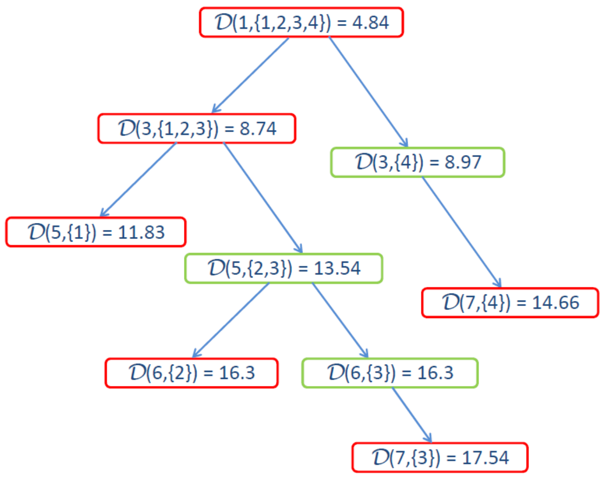

FIG. 3. *Decision tree and latest exit times for* $V = 1.62$.

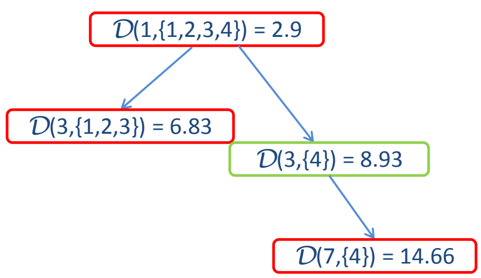

FIG. 4. *Decision tree and latest exit times for* $V = 1.61$.

If we pick $V = 1.61$ instead, we get the decision tree shown in Figure 4. For this choice, note that (4.2) is no longer satisfied. In this case, the maximum delay at UGS 1 with a capture guarantee reduces to ≈ 2.9. This is so because the slower moving pursuer has to capture the evader at UGS 3 itself, if the evader picks any path other than 4. As mentioned earlier, UGS 3 acts like an exit node under the reduced speed. Suppose that we explicitly include pursuer speed in the definition of the performance metric and let $\mathcal{D}_V(1, \{1, 2, 3, 4\})$ denote the maximal delay with capture guarantee at UGS 1. Below some critical speed, $\underline{V}$, Algorithm 1 will return $\mathcal{D}_{\underline{V}}(1, \{1, 2, 3, 4\}) = 0$, indicating that no initial delay can be tolerated at UGS 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 maximal delay at UGS 1 is ℓ_1, i.e., the earliest evader exit time. So, $\mathcal{D}_V(1, \{1, 2, 3, 4\})$ is a monotonically nondecreasing function of V and is bounded by 0 and ℓ_1—see Figure 5. The different colors in the figure reflect different pursuit policies and are demarcated by the corresponding critical pursuer speeds. At each of those critical speeds, we see a discontinuity in the maximal delay function. Furthermore, the green plot continues with no further discontinuity, with its asymptote shown by the brown line, which equals the value ℓ_1. The optimal controls from UGS 1 for the red, blue, and green regions are given by nodes 3, 3, and 5, respectively.

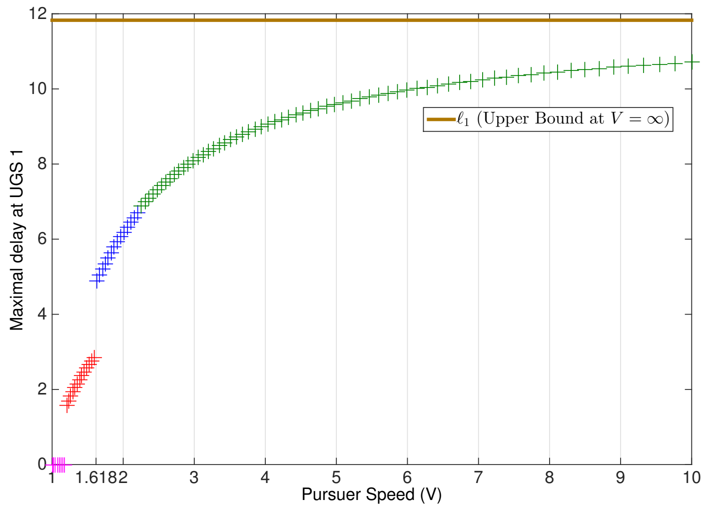

FIG. 5. *Maximal delay at UGS 1 as a function of pursuer speed (V).*

6. Conclusion. We have solved a deterministic pursuit-evasion game on a finite directed acyclic graph, where the evader's strategy is open-loop control and the pursuer has partial information. Due to the pursuer's information pattern, which is restricted to partial observations of the physical state of the system, the max-min optimization problem is not amenable to dynamic programming. However, we provide an optimal restriction on the search space that results in a tractable solution methodology. In particular, we show that the restriction results in an ordered recursive solution method whose complexity grows exponentially with the number of intruder paths. Fortunately, in the process of establishing the maximal delay at UGS 1 with a capture guarantee, the maximal delays for guaranteed capture at all the downstream UGSs are also calculated. Future work will focus on removing the constant speed assumption on the evader and also relaxing the directed acyclic graph assumption. In particular, extension to the case of varying evader speed with known (positive) bounds looks promising and is the focus of our current research.

REFERENCES

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

[2] H. Chen, K. Kalyanam, W. Zhang, and D. W. Casbeer, *Continuous time intruder isolation using UGSs on a general graph*, in Proceedings of the American Control Conference, Portland, OR, 2014, IEEE Press, Piscataway, NJ, 2014, pp. 5270–5275.

[3] T. H. Chung, G. A. Hollinger, and V. Isler, *Search and pursuit-evasion in mobile robotics*, Autonomous Robots, 31 (2011), pp. 299–316.
[4] N. E. Clarke, *A witness version of the cops and robber game*, Discrete Math., 309 (2009), pp. 3292–3298.
[5] M. Demirbas, A. Arora, and M. Gouda, *A pursuer-evader game for sensor networks*, in Self-Stabilizing Systems, Lecture Notes in Comput. Sci. 2704, Springer, New York, 2003, pp. 1–16.
[6] A. Dumitrescu, H. Kok, I. Suzuki, and P. Żyliński, *Vision-based pursuit-evasion in a grid*, SIAM J. Discrete Math., 24 (2010), pp. 1177–1204.
[7] G. T. Dzyubenko and B. N. Pshenichnyi, *Discrete differential games with information lag*, Cybernet. Systems Anal., 8 (1972), pp. 947–952.
[8] F. V. Fomin and D. M. Thilikos, *An annotated bibliography on guaranteed graph searching*, Theoret. Comput. Sci., 399 (2008), pp. 236–245.
[9] V. Isler and N. Karnad, *The role of information in cop-robber game*, Theoret. Comput. Sci., 399 (2008), pp. 179–190.
[10] K. Krishnamoorthy, D. W. Casbeer, and M. Pachter, *Pursuit on a graph under partial information*, in Proceedings of the American Control Conference, Chicago, IL, 2015, IEEE Press, Piscataway, NJ, 2015, pp. 4269–4275.
[11] K. Krishnamoorthy, S. Darbha, P. Khargonekar, D. W. Casbeer, P. Chandler, and M. Pachter, *Optimal minimax pursuit evasion on a Manhattan grid*, in Proceedings of the American Control Conference, Washington DC, 2013, IEEE Press, Piscataway, NJ, 2013, pp. 3427–3434.
[12] K. Krishnamoorthy, S. Darbha, P. Khargonekar, P. Chandler, and M. Pachter, *Optimal cooperative pursuit on a Manhattan grid*, in Proceedings of the AIAA Guidance, Navigation and Control Conference, Boston, MA, 2013, AIAA 2013-4633.
[13] T. D. Parsons, *Pursuit-Evasion in a Graph*, Lecture Notes in Math. 642, Springer, Berlin, Heidelberg, 1978, pp. 426–441.
[14] S. Rasmussen and D. Kingston, *Development and flight test of an area monitoring system using unmanned aerial vehicles and unattended ground sensors*, in Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, 2015, IEEE Press, Piscataway, NJ, 2015, pp. 1215–1224.
[15] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, *Distributed control applications within sensor networks*, Proc. IEEE, 91 (2003), pp. 1235–1246.
[16] K. Sugihara and I. Suzuki, *Optimal algorithms for a pursuit-evasion problem in grids*, SIAM J. Discrete Math., 2 (1989), pp. 126–143.
[17] M. A. M. Vieira, R. Govindan, and G. S. Sukhatme, *Scalable and practical pursuit-evasion with networked robots*, Intelligent Service Robotics, 2 (2009), pp. 247–263.