SMARTER LIONS: EFFICIENT COOPERATIVE PURSUIT IN GENERAL BOUNDED ARENAS

ZHENGYUAN ZHOU[*‡], JONATHAN SHEWCHUK[¶],
DUŠAN STIPANOVIĆ[†], HAOMIAO HUANG[§], AND CLAIRE TOMLIN[¶]

Abstract. We study a full-knowledge pursuit-evasion problem where cooperating pursuers attempt to capture a single evader of equal speed in a closed, bounded, two-dimensional arena whose boundaries may be curved. By a famous result of Besicovitch, a point-sized lion (pursuer) can evade a single point-sized man (evader) indefinitely. If the lion is endowed with a *capture radius* in the form of an outstretched paw of length r, by a result of Alonso, Goldstein, and Reingold, the man can evade the lion for time that is superlinear in the diameter of circular arena. We propose a pursuit algorithm by which two pursuers can capture an evader in a simply connected arena in time that is linear in the diameter of the arena, even when the capture radius is zero. This algorithm is asymptotically optimal, highlights the performance gap between one pursuer and two pursuers (even in a convex domain) and establishes that no more than two pursuers are needed for optimal pursuit in simply-connected domains. Furthermore, we propose a pursuit algorithm by which three pursuers are guaranteed to capture an evader in a general two-dimensional arena with h obstacles in time that is proportional to hd (when the capture radius is zero). To the best of our knowledge, this is the first algorithm that ensures guaranteed capture in an arbitrary two-dimensional domain in continuous-time (the hardest case) and that yields the best time-capture bounds.

Key words. Pursuit-evasion; robotics; differential games; control theory.

AMS subject classifications. 90D25.

1. Introduction.

In the late 1930s, Richard Rado introduced the *Lion and Man problem*, in which a lion and a man—each modeled as a single point—run about an arena, modeled as a closed circular disk. The lion tries to capture the man by occupying the same point, and the man tries not to be caught. Both players move continuously with the same maximum speed, and can react instantaneously to each other's positions. For more than a decade, it was wrongly believed that the lion could always catch the man in finite time by starting at the center and unfailingly staying on the same radial segment as the man. In 1952, Abram S. Besicovitch discovered a startling strategy by which the man can evade the lion forever, by running along a piecewise linear path whose segments have successively shorter lengths that sum to infinity. Although the lion can draw arbitrarily close to the man, the distance is never reduced to zero [30, 9].

In this article, we consider a pursuit-evasion game in which several cooperating pursuers attempt to capture an evader moving with equal speed in a bounded, two-dimensional arena. The arena may contain arbitrarily shaped obstacles that players must go around. The evader is captured when it comes within a distance $r \geq 0$ from the nearest pursuer, called the *capture radius*. We suppose the players (evader and pursuers) have *full knowledge*, meaning they know at all times the locations of all the other players and can react instantaneously to them. Several questions naturally arise. What pursuit strategies guarantee that the evader will be captured? How many pursuers are needed? How quickly can the evader be captured? Can we obtain

[*]Stern School of Business, New York University and IBM Research. Email: zyzhou@stanford.edu.
[‡]Corresponding author.
[¶]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.
[†]Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign.
[§]Kleiner Perkins.

(asymptotically) optimal bounds on the capture time?

The Lion and Man problem has sustained the interest of researchers for decades [4, 14, 15, 18, 37]. Alonso, Goldstein, and Reingold [5] consider the problem for a single lion in a circular arena of diameter d. To circumvent Besicovitch's proof of impossibility, they endow the lion with a *capture radius* in the form of an outstretched paw of length r. They describe a strategy for the lion that guarantees the capture of the man in $\mathcal{O}(d \cdot \log(d/r))$ time, and a strategy for the man that evades capture for $\Omega\left(d \cdot \sqrt{\frac{\log(d/r)}{\log\log(d/r)}}\right)$ time. It is an open problem to narrow the gap between these two bounds. Alexander, Bishop, and Ghrist [3] show that the simplest strategy, direct pursuit (i.e., always moving towards the current position of the evader at maximum speed), suffices eventually to capture a man in any bounded, simply connected arena if r is positive. Moreover, their result generalizes to higher-dimensional domains, and to many other topologies besides Euclidean domains. Kopparty and Ravishankar [28] describe the precise initial conditions for which multiple lions can capture a man in $\mathbb{R}^n$ or in an unbounded convex polyhedron. Alexander, Bishop, and Ghrist [4] consider pursuit-evasion games in arbitrary convex and unbounded domains. The above-mentioned works all assume that no obstacles reside in the domain. Bhadauria, Klein, Isler, and Suri [8] consider the problem for polygonal arenas, not necessarily convex, that may include polygonal obstacles. They show that three lions are sufficient and sometimes necessary for a successful capture in such environments. We compare in detail their work and ours in Section 1.1.

A rich literature studies pursuit-evasion games on discrete graphs, where players move discretely from vertex to vertex [1, 2, 6, 19, 20, 21, 36, 35]. These papers differ in how the players move, whether they know each others' positions, and what questions are studied. On the other hand, many pursuit-evasion games that are continuous in time and space, including those we study in this article, can be modeled as dynamical non-cooperative games [7, 10, 22, 26]. For example, Isaacs [26] finds optimal strategies for both the pursuer and the evader in the *homicidal chauffeur* game, in which a car with limited turning radius tries to run over a nimble pedestrian, by solving a Hamilton–Jacobi–Isaacs (HJI) equation backward in time. Flynn [17] and Lewin [29] employ HJI methods to solve the *Lion and Man problem.* More generally, HJI methods have been employed to solve different types of differential games [25, 16]. However, only simple games with very simple geometries can be solved analytically. [43] considered Voronoi-based pursuit strategies in simple convex domains (without obstacles) and derived an analytical pursuit strategy that guarantees capture under a single pursuer (with a positive radius). However, no capture time bound is given and the pursuit strategy does not extend to simply connected domains (let alone domains with obstacles). More complicated games such as capture-the-flag have been solved numerically [24, 34]. HJI methods offer a powerful conceptual framework, but they suffer from the curse of dimensionality; in practice, games with more than two players cannot be solved numerically.

We mention that dynamical non-cooperative games [22, 10] also model a number of other important and related problems, such as motion planning problems [23, 27, 39], reach-avoid problems [42, 11, 12, 13, 41] pursuit-evasion problems with a focus on evasion strategies [33, 32, 31] and so on. Finally, there is also an extensive literature in applying techniques developed in pursuit-evasion games to robotics: see [40] for an articulate article and the references therein.

1.1. Our Contributions. The main purpose of this article is to describe pursuit strategies that are asymptotically faster than previous strategies, and to establish guaranteed capture times for these strategies. We have three main contributions.

First, for arenas that are simply connected, we give a pursuit strategy called *orthographic pursuit* by which two pursuers can capture the evader in $\mathcal{O}(d)$ time, where d is the *intrinsic diameter* of the arena: the length of the longest shortest path connecting two points in the arena. More precisely, the *intrinsic distance* between two points is the length of the shortest path connecting them that stays in the arena, and the *intrinsic diameter* of an arena is the greatest intrinsic distance between two points in the arena. This capture time is clearly asymptotically optimal, and is guaranteed even if the capture radius is zero (recall that the evader can evade a single pursuer indefinitely). Recall also that even in the presence of a positive capture radius, Alonso et al. [5] gave a superlinear lower bound (in terms of the intrinsic diameter of the domain) for capture by a single pursuer. Consequently, our results in Section 3 highlight that two lions are asymptotically better than one; and to the best of our knowledge, this is the first asymptotically optimal capture bound for two pursuers in a simply connected domain. Furthermore, we also give explicit constants in the capture time bound, as we believe subsequent improvements in these constants can also be meaningful. In particular, for the important special cases of convex and bounded simply connected domains, we obtain capture times proportional to the domain diameter.

Second, we show that three pursuers are sufficient and sometimes necessary to guarantee capture in a bounded but otherwise general arena (i.e., possibly with any number of obstacles) by designing a novel three-pursuer cooperative pursuit strategy. To put this guaranteed capture result in perspective, we note that the existing state-of-the-art [8] gives a pursuit strategy and establishes that for polygonal arenas with a finite number of polygonal obstacles, three pursuers are sufficient and sometimes necessary for a successful capture[1]. This result leaves two important fronts open. First, the proof of guaranteed capture by Bhadauria et al. [8] relies inherently on the induction on the finite number of vertices of the polygons, which is not applicable to general-shaped arenas. In particular, it is not clear whether applying the pursuit strategy given by Bhadauria et al. would lead to guaranteed capture in general arenas. Consequently, the assumption of a polygonal environment is not only a modeling convenience, but also a crucial feature in establishing the theoretical guarantee. In comparison, we lift the polygonal restriction and work with general arenas. Second and perhaps more importantly, Bhadauria et al. adopt a discrete-time model where each player takes turns to move. Specifically, Bhadauria et al. assume that "in each move, a player can move to any position whose shortest path distance from its current position is at most one." One implication of this assumption is that this discretization of time also discretizes the space: pursuers are now effectively equipped with a positive radius of 1, which can be different from a radius of 0. (Our result handles the case $r = 0$; recall Besicovitch's result that an evader can evade a single pursuer forever when the capture radius is zero.) More broadly, this discrete-time model is a special case of our continuous-time model, where pursuers move continuously by selecting instantaneous control inputs at all times (rather than just at discrete time points). In particular, given any continuous-time pursuit strategy, a pursuer can easily convert it into a discrete-time pursuit strategy (by simply integrating the continuous-time

[1]The three-pursuer necessity proof of Bhadauria et al. [8] adapts a construction of Aigner and Fromme [2] on graphs to two-dimensional Euclidean spaces.

control over a unit time interval to obtain the final location and moving directly to it), whereas it seems difficult to convert a discrete-time pursuit strategy into a continuous-time strategy. As such, our result provides a meaningful generalization of the result of Bhadauria et al. [8] and settles the open problem raised by Aigner and Fromme [2]: how many pursuers are needed to guarantee capture in a general bounded arena? Our necessity proof adapts a similar construction as in [2, 8], and is included in the appendix for both completeness and the modeling differences.

Third, we analyze our proposed pursuit strategy for the general arena in detail and characterize the capture time bound. Specifically, we show that three pursuers can capture the evader in $\mathcal{O}(hd)$ time, where h (for "holes") is the number of obstacles in the arena. This capture time bound provides a substantial performance improvement over the best comparable previous work [8] mentioned above, which gives a strategy by which three pursuers in a polygonal arena are guaranteed to capture an evader in $\mathcal{O}(vd^2)$ time, where v is the number of vertices of the arena (obstacles included). Note that in addition to the linear dependence on the intrinsic diameter, we have $h \leq v/3$ for polygonal domains. We emphasize that for a polygonal representation to approximate the general-shaped obstacles, a large number of vertices are required in order to achieve a fine approximation, in which case $h \ll v$; in such cases, it is particularly important for the capture time bound to depend on the number of obstacles, rather than the number of vertices.

To highlight the connection between the orthographic pursuit strategy in simply connected domains and the pursuit strategy in general domains, we note that part of the sharp performance in $\mathcal{O}(hd)$ comes from utilizing the two-pursuer orthographic pursuit as a subroutine during the final phase of the pursuit process, where the evader has been trapped in an obstacle-free environment. Consequently, orthographic pursuit is not only interesting on its own, but also has broader implications.

2. Problem Formulation. Our planar pursuit-evasion game takes place in a bounded two-dimensional domain with a finite number of obstacles. Let $\Omega \subset \mathbb{R}^2$ be a closed, bounded, nonempty, simply connected *ambient space*. We use $\partial\Omega$ to denote the boundary[2] of the ambient space Ω. To rule out pathological structures such as fractals, we restrict $\partial\Omega$ to be of finite length and piecewise smooth with finitely many pieces and finitely many inflection points. In the ambient space there are h pairwise disjoint obstacles $\omega_i \subset \Omega$ that satisfy all the same restrictions as Ω except that they are open point sets, not closed. Their union is $\omega = \bigcup_{i=1}^{h} \omega_i$. Players move in a *free space* $\Omega_{\text{free}} = \Omega \setminus \omega$, which is a closed, connected point set. The boundary of the free space Ω_{free} is $\partial\Omega_{\text{free}} = \partial\Omega \cup \bigcup_{i=1}^{h} \partial\omega_i$: we emphasize that the boundary of the free space includes both the boundary of the ambient space and the boundary of the obstacles. Figure 1 depicts a representative game domain.

In this pursuit-evasion game, one or more *pursuers* chase a fleeing *evader* whose goal is to evade capture for as long as possible. Following the standard terminology and notation of control theory, all the players (a player is either a pursuer or an evader) have simple motion dynamics, where the control input is the instantaneous velocity. More specifically, let N be the total number of pursuers, and let $p^i(t) \in \mathbb{R}^2$ be the position of pursuer i $(1 \leq i \leq N)$ at time t. Let $e(t) \in \mathbb{R}^2$ be the position of

[2]This is often colloquially called a wall in the literature.

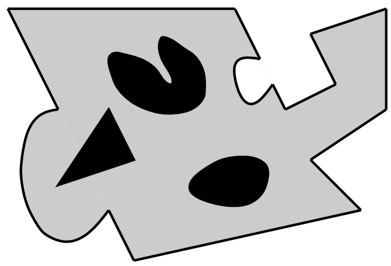

Fig. 1. *A pursuit-evasion game domain. The free space is gray, and includes its boundary. Obstacles appear in black.*

the evader at t. The equations of motion for all the players are:

$$\dot{e}(t) = d(t),\ e(0) = e^0,$$
$$\dot{p}^i(t) = u_i(t),\ p^i(0) = p^{i,0},\ i = 1, ..., N, \tag{2.1}$$

where $d(t) \in \mathbb{R}^2$ and $u_i(t) \in \mathbb{R}^2$ are the velocity control inputs of the evader and pursuer i, respectively, and $e^0, p^{i,0} \in \Omega$ are the initial evader and pursuer positions. We assume all players have equal maximum speed and hence, without loss of generality, we can normalize each player's maximum to be 1. Consequently, all players' control inputs are constrained to lie within the set $\mathbb{D} \subset \mathbb{R}^2$, where $\mathbb{D}$ is the unit disk.

$$d(t) \in \mathbb{D}, u_i(t) \in \mathbb{D}, \forall t \geq 0, \forall i; \mathbb{D} = \{d : \|d\|_2 \leq 1\}. \tag{2.2}$$

Further, the pursuers and the evader are restricted to the free space Ω_{free} and hence the motions of the evader and pursuers, as described by equation (2.1), are also constrained to lie within the region Ω_{free}, i.e.,

$$e(t),\ p^i(t) \in \Omega_{\text{free}},\ \forall t \geq 0. \tag{2.3}$$

Any velocity input $d(t)$ or $u_i(t)$ satisfying the constraint (2.2) and not causing a player to leave Ω_{free} is called an admissible input for the player. For simplicity, we assume that the control functions $d(\cdot)$ and $u_i(\cdot)$ are piecewise continuous in time, a weak and common assumption in the pursuit-evasion and robotics literatures[3].

The evader is captured if a pursuer comes within a disk of radius r centered on the evader; r is called the *capture radius*. Throughout the rest of this article, unless otherwise specified the capture radius r is 0. This is the hardest case: recall Besicovitch's result that an evader can evade a single pursuer (with equal maximum speed) forever when the capture radius is 0 in a bounded and circular arena. Finally, at all times, we assume each pursuer knows the position and velocity of the evader and can react instantaneously to it; we call this assumption *full-knowledge pursuit*. Our goal is to provide the pursuers with a pursuit strategy that guarantees capture; moreover, the worst-case capture time should be small. Intuitively, a (joint) pursuit strategy is a contingency plan that specifies how each pursuer should move based on the evader's current state (including both the current position and current velocity) as well as the game domain $\Omega_{\text{free}} = \Omega \setminus \omega$. More formally, a joint pursuit strategy

[3]This assumption can be further relaxed without altering the results in the article. For instance, we only need to require that $d(\cdot)$ and $u_i(\cdot)$ are measurable functions that are also Lebesgue integrable. However, doing so would overcomplicate the exposition and obscure the important ideas in the article.

is a function F that maps the evader's current position $e(t)$, current velocity $d(t)$, all pursuers' current position $p(t)$ and the game domain Ω_{free} to the joint control of all pursuers $F(e(t), d(t), p(t), \Omega_{\text{free}})$. That is, $F(e(t), d(t), p(t), \Omega_{\text{free}})$ represents the current velocities of all the pursuers. Pursuer i's control can then be conveniently denoted by $F_i(e(t), d(t), p(t), \Omega_{\text{free}})$. Consequently, in describing a pursuit strategy, we need only specify at each time t, prior to capturing the evader, what each pursuer's velocity is given the current configuration of the pursuit-evasion game.

If it seems strange that a pursuer can take into account the evader's current velocity to decide its own velocity, bear in mind that a pursuer has the challenging goal of exact capture (i.e., $r = 0$); it must coincide exactly with the evader at some point in time. If the pursuers cannot know the evader's exact velocity, the evader can evade capture forever through fractal random variations in its direction or speed of motion (even if its velocity is constrained to be continuous). If the capture radius is strictly positive (i.e., $r > 0$), our pursuit strategy can easily be modified to operate without knowing the evader's velocity, being a function only of all the players' positions and the domain. We omit the details.

For a pursuit strategy to be admissible, two requirements must be satisfied. 1) The resulting instantaneous control $F_i(e(t), d(t), p(t), \Omega_{\text{free}})$ for each pursuer i must be admissible. 2) The resulting control trajectory $F_i(e(\cdot), d(\cdot), p(\cdot), \Omega_{\text{free}})$ for each pursuer i must be piecewise continuous.

For the rest of this article, we develop efficient pursuit strategies that ensure capture of the evader, regardless of how the evader moves, where efficiency is measured in terms of the capture time. Of course, the number of pursuers and the number of obstacles influence whether such a strategy exists, and if so, its capture time. Consequently, to streamline the presentation of pursuit strategies and to obtain sharp capture time bounds, we divide the exposition on pursuit strategies into two categories: 1) simply connected arenas (i.e., domains where no obstacles exist); 2) general arenas with obstacles. This division of presentation is due to the differences between these two categories. First, general arenas with obstacles require three pursuers to guarantee capture while two pursuers are sufficient for capture in simply connected domains. Second, the pursuit strategy for simply connected arenas is much simpler than the strategy for general arenas. Third, the capture time bounds are different.

Despite these differences, an important building block shared by all the pursuit strategies (for different classes of domains) in the article lies in defending some path. We next introduce the notion of a defendable path.

Definition 1 *A path $s \subset \Omega_{\text{free}}$ is* defendable *if there is a strategy by which a pursuer can position himself on s in finite time, and thereafter move on s to guarantee that he will coincide with the evader at any time the evader moves onto s. A pursuer that is correctly positioned and following this strategy is said to be* defending *s.*

Under this definition, once the pursuer is correctly positioned on the path s, the evader cannot cross s without being captured: at the moment when the evader is crossing s, the pursuer will coincide with the evader, at which point *exact* capture occurs. It turns out that in any domain, every shortest path is defendable. However, the strategies used to defend a shortest path, as an important building block of the overall pursuit strategy, vary according to the category of the domain. In particular, even though a shortest path can be defended in different ways, a particular defending strategy will be chosen to synthesize a good pursuit strategy. We will discuss the different pursuit strategies as well as the corresponding defending strategies in the next two sections. We finish this section with a little more notation.

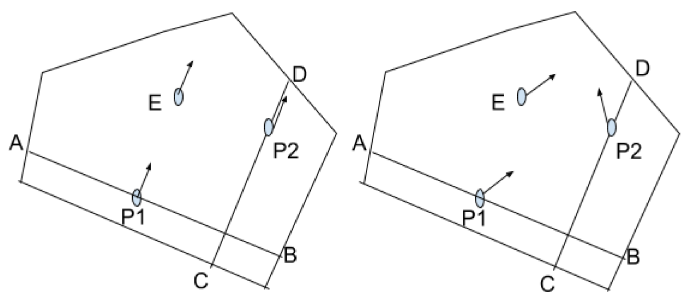

FIG. 2. *The orthographic pursuit strategy in a convex domain. Left: P_2 projects the evader's velocity onto CD. In this example, the projected velocity on CD has magnitude 1, so P_2 matches that velocity. By contrast, the projected velocity on AB is zero, so P_1 does not have a velocity component parallel to AB. Instead, it moves at full speed orthogonally to AB, advancing AB toward E. Right: An evader with a different velocity. Both P_1 and P_2 project the evader's velocity onto AB and CD, respectively. Each matches the projected evader velocity (parallel to the line segment), and any remaining speed (with maximum speed 1) is used to advance the line segments toward the invader.*

Definition 2 *Let $s(x, y) \subset \Omega_{\text{free}}$ denote the shortest path in the free space between two points $x, y \in \Omega_{\text{free}}$. The length of the path $s(x, y)$, denoted $d(x, y)$, is called the* intrinsic distance *between x and y. The* intrinsic diameter *of Ω_{free} is $\text{diam}(\Omega_{\text{free}}) = \max_{x,y\in\Omega_{\text{free}}} d(x, y)$.*

Remark 1 *We sometimes use subscripts to indicate the domain in which a path is shortest, e.g., $s_\Omega(x, y)$, or a path along which distance is measured, e.g., $d_s(x, y)$.*

3. Two Pursuers in a Simply Connected Arena.

In this section, we show that two pursuers are sufficient to capture an evader in a simply connected arena. (The simply connected topology implies that $\Omega = \Omega_{\text{free}}$.) We establish bounds for the capture time that are asymptotically optimal. We do so by designing a joint pursuit strategy we call *orthographic pursuit*. Our main result is that with orthographic pursuit, two pursuers can capture an evader of equal speed in a simply connected domain Ω in time $\mathcal{O}(\text{diam}(\Omega))$, which is clearly asymptotically optimal. We divide the presentation into two parts: one where the arena is convex, and another where no assumption is made beyond simply connectedness. We do so for two reasons: first, the exposition of orthographic pursuit is easier for convex arenas; second, the capture time bound is smaller when the arena is convex.

We introduce some definitions that will be used throughout this section. The *x-length* of a path $s \subset \Omega$ is $\int_s |dx|$, where dx is a measure of displacement along the x-coordinate. In other words, if we subdivide s into a set of x-monotone paths, the x-length of s is the sum of the horizontal widths of those paths. The *x-distance* $d_x(v, w)$ between two points $v, w \in \Omega$ is the minimum x-length among all paths connecting v to w in Ω. Define the *y-length* and the *y-distance* $d_y(v, w)$ analogously. It is straightforward to show that the intrinsic shortest path between two points in a simply connected domain also has the shortest x-length and the shortest y-length. (This is not true of a domain with obstacles.)

3.1. A Convex Arena.

We begin by explaining how to defend a line segment (a particular type of shortest path). Defending a line segment is simple: the pursuer need only stand at the point on the segment that is closest to the evader, and track that closest point as it moves along the segment. There are at least two different ways to define "closest point": we could use the Euclidean distance or the intrinsic distance. In a convex domain, the Euclidean and intrinsic distances between two points are always the same; but in a nonconvex domain, the intrinsic distance becomes more interesting. We discuss it further in Section 3.2.

If we choose the closest point by Euclidean distance, then regardless of the type of domain, the closest point is the orthogonal projection of the evader onto the line if that projection lies on the line segment; or an endpoint of the segment otherwise. Note that the closest-point operation onto a line segment is itself a projection (i.e., the projection of a projection is the same as the projection) called the *Euclidean projection*, which is not always the same as the orthogonal projection (unless the line segment is a line). In any domain, a pursuer can defend the evader's Euclidean closest point on a line segment, because the projection of the evader's velocity onto the line has magnitude at most 1.

Figure 2 illustrates two pursuers P_1 and P_2 pursuing an evader E. P_1 is defending a line segment AB by tracking the projection of E onto AB, and P_2 is defending a line segment CD (orthogonal to AB) by tracking the projection onto CD. P_2 achieves this by projecting E's velocity onto CD and matching that velocity. In the example at left, E's velocity is parallel to CD with magnitude 1, and so is P_2's velocity. Consequently, E can never cross CD without getting captured by P_2.

3.1.1. Orthographic Pursuit in a Convex Arena. Of course, just defending CD does not suffice to capture E. If E always moves parallel to CD, then so does P_2, and hence P_2 can never capture E. In our orthographic pursuit strategy, a second pursuer P_1 is defending a line segment AB orthogonal to CD. At any time, at least one of the two pursuers can make progress toward capture by *advancing* its line segment toward the invader. For example, in Figure 2, left, E's velocity is orthogonal to AB, so P_1 can move toward E at full speed. When it does so, the line segment AB is not fixed in place; rather, it advances toward E. In other words, we are not defining AB to be a fixed line segment (though its orientation is fixed); rather, at any point in time, we define AB to be the line segment currently passing through P_1 with its endpoints on the domain boundary.

A second example appears at right in Figure 2. In this example, E's velocity is parallel to neither AB nor CD. P_2 projects E's velocity onto CD and matches that velocity; then uses the leftover speed to advance CD toward E, so the velocity of P_2 has magnitude 1. P_1 does the same with AB. Consequently, P_1 and P_2 are simultaneously defending AB and CD and advancing the line segments toward E.

Observe that P_1 can guarantee that E will never be able to cross AB and P_2 can guarantee that E will never be able to cross CD. Moreover, at any given time, at least one of the two line segments is advancing toward E. Hence E is guaranteed to be captured.

A special case arises when a pursuer lies on the domain boundary $\partial\Omega$ and either the evader's orthogonal projection onto its line is outside the domain or the evader's motion would cause the pursuer to try to walk outside the domain. For example, in Figure 3 the orthogonal projection of E onto the line through AB does not lie on AB, and the Euclidean projection of E onto AB is the endpoint A. In this case, P_1 moves along the boundary $\partial\Omega$, thereby advancing the line while sufficing to ensure that E will not be able to cross AB. Before E could cross AB, first E's Euclidean projection onto AB must coincide with the endpoint A, whereupon P_1 would resume coinciding with E's projection.

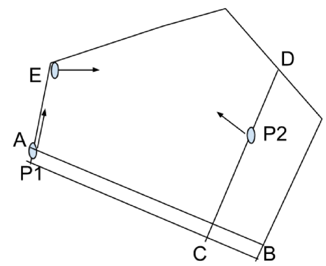

FIG. 3. *A demonstration of the orthographic pursuit strategy in a convex, simply connected domain. P_1 is at the boundary. In this case, the evader's Euclidean projection onto the line through AB is outside the domain, so P_1 moves along the boundary to guarantee that E will not be able to cross AB without getting captured. P_2 is again moving according to the same strategy.*

At the very beginning of pursuit, each pursuer is probably not defending a line segment: the evader's Euclidean projection onto the line segment may not coincide with the initial position of the pursuer. Hence the pursuit strategy begins by letting each pursuer sweep along its line segment until the pursuer coincides with the evader's Euclidean projection on the line segment. From that moment on, the pursuer defends its line segment.

Algorithm 1 in Figure 4 specifies our orthographic pursuit strategy. We divide it into two phases: in Phase I the pursuers move to start defending their line segments, and in Phase II they defend their line segments while advancing them. As written, the two pursuers enter Phase II simultaneously; but in practice, the two pursuers could enter Phase II asynchronously, perhaps speeding up the capture.

Since Euclidean projection is a continuous operator and the boundary of the domain is piecewise smooth, when the evader's control is piecewise continuous, the resulting control from orthographic pursuit for each pursuer is also piecewise smooth. Consequently, orthographic pursuit is admissible.

3.1.2. Upper Bound on Capture Time. We show that if the domain is convex, the worst-case capture time of the orthographic pursuit strategy is $4 \cdot \mathrm{diam}(\Omega)$. To proceed, let w_{1y} be the width of the range of y-coordinates that the pursuer p_1 has visited; that is, the largest y-coordinate p_1 has visited minus the smallest y-coordinate. Symmetrically, let w_{2x} be the width of the range of x-coordinates that the pursuer p_2 has visited. Observe that w_{1y} and w_{2x} are initially zero, never decrease, and can never exceed $\mathrm{diam}(\Omega)$.

Lemma 3 *If Ω is convex, then at any time when p_1 is moving along $\partial\Omega$ in a direction that is not precisely vertical or precisely horizontal, p_1 is visiting y-coordinates it has not previously visited, so w_{1y} is increasing.*

Proof Consider an instant in time when p_1 is moving along $\partial\Omega$. Suppose without loss of generality that p_1 is moving up and to the right; i.e., both its x- and y-coordinates are increasing. Let t be the line through p_1 parallel to p_1's direction of motion; hence t is tangent to a piece of $\partial\Omega$ at p_1. The fact that t is not precisely vertical implies that p_1 cannot move straight up without leaving Ω. Because Ω is convex, the interior of Ω lies entirely below t. The fact that p_1 is not moving precisely horizontally implies that e has a greater y-coordinate than p_1. The x-coordinate p_{1x} is always monotonically nondecreasing, and p_1 has never been above the line t, so p_1 is visiting y-coordinates it has not previously visited. ■

Algorithm 1 Orthographic Pursuit in a Convex, Simply Connected Domain

1: Choose an arbitrary coordinate system.
2: **Phase I:**
3: Direct p_1 to defend the vertical line (relative to the coordinate system) that passes through p_1^0; and direct p_2 to defend the horizontal line that passes through p_2^0.
4: **Phase II:**
5: Project evader's velocity onto the vertical line and set p_1's vertical velocity to be the same; i.e., $\dot{p}_{1y} = \dot{e}_y$. Choose the horizontal velocity $\dot{p}_{1x}$ so that $\|\dot{p}_1\|_2 = 1$, with the sign of $\dot{p}_{1x}$ chosen so the vertical line moves toward the evader.
6: **if** $p_1 \in \partial\Omega$ **and** ($p_{1y} \neq e_y$ **or** motion in direction $\dot{p}_1$ immediately exits Ω) **then**
7:     Set p_1's velocity $\dot{p}_1$ to trace $\partial\Omega$ with magnitude 1, with the direction chosen so the vertical line moves toward the evader.
8: **end if**
9: Project evader's velocity onto the horizontal line and set p_2's horizontal velocity to be the same; i.e., $\dot{p}_{2x} = \dot{e}_x$. Choose the vertical velocity $\dot{p}_{2y}$ so that $\|\dot{p}_2\|_2 = 1$, with the sign of $\dot{p}_{2y}$ chosen so the horizontal line moves toward the evader.
10: **if** $p_2 \in \partial\Omega$ **and** ($p_{2x} \neq e_x$ **or** motion in direction $\dot{p}_2$ immediately exits Ω) **then**
11:     Set p_2's velocity $\dot{p}_2$ to trace $\partial\Omega$ with magnitude 1, with the direction chosen so the horizontal line moves toward the evader.
12: **end if**

FIG. 4. *Algorithm 1.*

By symmetry, Lemma 3 applies also with p_1 and w_{1y} replaced by p_2 and w_{2x}.

Theorem 4 *With the orthographic pursuit strategy, two pursuers capture an evader of equal speed in a convex domain Ω in time at most $4 \cdot \mathrm{diam}(\Omega)$.*

Proof Let p_1^0 and p_2^0 be the initial positions of p_1 and p_2, and consider the objective $\phi = d_x(p_1^0, p_1) + d_y(p_2^0, p_2) + w_{1y} + w_{2x}$. Initially, all four terms are zero, and so is ϕ. None of the four terms ever decreases. None of the four terms can ever exceed $\mathrm{diam}(\Omega)$, so $\phi \leq 4 \cdot \mathrm{diam}(\Omega)$.

When p_1 initially moves to defend its line, $\dot{w}_{1y} = 1$; once p_1 reaches the closest-point projection of e onto its line, it will stay at e's projection thereafter. Likewise, when p_2 initially moves to defend its line, $\dot{w}_{2x} = 1$. Hence, $\dot{\phi} \geq 1$ throughout Phase I.

If one of the pursuers is moving along $\partial\Omega$ in a direction that is not precisely vertical or precisely horizontal, suppose without loss of generality that pursuer is p_1. By Lemma 3, the rate of change of $d_x(p_1^0, p_1) + w_{1y}$ is $|\dot{p}_{1x}| + |\dot{p}_{1y}| \geq 1$, so $\dot{\phi} \geq 1$.

Otherwise, both pursuers are defending the evader's projection onto their respective lines, and neither pursuer is moving diagonally along the boundary, so p_{1y} is tracking e_y and p_{2x} is tracking e_x. It follows that the rate of change of $d_x(p_1^0, p_1) + d_y(p_2^0, p_2)$ is at least $|\dot{p}_{1x}| + |\dot{p}_{2y}| \geq \dot{p}_{1x}^2 + \dot{p}_{2y}^2 = 2 - \dot{p}_{1y}^2 - \dot{p}_{2x}^2 = 2 - \dot{e}_y^2 - \dot{e}_x^2 \geq 1$. In every case, $\dot{\phi} \geq 1$, so ϕ will reach $4 \cdot \mathrm{diam}(\Omega)$ in time at most $4 \cdot \mathrm{diam}(\Omega)$. By that time (or sooner), both $d_x(p_1^0, p_1)$ and $d_y(p_2^0, p_2)$ have reached their maximum possible values, so the evader is captured. ■

3.2. A General Simply Connected Arena. In our pursuit algorithm for simply connected arenas, a pursuer defends a line segment not by tracking the evader's Euclidean closest point on the line segment, but rather by tracking the evader's *in-*

trinsic closest-point on the line segment.

Definition 5 *Given a shortest path $s \subset \Omega$ and an evader at a point $e \in \Omega$, The* intrinsic closest point *to e on s is the point $\bar{e} \in s$ that minimizes the intrinsic distance $d(e, \bar{e})$.*

If the domain is simply connected, then the intrinsic closest point to an evader on a line segment moves continuously with speed at most 1, so again a pursuer can defend the intrinsic closest point. However, in a domain with obstacles, the intrinsic closest point to the evader can change discontinuously, so following the closest point is not a valid strategy.

The intrinsic closest-point projection can be viewed as the right generalization of the Euclidean projection for a general simply connected domain: in a convex domain, whenever the evader's Euclidean projection onto a line segment lies inside the domain, that projection is also the intrinsic closest-point projection. In a general simply connected domain, tracking the evader's intrinsic closest-point projection is sufficient to defend a path, as formalized by the following lemma (whose proof is in the appendix).

Lemma 6 *Let ℓ be a line segment included in a simply connected domain Ω. A pursuer tracking the evader's intrinsic closest-point projection defends it.*

3.2.1. Orthographic Pursuit in a Simply Connected Arena. Orthographic pursuit in a general simply connected arena is similar to orthographic pursuit in a convex domain, with one important difference: a pursuer uses the intrinsic closest-point projection rather than the Euclidean projection in choosing its velocity.

As before, one pursuer, p_1, always tries to defend the vertical line segment that passes through p_1 from one wall to another. The second pursuer, p_2, always tries to defend the horizontal line segment through p_2. Both pursuers follow the evader's intrinsic closest-point projection onto their respective paths. They do not stick to stationary paths; both pursuers try to advance the paths they are defending so the evader becomes cornered.

At a fixed point in time, let $\ell_1 \subset \Omega$ be the longest vertical line segment through p_1 in Ω. The top priority of p_1 is to move along ℓ_1 until p_1 lies at the intrinsic closest-point projection e_1 of the evader e onto ℓ_1, and then to stay on the projection as it moves along ℓ_1. Subject to this priority, p_1 also tries to advance the line segment ℓ_1 so that it sweeps through the domain and forces the evader into a corner. The second pursuer p_2 follows the same strategy, except that p_2 always defends a horizontal line. Each pursuer always moves with a speed of 1, so any leftover speed not needed to follow the evader's projection is used to advance the line.

A key observation about this strategy in a simply connected domain is that once p_1 has reached the evader's projection on ℓ_1, the evader can never again cross ℓ_1 without being captured, even after the line p_1 is defending has advanced. Therefore, ℓ_1 slices Ω into two portions, one of which the evader can never visit again. When possible, p_1 moves to enlarge the latter portion and shrink the portion containing e. Sometimes p_1 advances its line to a "corner-turn" where the boundary $\partial\Omega$ is not x-monotone and the line ℓ_1 suddenly lengthens, as illustrated in the third and fifth drawings in Figure 5. At that moment, p_1 might no longer lie at the evader's projection on the newly lengthened ℓ_1, and therefore ℓ_1 is not properly defended until p_1 moves to the new projection. Fortunately, the evader cannot cross the "old" portion of ℓ_1, so no progress is lost. While p_1 is moving to the new projection e_1, e may cross the new portion of ℓ_1, but once p_1 reaches e_1 the evader must have chosen one side of ℓ_1, which determines the direction in which p_1 will continue to advance its line.

Consider p_1's strategy when it is at e_1. Let $s \subset \Omega$ be the intrinsic shortest path

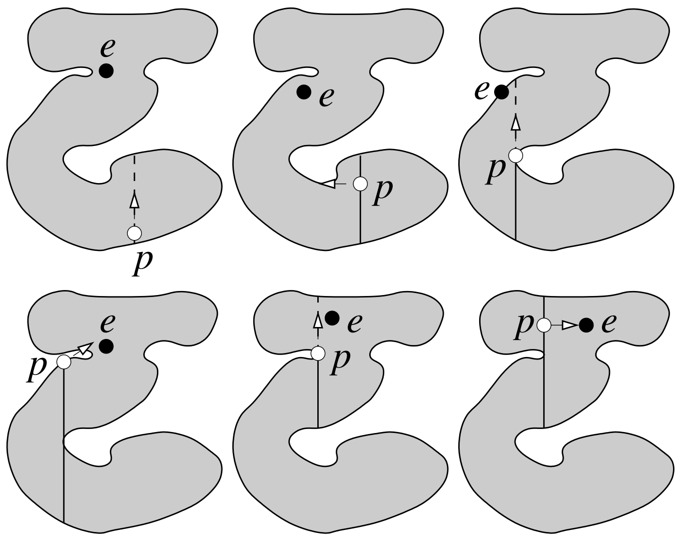

FIG. 5. *One pursuer's (p_1) role in the orthographic pursuit strategy.*

from e to ℓ_1. Then $p_1 = e_1 = s \cap \ell_1$ is the intrinsic closest-point projection of e onto ℓ_1. Because s is the shortest path, either s meets ℓ_1 at a right angle or e_1 is an endpoint of ℓ_1 (or both). In the former case, the most common action is that p_1 moves vertically at the same speed as e_1—call that speed $\dot{p}_{1y}$—and horizontally at a speed $\dot{p}_{1x}$, where $\dot{p}_{1x}^2 + \dot{p}_{1y}^2 = 1$. However, if e_1 is an endpoint of ℓ_1, then $p_1 = e_1 \in \partial\Omega$, and a movement by p_1 that matches the y-velocity of e may exit the domain Ω, in which case we readjust $\dot{p}_1$ to follow the boundary $\partial\Omega$; then its velocity is tangent to (a piece of) the boundary.

In the case where the path s does not meet ℓ_1 at a right angle, again $p_1 = e_1 \in \partial\Omega$, and again we move p_1 along the boundary $\partial\Omega$. This occurs in the fourth drawing in Figure 5 (and also sometime between the second and third drawings). Algorithm 2 in Figure 6 provides a formal description of the pursuit strategy in the above discussion.

Here we do not divide the pursuit strategy into two phases as we did in Algorithm 1, due to the nature of a nonconvex domain. Even after each pursuer has reached a projection of e and started defending its line segment, e's intrinsic closest-point projection can still suddenly jump because the line segment suddenly lengthens. (This never occurs in a convex domain.) In that case, a pursuer has to retreat to "Phase I," moving to a position where it can again start defending its line segment.

3.2.2. Capture Time Bound. Since intrinsic closest-point projection is a continuous operator, when the evader's control is piecewise continuous, the control from orthographic pursuit for each pursuer is also piecewise smooth. Consequently, orthographic pursuit in a general simply connected domain is an admissible pursuit strategy.

Theorem 7 *With the orthographic pursuit strategy, two pursuers capture an evader of equal speed in a simply connected domain Ω in time at most $(4 + 2\sqrt{2}) \cdot \mathrm{diam}(\Omega) \doteq 6.828 \cdot \mathrm{diam}(\Omega)$.*

Proof Let p_1^0 be the initial position of p_1. Let p_1 denote the position of pursuer p_1 at an arbitrary, current moment in time. Because p_1 defends a vertical path, advances the path horizontally, and never allows the path to retreat to a prior position,

Algorithm 2 Orthographic Pursuit in a Simply Connected Domain

Choose an arbitrary coordinate system.

Let ℓ_1 be the vertical line segment defended by p_1 and let ℓ_2 be the horizontal line segment defended by p_2. Let e_1 and e_2 be the intrinsic closest-point projections of e onto ℓ_1 and ℓ_2, respectively.

if e_1 does not coincide with p_1 **then**
    p_1 moves along ℓ_1 with speed 1 toward e_1.
else
    If the shortest path from p_1 to e is a horizontal line segment, let $\dot{p}_{1y} = \dot{e}_y$; otherwise, let $\dot{p}_{1y} = 0$.
    Choose the horizontal velocity $\dot{p}_{1x}$ so that $\|\dot{p}_1\|_2 = 1$, with the sign of $\dot{p}_{1x}$ chosen so the vertical line moves into the shortest path from p_1 to e.
    if $p_1 \in \partial\Omega$ **and** (the shortest path from p_1 to e does not meet p_1 at a right angle **or** motion in direction $\dot{p}_1$ immediately exits Ω) **then**
        Set p_1's velocity $\dot{p}_1$ to trace $\partial\Omega$ with magnitude 1, with the direction chosen so the vertical line moves into the shortest path from p_1 to e.
    end if
end if
if e_2 does not coincide with p_2 **then**
    p_2 moves along ℓ_2 with speed 1 toward e_2.
else
    If the shortest path from p_2 to e is a vertical line segment, let $\dot{p}_{2x} = \dot{e}_x$; otherwise, let $\dot{p}_{2x} = 0$.
    Choose the vertical velocity $\dot{p}_{2y}$ so that $\|\dot{p}_2\|_2 = 1$, with the sign of $\dot{p}_{2y}$ chosen so the horizontal line moves into the shortest path from p_2 to e.
    if $p_2 \in \partial\Omega$ **and** (the shortest path from p_2 to e does not meet p_2 at a right angle **or** motion in direction $\dot{p}_2$ immediately exits Ω) **then**
        Set p_2's velocity $\dot{p}_2$ to trace $\partial\Omega$ with magnitude 1, with the direction chosen so the horizontal line moves into the shortest path from p_2 to e.
    end if
end if

FIG. 6. *Algorithm 2.*

the x-length of the path p_1 has trodden so far is the minimum possible, namely, $d_x(p_1^0, p_1)$. However, this is not true of the y-length of p_1's path; in defending its line, p_1 may have done quite a bit of backtracking along the y-axis to follow e's projection. Symmetrically, the path that p_2 traces always has a y-length of $d_y(p_2^0, p_2)$, the minimum y-length possible, but there is no such guarantee for its x-length.

No two points in Ω can be separated by an x-distance greater than $\operatorname{diam}(\Omega)$, nor by a y-distance greater than $\operatorname{diam}(\Omega)$. Therefore, if either $d_x(p_1^0, p_1)$ or $d_y(p_2^0, p_2)$ grows as great as $\operatorname{diam}(\Omega)$, the evader is captured.

Consider the objective function $\phi = d_x(p_1^0, p_1) + d_y(p_2^0, p_2) - d_y(e, p_1) - d_x(e, p_2)$. Initially, the first two terms are zero and the last two terms have magnitude at most $\operatorname{diam}(\Omega)$, so initially $\phi \geq -2 \cdot \operatorname{diam}(\Omega)$. The terms $d_y(e, p_1)$ and $d_x(e, p_2)$ are nonnegative, so the evader is captured by the time ϕ reaches $2 \cdot \operatorname{diam}(\Omega)$.

The rate of change of $d_x(p_1^0, p_1)$ is the x-speed $|\dot{p}_{1x}|$ of pursuer p_1, and the rate of change of $d_y(p_2^0, p_2)$ is the y-speed $|\dot{p}_{2y}|$ of pursuer p_2. The rate of change of $d_y(e, p_1)$ is $-|\dot{p}_{1y}| \pm |\dot{e}_y|$, and the rate of change of $d_x(e, p_2)$ is $-|\dot{p}_{2x}| \pm |\dot{e}_x|$. Therefore, the rate of change of ϕ is $\dot{\phi} = |\dot{p}_{1x}| + |\dot{p}_{1y}| + |\dot{p}_{2x}| + |\dot{p}_{2y}| \pm |\dot{e}_x| \pm |\dot{e}_y| \geq 2 - \sqrt{2}$, and the evader is captured in time at most $4 \cdot \mathrm{diam}(\Omega)/(2 - \sqrt{2}) = (4 + 2\sqrt{2}) \cdot \mathrm{diam}(\Omega)$. ■

4. Three Pursuers in a General Arena: A Divide-and-Conquer Pursuit Strategy. In this section we propose a *divide-and-conquer strategy* for three pursuers chasing an evader of equal speed in a domain Ω_{free} with obstacles. We start by describing a strategy for defending a shortest path in a general arena in Section 4.1 and some concepts related to game subdomains in Section 4.2. Both serve as important building blocks in our pursuit strategy discussed in Section 4.3. We provide a demonstration of the pursuit strategy in action in Section 4.4.

4.1. Defending a Shortest Path via Level-Set Projection. Unlike in the previous section, neither a Euclidean projection nor an intrinsic closest-point projection provides the correct vehicle to defend an arbitrary shortest path in a general arena. Instead, we use a level-set projection.

Definition 8 *Given a shortest path $s \subset \Omega_{\text{free}}$ with two endpoints x, y and an evader $e \in \Omega_{\text{free}}$, the* level-set projection *of e onto s is the point $\bar{e} \in s$ such that $d(x, \bar{e}) = d(x, e)$, if such a point exists. No such point exists if $d(x, y) < d(x, e)$, in which case the level-set projection is defined to be y.*

Remark 2 *Level-set projection is closely related to the concept of level sets: Given a point $x \in \Omega_{\text{free}}$ and a real value $\alpha \geq 0$, the α-*level set *of the distance function $d(x, \cdot)$ is the set of all points whose shortest path to x has length α. The level sets are piecewise smooth curves. A pursuer p tracking an evader's level-set projection tries to stay on the same level set as the evader, subject to the constraint that p must stay on the path.*

We then have following result. (The proof is in the appendix.)

Lemma 9 *For any two points $x, y \in \Omega_{\text{free}}$, let $s(x, y) \subset \Omega_{\text{free}}$ be a shortest path connecting x and y. Then $s(x, y)$ is defendable, and a pursuer tracking the evader's level-set projection defends it.*

Remark 3 *Consider two different points on a path that are different projections of the same evader; for instance, one point might be the intrinsic closest-point projection, and the other might be the level-set projection. Then all the points on the interval between those projections are projections too. Therefore, a pursuer can switch from shadowing one type of projection to another type at any time without allowing the evader to cross s.*

The following question will be useful in analyzing the capture time later: Given an arbitrary shortest path s of length λ, how long (in worst case) does it take for a pursuer to start defending it? Our usual procedure is to first move to the path's midpoint (by length), taking time at most $\mathrm{diam}(\Omega_{\text{free}})$, then move to the evader's projection $\bar{e}$, taking time at most $\lambda/2$. Typically this is fast, but we will see that occasionally λ is much longer than the diameter $\mathrm{diam}(\Omega_{\text{free}})$. In the worst case, it is hard to improve this time: the evader could wait at the midpoint until the pursuer comes arbitrarily close, then run along the path. However, if a second pursuer is free, the following *interval trap* takes time at most $2 \cdot \mathrm{diam}(\Omega_{\text{free}})$. Let $a, b \in s$ be the points at a distance (measured along s) of $\mathrm{diam}(\Omega_{\text{free}})$ from the initial projection $\bar{e}$ in each direction, taking endpoints of s if they are closer. In time at most $\mathrm{diam}(\Omega_{\text{free}})$, two pursuers can move to a and b; then in time at most $\mathrm{diam}(\Omega_{\text{free}})$, they can move along s until one of them meets $\bar{e}$. That pursuer continues defending s, while the other pursuer is free to leave.

4.2. Game Subdomains. The high-level summary of the divide-and-conquer strategy is as follows. Pursuers defend paths that subdivide the game domain into subdomains. While two pursuers prevent the evader from escaping a subdomain, the third pursuer subdivides it further, eventually freeing one of the first two pursuers to do the same in turn. Consequently, a game subdomain is an important object that runs through the design of the pursuit strategy.

Definition 10 *A closed point set* Ψ *is a* game subdomain *of a game domain* $\Omega = \Omega_{\text{free}} \cup \omega$ *if* $\Psi \subseteq \Omega$ *and its boundary* $\partial\Psi$ *is a piecewise smooth Jordan curve of finite length with* $\partial\Psi \subset \Omega_{\text{free}}$*. A game subdomain inherits the obstacles* $\psi = \bigcup_{\omega_i \subseteq \Psi} \omega_i$ *and the free space* $\Psi_{\text{free}} = \Psi \setminus \psi \subseteq \Omega_{\text{free}}$*.*

A typical game subdomain is obtained by choosing a shortest path that connects two distinct points on $\partial\Omega$, thereby dividing Ω into two or more pieces, each of which may include some of the original obstacles. Each piece is a game subdomain of Ω, and a domain in its own right. The pursuers defend subdomains of a particular configuration, illustrated in Figure 7. Let $\Omega = \Omega_{\text{free}} \cup \omega$ be a game domain, and let $\Psi = \Psi_{\text{free}} \cup \psi$ be a game subdomain of Ω. The subdomain Ψ is of the *triangle configuration* if there are three points a, b, and c on $\partial\Psi$ that subdivide $\partial\Psi$ into three paths s_1, s_2, and w that are disjoint except at their endpoints, such that s_1 is a shortest path from a to b in Ψ_{free}, s_2 is a shortest path from a to c in Ψ_{free}, and w is a natural boundary path included in $\partial\Omega_{\text{free}}$. (Some of these paths may have length zero.) If the evader is in a game subdomain Ψ of the triangle configuration and two pursuers are defending s_1 and s_2, then by Lemma 9, s_1 and s_2 are defendable, even if they are not shortest paths in Ω_{free}; it suffices that they are shortest paths in Ψ_{free}. (If the evader could escape Ψ, those paths might no longer be defendable.) It is possible that b and c are the same point, and there are two different shortest paths from that point to a.

A path s *touches* an obstacle ψ_i if the relative interior of s intersects $\partial\psi_i$. (The endpoints of s do not count.) A *touching point* is a point in the intersection. A path w is a *natural boundary path* if w is a connected subset of $\partial\Omega_{\text{free}}$. The intersection of a path s with an obstacle can be one point, a path, or a union of paths and points. While two pursuers defend the boundary of Ψ, the third pursuer subdivides a subdomain by choosing a shortest path within the subdomain and defending it. To ensure that this shortest path is not identical to a path defended by the other two pursuers, we modify the free space to remove points where a defended path touches an obstacle. We thereby connect such obstacles to the infinite space outside Ψ.

Definition 11 *The* blocked free space *of a subdomain* Ψ_{free} *whose boundary includes the defended paths* $s_1, s_2 \subset \partial\Psi$ *is* $\Psi_{\text{block}} = \Psi_{\text{free}} \setminus (\partial\psi \cap (s_1 \cup s_2))$*.*

In general, Ψ_{block} is a neither closed nor open point set. In Figure 7, Ψ_{block} lacks points that Ψ_{free} possesses where the obstacles touch s_1 and s_2. The removal of these points prevents any new shortest path from a to b from taking the same path as s_1. (Obstacles outside Ψ do not affect Ψ_{block}.)

4.3. A Divide-and-Conquer Pursuit Strategy. We now present our divide-and-conquer pursuit strategy for three pursuers. The strategy repeatedly subdivides a game subdomain of the triangle configuration so the evader is trapped in a smaller game subdomain of the triangle configuration, each time reducing the number of obstacles or increasing the number of obstacles touched by the defended paths. Suppose p_1 and p_2 are defending s_1 and s_2, respectively. There are three cases.

1. The subdomain Ψ encloses no obstacle.
2. Either s_1 or s_2 touches an obstacle in Ψ, as in Figure 7.

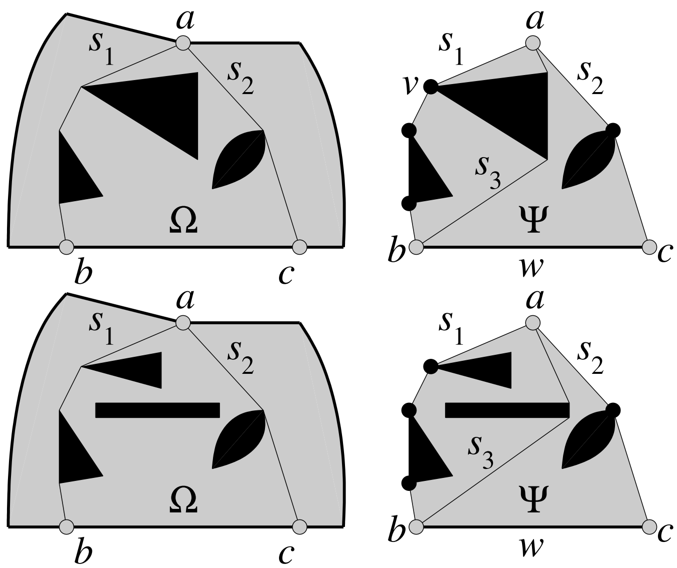

FIG. 7. *The upper left drawing shows the original domain Ω, where pursuer p_1 is defending path s_1 and pursuer p_2 is defending path s_2. The upper right drawing shows the subdomain Ψ bounded by s_1, s_2, and a natural boundary path w connecting b to c. In this subdomain, the shortest path between a and b is s_3, which touches an obstacle s_1 already touches. Note that the blocked free space Ψ_{block} (i.e., the free space in Ψ) omits the points where the obstacles in Ψ meet s_1 and s_2: this is why the shortest path s_3 between a and b in Ψ_{block} takes a different path from s_1. The bottom two drawings show another domain Ω (and Ψ), where taking the shortest path s_3 between a and b therein touches an obstacle that neither s_1 nor s_2 touches.*

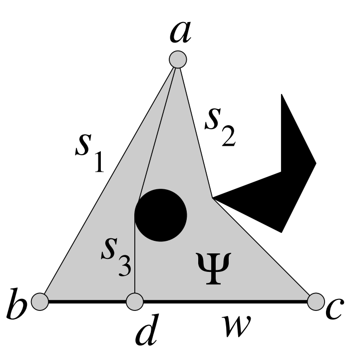

FIG. 8. *Case 3. Neither s_1 nor s_2 touches an obstacle in Ψ, and the blocked free space is the same as the free space. The point d is chosen on w so that the shortest path between a and d touches an obstacle.*

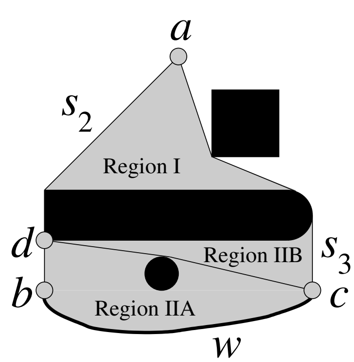

FIG. 9. *A subdomain whose blocked free space has more than one connected component.*

3. Ψ includes one or more obstacles, but none touches s_1 or s_2, as in Figure 8. (As the figure suggests, obstacles outside Ψ do not count.)

In case 1, the third pursuer executes direct pursuit. Direct pursuit means that at any given point, the pursuer always moves towards the evader at full speed 1. As mentioned in the introduction, it is known that direct pursuit with a single pursuer is guaranteed to capture the evader with any positive capture radius. In Section 5.1.1 we will replace direct pursuit with a better endgame strategy that both is faster and guarantees capture with capture radius zero. However, for simplicity, we settle for direct pursuit here.

In case 2, suppose without loss of generality that s_1 touches at least one obstacle. We find the shortest path s_3 between a and b in Ψ_{block}, as illustrated in Figure 7. In Section 5.1.2, we show (Lemma 12) that s_3 touches at least one obstacle in Ψ. Although s_3 is probably not a shortest path in Ψ_{free} (as s_1 is), it is a shortest path in Ψ_{block}, and therefore is defendable so long as the evader is trapped in Ψ_{block}. We direct p_3 to defend s_3. Once p_3 reaches the evader's projection on s_3 and is defending s_3, the evader cannot cross s_3, which splits Ψ into two subdomains of the triangle configuration and frees one of the pursuers p_1 or p_2 to defend a new path.

In case 3, we use the following *path shooting procedure* to find a point d on w such that the shortest path s_3 between a and d in Ψ_{free} touches an obstacle, as illustrated in Figure 8. Let q be any point in the closure of ψ. Let $s_\Psi(a, q) \subset \Psi$ be the shortest path from a to q in Ψ. (We emphasize that this is a path in Ψ, not in Ψ_{free}, so the path can pass through an obstacle's interior.) If q lies on the boundary of Ψ, let $d = q$; otherwise, observe that a sufficiently small neighborhood of q in $s_\Psi(a, q)$ is a straight line segment, and let $s_\Psi(a, d)$ be the path found by extending that line segment until it strikes a point d on the boundary of Ψ. The point d always lies on the natural boundary path w, by the following reasoning. In case 3, neither s_1 nor s_2 touches any obstacle inside Ψ, so both paths can bend outward (turning away from the domain) but can nowhere bend inward. Hence the point d does not lie on s_1 nor on s_2; it must lie on w. In Section 5.1.2 we show (Lemma 14) that $s_\Psi(a, d)$ is a shortest path and the related shortest path $s_3 = s_{\Psi_{\text{free}}}(a, d)$ touches an obstacle (albeit not necessarily the same obstacle that touches q). The path s_3 also splits Ψ into two subdomains of the triangle configuration and frees one of the pursuers p_1 or p_2.

In either case 2 or case 3, if the new, smaller subdomain has a blocked free space with two or more connected components, as illustrated in Figure 9, then we reduce the subdomain to the connected component that contains the evader. If the evader is trapped in the connected component that contains a, labeled Region I in Figure 9, then Region I becomes the new subdomain; we shorten the pursuer's paths accordingly and iterate. If the evader is in a connected component of a quadrangle configuration like Region II, the freed pursuer moves to defend the shortest path in Region II between the points labeled c and d in Figure 9. Once the pursuer has reached the evader's projection on $s(c, d)$ and is defending that path, the evader is in Region IIA or Region IIB. Either region is of the triangle configuration, and frees one of the pursuers to defend a new path. Regardless of which connected component the evader is in, any obstacle that touches two of s_1, s_2, and s_3 is not inside the new subdomain. To bootstrap the pursuit, we simply choose a point $a = b = c$ on the boundary of the ambient space Ω as a degenerated triangle configuration. The discussion above is summarized in Algorithm 3 in Figure 10.

4.4. Demonstration of the Capture Strategy. In this section we demonstrate how the proposed pursuit strategy may be applied to a general game domain. The example we use is the domain shown in Figure 11. In each step, one of the pursuers acts to partition the game domain into subdomains, one of which contains the evader. This strategy is iterated until the evader is trapped within a simply connected subdomain. Then the evader is captured by a pursuer using direct pursuit.

In the first drawing in Figure 11, a defendable path is established between points a and b by p_1, partitioning the domain into three subdomains: two simply connected regions and a third containing two obstacles. Note that this effectively removes the

Algorithm 3 Divide-and-Conquer Pursuit in a General Arena

Initialize the game subdomain $\Psi = \Omega$.
Initialize $a = b = c$. Set the shortest paths $s_1 = ab$ and $s_2 = ac$ to be the paths defended by two pursuers.
while Ψ_{free} is not simply connected **do**
  if either s_1 or s_2 touches an obstacle in Ψ **then**
    Find a shortest path s_3 between a and b (or between a and c) in Ψ_{block}. The path s_3 splits Ψ into two subdomains.
    Direct the third, free pursuer to defend s_3.
    Once s_3 is defended, update Ψ to be the subdomain that contains the evader. Free one of the two pursuers on s_1 and s_2 accordingly. Rename s_3 to be either s_1 or s_2 defending on which path is no longer needed.
  else
    Find a point $d \in w$ such that the shortest path $s_3 \in \Psi_{\text{free}}$ between a and d touches an obstacle.
    Direct the third, free pursuer to defend s_3.
    Once s_3 is defended, update Ψ to be the subdomain that contains the evader. Free one of the two pursuers on s_1 and s_2 accordingly. Rename s_3 to be either s_1 or s_2 defending on which path is no longer needed.
  end if
end while
The free pursuer executes direct pursuit.

Fig. 10. *Algorithm 3.*

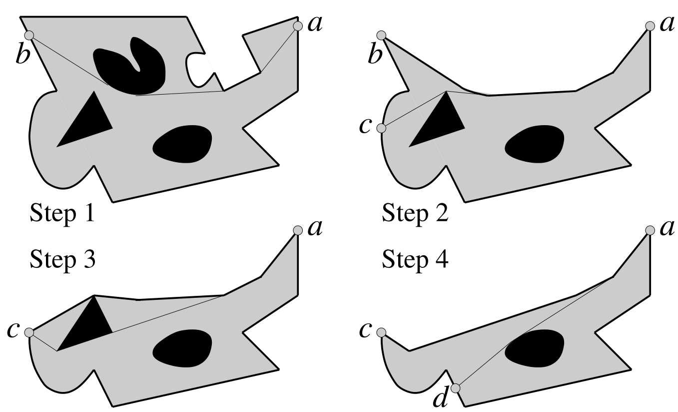

Fig. 11. *Demonstration of the divide-and-conquer pursuit strategy.*

large irregular obstacle from the game, as it touches the boundary. In the second drawing, p_2 establishes a defendable path between a and a point c, selected according to the procedure outlined in the forthcoming Lemma 14 and freeing p_1. In the third drawing, a second path between a and c established by p_1 partitions the resulting subdomain, removing the triangular obstacle. If the evader is not in one of the simply connected regions, p_2 is free to establish a path between a and a new point d, removing

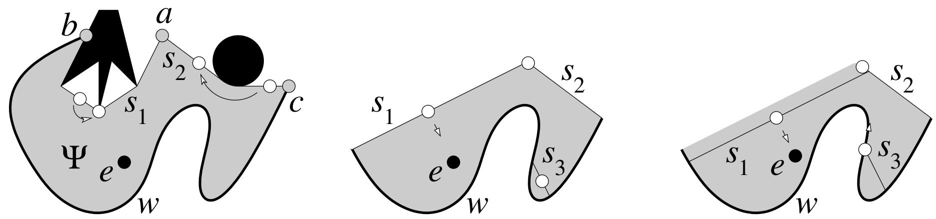

FIG. 12. *The endgame strategy. At left, pursuers p_1 and p_2 move from the evader's level-set projections to the evader's intrinsic closest-point projections. At center, the two pursuers switch to defending straight line segments. The third pursuer arrives in the shrunken domain. At right, pursuers p_1 and p_3 carry out the orthographic pursuit strategy while p_2 continues to defend a fixed path s_2.*

the elliptical obstacle, as in the fourth drawing.

5. Analysis of the Divide-and-Conquer Pursuit Strategy. In this section, we establish the correctness and the efficiency of our divide-and-conquer pursuit strategy for a general arena with obstacles.

5.1. Correctness of the Divide-and-Conquer Pursuit Strategy. Here, we establish that the divide-and-conquer pursuit strategy guarantees the capture. To do so, we start by providing in Section 5.1.1 an improved pursuit strategy in endgame (i.e. the last phase of the game where no obstacles are present in the game subdomain). We do so for two reasons: 1) As we mentioned in the previous section, the original strategy of the third free pursuer performing direct pursuit is sufficient only for capture with a positive capture radius (i.e., $r > 0$), but not for capture with capture radius zero. Second, the improved strategy in the endgame will be provably faster than direct pursuit.

5.1.1. Improved Capture Strategy in the Endgame. When the evader is trapped in a subdomain Ψ of the triangle configuration that encloses no obstacle, the pursuers execute the following *endgame strategy*, illustrated in Figure 12.

Unfortunately, it is possible that a subdomain Ψ_{free} might have diameter substantially greater than the diameter of the complete domain Ω_{free}, and it is possible that the paths s_1 and s_2 might be substantially longer than that diameter; see Section 5.2.2 for examples. Accordingly, our first task is to shrink the subdomain so its intrinsic diameter is at most $2d$, where $d = \text{diam}(\Omega_{\text{free}})$. The pursuers p_1 and p_2 stand at the level-set projections of the evader e onto s_1 and s_2, respectively. In the first step of the endgame strategy, p_1 and p_2 simultaneously move to the intrinsic closest-point projections instead. (Note that all projections are with respect to intrinsic distances in Ψ, which is simply connected, therefore the intrinsic closest-point projection defends the path.) As Remark 3 discusses, this switch from one projection to another does not permit the evader to cross the paths.

Because the subdomain encloses no obstacle, the two defended paths s_1 and s_2 are concave: they curl away from the domain if they turn at all. Upon reaching the intrinsic closest-point projection of e onto s_1, pursuer p_1 instantly switches to defending the straight line-segment path that meets the shortest path $s_\Psi(e, p_1)$ at a right angle (Figure 12, center). Observe that p_1 is already at the intrinsic closest-point projection of e onto the new, straight path, so this switch takes no time. Pursuer p_2 does likewise. These path switches can reduce but not increase the subdomain in which the evader is trapped. After the switch, the evader is caught in a subdomain of the triangle configuration whose boundary consists of a portion of the natural boundary and two defended line segments—or sometimes just one line segment, because a pursuer may single-handedly trap the evader in a pocket. (In the latter case, assume without loss of generality that pursuer is p_1.) Either way, Theorem 18 in Section 5.2 shows that this subdomain has the intrinsic diameter at most $2d$.

In the second step, p_3 moves to a point in the evader's subdomain in time at most d. In the third and final step, p_1 and p_3 use the orthographic pursuit strategy to capture the evader in that subdomain (Figure 12, right). Rather than defend horizontal and vertical paths, p_1 defends paths that are parallel to the path it started within the third step, and p_3 defends paths perpendicular to that path. Observe that although part of the boundary of the subdomain is not directly defended—while p_2 continues to defend a fixed path, p_1 advances the path it defends—the evader never has an opportunity to leave the subdomain. (Optionally, p_2 can advance its path too.)

The first of these three steps can be slow if s_1 or s_2 is very long. To help speed up the change in projections, p_3 creates an interval trap. Let $\bar{e}$ be the point on $s_1 \cup s_2$ nearest e by intrinsic distance—that is, e's closest-point projection onto s_1 or s_2, whichever is closest. Suppose without loss of generality that $\bar{e}$ lies on s_1. To create the trap, let b be the point at a distance of d from $\bar{e}$ along s_1 such that $\bar{e}$ is trapped between p_1 and b. (If there is no such point, let b be the corresponding endpoint of s_1.) While p_1 slides along s_1 toward $\bar{e}$, p_3 moves to the point b in time at most d. If p_1 has not reached $\bar{e}$ by then, p_1 and p_3 move toward each other until one of them reaches $\bar{e}$. If p_3 gets there first, the two pursuers swap identities. By Lemma 17 in Section 5.2, it takes time at most $3d/2$ for one of the pursuers to reach $\bar{e}$ and start defending the straight line-segment path that replaces s_1.

At that time, the evader may be trapped in a subdomain bounded solely by the new, straight s_1 and a natural boundary path, in which case the second step of the endgame begins. Otherwise, the pursuer p_2 might or might not still be chasing the evader's projection onto s_2. If it is, p_3 again creates an interval trap that catches it in time at most $3d/2$. If p_3 gets there first, the two pursuers swap identities.

5.1.2. Guaranteed Capture in the Divide-and-Conquer Strategy. We start by studying paths that are locally shortest in the sense that they cannot be made shorter by a small perturbation, although it might be possible to find a shorter path by taking a different route around the obstacles. A path s is *taut* if there is an $\epsilon > 0$ such that no point $p \in s$ has an open neighborhood in s of diameter less than ϵ that could be shortened by replacing it with a straight line segment included in Ω_{free}. If a taut path were a piece of string, pulling its ends would not change it. Every shortest path is taut, but not every taut path is a globally shortest path. A point $p \in s$ is a *turning point* if no open neighborhood of p in s is a straight line segment. Turning points can be *curve points* at which there is a well-defined line tangent to s though s is not straight, or *corners* at which the line tangent to s is not defined. If we suppose s is *directed* from x to y, then every turning point is either a left turn or a right turn. At a left turn, a taut path must touch a component of $\mathbb{R}^2 \setminus \Omega_{\text{free}}$ on its left side, and at a right turn it must touch a component of $\mathbb{R}^2 \setminus \Omega_{\text{free}}$ on its right side.

Lemma 12 *Let $s_1, s_2 \subset \Omega_{\text{free}}$ be two distinct taut paths connecting two points $x, y \in \Omega_{\text{free}}$, with s_2 longer than s_1 or equally long. Then $s_1 \cup s_2$ encloses an obstacle in Ω_{free} that touches s_2.*

Corollary 13 *If Ω_{free} has no obstacles, then any two points $x, y \in \Omega_{\text{free}}$ are connected by exactly one taut path, which is the unique shortest path connecting them.*

Lemma 14 *Let $\Omega = \Omega_{\text{free}} \cup \omega$ be a game domain, with Ω a topological disk, and* *suppose ω is nonempty. Then for every point $x \in \Omega_{\text{free}}$, there exists a point y on the boundary of Ω such that every taut path $s \subset \Omega_{\text{free}}$ connecting x and y touches an obstacle, and the path shooting procedure described in Section 4 finds such a y.*

Our strategy always makes progress toward an obstacle-free subdomain by induction on the characteristic number $\chi(\Psi)$ of a subdomain Ψ, which is the number of obstacles in Ψ plus the number of obstacles in Ψ that neither s_1 nor s_2 touches. (Thus, the latter obstacles are counted twice.) For example, in Figure 9, $\chi(\Psi) = 3$, because the subdomain encloses one obstacle that touches s_1 and s_2 and one obstacle that touches neither. The square obstacle outside the subdomain is not counted.

Theorem 15 *Let $\Psi = \Psi_{\text{free}} \cup \psi$ be a game subdomain of the triangle configuration, with respect to a game domain $\Omega = \Omega_{\text{free}} \cup \omega$. Suppose that at some point in time the evader is in Ψ, one pursuer p_1 is defending s_1, a second pursuer p_2 is defending s_2, and a third pursuer is available. Then the divide-and-conquer strategy guarantees that the three pursuers capture the evader.*

Proof If $\chi(\Psi) = 0$, the subdomain encloses no obstacle, and the evader is caught as described in Section 5.1.1. For cases 2 and 3, let s_3 be the newly defended path, and let $t \geq 1$ be the number of obstacles that s_3 touches in Ψ. For each obstacle that touches s_3 but not s_1 nor s_2 (see Figures 7 and 8), χ decreases by at least one, because each such obstacle is counted twice in $\chi(\Psi)$ but at most once in each of its two subdomains. Likewise, for each obstacle that touches s_3 and s_1, or s_3 and s_2, χ decreases by one (see Figure 7, top). Each such obstacle either will be on the opposite side of s_3 as the evader or will subdivide the new subdomain into connected components; in the latter case, the obstacle will not be inside the connected component that contains the evader. Overall, χ decreases by at least t. By induction on the sequence of iterations, χ is eventually reduced to zero and the evader is caught. ■

5.2. Efficiency of the Divide-and-Conquer Pursuit Strategy. In this section, we characterize the efficiency of the proposed pursuit strategy.

5.2.1. Trimming the Paths. The divide-and-conquer pursuit strategy we have presented works, but it is not always as fast as we want. Sometimes, the path s_3 shares portions of s_1, s_2, w, or all three. By shortening s_3 to reduce the overlap, we help p_3 to reach the evader's projection and start defending the path more quickly.

Imagine walking along s_3 from a until you touch an obstacle, then continuing until you touch the natural boundary path w (possibly simultaneously). At that point we trim s_3 so it goes no further, yielding a path s_3^*. (Ideally we would not let s_3^* touch w except at its final endpoint, but s_3^* must touch at least one obstacle.)

To guarantee a three-pursuer capture time of $\mathcal{O}(hd)$, the subdividing pursuer p_3 must defend a path that intersects the boundary paths s_1 and s_2 nowhere but at its endpoints. Accordingly, we further trim the path by setting $s_3^* \leftarrow s_3^* \setminus (s_1 \cup s_2)$. If that set has multiple connected components, we set s_3^* to be one of the connected components that touches an obstacle. This is the path that p_3 defends. In the single-component case, the evader cannot cross any part of the untrimmed s_3, but part of it might be defended by p_1 or p_2.

Trimming the path introduces a new difficulty: if p_3 defends only the trimmed path s_3^*, can any pursuer be relieved? The answer is yes, but often one of the pursuers has to modify the path it is defending. The easy case is when the evader's projection onto s_3 lies on the trimmed portion s_3^*: once p_3 arrives at that projection, p_3 can immediately start defending the entire path s_3, and the pursuit strategy proceeds as usual. The fast case is when the evader's projection onto s_3 does not lie on s_3^*: we modify s_1 or s_2 by patching s_3^* over the portion of s_1 or s_2 that the evader can no

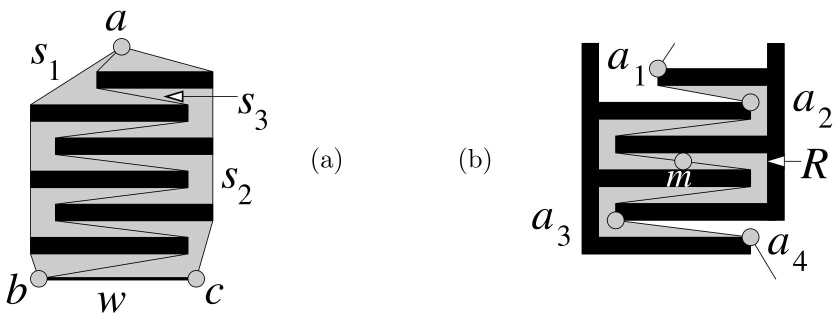

FIG. 13. *(a) A shortest path s_3 touching h obstacles in a game subdomain can have length $\Theta(hd)$. (b) A convoluted shortest path between two obstacles implies that the intrinsic diameter of Ω_{free} is at least half the length of the* inner path *connecting a_2 to a_3, because the shortest path in Ω_{free} from the midpoint m to any point outside the shaded region R has length at most d.*

longer reach, freeing p_3 immediately to defend a new path. (Note that we do not even need to wait for p_3 to reach s_3^*!)

5.2.2. The Capture Time for Three Pursuers. In a domain of intrinsic diameter $d = \text{diam}(\Omega_{\text{free}})$ with h holes, the main difficulty in showing that three pursuers can capture an evader in $\mathcal{O}(hd)$ time is that the paths the pursuers defend can be very long. Although a shortest path in the original domain Ω_{free} has length at most d, the subdomains that arise later can have intrinsic diameters as great as $\Omega(hd)$, as Figure 13(a) shows—insinuating that our pursuit strategy might have a worst-case capture time of $\Omega(h^2 d)$.

We meet this difficulty by showing that a shortest path whose relative interior touches t obstacles, but not the subdomain boundary, has length $\mathcal{O}(td)$. The path cuts a subdomain into two subdomains whose characteristic numbers are smaller by at least t. As the characteristic number of Ω is $2h$, it takes only $\mathcal{O}(hd)$ pursuit time to reduce the playing field to an obstacle-free subdomain.

Another difficulty is that paths between nonconvex obstacles can be twisty and very long; imagine interlocking gears or combs, as illustrated in Figure 13(b). A path may revisit the same obstacle arbitrarily many times. We handle this difficulty by showing that long passages imply that d is large.

Lemma 16 *Let $s = s(x, y)$ be a shortest path in a blocked free space Ψ_{block} of the triangle configuration. Suppose s intersects the defended paths s_1 and s_2 only at its endpoints x and y. Let $t \geq 1$ be the number of obstacles that s touches. (Obstacles intersecting only endpoints of s do not count.) The length of s is at most* $10td - 6d$.

Proof The path s subdivides Ψ_{block} into two subdomains, partitioning its obstacles. Consider obstacles ω_i and ω_j on opposite sides of s. Suppose that as you walk along s, you touch ω_i, then ω_j (possibly simultaneously); then you subsequently touch both obstacles again (possibly simultaneously). (The path may touch other obstacles in between those four contacts.) Let a_1 and a_2 be the first points where s touches ω_i and ω_j, respectively; let a_4 be the last point where s touches one of those two obstacles; and let a_3 be the last point where s touches the other obstacle, as illustrated in Figure 13(b). The shortest (in Ψ_{block}) paths $s(a_1, a_2)$, $s(a_2, a_3)$, and $s(a_3, a_4)$ are subpaths of s. Consider the region R bounded by portions of $\partial\omega_i$, $\partial\omega_j$, $s(a_1, a_2)$, and $s(a_3, a_4)$, shaded in Figure 13(b). Let m be the midpoint (by length) of $s(a_2, a_3)$. We see that $R \subset \Psi_{\text{block}}$ because the subdomain boundary paths s_1 and s_2 do not cross s or any obstacle boundaries. We ask, can there be a path from m to a_2 in Ω_{free} shorter than the shortest path $s(m, a_2)$ in Ψ_{block}? If such a path existed, it would have to leave R through $s(a_1, a_2)$ or $s(a_3, a_4)$ before reaching a_2; but no shorter path can reach $s(a_1, a_2)$ or $s(a_3, a_4)$. Therefore, the path $s(a_2, m)$ is a shortest path in Ω_{free} as well as in Ψ_{block}, so $d = \text{diam}(\Omega_{\text{free}})$ is at least as long as $s(a_2, m)$. Symmetrically, the length of $s(m, a_3)$ is at most d, thus the length of $s(a_2, a_3)$ is at most $2d$.

We call $s(a_2, a_3)$ an *inner path.* Observe that every inner path starts at a point a_2 where s touches an obstacle for the first time, and an inner path cannot start where s touches its first obstacle unless it also touches its second obstacle at the same point. Therefore, there are $t-1$ points on s where an inner path can start, so inner paths contribute at most $2(t-1)d$ to the length of s. If we remove the inner paths from s, the remainder of s can be subdivided into two types of subpaths. An *interpath* is a straight line segment whose endpoints lie on two different obstacles, or on one obstacle and one endpoint of s. An *intrapath* is a subpath that starts and ends on the boundary of the same obstacle, turns or curves only to one side, and turns solely on account of that obstacle.

We use amortized analysis to show there are at most $2t$ interpaths (excluding inner paths). We claim that where s includes a straight subpath (that doesn't overlap an inner path) from a point on $\partial\omega_i$ to a point on $\partial\omega_j$, s either touches ω_j for the first time or touches ω_i for the last time. We charge the cost of the subpath to ω_j in the former case; to ω_i in the latter case. If there are straight subpaths from the endpoints of s to the first and last obstacles s touches, we charge those obstacles for that. Each obstacle is charged at most twice: the first time and last time s touches it. Hence, there are at most $2t$ interpaths with total length $2td$. To justify this claim, observe that if ω_i and ω_j are on the same side of s, then s cannot touch ω_j, ω_i, ω_j again, and ω_i again in that order, because the obstacles are disjoint and each is connected.[4] If ω_i and ω_j are on opposite sides of s, and there is a straight subpath of s from ω_i to ω_j where s neither touches ω_j for the first time nor ω_i for the last time, then that subpath is part of an inner path.

Intrapaths are flanked by inner paths and interpaths, so there are at most $3t-2$ intrapaths. Each intrapath has length at most $2d$. To see this, consider an intrapath $s(x, y)$ connecting two points x and y on the boundary of an obstacle ω_i, and let m be its midpoint. By definition, s turns in one direction only, and only where it touches ω_i. Because $s(m, x)$ is a shortest path in Ψ_{block}, the only circumstance in which there might be a shorter path connecting m to x in Ω_{free} is if the path went around ω_i the other way; but such a path would have to go around $s(m, y)$ before reaching x, and therefore cannot be shorter than $s(m, y)$. It follows that $s(x, m)$ and (symmetrically) $s(m, y)$ are shortest paths in Ω_{free}, their lengths are at most d, and the length of $s(x, y)$ is at most $2d$. Summing the contributions from inner paths, interpaths, and intrapaths, we find that the length of s is at most $10td - 6d$. ■

The following two lemmas help to establish the pursuit time in the endgame.

Lemma 17 *Each interval trap described in Section 5.1.1 guarantees that a pursuer reaches the evader's intrinsic closest-point projection $\bar{e}$ in time at most $3d/2$, where $d = \text{diam}(\Omega_{\text{free}})$.*

Lemma 18 *Let Ψ_{free} be a subdomain of the triangle configuration. Suppose that the*

---

[4] The sequence of obstacles that s touches on one side only is a Davenport–Schinzel sequence of order two, which is known to have length at most $2o-1$ where o is the number of obstacles on that side [38]. The sequence of obstacles touched by s on both sides, after deleting inner paths, is not quite a Davenport–Schinzel sequence of order two, but it is close enough that we are using an amortization argument designed for such sequences.

outer boundary of Ψ_{free} *consists of two straight line segments* $s_1, s_2 \subset \Omega_{\text{free}}$ *(defended by pursuers) and a natural boundary path* $w \in \partial\Omega_{\text{free}}$*. Then* $\text{diam}(\Psi_{\text{free}}) \leq 2d$*, where* $d = \text{diam}(\Omega_{\text{free}})$*.*

Theorem 19 *With the divide-and-conquer pursuit strategy, three pursuers capture an evader of equal speed in a domain* Ω_{free} *with* h *obstacles in time at most* $(10h + 8 + 4\sqrt{2}) \cdot \text{diam}(\Omega_{\text{free}}) \doteq (10h + 13.657) \cdot \text{diam}(\Omega_{\text{free}})$*.*

Proof The initial subdomain Ω_{free} has a characteristic number of $2h$. In each iteration of the pursuit strategy, a pursuer moves to defend a path of length $10td - 6d$ whose relative interior touches $t \geq 1$ obstacles inside the subdomain, where $d = \text{diam}(\Omega_{\text{free}})$. It takes time at most d to move to the midpoint of the path and time at most $5td - 3d$ to move to the evader's projection, for a total of $5td - 2d$. Afterward, the characteristic number of the new subdomain containing the evader is reduced by at least t. Therefore, after time $10hd - 4d$ the characteristic number is reduced to zero and the subdomain is simply connected. (The worst case is two successive iterations, each taking time $5hd - 2d$.)

Next, it takes time at most $3d$ for the two pursuers defending the subdomain to switch from a level-set projection to an intrinsic closest-point projection. These two pursuers immediately switch to defending straight paths, whereupon the shrunken subdomain containing the evader has the diameter at most $2d$. It takes time at most d for the third pursuer to reach a point in that subdomain. Finally, it takes time at most $(8 + 4\sqrt{2})d$ to capture the evader with the orthographic pursuit strategy. ∎

If the convex hulls of the obstacles have mutually disjoint interiors, ruling out the circumstances illustrated in Figure 13(b), the bound of Lemma 16 improves to $3td + d$, and the capture time improves to $(4h + 12 + 4\sqrt{2}) \cdot \text{diam}(\Omega_{\text{free}})$. We omit the proof.

6. Future Work. It remains an open problem to find a pursuit strategy with an asymptotic capture time better than $\mathcal{O}(hd)$, given three pursuers in a bounded arena of diameter d with h obstacles. It is conceivable that there might be a strategy with an $\mathcal{O}(d)$ capture time, but that would probably require a new pursuit strategy very different from ours. If no such strategy exists, then understanding the fundamental limits of pursuit is an important problem. We conjecture that there exists a strategy with $\mathcal{O}(d \log h)$ capture time, which we would expect if there were an efficient way to repeatedly partition a subdomain into two subdomains each having half the obstacles; but we have not been able to find one. We view the question of the optimal capture time as a fundamental open problem in pursuit-evasion problems.

Acknowledgments. This work was supported in part by the National Science Foundation under Awards CPS-0931843, CCF-1423560, and CCF-1909204, and in part by the Office of Naval Research Basic Research Challenge on Multibody Control Systems N00014-17-S-BA13. The first author also gratefully acknowledges the support of IBM Goldstine Fellowship.

Appendix A. Deferred Proofs.

Proof of Lemma 6 So long as the shortest path through Ω from e to ℓ is a straight line segment perpendicular to ℓ, the projection $\bar{e}$ is the orthogonal projection of e onto ℓ, which moves continuously with speed at most 1. Otherwise, $\bar{e}$ does not move at all. When the shortest path from e to ℓ does not meet ℓ orthogonally, $\bar{e}$ remains fixed at an endpoint of ℓ. When the shortest path is not straight, the straight portion of that path that leaves ℓ perpendicularly remains fixed until e moves so that the straight portion is the entire path.

Proof of Lemma 9 A pursuer can move to a point on $s(x,y)$ in time $\mathrm{diam}(\Omega_{\mathrm{free}})$. It can then move to the projection $\bar{e}$ of an evader in finite time, regardless of the motion of $\bar{e}$, because the path is acyclic. By the following reasoning, the projection $\bar{e}(t)$ cannot move faster than unit speed on the path $s(x,y)$. The distance function $d(\cdot,\cdot)$ satisfies the triangle inequality: for any two points $p,q \in \Omega_{\mathrm{free}}$, $d(x,p) \le d(x,q) + d(p,q)$. This implies that where the gradient $\nabla d(x,\cdot)$ is defined, its magnitude nowhere exceeds one. For any $\Delta t > 0$, $|d(x, e(t+\Delta t)) - d(x, e(t))| \le d(e(t+\Delta t), e(t)) \le \Delta t$. The second inequality follows because the evader has unit speed. The projection $\bar{e}(t)$ is defined so that $d(x,\bar{e}(t)) = \min\{d(x,e(t)), d(x,y)\}$, so $|d(x,\bar{e}(t+\Delta t)) - d(x,\bar{e}(t))| \le \Delta t$. Therefore, the projection $\bar{e}$ of the evader cannot travel a distance greater than Δt along $s(x,y)$ in time Δt. This holds true for any positive Δt, even infinitesimal. It follows that $\bar{e}$ never moves faster than unit speed, even instantaneously, and once a pursuer coincides with $\bar{e}$, he can stay with $\bar{e}$ indefinitely.

Proof of Lemma 12 We call a Jordan curve (1-manifold without boundary) in the plane a *loop*. Because s_1 and s_2 are distinct with shared endpoints, $s_1 \cup s_2$ includes one or more loops; moreover, in at least one of these loops, the portion of s_2 in the loop is as least as long as that of s_1. At least one turning point $p \in s_2$ on that loop has curvature toward the region enclosed by the loop. Therefore, p lies on the boundary of a component of $\mathbb{R}^2 \setminus \Omega_{\mathrm{free}}$ enclosed by the loop. This component is bounded and therefore is an obstacle.

Proof of Lemma 14 Recall that the path shooting procedure chooses a point q in the closure of ω and extends $s_\Omega(x,q)$ until it meets $\partial\Omega$ at a point y. Because $s_\Omega(x,q)$ is a taut path in Ω, $s_\Omega(x,y)$ is also a taut path in Ω. As Ω has no obstacles, $s_\Omega(x,y)$ is a shortest path by Corollary 13, and no other path connecting x to y in Ω is equally short. Let $s \subset \Omega_{\mathrm{free}}$ be a taut path connecting x and y in Ω_{free}. If $q \in s$, it immediately follows that s touches an obstacle. Otherwise, s must differ from $s_\Omega(x,y)$, and therefore must be longer than $s_\Omega(x,y)$. The union $s \cup s_\Omega(x,y)$ includes one or more loops; in one of these loops, the portion provided by s is longer than the portion provided by $s_\Omega(x,y)$. Therefore, some turning point $p \in s$ on that loop has curvature toward the region enclosed by the loop. As s is taut, p lies on the boundary of a component of ω.

Proof of Lemma 17 The shortest path from e to $s_1 \cup s_2$ in Ω_{free} has length at most d and, as it cannot cross $s_1 \cup s_2$, lies entirely in Ψ. Hence $d_\Psi(e,\bar{e}) \le d$ during the first interval trap described in Section 5.1.1, where $\bar{e}$ is the point on $s_1 \cup s_2$ closest to e. Without loss of generality, assume $\bar{e}$ lies on s_1. The pursuer p_1 is initially at some projection of e onto s_1; by definition, $d_{s_1}(p_1,\bar{e}) \le d_\Psi(e,\bar{e}) \le d$. While p_1 chases $\bar{e}$, p_3 moves to a point d away from the initial position of $\bar{e}$ in time at most d, then chases $\bar{e}$. Hence, it takes time at most $3d/2$ for p_1 and p_3 to meet. During the second interval trap described in Section 5.1.1, s_1 is a straight line segment and $\bar{e}$ is the point on s_2 closest to e. Let a be the point where s_1 and s_2 meet. The shortest path from e to s_2 in Ω_{free} has length at most d and cannot cross s_2. If it crosses s_1 at some point x, we can shorten the shortest path by replacing the remainder of the path with the line segment xa; therefore the shortest path does not cross s_1, and it lies entirely in Ψ. Hence, $d_\Psi(e,\bar{e}) \le d$ during the second interval trap. By a repetition of the reasoning above, it takes time at most $3d/2$ for p_2 and p_3 to meet.

Proof of Lemma 18 Let x be a point in Ψ_{free}, and let a be the apex where s_1 and s_2 meet. Let s be the shortest path from x to a in Ω_{free}; its length is at most d. The path s is included in Ψ_{free}; if it were not, s would leave Ψ_{free} through some point q that lies on $s_1 \cup s_2$, and s could be shortened by replacing its exterior portion with a

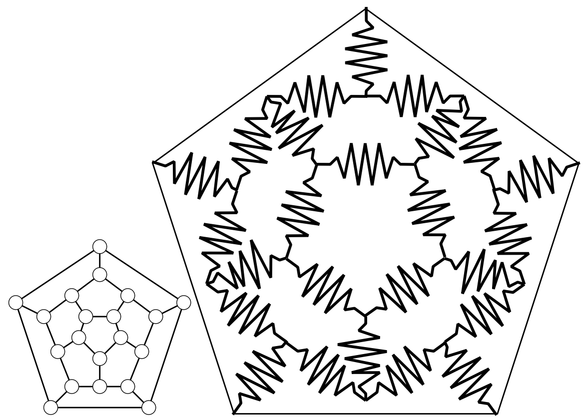

FIG. 14. *On the left is the discrete dodecahedron graph. On the right is a domain with the graph of a dodecahedron. Each tunnel between two degree-three junctions has the same travel time.*

straight line segment qa, a contradiction. Therefore, the shortest path through Ψ_{free} from a to any point in Ψ_{free} has length at most d. By the triangle inequality, the shortest path through Ψ_{free} connecting any two points in Ψ_{free} has length at most $2d$.

Appendix B. The Necessity of Three Pursuers. We show that three pursuers are necessary for an arbitrary game domain via an example in which an evader can evade two pursuers indefinitely, regardless of the controls chosen by the two pursuers. Consider the game domain Ω in Figure 14. The free space in this domain consists of the straight and wiggly black lines, representing tunnels of width zero. All tunnels, straight or wiggling, have the same travel time of 1 to get from one junction to a neighboring junction. Let the initial position of the evader be a junction. Suppose the two pursuers p_1 and p_2 begin at positions with distance at least 0.5 from the evader. The following is a strategy for the evader: (1) The evader remains stationary at a junction until at least one pursuer comes within an intrinsic distance of 0.5 (half a tunnel length) from the evader. (2) If a pursuer—assume without loss of generality it is p_1—is within a distance of 0.5 or less from the evader, the evader flees. The evader chooses between the two tunnels that do not contain p_1. The evader commits to the tunnel which, upon arriving the other end, would maximize the distance of the evader from the current position of p_2.

Proposition 20 *The preceding strategy enables the evader to evade indefinitely.*

Proof: The evader is safe resting at a junction so long as no pursuer is within a distance of 0.5 from the evader. When the evader is required to flee a pursuer p_1, that pursuer cannot get closer to the evader than a distance of 0.5, as both run at the same speed. The two tunnels that do not contain p_1 are part of a cycle of length five (with no shortcuts between any two points of the cycle), so no matter where the other pursuer p_2 moves, the evader chooses a tunnel such that when it reaches the junction at the other end of the tunnel, p_2 is still at least a distance of 0.5 away. Therefore, no pursuer ever gets closer to the evader than a distance of 0.5.

REFERENCES

[1] M. ADLER, H. RACKE, N. SIVADASAN, C. SOHLER, AND B. VOCKING, *Randomized pursuit-evasion in graphs*, Combinatorics Probability and Computing, 12 (2003), pp. 225–244.

[2] M. Aigner and M. Fromme, *A Game of Cops and Robbers*, Discrete Applied Mathematics, 8 (1984), pp. 1–12.
[3] S. Alexander, R. Bishop, and R. Ghrist, *Pursuit and Evasion in Non-Convex Domains of Arbitrary Dimension*, in Robotics: Science and Systems II, Philadelphia, Pennsylvania, Aug. 2006.
[4] S. Alexander, R. Bishop, and R. Ghrist, *Capture pursuit games on unbounded domains*, L'Enseignement Mathématique, 55 (2009), pp. 103–125.
[5] L. Alonso, A. S. Goldstein, and E. M. Reingold, *'Lion and Man': Upper and Lower Bounds*, ORSA Journal on Computing, 4 (1992), pp. 447–452.
[6] T. Andreae, *Note on a pursuit game played on graphs*, Discrete Applied Mathematics, 9 (1984), pp. 111–115.
[7] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, second ed., 1999.
[8] D. Bhadauria, K. Klein, V. Isler, and S. Suri, *Capturing an Evader in Polygonal Environments with Obstacles: The Full Visibility Case*, The International Journal of Robotics Research, 31 (2012), pp. 1176–1189.
[9] B. Bollobás, *The Lion and the Christian, and Other Pursuit and Evasion Games*, in An Invitation to Mathematics, D. Schleicher and M. Lackmann, eds., Springer-Verlag, Berlin, 2011, pp. 181–193.
[10] M. Breton and K. Szajowski, *Advances in Dynamic Games: Theory, Applications, and Numerical Methods for Differential and Stochastic Games*, Annals of the International Society of Dynamic Games, Birkhäuser Boston, 2010.
[11] M. Chen, Z. Zhou, and C. J. Tomlin, *Multiplayer reach-avoid games via low dimensional solutions and maximum matching*, in 2014 American control conference, IEEE, 2014, pp. 1444–1449.
[12] ———, *A path defense approach to the multiplayer reach-avoid game*, in 53rd IEEE conference on decision and control, IEEE, 2014, pp. 2420–2426.
[13] ———, *Multiplayer reach-avoid games via pairwise outcomes*, IEEE Transactions on Automatic Control, 62 (2016), pp. 1451–1457.
[14] H. T. Croft, *"Lion and man": A postscript*, Journal of the London Mathematical Society, 1 (1964), pp. 385–390.
[15] A. Dumitrescu, I. Suzuki, and P. Zylinski, *Offline variants of the lion and man problem*, in Proceedings of the Twenty-Third Annual Symposium on Computational Geometry, ACM, 2007, pp. 102–111.
[16] A. Festa and R. B. Vinter, *Decomposition of differential games with multiple argets*, Journal of Optimization Theory and Applications, 169 (2016), pp. 848–875.
[17] J. Flynn, *Lion and man: The general case*, SIAM Journal on Control, 12 (1974), pp. 581–597.
[18] J. O. Flynn, *Lion and man: The boundary constraint*, SIAM Journal on Control, 11 (1973), pp. 397–411.
[19] P. Frankl, *Cops and robbers in graphs with large girth and Cayley graphs*, Discrete Applied Mathematics, 17 (1987), pp. 301–305.
[20] ———, *On a pursuit game on Cayley graphs*, Combinatorica, 7 (1987), pp. 67–70.
[21] A. S. Goldstein and E. M. Reingold, *The complexity of pursuit on a graph*, Theoretical Computer Science, 143 (1995), pp. 93–112.
[22] A. Haurie, J. Krawczyk, and G. Zaccour, *Games and Dynamic Games*, World Scientific, 2012.
[23] J. Hu, M. Prandini, and S. Sastry, *Optimal coordinated motions of multiple agents moving on a plane*, SIAM Journal on Control and Optimization, 42 (2003), pp. 637–668.
[24] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, *A Differential Game Approach to Planning in Adversarial Scenarios: A Case Study on Capture-the-Flag*, in 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, May 2011, pp. 1451–1456.
[25] G. I. Ibragimov, *Optimal pursuit with countably many pursuers and one evader*, Differential Equations, 41 (2005), pp. 627–635.
[26] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, John Wiley & Sons, New York, 1965.
[27] S. Karaman and E. Frazzoli, *Sampling-based algorithms for optimal motion planning*, The International Journal of Robotics Research, 30 (2011), pp. 846–894.
[28] S. Kopparty and C. V. Ravishankar, *A Framework for Pursuit Evasion Games in* $\mathbb{R}^n$, Information Processing Letters, 96 (2005), pp. 114–122.
[29] J. Lewin, *The lion and man problem revisited*, Journal of Optimization Theory and Applications, 49 (1986), pp. 411–430.
[30] J. E. Littlewood, *Littlewood's Miscellany*, Cambridge University Press, Cambridge, United

Kingdom, 1986.

[31] S.-Y. Liu, Z. Zhou, C. Tomlin, and J. K. Hedrick, *Evasion of a team of dubins vehicles from a hidden pursuer*, in 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 6771–6776.

[32] S.-Y. Liu, Z. Zhou, C. Tomlin, and K. Hedrick, *A gradient-based method for team evasion*, in ASME 2013 Dynamic Systems and Control Conference, American Society of Mechanical Engineers Digital Collection.

[33] ———, *Evasion as a team against a faster pursuer*, in 2013 American control conference, IEEE, 2013, pp. 5368–5373.

[34] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, *A Time-Dependent Hamilton–Jacobi Formulation of Reachable Sets for Continuous Dynamic Games*, IEEE Transactions on Automatic Control, 50 (2005), pp. 947–957.

[35] S. W. Neufeld, *A pursuit-evasion problem on a grid*, Information Processing Letters, 58 (1996), pp. 5–9.

[36] T. D. Parsons, *Pursuit-Evasion in a Graph*, in Theory and Applications of Graphs, Y. Alavi and D. R. Lick, eds., vol. 642 of Lecture Notes in Mathematics, Springer, Berlin, 1978, pp. 426–441.

[37] J. Sgall, *Solution of David Gale's lion and man problem*, Theoretical Computer Science, 259 (2001), pp. 663–670.

[38] M. Sharir and P. K. Agarwal, *Davenport–Schinzel Sequences and their Geometric Applications*, Cambridge University Press, New York, 1995.

[39] R. Takei, R. Tsai, Z. Zhou, and Y. Landa, *An efficient algorithm for a visibility-based surveillance-evasion game*, Communications in Mathematical Sciences, 12 (2014), pp. 1303–1327.

[40] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, *Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation*, IEEE Transactions on Robotics and Automation, 18 (2002), pp. 662–669.

[41] Z. Zhou, J. Ding, H. Huang, R. Takei, and C. Tomlin, *Efficient path planning algorithms in reach-avoid problems*, Automatica, 89 (2018), pp. 28–36.

[42] Z. Zhou, R. Takei, H. Huang, and C. J. Tomlin, *A general, open-loop formulation for reach-avoid games*, in 51st Annual Conference on Decision and Control, IEEE, 2012, pp. 6501–6506.

[43] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin, *Cooperative pursuit with voronoi partitions*, Automatica, 72 (2016), pp. 64–72.