A Visibility-Based Escort Problem

Lance G. Fletcher[†] Priyankari Perali[†] Andrew Beathard Jason M. O'Kane

***Abstract*— This paper introduces and solves a visibility-based escort planning problem. This novel problem, which is closely related to the well-researched family of visibility-based pursuit-evasion problems in robotics, entails an escort agent tasked with escorting a vulnerable agent, called the VIP, in a 2-dimensional environment. The escort protects the VIP from adversaries that pose line-of-sight threats. We describe a correct and complete planning algorithm whose inputs are a simply-connected polygonal map of the environment, starting locations for the escort and the VIP, along with a goal location to which the VIP agent should be safely moved. The algorithm computes trajectories for the escort and VIP which allow the VIP to reach its goal without coming into the line-of-sight of the adversary at any time. During the execution of these trajectories, the adversary is allowed to move along any continuous path that does not enter into the line-of-sight of the escort. The algorithm proceeds by dividing the environment into a collection of conservative regions and planning the escort's movements as a sequence of these regions via breadth-first search over an information graph. The trajectory of the VIP can then be constructed by tracing the 'safe zones' swept out by the escort's trajectory. We describe an implementation of this algorithm and present computed examples of escort agent strategies in diverse environments.**

I. Introduction

We consider a *visibility-based escort problem*, a geometric planning problem in which an escort robot has the mission of safely escorting another vulnerable agent, the VIP, through an environment while ensuring that the VIP remains out of the line-of-sight of adversaries. This sort of escorting problem is relevant, for example, in scenarios where the safety or privacy of mobile agents must be preserved in the presence of adversaries.

The problem we address is inspired by —and closely related to— the well-researched visibility-based pursuit-evasion problems in robotics, which have been studied in many different variations [4], [6], [8], [21], [24]. That established formulation is a game between two types of agents, called *pursuers* and *evaders*. The goal is to find a path for the pursuers which guarantees that the evaders are located within the environment. The escorting problem we introduce here expands the setting to three types of agents: (i) a *VIP*, whose goal is to move safely through the environment to a specified goal, (ii) an *adversary*, whose goal is to establish a line-of-sight with the VIP without being seen by the escort, and (iii) an *escort*, who works in cooperation with the VIP to enable the VIP to reach its goal.

[†]The contributions of the first two authors are so close to equal as to make futile any attempt at a confident ordering of their relative contributions. All of the authors are with the Department of Computer Science and Engineering, Texas A&M University, College Station, Texas, USA. {lance.fletcher, perali, drewbeathard, jokane}@tamu.edu

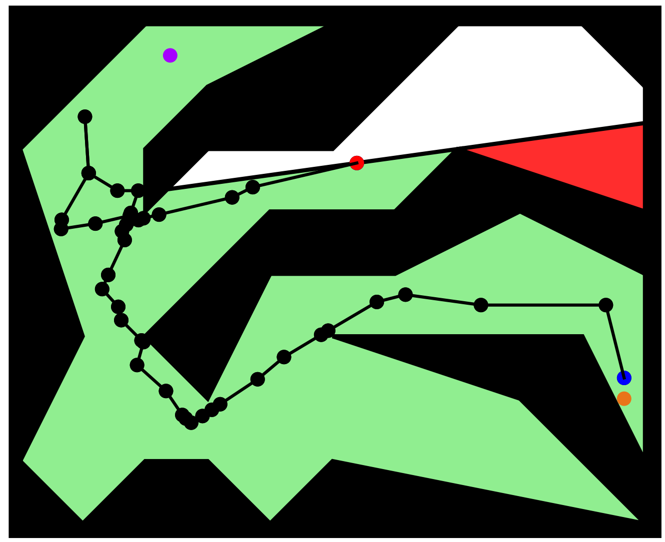

Fig. 1: An escort path, black line, which solves the escort problem. The blue dot is the escort's starting position. The red dot is the escort's end position. The orange dot is the VIP's starting position. The purple dot is the VIP's goal position. The green shaded region is a VIP-reachable safe zone. The red shaded region is a contaminated shadow.

The novel contribution of this paper is to introduce this visibility-based escort problem, to describe a complete algorithm for solving it, and to present an implementation of the algorithm with computed examples for the generated paths.

We present a complete algorithm that generates trajectories for both the escort and the VIP, ensuring that the VIP reaches its goal without being visible to any adversary that has not first been visible to the escort. See Figure 1. The algorithm works by first generating a trajectory for the escort, constructed to ensure that a corresponding safe trajectory for the VIP exists. For each partial escort trajectory that the algorithm considers, we identify the *shadows*, which are portions of the environment that are not visible to the escort from a given position. Each shadow may be either *contaminated* or *clear*, indicating whether an adversary not yet seen by the escort may be concealed therein. Based on the shadows, the algorithm also computes *safe zones*, which are portions of the environment not visible from any point in any of the contaminated shadows. Analogously to the contaminated/clear labels for the shadows, each safe zone is labeled *reachable* or *unreachable*, to track whether any VIP trajectory exists to reach that safe zone, given the escort trajectory so far. The combination of the escort position, the contaminated/clear label for each shadow, and the reachable/unreachable label for each safe zone is sufficient to track the state of the problem.

The overall planning algorithm works by dividing the environment into *conservative regions* within which this problem state is invariant, and then performing a breadth-first search over the resulting information graph. The search concludes when it finds a path that reaches a problem state in which the goal position is contained in a reachable safe zone. As a final step, the VIP trajectory is generated by tracing through the reachable safe zones generated by the escort's trajectory.

The remainder of this paper consists of a review of related work (Section II), a precise problem statement (Section III), a description of the division of the environment into conservative regions (Section IV), an algorithm that uses those conservative regions to solve the escort problem (Section V), experimental results (Section VI), and concluding remarks (Section VII).

II. Related Work

A. Escort with robots

A few other escorting problems have been considered in the literature, often in the context of human-robot interaction [5], [11], and also occasionally in the context of defense against various threats [7], [13]. Antonelli, Arrichiello, and Chiaverini describe an approach in which the escorting is done by surrounding an entity whose path is not known with multiple robots [1], [2]. The goal is to limit the escape windows of the entity by equally distributing the robots around it. More closely related to our problem, Bhatia, Solmaz, Turgut, Bölöni account for unknown adversaries having a view of the VIP [3]; however, they attempt to limit the adversaries' view of the VIP by surrounding it with multiple robots. Our approach utilizes a single escort robot to clear areas in an environment of adversaries to create a safe path for the VIP to take.

B. Pursuit and evasion

The escort problem presented has many similarities with the visibility-based pursuit-evasion problem in which one or multiple pursuer agents work together to locate mobile adversaries. The visibility-based pursuit-evasion problem has been a well studied problem within the robotics community [10], [12], [14], [18], [20]. Given an environment, a solution to the pursuit-evasion problem will specify a path for each pursuer to take such that it is guaranteed that any adversaries will eventually be discovered. In the escort problem, the escort has a similar role to the pursuer, since both are capable of clearing areas within the environment. Early forms of the pursuit-evasion problem utilized a graph representation of the environment [19]. An algorithm for a geometric setting, in which the environment is modeled as a polygon, was introduced by Guibas, Latombe, LaValle, Lin, and Motwani [9]. Provably complete and optimal solutions to the pursuit-evasion problem have been found [9], [23]. A generalization of this approach for multiple pursuers is presented by Stiffler and O'Kane [21]. The more general problem of reasoning about agents moving among unobservable regions was tackled by Yu and LaValle using the concept of shadow information spaces [25].

Our approach to the escort problem is based on concepts from this prior work, specifically the concept of an information space which allows for the environment to be discretized into conservative regions. A conservative region is an area within the environment in which the pursuer can move freely while maintaining the same shadow information [9]. A major difference between the pursuit-evasion problem and the escort problem is the latter does not require every area within the environment to be viewed. This is because solutions to the escort problem only require guaranteeing the safety of the VIP along a path at all times, rather than locating an adversary.

III. Problem Statement

This section introduces and formalizes the visibility-based escort problem. The basic idea is that the VIP seeks to travel safely to a goal position, with the assistance of a cooperative escort. The VIP is threatened by an adversary, whose goal is to establish visibility with the VIP without first being visible to the escort. This interaction forms a two-player game: If the adversary sees the VIP without first being seen by the escort, the adversary wins; otherwise, the VIP and escort win. The following definitions make this idea precise.

The environment is modeled as a two-dimensional simply-connected polygonal closed free space $F \subset \mathbb{R}^2$. For a given position $q \in F$, the *visibility polygon* $V(q) \subseteq F$ is the region in F visible from q, $V(q) = \{p \in F \mid \overline{pq} \subset F\}$.

Three agents, each modeled as a single point, move within the environment F:

- The *VIP* agent is considered 'important' in some sense, and thus needs to be protected at all times and escorted to the goal location.[1] We write $v(t)$ to denote the position of the VIP at time t. A start position $v(0)$ and a goal position v_G are given as input.
- The *escort* agent works in cooperation with the VIP to enable the VIP to reach its goal, as detailed below. The escort has omnidirectional vision with unlimited range. Let $e(t) \in F$ represent the escort agent's position at time t.
- The *adversary* agent poses a line-of-sight threat to the VIP. Following the tradition of prior pursuit-evasion work, we adopt a worst-case approach, in which the adversary may follow any continuous trajectory within F, with no limitations on its speed. Because we hope to generate trajectories for the VIP and escort that achieve the goal for any adversary movements, it is sufficient to consider only a single adversary; any strategy that protects the VIP from a single adversary in this worst case sense will also protect it protect the VIP from multiple simultaneous adversaries. The adversary's position

[1] Though for simplicity we refer to the VIP in the singular in this paper, note that no changes to the approach are needed to account for multiple VIP agents all sharing the same goal, and that only a slight generalization is needed for multiple VIPs with distinct goals.

at time t is denoted $a(t)$, but the adversary's location is unknown to the escort and VIP agents.

As these three agents move, there are three termination conditions, based on the agents' interactions:

1) If the escort and adversary share a line-of-sight, the adversary is 'neutralized', and the VIP/escort team wins.
2) If the adversary and the VIP share a line-of-sight, the VIP is assumed to experience harm of some kind. The adversary wins.
3) If the VIP reaches its goal, the objective is complete and the VIP/escort team wins.

In that setting, we can state the algorithmic problem addressed in this paper:

Input: An environment F, the starting positions $v(0)$ and $e(0)$ for the VIP and escort respectively, and the VIP goal position v_{G}.

Output: A non-negative real number finishing time T and trajectories $v : [0,T] \to F$ and $e : [0,T] \to F$, such that $v(T) = v_{\mathrm{G}}$, and for any continuous adversary trajectory $a : [0,T] \to F$ and any $t \in [0,T]$ for which $v(t) \in V(a(t))$, there exists $t' \in [0,t)$ for which $a(t') \in V(e(t'))$.

That is, we seek to generate trajectories for the VIP and escort that ensure that, regardless of the trajectory followed by the adversary, the adversary does not see the VIP, except possibly when the adversary has first been seen by the escort.

IV. Forming a discrete state space

The problem introduced in Section III is a fundamentally continuous problem: It deals with agents moving in continuous time along paths within a continuous space. This section discusses how to treat the problem in a discrete way by identifying finite sets of regions sufficient to represent the full range of possible movements for each of the three agents. This 'lossless' discretization forms the essential foundation for the main algorithm described in Section V. Specifically, we consider shadows (Section IV-A), safe zones (Section IV-B) and conservative regions (Section IV-C) to describe the possible movements of the adversary, VIP, and escort, respectively.

A. Shadows

Consider some particular time t, when the escort is at some position $e(t)$.

Definition 1. *For a given escort position $e(t)$, a* shadow *is a maximal path-connected region in $F \setminus V(e(t))$.*

Since F is simply-connected, we can characterize each shadow by an *anchor point* (an environment vertex around which the escort cannot see) and an *incident point* (the opposite endpoint of the shadow's boundary segment).

The notion of shadows is relevant because, assuming that the adversary remains unseen by the escort, the adversary's movements are restricted to the portions of the environment not visible from $e(t)$; these hidden regions are the shadows. Crucially, since the adversary can move arbitrarily quickly, there is no need to reason about the adversary's specific location within a shadow. It can move freely within a shadow, but must remain within that shadow lest it be eliminated by the escort. Thus, given a trajectory prefix $e : [0,t] \to F$ for the escort, we can classify each shadow as either *contaminated* or *cleared*.

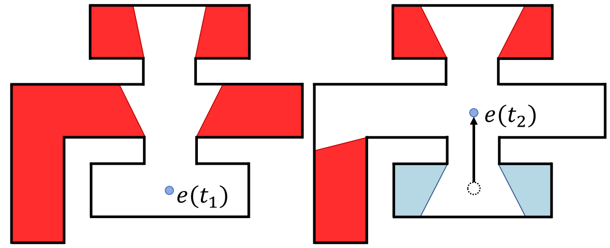

Fig. 2: [left] The initial state of shadows for the shown escort position e. There are four shadows, all contaminated, shown in red. [right] The shadows change after an upward movement by the escort. One shadow had disappeared and two new cleared shadows, shown in blue, have been created.

Definition 2. *A* contaminated shadow *$S \subseteq F \setminus V(e(t))$ is a shadow for which there exists some continuous adversary trajectory $a : [0,t] \to F$ ending in S that is never visible to the escort. That is, a contaminated shadow S has some trajectory a with $a(t) \in S$ and $a(t') \notin V(e(t'))$ for all $t' \le t$. A* cleared shadow *is a shadow that is not contaminated.*

Figure 2 illustrates the concept. The important idea is that, to find trajectories for the escort and VIP, we only need to keep track of the set of contaminated shadows, because the rules of the system require precisely that the VIP must never be visible from any point within a contaminated shadow.

The way the shadows and their contaminated/cleared status change is well-known from the pursuit-evasion literature [9]. In short, depending on the geometry of the environment, shadows can *appear* or *disappear*, a pair of shadows can *merge*, or a single shadow can *split* into two separate shadows. The contaminated/clear status of a shadow can be correctly tracked across these changes according to intuitive rules: any newly-appeared shadow is cleared; any shadow that disappears has its contaminated/cleared label discarded; a merged shadow is contaminated only if either of the original shadows was contaminated; and new shadows resulting from a split inherit the same label as the original shadow. Details appear in Guibas et al. [9].

B. Safe zones

Next, consider the possible locations for the VIP, given an escort position $e(t)$ and contaminated labels on some of the resulting shadows. In such a situation, what locations are safe for the VIP?

Definition 3. *For a given escort trajectory $e : [0,t] \to F$, let $\mathcal{S} = \{S_1, \ldots, S_m\}$ denote the set of contaminated shadows. A* safe zone *is a maximal path-connected subset of $F \setminus \bigcup_{S \in \mathcal{S}} \bigcup_{a \in S} V(a)$.*

Simply put, the safe zones are the regions where a VIP

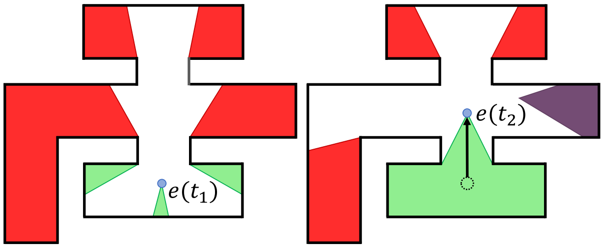

Fig. 3: A continuation of the example from Figure 2. Recall that after the escort's upward movement, there are three contaminated shadows, shown in red. At this point, there are two safe zones: One shown in green which is VIP-reachable and another shown in purple which is VIP-unreachable. The two cleared shadows, which do not impact the safe zones, are not shown.

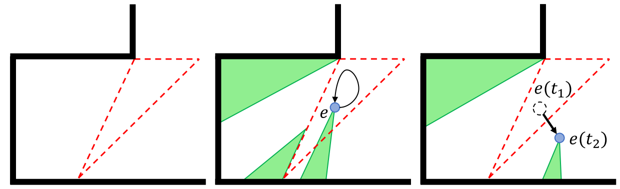

Fig. 4: [left] An example of a conservative region, shown by dashed red lines. [center] A path for the escort within the conservative region. The three safe zones, shown in green, are unaffected by this escort movement; each exists and is VIP-reachable before the movement and each remains VIP-reachable during and after the movement. [right] A movement for the escort that leaves a conservative region, creating a change in the safe zones. In this case, the middle safe zone disappears. If the escort were to return to the original position, that middle safe zone would reappear, but would be labeled as VIP-unreachable.

agent can safely exist at a particular time, based on the current position and past movements of the escort. As with adversaries in shadows, note that we need only keep track of which safe zone contains the VIP, and need not attend to the VIP's position within that safe zone. This occurs because changes to the safe zones occur as the escort moves; the timing of the escort's and VIP's movements can be coordinated to ensure that the VIP can remain within the safe zone as these changes occur. However, only some safe zones can be safely reached by the VIP without crossing through unsafe parts of the environment.

Definition 4. *A safe zone Z is* VIP-reachable *if there exists some continuous trajectory $v : [0, t] \to F$ ending in Z that is never visible to any point in any contaminated shadow. A safe zone is* VIP-unreachable *if it is not VIP-reachable.*

The idea is that VIP-reachable safe zones are those that can be reached by the VIP, considering the VIP's starting position and the previous actions of an escort agent up to a point in time. Figure 3 provides an example of a configuration with both VIP-reachable and VIP-unreachable safe zones. Our algorithm assigns a reachable/unreachable label to each safe zone during the planning process. The concept of VIP-reachability is somewhat analogous to the contaminated/clear labels for each shadow, but in reference to possible locations for the VIP, rather than for the adversary; this generalized concept does not have a direct analogue in prior pursuit-evasion work. Nonetheless, as the escort moves, the reachable/unreachable labels can be updated in a manner equivalent to updates to the shadows' contaminated/cleared labels.

C. Conservative regions

Finally, we turn to the question of how movements for the escort can be generated as a sequence of discrete steps. The key idea is the concept of *conservative regions*, which is inspired by a similar notion in the original Guibas et al. paper [9].

As articulated in Sections IV-A and IV-B, the relevant information about where the adversary and the VIP can travel can be captured by tracking the contaminated/cleared and VIP-reachable/VIP-unreachable status of the shadows and safe zones, respectively. Though even small movements of the escort will change the precise geometry of the shadows —and, therefore, the precise geometry of the safe zones—, we show in this section that only escort movements that cross certain critical boundaries affect any change to these labels. The term 'conservative region' refers to a portion of the environment in which the escort can move without triggering any of these discrete changes. Thus, the planning algorithm of Section V only needs to consider escort motions in terms of a sequence of successively-adjacent conservative regions. This leads to a discrete search space for the algorithm while maintaining its completeness. In the remainder of this subsection, we define this idea precisely, describe a method of partitioning the environment into conservative regions, and argue that this partition yields regions that are indeed conservative.

1) Partitioning F into conservative regions: We begin by defining what it means for a region to be conservative.

Definition 5. *A region $C \subseteq F$ is* conservative *if any continuous escort trajectory $e : [t_1, t_2] \to C$ that remains in C leaves the contaminated/cleared and VIP-reachable/VIP-unreachable labels unchanged.*

The intuition is that escort movements within a conservative region are 'irrelevant', in the sense that they do not make any meaningful change to the problem state. Figure 4 shows an example.

The collective set of conservative regions depends on the geometric structure of the environment.

Construction 1. *Partition F by adding dividing line segments via two distinct types of* ray shooting *operations:*

1) *From each pair of mutually visible environment vertices, extend rays outward, along the direction between the two vertices, anywhere those rays do not immediately leave the environment. Figures 6a–c show examples of these ray extensions, which match the visibility cell decomposition used in prior work [9]. Note the particular case shown in Figure 6c, in which the two vertices are both reflex vertices; this case is called a* bitangent.
2) *For each ray extension from a bitangent, let i denote the*

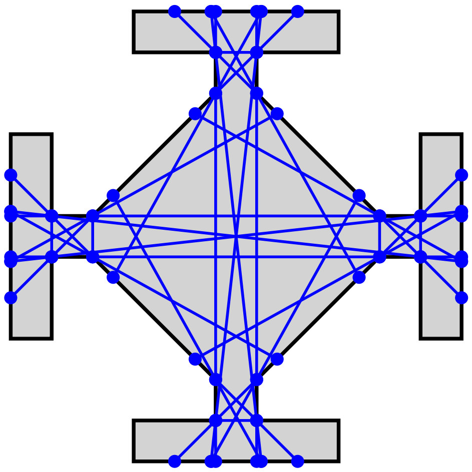

Fig. 5: An example environment, subdivided into conservative regions.

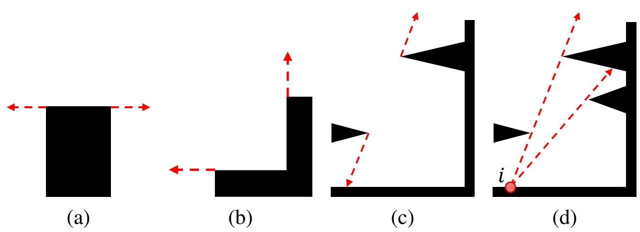

Fig. 6: Ray shooting methods to obtain conservative regions. Cases (a), (b), and (c) are part of the visibility cell decomposition used in many pursuit-evasion problems. Case (d) is a new case arising from potential changes in the safe zones for the VIP.

incident point where this ray reaches the boundary of F. From each such i, extend rays toward each reflex vertex visible from i. See Figure 6d. Note that this includes a ray extension back through the original reflex vertex from which i was created, creating a division along the segment between the two mutually visible reflex vertices that form the bitangent.

Informally, the second class of ray extensions in Construction 1 captures a form of "visibility of the visibility", marking places where a movement of the escort would lead to changes in the safe zones. The intuition for these divisions is that the escort problem is based upon regions that are visible from the shadows, i.e. locations not visible by the escort. This strongly suggests the idea of shooting rays from incident points of other shadows. See Figure 5 for an complete example of Construction 1.

2) Correctness of the partition: Now we argue that the partition of F formed by Construction 1 is indeed a partition of F into conservative regions. The argument is similar to the approach taken by Stiffler and O'Kane [21] for a multi-pursuer variant of the pursuit-evasion problem, focused on characterizing locations for the escort at which the boundary of the safe zones can change.

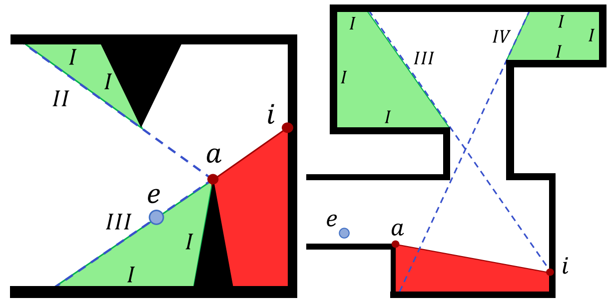

Fig. 7: Example environments illustrating the four types of safe zone boundary edges, determined by the escort's position e and a shadow's anchor point a and incident point i. Contaminated shadows are shown in red; safe zones are shown in green. [left] An environment with safe zone boundary edge Types I, II, and III. [right] An environment with safe zone boundary edge Types I, III, and IV.

Notice that each safe zone is a polygonal region within F. Each of the edges of its boundary can be classified into one of four types, as shown in Figure 7:

Type I: Portions of the environment boundary. Several examples appear in Figure 7.

Type II: Edges formed by rays passing through a contaminated shadow's anchor point and an environment vertex. In Figure 7a, this case occurs along the ray from the contaminated shadow anchor point a, through the upper reflex vertex of the environment, ending at the environment boundary.

Type III: Edges formed by rays passing through a shadow's incident vertex and an environment vertex. In Figure 7b, this case occurs on the left side, on the ray for the shadow's incident point i, past the reflex vertex of the environment, to the environment boundary.

Type IV: Edges along the line between two environment vertices. In Figure 7b, this occurs in the upper right. Notice that the line forming the Type IV edge intersects with the contaminated shadow boundary: This is characteristic of Type IV edges, which informally correspond to scenarios where the adversary can see certain regions only from the interior of a particular shadow, rather than at its anchor or incident points.

These edge types are relevant because we can characterize changes to a safe zone in terms of changes to the cyclic sequence of edge types around its boundary. To analyze those changes, we start by utilizing a classic result.

Lemma 1. *Let $e : [t_1, t_2] \rightarrow F$ denote a portion of a trajectory for the escort. If e does not cross any of the ray extensions generated by Construction 1 along this trajectory, then the set of contaminated shadows does not change, nor does the anchor point for any of of those shadows.*

Proof. Our partition of F is a refinement of the partition used by Guibas et al. [9], whose Lemma 4 demonstrates that their partition conserves shadow contamination. □

Lemma 1 ensures us that, since the set of contaminated shadows will not change within a single region of our parti-

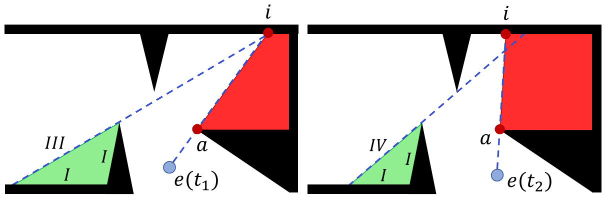

Fig. 8: Changes to the visibility polygon of the incident point of a shadow lead to changes in a safe zone.

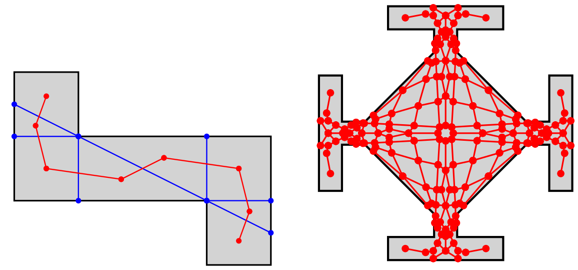

Fig. 9: Example position graphs, shown in red. Blue lines in the left example represent boundaries of the conservative regions.

tion, changes to the VIP-reachable safe zones as the escort moves can only be induced by movements of the incident points of each shadow. Therefore, meaningful changes to the safe zones —that is, changes to the cyclic sequence of edge types around a safe zone's boundary— cannot occur unless there is a change to the set of environment vertices visible from some shadow's incident point. Notice, however, that ray extensions introduced in the second step of Construction 1 ('visibility of the visibility') are precisely the escort locations where this occurs. An example appears as Figure 8.

Consequently, the result we need follows directly:

Lemma 2. *Each region of the partition of F described above is conservative.*

Informally, Lemma 2 confirms that the partitioning we propose is lossless, in the sense that our algorithm can consider only the sequence of conservative regions visited by the escort, rather than the full space of continuous trajectories, without sacrificing completeness. The next section describes our planning algorithm for the escort problem leveraging this structure.

V. Algorithm Description

This section introduces an algorithm to generate paths for the escort and the VIP to solve the escort problem, leveraging the decomposition of the environment into conservative regions from Section IV. The method consists of four main components. First, a graph of positions is constructed from the conservative region discretization of the environment. Second, an information graph is formed that encodes the way the contaminated/cleared and VIP-reachable/VIP-unreachable labels change as the escort moves between those conservative regions. Then, a breadth first search (BFS) algorithm is applied to the information graph to find a path for the escort that is capable of escorting the VIP to its goal location. In the final step, a specific path for the VIP is constructed by working backward from the final problem state of the BFS. Details of each of these steps follow.

A. The position graph

Recall from Section IV that the information about the escort's trajectory that is relevant for the escort problem — specifically, the changes to the contaminated/cleared labels of the shadows or to the VIP-reachable/VIP-unreachable labels of the safe zones— is fully captured by the sequence conservative regions visited by the escort. Thus, our algorithm considers the escort's movement across a *position graph* G_P, which has a vertex at the center of each conservative region. Edges in G_P connect vertices whose conservative regions share a boundary edge. See Figure 9. The idea is the G_P encodes both the conservative regions that can be visited by the escort and the transitions that can be made between them.

B. The information graph

To construct a path for the escort, our algorithm must track not only the conservative regions visited by the escort, but also the status of the shadows and safe zones, contaminated/cleared and VIP-reachable/VIP-unreachable respectively. This information is collectively referred to as a *problem state*.

Notice that problem states are not solely based on the position of the escort, but also the path taken by the escort leading up to that position. In fact, the escort may revisit the same position multiple times, with dramatically different problem states each time. This occurs because the problem state's set of contaminated shadows and VIP-reachable safe zones are dependent on the previous movements of the escort. Problem states capture precisely the information needed to handle this complexity correctly.

For a given problem state, we enumerate the choices for the possible next problem states by considering each of the neighboring conservative regions in the position graph, and computing the resulting shadow and safe zone labels, as described in Section IV. This, the problem of searching for a trajectory for the escort is cast as the problem of finding a path through a directed graph of such transitions, wherein each vertex represents a distinct problem state. We call this graph the *information graph*, denoted G_I.

C. A trajectory for the escort

To find a trajectory for the escort, we perform a breadth-first search on G_I. The search starts at the initial problem state, namely a problem state at the conservative region containing the initial escort position, with all shadows contaminated and only the safe zone containing the VIP's initial position marked as VIP-reachable. The algorithm terminates when a path is found which ends at a problem state with a VIP-reachable safe zone containing the VIP's goal position. Given such a path, an escort trajectory can be constructed by generating motions, at some arbitrary constant velocity,

TABLE I: Algorithm computation time in various environments.

Environment	Time (sec)
Figure 1	551.32
Figure 10a	222.46
Figure 10b	289.29
Figure 10c	536.13
Figure 10d	1500.00
Figure 10e	2753.57
Figure 10f	52.22

that visit the centroids of the conservative regions for each problem state on the path through G_I. On the other hand, if the search queue is exhausted before finding such a path, the algorithm returns failure.

D. A trajectory for the VIP

The final step of the algorithm is to construct a trajectory for the VIP. Recall that the constructed escort trajectory leads to a final state in which the VIP's goal state is contained within a VIP-reachable safe zone. Thus, if an escort trajectory is generated, by construction there also exists a corresponding VIP trajectory. This trajectory is synthesized starting from the goal, working backward to the starting condition, remaining within the VIP-reachable safe zone that leads eventually to the goal.

VI. Results

We implemented the proposed algorithm in Python, using the geometric primitives provided by the `scikit-geometry` library. Figure 10 shows some computed results, which confirm that the algorithm successfully finds solutions in a variety of scenarios involving environments which vary in complexity and size. Each example in the figure shows the escort's path along with the final problem state reached from the found solution. In each case, the final problem state shows a VIP-reachable safe zone (green) containing the VIP's goal location (green dot), indicating the VIP's ability to safely reach its destination. Notice that the escort's paths are non-trivial and often require the clearing of many areas within the environment as part of the solution.

A notable property of the computed solutions is that not every contaminated shadow (red) was cleared in the final solution. This occurs, for example, in Figure 10f. This distinguishes the solutions found by our method from the solutions to a typical pursuit-evasion problem.

Table I shows the computation time for our implementation to find paths for several different scenarios, including those depicted in Figure 10. As one might expect, instances which require the escort to clear more distinct contaminated shadows take longer to solve. Anecdotally, a large fraction of the computation time is due to geometric computations performed to find transitions in the information graph G_I, specifically in determining which shadows will be contaminated and which safe zones will be VIP-reachable.

VII. Conclusion

This paper introduced a novel visibility-based escort problem and provided an algorithm that solves it. The algorithm utilizes an environment discretization method and a forward search planning algorithm to determine a non-trivial path for the escort, resulting in a safe path for the VIP to follow. Moreover, our approach demonstrates that the escort does not need to clear the entire environment of adversaries to find a safe path for the VIPs.

Several additional questions remain to be addressed in future research. One possibility is to allow environments that are not simply-connected, i.e. environments with holes. In most of the escort scenarios shown in this paper, during the escort's execution of the solution path, the VIP-reachable safe zones gradually expand. We anticipate that environments with holes are likely to have solutions which would cause the safe zones to relocate across the environment rather than expand. That is, holes seem likely to make the problem more challenging by allowing the adversary an opportunity to seek a line-of-sight with the VIP from either ahead or behind.

Another avenue for further research is the case of multiple escorts. This scenario would allow safe paths for VIPs to be discovered for more complex environments. However, we anticipate that the size of the state space to be search is likely to grow exponentially with the number of escorts. This very generalization has led to massive computational complexity issues in related multi-agent problems [21], for which specialized sampling approaches have proved successful [17], [22].

Finally, other interesting questions arise if there is a need to recover from escort robot failures [16] or to form plans that are robust-by-construction to such failures [15]. In such cases, judicious expansions of the problem state may be effective.

Acknowledgment The authors are grateful to Nicholas Stiffler for assistance with the implementation of our method.

References

[1] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment/escorting mission for a multi-robot system: Theory and experiments," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, 2007, pp. 1–6.

[2] ——, "The entrapment/escorting mission," *IEEE Robotics & Automation Magazine*, vol. 15, no. 1, pp. 22–29, 2008.

[3] T. S. Bhatia, G. Solmaz, D. Turgut, and L. Boloni, "Two algorithms for the movements of robotic bodyguard teams," in *Workshops at the AAAI Conference on Artificial Intelligence*, 2015.

[4] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics: A survey," *Autonomous Robots*, vol. 31, pp. 299–316, 2011.

[5] D. Conte and T. Furukawa, "Autonomous robotic escort incorporating motion prediction and human intention," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3480–3486.

[6] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion without mapping or global localization via local frontiers," *Autonomous Robots*, vol. 32, pp. 81–95, 2012.

[7] A. Garg, Y. A. Hasan, A. Yañez, and L. Tapia, "Defensive escort teams via multi-agent deep reinforcement learning," *arXiv preprint arXiv:1910.04537*, 2019.

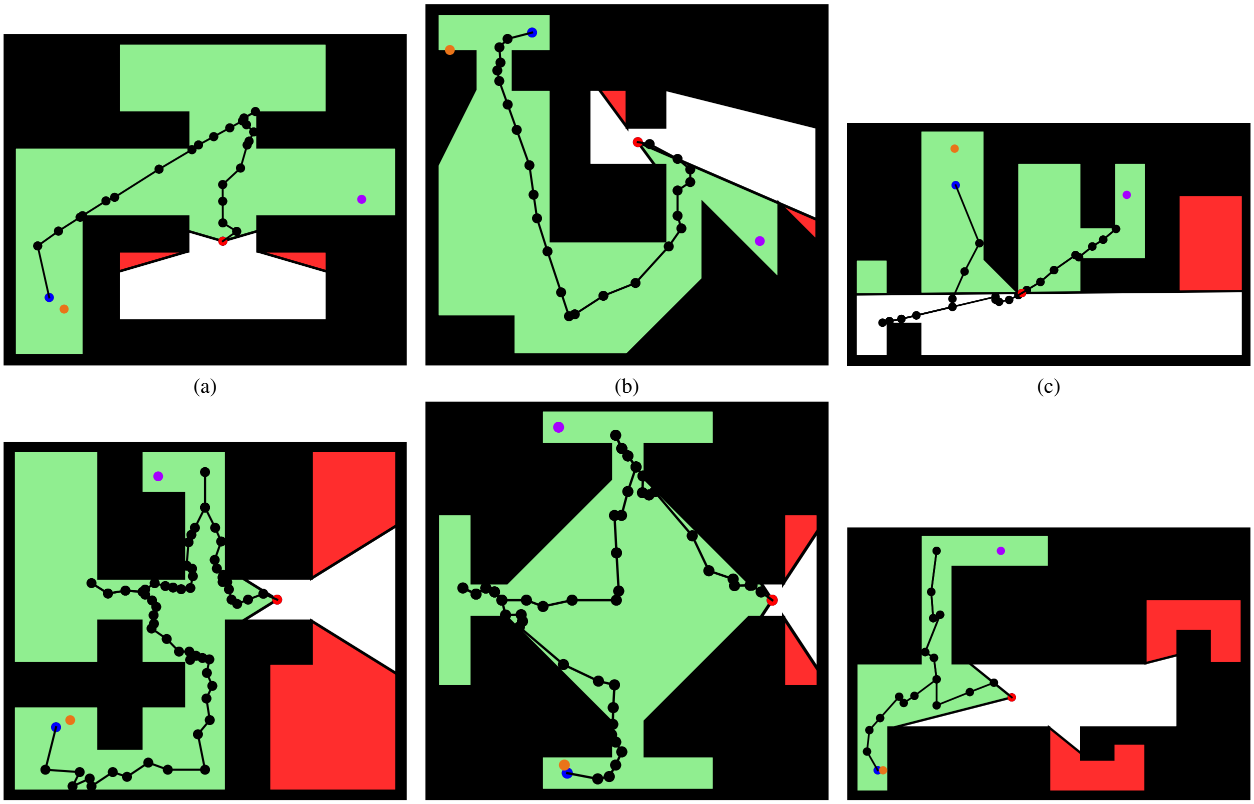

Fig. 10: Escort paths and final problem states computed by our algorithm in a variety of problem instances. Blue dots represent the starting positions of the escort, orange dots show the starting position of the VIP, red dots depict the end position of the escort, and purple dots mark the end/goal position for the VIP. The black lines depict the path taken by the escort. Green regions represent VIP-reachable safe zones. Red regions represent contaminated shadows.

[8] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.

[9] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani, "A visibility-based pursuit-evasion problem," *International Journal of Computational Geometry & Applications*, vol. 9, no. 04n05, pp. 471–493, 1999.

[10] H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *Proc. IEEE Conference on Decision and Control and European Control Conference*, 2011.

[11] K. Ichihara, T. Hasegawa, H. Ichikawa, and Y. Naruse, "Waypoint-based human-tracking navigation for museum guide robot," *Journal of Robotics and Mechatronics*, vol. 34, no. 5, pp. 1192–1204, 2022.

[12] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 875–884, 2005.

[13] M.-h. Jiao, H.-x. Wei, B.-w. Zhang, J.-q. Jin, Z.-q. Jia, and J.-l. Yan, "Path planning of escort robot based on improved quantum particle swarm optimization," in *Chinese Control And Decision Conference*. IEEE, 2019, pp. 3730–3735.

[14] S. LaValle and J. Hinrichsen, "Visibility-based pursuit-evasion: the case of curved environments," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 196–202, 2001.

[15] T. Olsen, N. M. Stiffler, and J. M. O'Kane, "Robust-by-design plans for multi-robot pursuit-evasion," in *Proc. International Conference on Robotics and Automation*, 2022, pp. 10 716–10 722.

[16] T. Olsen, N. M. Stiffler, and J. M. O'Kane, "Rapid recovery from robot failures in multi-robot visibility-based pursuit-evasion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 9734–9741.

[17] T. Olsen, A. M. Tumlin, N. M. Stiffler, and J. M. O'Kane, "A visibility roadmap sampling approach for a multi-robot visibility-based pursuit-evasion problem," in *Proc. IEEE International Conference on Robotics and Automation*, 2021, pp. 7957–7964.

[18] S. Pan, H. Huang, J. Ding, W. Zhang, D. M. S. vić, and C. J. Tomlin, "Pursuit, evasion and defense in the plane," in *Proc. American Control Conference*, 2012, pp. 4167–4173.

[19] T. D. Parsons, "Pursuit-evasion in a graph," in *Theory and Application of Graphs*, Y. Alavi and D. R. Lick, Eds. Berlin: Springer-Verlag, 1976, pp. 426–441.

[20] F. Shkurti and G. Dudek, "On the complexity of searching for an evader with a faster pursuer," in *Proc. IEEE International Conference on Robotics and Automation*, 2013.

[21] N. M. Stiffler and J. M. O'Kane, "A complete algorithm for visibility-based pursuit-evasion with multiple pursuers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1660–1667.

[22] ——, "A sampling-based algorithm for multi-robot visibility-based pursuit-evasion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1782–1789.

[23] N. M. Stiffler and J. M. O'Kane, "Complete and optimal visibility-based pursuit-evasion," *International Journal of Robotics Research*, vol. 36, no. 8, pp. 923–946, 2017.

[24] B. Tovar and S. M. LaValle, "Visibility-based pursuit—evasion with bounded speed," *International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1350–1360, 2008.

[25] J. Yu and S. M. LaValle, "Shadow information spaces: Combinatorial filters for tracking targets," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 440–456, 2011.