Dynamic Games and Applications (2020) 10:391–416
https://doi.org/10.1007/s13235-019-00327-x

Evasive Path Planning Under Surveillance Uncertainty

Marc Aurèle Gilles[1] · **Alexander Vladimirsky**[2]

Published online: 21 October 2019

Abstract
The classical setting of optimal control theory assumes full knowledge of the process dynamics and the costs associated with every control strategy. The problem becomes much harder if the controller only knows a finite set of possible running cost functions, but has no way of checking which of these running costs is actually in place. In this paper we address this challenge for a class of evasive path planning problems on a continuous domain, in which an evader needs to reach a target while minimizing his exposure to an enemy observer, who is in turn selecting from a finite set of known surveillance plans. Our key assumption is that both the evader and the observer need to commit to their (possibly probabilistic) strategies in advance and cannot immediately change their actions based on any newly discovered information about the opponent's current position. We consider two types of evader behavior: in the first one, a completely risk-averse evader seeks a trajectory minimizing his *worst-case* cumulative observability, and in the second, the evader is concerned with minimizing the *average-case* cumulative observability. The latter version is naturally interpreted as a semi-infinite strategic game, and we provide an efficient method for approximating its Nash equilibrium. The proposed approach draws on methods from game theory, convex optimization, optimal control, and multiobjective dynamic programming. We illustrate our algorithm using numerical examples and discuss the computational complexity, including for the generalized version with multiple evaders.

Keywords Path planning · Semi-infinite games · Nash equilibrium · Surveillance evasion · Convex optimization · Hamilton–Jacobi PDEs

Mathematics Subject Classification 49N75 · 49N90 · 49K35 · 91A05 · 90C29

✉ Alexander Vladimirsky
vladimirsky@cornell.edu

Marc Aurèle Gilles
mtg79@cornell.edu

[1] Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA

[2] Department of Mathematics, Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA

1 Introduction

Path planning is a problem of interest for many communities: traffic engineering, autonomous driving, robotics, and military. In the classical setting, the path planner is assumed to have full information about the environment and chooses a path minimizing some undesirable quantity, e.g., time-to-target, distance traveled, fuel consumption, or threat exposure. A particular type of continuous path planning problems is surveillance-evasion applications. In the simplest scenario, an evader (E) is choosing a path to minimize its exposure to an observer (O) whose surveillance plan is fixed and fully known to E in advance. This formulation is conveniently treated in the framework of *optimal control theory*, reviewed in Sect. 2, with the evader's optimal policy recovered by solving a Hamilton–Jacobi–Bellman (HJB) partial differential equation (PDE). But the real focus of this paper is on path planning under uncertainty, where E knows the full list of different surveillance plans available to O but does not know which of them is currently in use.

The key assumption of our model is that neither E nor O can change their respective strategy in real time based on the opponent's discovered position or actions. In practical contexts (e.g., in satellite-based surveillance), this restriction might be due to either a delayed analysis of observations or due to logistical needs of committing to a strategy in advance. As in many other optimization under uncertainty situations, it is natural for E to treat this as an *adversarial* problem—either because the prior statistics on the frequency of use for specific surveillance plans are unreliable or because O might be actively adjusting these frequencies in response to E's routing choices.

In considering each potential path to its destination, E needs to evaluate the trade-offs in observability with respect to different surveillance plans. This naturally brings us to the notion of *Pareto optimality* [23] and the numerical methods developed for multiobjective optimal control problems [13,18,19,25]. As we show in Sect. 3, the method introduced in [19] can be used to find the deterministic optimal policy for a completely risk-averse evader (i.e., minimizing the worst-case observability). Unfortunately, the computational cost of this approach grows exponentially with the number of surveillance plans available to O. But if the goal for both players is to optimize the average-case/expected observability, we show that this can be accomplished by adopting a much more computationally affordable method from [25], despite its significant drawbacks for general multiobjective control problems. Moreover, we show that if the evader's average-case optimal strategy is deterministic, then that same strategy is also worst-case optimal.

For the rest of the paper, we concentrate on the average-case observability formulation using a semi-infinite zero-sum game [35] between E and O, each of them searching for the best randomized/mixed strategy—an optimal probability distribution over that player's available deterministic/"pure" options. We refer to these as "surveillance-evasion games" (SEGs), although the same terminology was previously used in the 1960s and 1970s to describe a very different class of problems, where the evader needs to escape from the observer's surveillance zone as quickly as possible [15,20–22]. Aside from this terminological overlap, those earlier papers have little in common with our context since in them E and O operated with full information on their opponent's current state, reacted in real time, and sought optimal feedback policies recovered by solving Hamilton–Jacobi–Isaacs equations.

In classical (finite zero-sum two-player) strategic games, the Nash equilibrium is typically obtained using linear programming [26]. But the fact that E's set of pure strategies is uncountably infinite makes this approach unusable in our SEGs. Instead, we show how to compute the Nash equilibrium in Sect. 4 by combining convex optimization with fast numer-

ical methods for HJB equations. The computational cost of the resulting method scales at most linearly with the number of surveillance plans. We illustrate this approach on a large number of examples, with the details of our numerical implementation covered in Sect. 5.

We note that the same ideas are also useful outside of surveillance-evasion context, whenever the path planner cannot assess the actually incurred running cost until it reaches the target. In fact, the same PDEs and semi-infinite zero-sum games can be used to model civilians' routing choices in war zones and other dangerous environments, minimizing their exposure to bomb threats.

Our modeling approach is quite general, but to simplify the exposition we will assume that the evader is moving in a two-dimensional domain with occluding/impenetrable obstacles, both the observability and E's speed are *isotropic* (i.e., independent of E's chosen direction of motion), and all O's surveillance plans are stationary (i.e., the observer is choosing among possible stationary locations). This further simplifies the PDE aspect of our problem from a general HJB to stationary *Eikonal* equations, the efficient numerical methods for which are particularly well developed in the last 25 years (e.g., [30]).

In Sect. 6, we generalize the problem by considering multiple evaders. We first treat this as a two-player game between a single observer and a centralized controller of all evaders. But we also show that the resulting set of strategies is a Nash equilibrium even from the point of view of individual/selfish evaders. We conclude by discussing further extensions and limitations of our approach in Sect. 7.

2 Continuous Path Planning

The case where the observer's strategy is fixed and known can be easily handled by methods of classical optimal control theory. The goal is to guide an evader (E) from its starting position $\boldsymbol{x}_S$ to its desired target $\boldsymbol{x}_T$ while minimizing the "cumulative observability" (also called "cumulative cost") along the way through its state space represented by some compact set $\Omega \subset \mathbb{R}^d$. More precisely, we will suppose that A is a compact set of control values, and $\mathcal{A}$ is the set of E's admissible controls which are measurable functions $\boldsymbol{a} : \mathbb{R} \mapsto A$. The evader's dynamics are defined by $\boldsymbol{y}'(t) = \boldsymbol{f}(\boldsymbol{y}(t), \boldsymbol{a}(t))$, with the initial state $\boldsymbol{y}(0) = \boldsymbol{x} \in \Omega$. (Even though E only cares about the optimal trajectory from $\boldsymbol{x}_S$, the method we use encodes optimal trajectories to $\boldsymbol{x}_T$ from all starting positions $\boldsymbol{x}$.) In all of our numerical examples, we will assume that E's state is simply its position, $\boldsymbol{f}$ is its velocity defined on a known map Ω that excludes (impenetrable, occluding) obstacles, and E is allowed to travel along $\partial\Omega$ (including the obstacle boundaries). Suppose $T_{\boldsymbol{a}} = \min\{t \geq 0 \mid \boldsymbol{y}(t) = \boldsymbol{x}_T\}$ is the travel-time-through-Ω associated with this control. A pointwise observability function (also called cost function) $K : \Omega \times A \mapsto \mathbb{R}$ is then defined to reflect O's surveillance capabilities for different parts of the domain, taking into account all obstacles/occluders and E's current position and direction. The cumulative observability is then defined by integrating K along a trajectory corresponding to $\boldsymbol{a}(\cdot)$ with initial position $\boldsymbol{x}$

$$\mathcal{J}(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \int_0^{T_a} K(\boldsymbol{y}(t), \boldsymbol{a}(t))\, \mathrm{d}t, \tag{2.1}$$

which we will also denote as $\mathcal{J}(\boldsymbol{a}(\cdot))$ when $\boldsymbol{x}$ is clear from the context. As usual in dynamic programming, the *value function* is then defined by minimizing the cumulative observability: $u(\boldsymbol{x}) = \inf_{\boldsymbol{a}(\cdot)} \mathcal{J}(\boldsymbol{x}, \boldsymbol{a}(\cdot))$, with the infimum taken over controls leading to $\boldsymbol{x}_T$ without leaving Ω (i.e., $T_{\boldsymbol{a}} < \infty$ and $\boldsymbol{y}(t) \in \Omega, \forall t \in [0, T_{\boldsymbol{a}}]$ along the corresponding trajectory). Under

suitable "small-time controllability" assumptions [2], it is easy to show that u is locally Lipschitz on Ω. If it is also smooth, a Taylor series expansion can be used to show that u satisfies a static Hamilton–Jacobi–Bellman PDE:

$$\min_{\boldsymbol{a}\in A}\{K(\boldsymbol{x},\boldsymbol{a}) + \nabla u(\boldsymbol{x})\cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\} = 0, \quad \forall \boldsymbol{x}\in\Omega\backslash\{\boldsymbol{x}_T\}; \quad u(\boldsymbol{x}_T)=0, \tag{2.2}$$

with the special treatment at $\partial\Omega\backslash\{\boldsymbol{x}_T\}$ where the minimum is taken over the subset of control values A that ensure staying inside Ω.

Unfortunately, the value function u is generically non-smooth, and there usually are starting positions with multiple optimal trajectories—these are the locations where the characteristics cross and ∇u is undefined. Thus, a classical solution to (2.2) usually does not exist. The theory of *viscosity solutions* introduced by Crandall and Lions [11] overcomes this difficulty by selecting the unique weak solution coinciding with the value function of the original control problem. Restricting the process dynamics to Ω is similarly handled by switching to domain-constrained viscosity solutions [2,33].

To simplify the exposition, we focus on *isotropic* problems, where the observability K and the speed of motion f depend only on $\boldsymbol{x}$. In this case, we choose $A=\{\boldsymbol{a}\in\mathbb{R}^d \mid |\boldsymbol{a}|=1\}$ and interpret $\boldsymbol{a}$ as the direction of motion. Then, $K(\boldsymbol{x},\boldsymbol{a})=K(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})=f(\boldsymbol{x})\boldsymbol{a}$, with f encoding the speed of motion through the point $\boldsymbol{x}$. In this case, the optimal direction is known analytically: $\boldsymbol{a}^*=-\nabla u/|\nabla u|$ and (2.2) reduces to an *Eikonal equation*

$$|\nabla u(\boldsymbol{x})| f(\boldsymbol{x}) = K(\boldsymbol{x}), \quad \forall \boldsymbol{x}\in\Omega\backslash\{\boldsymbol{x}_T\}; \quad u(\boldsymbol{x}_T)=0. \tag{2.3}$$

The characteristics of these static PDEs are precisely the optimal trajectories, which define the direction of "information flow". This is quite useful once (2.3) is discretized on a grid (e.g., substituting upwind divided differences for partial derivatives, while taking $u=+\infty$ for all gridpoints outside of Ω to enforce the state constraints). The discretization yields a large coupled system of nonlinear equations. Knowing the characteristic direction for every gridpoint, one could, in principle, reorder the equations, effectively decoupling this system. But since the PDE is nonlinear, its characteristic directions are not known in advance. One path[1] to computational efficiency is to determine those characteristic directions simultaneously with solving the discretized system, in the spirit of Dijkstra's classical algorithm for finding shortest paths on graphs [14]. Two such non-iterative methods (Tsitsiklis' algorithm [37] and Sethian's fast marching method [29]) are applicable to this special isotropic case. Once (2.3) is solved, the optimal trajectory may be recovered by finding the path orthogonal to the level curves of $u(x)$. This can be achieved numerically by the steepest descent method on $u(x)$. An example of the solution of (2.3) is shown in Fig. 1.

3 Multiple Observer Locations and Different Notions of Optimality

We now transition to the setting where the observer has a choice of multiple surveillance plans. Assuming that the observer remains stationary, this is equivalent to choosing its position from a fixed set of r locations $\mathcal{X}=\{\hat{\boldsymbol{x}}_1,\ldots,\hat{\boldsymbol{x}}_r\}$ known to the evader. Each location is associated

[1] Fast sweeping [39] is another popular approach for gaining efficiency in solving Eikonal equations. We refer readers to [7,8] for a review of many other "fast" techniques, including the hybrid marching/sweeping methods that aim to combine the best features of both approaches. Even though our own implementation is based on fast marching, any of these methods can be used to solve isotropic control problems arising in subsequent sections. Which one will be faster depends on the domain geometry and the particular pointwise observability functions.

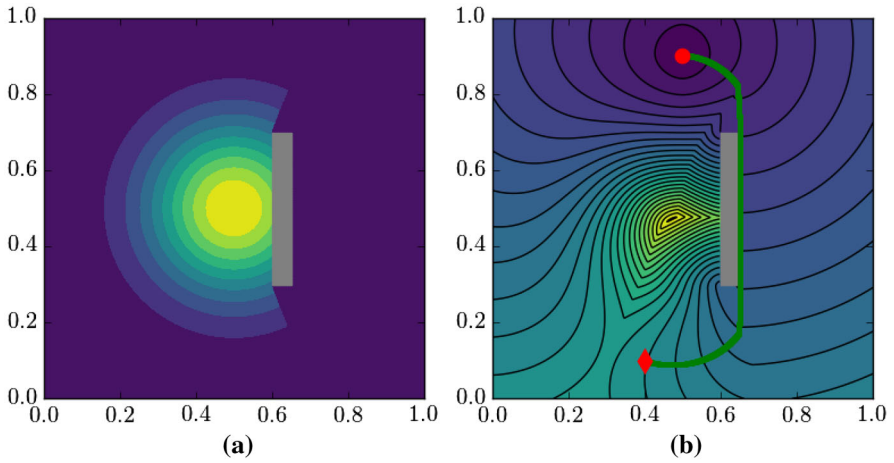

Fig. 1 **a** Observability function $K(\boldsymbol{x})$ for an observer position (0.5, 0.5). The gray rectangle is an obstacle, which obstructs the vision of the observer. The shadow zones created by the obstacle can be computed using the solution of the Eikonal equation (see Sect. 5.1). **b** Contour plot of the solution of (2.3) for $f(\boldsymbol{x}) = 1$ and the cost function in (**a**). The red diamond is the starting position, the red circle is the target position, and the green curve is the optimal trajectory, which is orthogonal to the level curves of $u(x)$ and follows a part of the obstacle boundary. See Sect. 5 for additional information and parameters used (Color figure online)

with a pointwise observability function $K_i(\boldsymbol{x})$ for an evader moving through $\boldsymbol{x} \in \Omega$ and an observer stationed at $\hat{\boldsymbol{x}}_i$. (Typically, K_i is a decreasing function of $|\boldsymbol{x} - \hat{\boldsymbol{x}}_i|$ when $\boldsymbol{x}$ is visible from $\hat{\boldsymbol{x}}_i$ or a small constant $\sigma > 0$ if $\boldsymbol{x}$ is in a "shadow zone"; see Sect. 5 for further details.) This results in r different definitions of the cumulative observability $\boldsymbol{\mathcal{J}} = \left[\mathcal{J}_1, \ldots, \mathcal{J}_r\right]^T$ for a particular control. Ideally, a rational evader would prefer a path that minimizes its exposure to all possible observer locations $\hat{\boldsymbol{x}}_i$. Unfortunately, there usually does not exist a single control minimizing all $\mathcal{J}_i$'s simultaneously. This naturally leads us to a notion of Pareto optimal trajectories and the methods for computing them efficiently. We review two such methods[2] in Sect. 3.1 and explain how they can be used for planning by an evader optimizing either the worst-case or average-case observability in Sect. 3.2.

3.1 Multiobjective Path Planning

For a fixed starting position $\boldsymbol{x} \in \Omega$, a control $\boldsymbol{a}(\cdot)$ is *dominated* by a control $\hat{\boldsymbol{a}}(\cdot)$ if $\mathcal{J}_i(\boldsymbol{x}, \hat{\boldsymbol{a}}(\cdot)) \leq \mathcal{J}_i(\boldsymbol{x}, \boldsymbol{a}(\cdot))$ for all i and the inequality is strict for at least one of them. We call $\boldsymbol{a}(\cdot)$ *Pareto optimal* if it is not dominated by any other control. In other words, Pareto optimal controls are the ones that cannot be improved with respect to any one criterion without making them worse with respect to another. The vector of costs associated with each Pareto optimal control defines a point in $\mathbb{R}^r$ and the set of all such points is the Pareto front (PF). In path planning applications, the PF is typically used to carefully evaluate all trade-offs. (For example, what is the smallest attainable $\mathcal{J}_1$ given the desired upper bounds on $\mathcal{J}_2, \ldots, \mathcal{J}_r$?)

[2] Here, we describe these methods in terms of exposure to different observer's positions, but both of them were introduced for much more general multiobjective control problems. In many applications, it is necessary to balance completely different criteria, e.g., time vs fuel vs money vs threat, etc. Other methods for approximating the full PF can be found in [13] and [18].

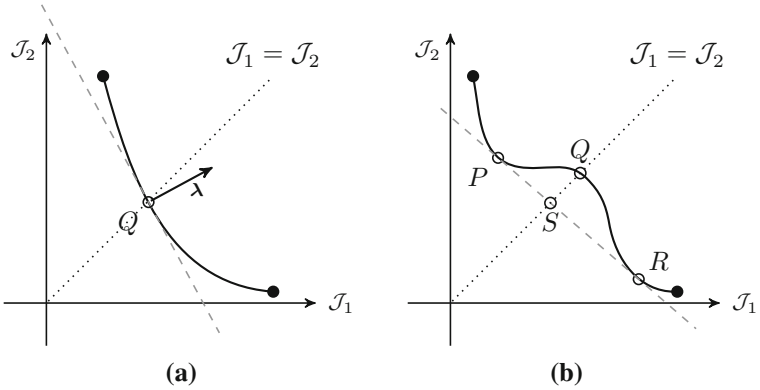

Fig. 2 **a** Convex smooth Pareto front with a point Q corresponding to the worst-case optimal $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) \in [0, 1]^2$. The line perpendicular to $\boldsymbol{\lambda}$ is tangent to PF at Q. If any part of PF fell below it, the path corresponding to Q would not be $\boldsymbol{\lambda}$-optimal. The dotted line is the central ray (where $\mathcal{J}_1 = \mathcal{J}_2$)). **b** Non-convex smooth Pareto front. Points P and R correspond to two different $\boldsymbol{\lambda}$-optimal paths. The portion of PF between P and R (including the worst-case optimal point Q) cannot be found by scalarization. Point S, found as a convex combination of P and R, is average-case optimal

Mitchell and Sastry developed a method for multiobjective path planning [25] based on the usual *scalarization* approach to multiobjective optimization [23]. Let $\Delta_r = \{\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_r) \mid \sum_{i=1}^r \lambda_i = 1, \text{ and all } \lambda_i \geq 0\}$. For each $\boldsymbol{\lambda} \in \Delta_r$, one can define a new pointwise observability function $K^{\boldsymbol{\lambda}} = \sum_{i=1}^r \lambda_i K_i$ and a new cumulative observability function $\mathcal{J}^{\boldsymbol{\lambda}} = \sum_i \mathcal{J}_i$. A weighted cost Eikonal equation

$$|\nabla u^{\boldsymbol{\lambda}}(\boldsymbol{x})| f(\boldsymbol{x}) = K^{\boldsymbol{\lambda}}(\boldsymbol{x}) \tag{3.1}$$

is then solved for a fixed $\boldsymbol{\lambda}$, providing a control function $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)$ satisfying $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot) \in \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{x}_S, \boldsymbol{a}(\cdot))$. We call such a control function $\boldsymbol{\lambda}$-optimal. If $\lambda_i > 0$ for all i, the obtained $\boldsymbol{\lambda}$-optimal control is also guaranteed to be Pareto optimal; see Fig. 2. However, if at least one $\lambda_i = 0$ and multiple $\boldsymbol{\lambda}$-optimal strategies exist for a specific $\boldsymbol{\lambda}$, then some of the $\boldsymbol{\lambda}$-optimal strategies may not be Pareto optimal. Such corner cases (illustrated in Fig. 5) might require additional pruning; alternatively, one can rule out such non-Pareto trajectories by perturbing $\boldsymbol{\lambda}$ to ensure that all components are positive.

Additional linear PDEs can be solved simultaneously to compute the individual costs $(\mathcal{J}_1, \ldots, \mathcal{J}_r)$ incurred along any $\boldsymbol{\lambda}$-optimal trajectory; for example, when f and all K_i's are isotropic, the corresponding linear equations are

$$\nabla v_i^{\boldsymbol{\lambda}} \cdot \nabla u^{\boldsymbol{\lambda}} = K_i K^{\boldsymbol{\lambda}} / f^2, \tag{3.2}$$

where $v_i^{\boldsymbol{\lambda}}(\boldsymbol{x}) = \mathcal{J}_i\left(\boldsymbol{x}, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)\right)$.

To approximate the PF, this procedure is repeated for a large number of $\boldsymbol{\lambda} \in \Delta_r$. Unfortunately, as shown in Fig. 2, scalarization-based methods can only recover the convex portion of PF [12]. This is an important drawback since in many optimal control problems, the non-convex parts of PF are very common and equally important. An alternative approach was developed in [19] to address this limitation and produce the entire PF for all $\boldsymbol{x} \in \Omega$ simultaneously. The method is applicable for any number of observer positions, but to simplify the notation we explain it here for $r = 2$ only. We expand the state space to $\Omega_e = \Omega \times [0, B]$

and define the new value function $w(\boldsymbol{x}, b) = \inf \mathcal{J}_1(\boldsymbol{x}, \boldsymbol{a}(\cdot))$, with the infimum taken over all controls satisfying $\mathcal{J}_2(\boldsymbol{x}, \boldsymbol{a}(\cdot)) \leq b$. Thus, b is naturally interpreted as the current "budget" for the secondary criterion. The value function is then recovered by solving an augmented PDE

$$\min_{\boldsymbol{a}\in A} \left\{ K_1(\boldsymbol{x}, \boldsymbol{a}) + \nabla_{\boldsymbol{x}} w \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) \; - \; K_2(\boldsymbol{x}, \boldsymbol{a}) \frac{\partial w}{\partial b} \right\} \; = \; 0. \tag{3.3}$$

The method in [19] uses a first-order accurate semi-Lagrangian discretization [16] to compute the discontinuous viscosity solution of (3.3) for a range of problems in multi-criterion path planning. The method was later generalized to treat constraints on reset-renewable resources [34]. The same approach was also adapted to Probabilistic RoadMap graphs and field tested on robotic platforms at the United Technologies Research Center [10].

Aside from approximating the entire PF, the key computational advantage is the *explicit causality*: since K_2 is positive, all characteristics are monotone in b and methods similar to the explicit "forward marching" in b-direction are applicable (i.e., the system of discretized equations is trivially de-coupled). Of course, the main drawback of the above idea is the higher dimensionality of Ω_e. For r observer locations, the scalarization approach [25] requires solving $(r + 1)$ PDEs on $\Omega \subset \mathbb{R}^d$, but the parameter space Λ_r is $(r - 1)$-dimensional. In contrast, with $w(\boldsymbol{x}, b)$ there are no parameters, but the computational domain is $(d + r - 1)$-dimensional. Several techniques for restricting the computations to a relevant part of Ω_e were developed in [19], but the computational cost and memory requirements are still significantly higher than for any (single) HJB solve in Ω.

3.2 Different Notions of Adversarial Optimality

The Pareto front allows us to answer one version of the surveillance-evasion problem: if the evader is completely risk-averse, he may choose the *worst-case optimal* strategy. That is, E will pick a control $\boldsymbol{a}_W(\cdot)$ that minimizes the observability from its "worst" observer position $\hat{\boldsymbol{x}}_i$:

$$\max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_W(\cdot)) \; \leq \; \max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}(\cdot)), \quad \forall \boldsymbol{a}(\cdot) \in \mathcal{A}.$$

This corresponds to the version of the problem where E is forced to "go first", with O selecting the maximizing $\hat{\boldsymbol{x}}_i \in \mathcal{X}$ in response. The following result shows that the intersection of Pareto front with the "central ray" (i.e., the line where $\mathcal{J}_1 = \mathcal{J}_2 \cdots = \mathcal{J}_r$) yields the worst-case optimal strategy for E:

Theorem 3.1 *If $\boldsymbol{a}_=(\cdot)$ is a Pareto optimal control satisfying $\mathcal{J}_i(\boldsymbol{a}_=(\cdot)) = \mathcal{J}_j(\boldsymbol{a}_=(\cdot))$ for all $i, j \in \{1, \ldots, r\}$, then $\boldsymbol{a}_=(\cdot)$ is also worst-case optimal.*

Proof Suppose there exists $\boldsymbol{a}'(\cdot)$ s.t.

$$\max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}'(\cdot)) \; < \; \max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_=(\cdot)).$$

Then, for all j we have:

$$\mathcal{J}_j(\boldsymbol{a}'(\cdot)) \leq \max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}'(\cdot)) < \max_{\hat{\boldsymbol{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_=(\cdot)) = \mathcal{J}_j(\boldsymbol{a}_=(\cdot)),$$

which contradicts the Pareto optimality of $\boldsymbol{a}_=(\cdot)$. □

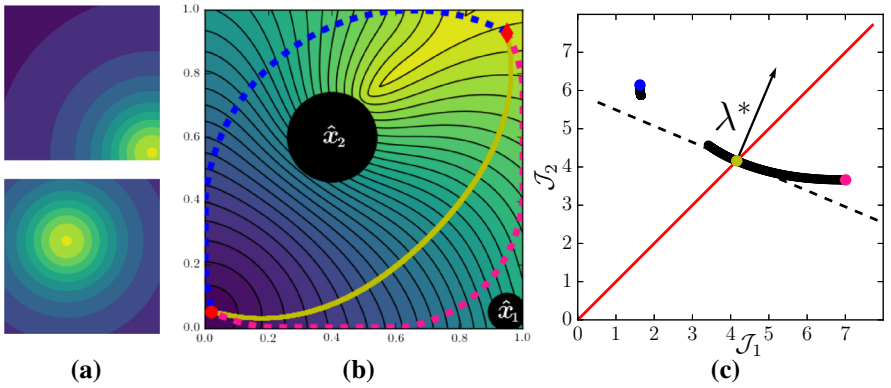

Fig. 3 **a** Two observer positions and the corresponding observability maps K_i. **b** $\boldsymbol{\lambda}^*$-optimal path corresponding to $\boldsymbol{\lambda}^* \approx (0.30, 0.70)$ is shown in yellow over the level sets of $u^{\boldsymbol{\lambda}^*}$. The radii of black disks centered at $\hat{\boldsymbol{x}}_i'$s are proportional to the corresponding components of $\boldsymbol{\lambda}^*$. The two best-response trajectories used when O chooses $\hat{\boldsymbol{x}}_1$ or $\hat{\boldsymbol{x}}_2$ are shown in blue and pink, respectively. The trajectory in yellow is worst-case optimal for the evader as it is equally observable from both locations. **c** The convex part of Pareto front (computed using the scalarization approach) intersects the "central ray" ($\mathcal{J}_1 = \mathcal{J}_2$, shown in red). The worst-case optimal vector $\boldsymbol{\lambda}^*$ is orthogonal to PF at the point of intersection (in yellow), whose coordinates correspond to the partial costs of the optimal path. The probability distribution $\boldsymbol{\lambda}^*$, together with the yellow trajectory, form a Nash equilibrium of the strategic game between the evader and the observer described in Sect. 4. See Sect. 5 for additional information and parameters used (Color figure online)

The corresponding vector of costs $\boldsymbol{\mathcal{J}}(\boldsymbol{a}_=(\cdot))$ may lie on the convex portion of PF, as in Figs. 2a and 3, in which case $\boldsymbol{a}_W = \boldsymbol{a}_=$ can be found by scalarization [25]. But if $\boldsymbol{\mathcal{J}}(\boldsymbol{a}_=(\cdot))$ lies on the non-convex portion of PF, as in Figs. 2b and 4, the computational cost of finding the evader's worst-case optimal strategy grows exponentially with r as it involves solving a nonlinear differential equation in $(r + d - 1)$ dimensions [19]. As it will be shown in Sects. 4–6, the latter scenario is particularly common on domains with obstacles.

Luckily, another interpretation of evader's objectives proves much more computationally tractable. Even though $\boldsymbol{a}_=(\cdot)$ yields the lowest *worst-case* observability that E can achieve if he must choose a single control function deterministically, E might be able to attain an even lower expected (or *average-case*) observability if he switches to "mixed policies", choosing a probability distribution over a set of Pareto optimal controls. This is illustrated in Fig. 2b: by choosing probabilistically a path corresponding to the point P and another corresponding to point R, E obtains a new point S on the central ray, whose expected observability is lower than for the worst-case optimal Q regardless of O's selected location. This, of course, assumes that O's location is selected without knowing in advance which of the two paths will be used by E. Indeed, for any single run from $\boldsymbol{x}_S$ to $\boldsymbol{x}_T$, the *worst-case* observability of this probabilistic policy is based on the worst cases for P and R, which (from the point of view of a completely risk-averse evader) would make the average-case optimal S inferior to the worst-case optimal Q. This scenario is fully realized in Fig. 4, where $J_1(\boldsymbol{a}_=(\cdot)) = J_2(\boldsymbol{a}_=(\cdot)) \approx 4.94$, the expected observability corresponding to the optimal "probabilistic mix" of yellow and green trajectories is ≈ 4.83, but the worst case associated with this mixed policy is $J_1(\text{yellow}) \approx 6.03$.

We note that O could also consider using a mixed strategy. In this case, $K^{\boldsymbol{\lambda}}$ can be interpreted as the expected pointwise observability when using the probability density $\boldsymbol{\lambda} \in \Delta_r$ over the positions $\mathcal{X}$. Similarly, $\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))$ is the expected cumulative observability when using

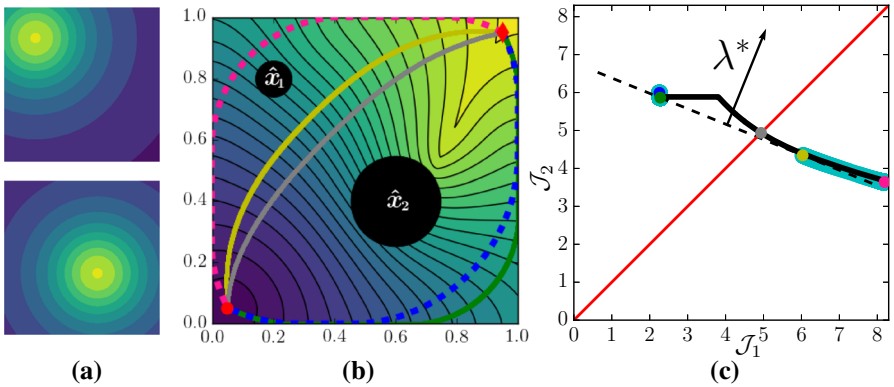

Fig. 4 **a** Two observer positions and the corresponding observability maps K_i. **b** Two $\boldsymbol{\lambda}^*$-optimal trajectories corresponding to $\boldsymbol{\lambda}^* \approx (0.29, 0.71)$ are shown in yellow and green over the level sets of $u^{\boldsymbol{\lambda}^*}$. The two best-response trajectories used when O chooses $\hat{\boldsymbol{x}}_1$ or $\hat{\boldsymbol{x}}_2$ are shown in blue and pink, respectively. The worst-case optimal trajectory is plotted in gray. **c** Convex part of the Pareto front (in cyan) computed using the scalarization approach, and the whole Pareto front (in black) computed using the method in [19]. The convex part of the Pareto front does not intersect the central ray (shown in red). The worst-case optimal strategy (in gray) lies on the non-convex part of the Pareto front and thus cannot be computed using scalarization. The Nash equilibrium pair of strategies consists of using positions $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ with probabilities $\boldsymbol{\lambda}^*$ for O and using the yellow and green trajectories (both of which lie on the convex part of the PF) with probability $\big[p(\text{yellow}),\, p(\text{green})\big] = [0.29, 0.71]$ for E (see Sect. 4). The latter mixed strategy is average-case optimal for E. See Sect. 5 for additional information and parameters used (Color figure online)

the control function $\boldsymbol{a}(\cdot)$. Figure 2 shows that when we are interested in the average-case optimal behavior for both O and E, we only need to consider a convex hull of PF (denoted $co(\text{PF})$), and the scalarization is thus adequate. Note that in Figs. 3, 4, and 6, the set $co(\text{PF})$ was approximated by imposing a fine grid on Δ_r and resolving (3.1) for each sampled $\boldsymbol{\lambda}$. Since we only care about the intersection of $co(\text{PF})$ with the central ray, this procedure is wasteful—and prohibitively expensive for high r. In the next section, we consider the case where both E and O optimize the expected/average-case performance by reformulating this as a semi-infinite strategic zero-sum game. We show that such surveillance-evasion games (SEGs) can be solved through scalarization combined with convex optimization, without approximating the (convex hull of the) entire Pareto front.

Remark 3.2 Up till now, our geometric interpretation in Figs. 3, 4, and 6 assumed that either PF or at least the $co(\text{PF})$ must intersect the central ray. If this is not the case, O will avoid using some of his positions. For example, Fig. 5 shows the pink and yellow trajectories corresponding to $\boldsymbol{a}_1(\cdot)$ and $\boldsymbol{a}_2(\cdot)$, which are optimal with respect to the observer positions $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$. Since $\mathcal{J}_1(\boldsymbol{a}_2(\cdot)) \leq \mathcal{J}_2(\boldsymbol{a}_2(\cdot))$, the E's worst case for $\boldsymbol{a}_2(\cdot)$ is actually the observer location $\hat{\boldsymbol{x}}_2$. A generalization of this scenario for $r > 2$ is covered in Theorem 4.2.

4 Surveillance-Evasion Games (SEGs)

In this section, we reformulate the problem of evasive path planning under surveillance uncertainty as a strategic game. This can model either the actual adversarial interactions between two players or the risk-averse logic of the evader even if the surveillance patterns

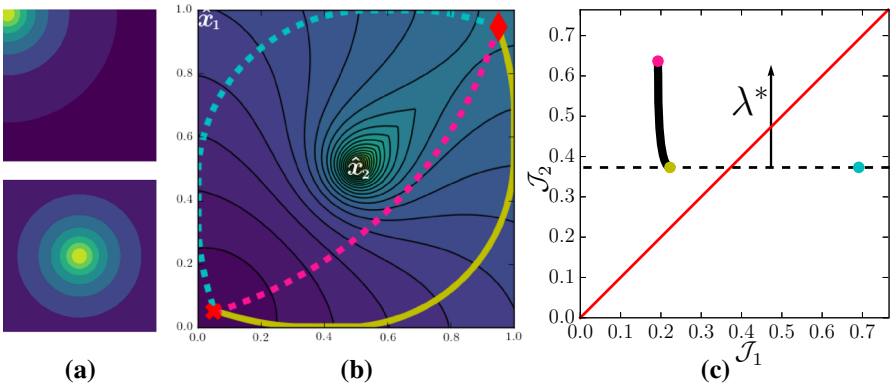

Fig. 5 **a** Two observer positions and the corresponding observability maps K_i plotted in logarithmic scale. **b** Value function $u^{\boldsymbol{\lambda}^*}$ at $\boldsymbol{\lambda}^* = (0, 1)$. The worst-case optimal strategy for O is the yellow trajectory, but both the yellow trajectory and the light blue trajectories are $\boldsymbol{\lambda}^*$-optimal. The pink trajectory is the best response when the observer uses position $\hat{\boldsymbol{x}}_1$. **c** Pareto front does not intersect the central ray. The worst-case optimal trajectory is the one point on the Pareto front that is closest to the central ray: the yellow point. The blue point is $\boldsymbol{\lambda}^*$-optimal, but it is not Pareto optimal as it is dominated by the yellow point. The Nash equilibrium strategy consists of the position $\hat{\boldsymbol{x}}_2$ for O, and the yellow trajectory for E (see Sect. 4). See Sect. 5 for additional information and parameters used (Color figure online)

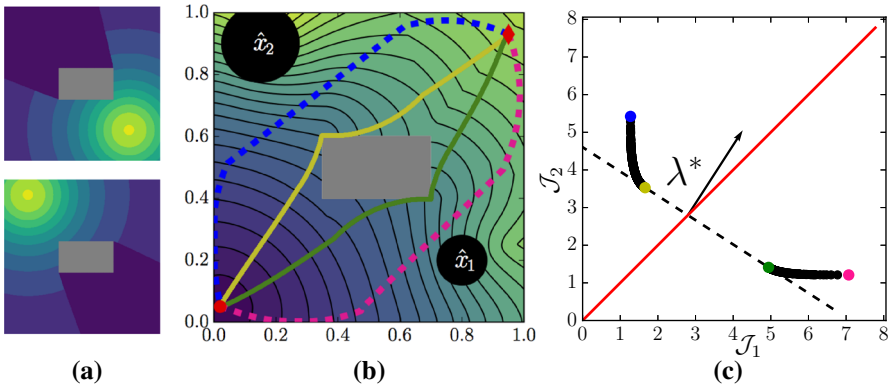

Fig. 6 **a** Two observer positions and the corresponding observability maps K_i on a domain with a single obstacle (shown in gray). **b** Two $\boldsymbol{\lambda}^*$-optimal trajectories corresponding to $\boldsymbol{\lambda}^* \approx (0.39, 0.61)$ are shown in yellow and green over the level sets of $u^{\boldsymbol{\lambda}^*}$. The two best-response trajectories used when O chooses $\hat{\boldsymbol{x}}_1$ or $\hat{\boldsymbol{x}}_2$ are shown in blue and pink, respectively. The trajectories in yellow and green are not worst-case optimal for the evader but are used in E's mixed Nash equilibrium strategy. **c** The convex part of the Pareto front does not intersect the central ray (shown in red). This is the same situation already observed in Fig. 4, but it is even more common on domains with obstacles. The Nash equilibrium pair of strategies consists of using positions $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ with probabilities $\boldsymbol{\lambda}^*$ for O and using the yellow and green trajectories with probability $\big[p(\text{yellow}),\, p(\text{green})\big] = [0.65, 0.35]$ for E. See Sect. 5 for additional information and parameters used (Color figure online)

are not likely to change in response to that evader's strategy. (The latter case is typically interpreted as a "game against nature".)

We assume that the evader is attempting to minimize (while the observer is attempting to maximize) the total expected observability integrated over E's trajectories and dependent

on O's positions. We further assume that O is aware of E's initial location $\boldsymbol{x}_S$ and its target location $\boldsymbol{x}_T$ but not of the trajectories chosen by E. Similarly, E is aware of the predefined locations of O, but not of the realized positions chosen by O. This game may be stated deterministically or stochastically. In the deterministic case, each player chooses a single pure strategy. That is, the observer chooses a single location $\hat{\boldsymbol{x}}_i \in \mathcal{X}$ and the evader chooses a single control function $\boldsymbol{a}(\cdot) \in \mathcal{A}$. In the probabilistic setting, each player chooses a mixed strategy, i.e., a probability distribution over the pure strategies. In other words, O chooses a probability distribution $\boldsymbol{\lambda} \in \Delta_r$ over positions and E chooses a probability distribution $\boldsymbol{\theta} \in \Delta_{\mathcal{A}}$ over control functions. The mixed strategy $\boldsymbol{\lambda}$ of the observer can be interpreted in several different ways:

1. O chooses a single position $\hat{\boldsymbol{x}}_i$ according to the probability distribution $\boldsymbol{\lambda}$ before E starts moving and remains at that position until the end of the round (that is, until E reaches the target).
2. O can randomly teleport between its positions at any time, and each λ_i reflects the proportion of time spent at the corresponding position $\hat{\boldsymbol{x}}_i$.
3. O has a budget of "observation resources", and $\boldsymbol{\lambda}$ reflects the fraction of these resources spent at each location. In this case, K_i reflects the pointwise observability corresponding to 100% of resources allocated to the position $\hat{\boldsymbol{x}}_i$.

Since we assume that neither player has access to the realization of the opponent's strategy in real time, these three interpretations are equivalent (and lead to the same optimal strategies) in our context. The payoff function of the game is the cumulative expected observability and can be expressed as $P(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right]$ where $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ denotes the expectation over the mixed strategy $\boldsymbol{\theta}$.

This SEG is a two-player *zero-sum game* [26], as each player's gains or losses are exactly balanced by the losses or gains of the opponent. Furthermore, it is *semi-infinite* as the set of pure strategies for O is a finite number r, whereas the set of pure strategies for E is uncountably infinite. A central notion of solution for strategic games is a Nash equilibrium [26], a pair of strategies for which neither player can improve his payoff by unilaterally changing his strategy. That is, a pair of strategies $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium if both of the following conditions hold:

$$\begin{aligned} P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) &\leq P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) \quad \text{for all } \boldsymbol{\theta} \in \Delta_{\mathcal{A}}\,, \\ P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) &\geq P(\boldsymbol{\lambda}, \boldsymbol{\theta}^*) \quad \text{for all } \boldsymbol{\lambda} \in \Delta_r\,. \end{aligned} \tag{4.1}$$

A pure strategy Nash equilibrium does not always exist, and therefore, we focus on finding a mixed strategy Nash equilibrium. In our setting, the minimax theorem for semi-infinite games [28] assures that a mixed strategy Nash equilibrium $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ exists, that all Nash equilibria have the same payoff, and that they are attained at the minimax (which is also equal to the maximin):

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \min_{\boldsymbol{\theta} \in \Delta_{\mathcal{A}}} \max_{\boldsymbol{\lambda} \in \Delta_r} \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right] = \max_{\boldsymbol{\lambda} \in \Delta_r} \min_{\boldsymbol{\theta} \in \Delta_{\mathcal{A}}} \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right]\,. \tag{4.2}$$

Although $\boldsymbol{\theta}$ is a probability distribution over the uncountable set $\Delta_{\mathcal{A}}$, there always exists an optimal mixed strategy $\boldsymbol{\theta}^*$ which is a mixture of at most r pure strategies, where r is the maximum number of positions for the observer [28]. In fact, it is easy to show that there will always exist a Nash equilibrium $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ with the number of pure strategies used in $\boldsymbol{\theta}^*$ not exceeding the number of nonzero components in $\boldsymbol{\lambda}^*$.

In the case of finite two-player zero-sum games, computing the Nash equilibrium is easily achieved by linear programming. For our SEGs, the challenge in computing a Nash equilibrium arises from enumerating the control functions $\boldsymbol{a}(\cdot) \in \mathcal{A}$ which are part of E's mixed

strategy. Indeed, we do not possess a useful parametrization of the set of control functions $\mathcal{A}$, and our only computational kernel to generate a single $\boldsymbol{\lambda}$-optimal control function $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)$ is to solve the weighted cost Eikonal equation in (3.1). For that reason, our solution strategy to compute the Nash Equilibrium involves two steps:

1. Find an approximate optimal strategy of the observer $\boldsymbol{\lambda}^*$ using convex optimization (see Sect. 4.1).
2. Find an approximate optimal strategy of the evader $\boldsymbol{\theta}^*$ by generating near-optimal control functions (see Sect. 4.2).

4.1 Optimal Strategy of the Observer

In order to compute an optimal strategy $\boldsymbol{\lambda}^*$ of the observer, we consider the following problem:

$$\max_{\boldsymbol{\lambda}\in\Delta_r} \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}\left(\boldsymbol{x}_S, \boldsymbol{a}(\cdot)\right) = \max_{\boldsymbol{\lambda}\in\Delta_r} u^{\boldsymbol{\lambda}}(\boldsymbol{x}_S)\ . \tag{4.3}$$

For any fixed strategy $\boldsymbol{\lambda}$ for O, the inner minimization represents the optimal response of player E to that fixed strategy. Therefore, the maximin problem answers the question: what is the optimal strategy for O given that E gets to observe that strategy and respond? We call this problem the *E-response* problem. Note that although E could use a mixed strategy, there always exists a pure strategy which is optimal. That is:

$$\min_{\theta\in\Delta_{\mathcal{A}}} \mathbb{E}_\theta\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right] = \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\ . \tag{4.4}$$

This implies that any optimal $\boldsymbol{\lambda}$ for (4.3) is also an optimal $\boldsymbol{\lambda}$ for (4.2). Consequently, the optimal $\boldsymbol{\lambda}$ for (4.3) is one half of a Nash equilibrium pair. However, the optimal pair $(\boldsymbol{\lambda}, \boldsymbol{a}(\cdot))$ of (4.3) is not a Nash equilibrium, except in a specific situation described in the following theorem.

Theorem 4.1 *Suppose there exists $\boldsymbol{\lambda}_= \in \Delta_r$ with associated $\boldsymbol{\lambda}_=$-optimal control function $\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)$ which satisfies $\mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \mathcal{J}_j(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot))$ for all $i, j \in \{1, \ldots, r\}$. Then, $(\boldsymbol{\lambda}_=, \boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot))$ is a Nash equilibrium.*

Proof The fact that E cannot improve his payoff follows from the definition of $\boldsymbol{a}^{\boldsymbol{\lambda}_=} \in \arg\min_{\boldsymbol{a}(\cdot)} \mathcal{J}^{\boldsymbol{\lambda}_=}(\boldsymbol{a}(\cdot))$. O may not improve his payoff either as for all $\boldsymbol{\lambda}$,

$$\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \sum \boldsymbol{\lambda}_i \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \sum \boldsymbol{\lambda}_{=,i} \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \mathcal{J}^{\boldsymbol{\lambda}_=}(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot))\ .$$

□

This situation corresponds to the case when the convex part of the Pareto front intersects the central ray, such as in the example in Fig. 3. Theorem 3.1 implies that in this case, the worst-case optimal strategy for E coincides with E's half of the Nash equilibrium. Note that in general such a $\boldsymbol{\lambda}_=$ does not have to exist; for example, in Figs. 4 and 6 the convex part of the Pareto front does not intersect the central ray. In such situations, the worst-case optimal strategy for E and the Nash equilibrium are different. Moreover, the latter involves a mixed strategy for E covered in Sect. 4.2.

We now direct our attention to solving the E-response problem numerically. Equation (4.3) may be stated as the following optimization problem:

$$\max_{\boldsymbol{\lambda}} G(\boldsymbol{\lambda})$$

$$\text{s.t. } \lambda_i \geq 0, \quad \sum_{i=1}^{r} \lambda_i = 1 \,. \tag{4.5}$$

The objective function $G(\boldsymbol{\lambda}) = \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \sum_{i=1}^{r} \boldsymbol{\lambda}_i \mathcal{J}_i(\boldsymbol{a}(\cdot))$ is concave as it is the pointwise minimum of linear functions. Furthermore, the vector of individual cumulative costs $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))$, where $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot) \in \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))$, is a supergradient of G (denoted as $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) \in \partial G(\boldsymbol{\lambda})$). A supergradient provides an ascent direction of a concave function, i.e., $\boldsymbol{w} \in \partial G(\boldsymbol{\lambda})$ if for all $\hat{\boldsymbol{\lambda}} \in \Delta_r$,

$$G(\hat{\boldsymbol{\lambda}}) - G(\boldsymbol{\lambda}) \leq \boldsymbol{w}^T(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}) \,.$$

The fact that $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) \in \partial G(\boldsymbol{\lambda})$ is seen from the following computation: for any $\hat{\boldsymbol{\lambda}}$,

$$\begin{aligned} G(\hat{\boldsymbol{\lambda}}) - G(\boldsymbol{\lambda}) &= \left(\min_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{r} \hat{\lambda}_i \mathcal{J}_i(\boldsymbol{a}(\cdot)) \right) - \sum_{i=1}^{r} \lambda_i \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) \\ &\leq \sum_{i=1}^{r} \hat{\lambda}_i \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) - \sum_{i=1}^{r} \lambda_i \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) \\ &= \boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))^T(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}) \,. \end{aligned}$$

Evaluating the vector $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))$ can be challenging computationally; we show how this can be done in Sect. 5.2. Once this ascent direction is known, one still needs to ensure that $\boldsymbol{\lambda}$ remains a feasible probability distribution over $\mathcal{X}$, and we use the orthogonal projection operator $\Pi : \mathbb{R}^r \to \Delta_r$. The operator Π can be computed in $\mathcal{O}(r \log r)$ operations [4,38] as summarized in Algorithm 4.1. The resulting projected supergradient method [3, Chap. 8] is shown in Algorithm 4.2. The iterates of Algorithm 4.2 for the example from Fig. 6 are illustrated in Fig. 7.

Algorithm 4.1 Orthogonal projection onto the probability simplex

Input $\boldsymbol{\lambda} \in \mathbb{R}^r$
Sort $\boldsymbol{\lambda}$ into $\boldsymbol{u}$: $u_1 \geq u_2 \geq \cdots \geq u_r$
Find $\rho = \max\{1 \leq j \leq r : u_j + \frac{1}{j}\left(1 - \sum_{i=1}^{j} u_i\right) > 0\}$
$\tau \leftarrow \frac{1}{\rho}\left(1 - \sum_{i=1}^{\rho} u_i\right)$
return x s.t. $x_i = \max\{\lambda_i + \tau, 0\}, i = 1, \ldots, r$.

Algorithm 4.2 Projected supergradient method for finding the maximum of G over the set Δ_r

Input Initial guess $\boldsymbol{\lambda}_0$, stepsizes α_k, number of iterations n
for $k = 0 : (n-1)$ **do**
  Compute $G(\boldsymbol{\lambda}_k) = u^{\boldsymbol{\lambda}_k}(\boldsymbol{x}_S)$ and find some $\boldsymbol{g} \in \partial G(\boldsymbol{\lambda}_k)$
  $\boldsymbol{\lambda}_{k+1} \leftarrow \Pi\left(\boldsymbol{\lambda}_k + \alpha_k \boldsymbol{g}\right)$
end for
return $\underset{\boldsymbol{\lambda}\in\{\boldsymbol{\lambda}_0,\ldots,\boldsymbol{\lambda}_n\}}{\arg\max} \; G(\boldsymbol{\lambda})$

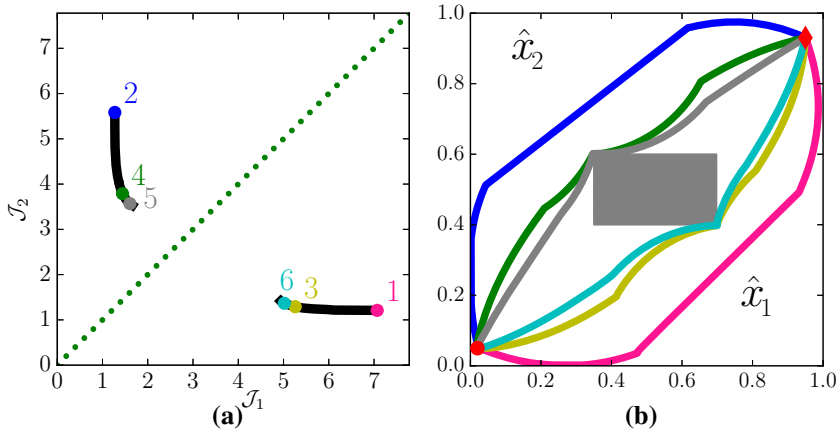

Fig. 7 **a** Convex part of PF and the individual costs of the first six iterates $\boldsymbol{\lambda}_k$ of Algorithm 4.2 (with stepsizes $\alpha_k = 3/k$) for the problem shown in Fig. 6. **b** The $\boldsymbol{\lambda}_k$-optimal trajectories of the first six iterates. We note that only a few iterates are needed to obtain trajectories which are close to the central ray. Thus, it does not require computing the whole PF which saves computational time (Color figure online)

4.2 Optimal Strategy of the Evader

Computing the evader's half of the Nash equilibrium is more challenging due to the fact that the set of E's pure strategies, i.e., the set of control functions $\boldsymbol{a}(\cdot)$ leading from the source $\boldsymbol{x}_S$ to the target $\boldsymbol{x}_T$, is uncountably infinite. We propose a heuristic strategy to approximate $\boldsymbol{\theta}^*$ which relies on two properties of the Nash equilibrium in semi-infinite games:

1. There exists a Nash mixed strategy for E which uses only r pure strategies[3] [28].
2. All pure strategies employed with positive probability in the Nash equilibrium have the same expected payoff, with the expectation taken over the other half of the Nash. In particular, all control functions used with positive probability in the Nash equilibrium are $\boldsymbol{\lambda}^*$-optimal.

The following characterization of the Nash equilibrium helps us generate a good candidate set of $\boldsymbol{\lambda}^*$-optimal trajectories.

Theorem 4.2 *Let* $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) \in \Delta_r \times \Delta_{\mathcal{A}}$ *and* $\mathcal{I} = \{i \mid \lambda_i^* \neq 0\}$. $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ *is a Nash equilibrium if and only if the following three conditions hold:*

1. $\boldsymbol{\lambda}^*$ *is a constrained maximizer of* $G(\boldsymbol{\lambda})$ *in* (4.5),
2. *if* $i \in \mathcal{I}$, *then* $\mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] = G(\boldsymbol{\lambda}^*)$, *and*
3. *if* $i \notin \mathcal{I}$, *then* $\mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] \leq G(\boldsymbol{\lambda}^*)$.

Proof ($\Rightarrow$)

Suppose $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium. Item 1 follows from the minimax theorem for semi-infinite game and (4.4). Assume Item 2 does not hold, then there must exist $i, j \in \mathcal{I}$

[3] This result assumes that the set $S = \{(s_1, s_2, \ldots, s_r) \mid s_i = P(\hat{\boldsymbol{x}}_i, \boldsymbol{a}(\cdot)); i = 1, 2, \ldots, r; \boldsymbol{a}(\cdot) \in \mathcal{A}\} \subset \mathbb{R}^r$ is bounded and $co(S)$ is closed. In our case, S is not bounded for the full set of control functions in $\mathcal{A}$ but becomes bounded if we restrict our attention to Pareto optimal control functions.

s.t. $\mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] > \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_j(\boldsymbol{a}(\cdot))\right]$. Consider the strategy $\hat{\boldsymbol{\lambda}} \in \Delta_r$:

$$\hat{\lambda}_k = \begin{cases} \lambda_i^* + \lambda_j^* & \text{if } k = i \\ 0 & \text{if } k = j \\ \lambda_k^* & \text{otherwise} \end{cases} .$$

Then, we have that:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \sum_{i=1}^{r} \lambda_i^* \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] < \sum_{i=1}^{r} \hat{\lambda}_i \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] = P(\hat{\boldsymbol{\lambda}}, \boldsymbol{\theta}^*) .$$

This contradicts that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium, and thus, Item 2 must hold. A similar argument can be used to demonstrate Item 3: assume there exists $i \notin \mathcal{I}$ with $\mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] > G(\boldsymbol{\lambda}^*)$. Let $j \in \mathcal{I}$ and consider the strategy $\hat{\boldsymbol{\lambda}}$:

$$\hat{\lambda}_k = \begin{cases} \lambda_j^* & \text{if } k = i \\ 0 & \text{if } k = j \\ \lambda_k^* & \text{otherwise} \end{cases}$$

Once again, this implies that $P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) < P(\hat{\boldsymbol{\lambda}}, \boldsymbol{\theta}^*)$ which contradicts that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium.

($\Leftarrow$) Assume Items 1–3 hold and suppose there exists $\boldsymbol{\theta}$ s.t. $P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) < P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$, then there must exist $\boldsymbol{a}(\cdot)$, used with nonzero probability in $\boldsymbol{\theta}$ such that:

$$\mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}(\cdot)) < P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = G(\boldsymbol{\lambda}^*) .$$

This contradicts the definition of $G(\boldsymbol{\lambda}^*) = \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}(\cdot))$. Thus, for all $\boldsymbol{\theta} \in \Delta_{\mathcal{A}}$ we have that:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) \leq P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) . \tag{4.6}$$

From Items 2 and 3, it follows that for all $\boldsymbol{\lambda} \in \Delta_r$:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \sum_{i=1}^{r} \lambda_i^* \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] \geq \sum_{i=1}^{r} \lambda_i \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] = P(\boldsymbol{\lambda}, \boldsymbol{\theta}^*) . \tag{4.7}$$

Equations (4.6) and (4.7) imply that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium. □

Any mix of $\boldsymbol{\lambda}^*$-optimal trajectories forms a $\boldsymbol{\lambda}^*$-optimal strategy for the evader. However, that mix is part of a Nash equilibrium only if the observer has no incentive to change his strategy in response. Theorem 4.2 says that this is the case when the $\boldsymbol{\theta}^*$ defining the mix of individual observability of $\boldsymbol{\lambda}^*$-optimal trajectories lies on the central ray of the Pareto front for a reduced problem. That is, the PF for the SEG where the observer has a potentially smaller number of positions (the ones which are used with positive probability in $\boldsymbol{\lambda}^*$). This PF is in an s-dimensional criterion space, where $s = |\mathcal{I}| \leq r$. In Fig. 3, the number of observer positions is $r = 2$, and the dimension of the "reduced" problem is also $s = 2$ since both positions are used with positive probability. In this example, a single $\boldsymbol{\lambda}^*$-optimal trajectory exists and corresponds to the intersection of the central ray and the convex part of the PF. In the examples from Figs. 4 and 6, we still have $r = 2$ and $s = 2$; however, there are two $\boldsymbol{\lambda}^*$-optimal trajectories. The Nash mixed strategy for E is thus obtained by finding a probability distribution $(\omega_1, \omega_2) \in \Delta_2$ over these two trajectories $(\boldsymbol{a}_1(\cdot), \boldsymbol{a}_2(\cdot))$ such that the linear combination of their individual costs lies on the central ray, i.e., such

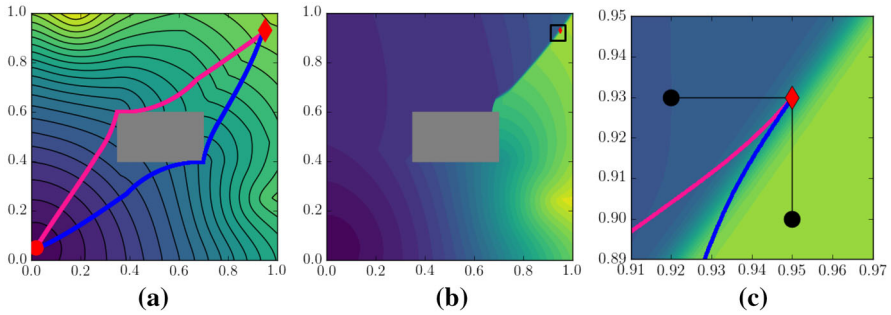

Fig. 8 **a** Two $\boldsymbol{\lambda}^*$-optimal trajectories in pink and blue plotted over the level sets of $u^{\boldsymbol{\lambda}^*}(x)$. The source location $\boldsymbol{x}_S$ is on a shockline of $u^{\boldsymbol{\lambda}^*}(x)$, and the two trajectories have the same expected cumulative observability, but different individual cumulative observability. **b** The individual cost function $v_1^{\boldsymbol{\lambda}^*}(x)$ is discontinuous at the source $\boldsymbol{x}_S$. The black square is the region displayed on (**c**). **c** The individual cost function $v_1^{\boldsymbol{\lambda}^*}(x)$ zoomed in around the source and a depiction of the upwind stencil. The stencil (displayed larger for the sake of visualization) contains a point on either side of the line of discontinuity of $v_1^{\boldsymbol{\lambda}^*}(x)$ (Color figure online)

that $\omega_1 \mathcal{J}_1(\boldsymbol{a}_1(\cdot)) + \omega_2 \mathcal{J}_1(\boldsymbol{a}_2(\cdot)) = \omega_1 \mathcal{J}_2(\boldsymbol{a}_1(\cdot)) + \omega_2 \mathcal{J}_2(\boldsymbol{a}_2(\cdot))$. In the example from Fig. 5, $r = 2$ and $s = 1$. The PF of the reduced problem is a single point and thus trivially lies on the "central ray", yielding a pure Nash equilibrium strategy for E. In Sect. 5, we show additional examples with $r = 3$, $s = 3$, and $r = 6$, $s = 4$. Computationally, Theorem 4.2 means that if we are able to find a set of g $\boldsymbol{\lambda}^*$-optimal control functions $\mathcal{A}(\boldsymbol{\lambda}^*) = \{\boldsymbol{a}_j(\cdot)\}_{j=1}^{j=g}$, such that Items 2 and 3 hold for some probability distribution $\boldsymbol{\omega} \in \Delta_g$, then $\boldsymbol{\lambda}^*$ is O's optimal response to $\boldsymbol{\omega}$ and we have found a Nash equilibrium pair. Note that the minimum number of trajectories g needed to form a Nash equilibrium is bounded above by s.

One remaining task is finding such a set $\mathcal{A}(\boldsymbol{\lambda}^*)$. Multiple $\boldsymbol{\lambda}^*$-optimal controls only exist if $\boldsymbol{x}_S$ lies on a shockline of $u^{\boldsymbol{\lambda}^*}$, where the gradient is undefined (e.g., the $\lim_{\boldsymbol{x}_i \to \boldsymbol{x}_S} \nabla u(\boldsymbol{x}_i)$ can be different depending on the sequence $\{x_i\}_i$). Numerically, our approximation of $u^{\boldsymbol{\lambda}^*}$ will yield a single upwind approximation of $\nabla u^{\boldsymbol{\lambda}^*}$, yielding a single $\boldsymbol{\lambda}^*$-optimal trajectory. As we show in Fig. 8, multiple optimal trajectories might lie in the same upwind quadrant and any numerical implementation of gradient descent will find only one of them. (In theory, one can approximate the other by perturbing $\boldsymbol{x}_S$, but the direction of perturbation is unobvious, particularly when $\boldsymbol{x}_S$ lies on an intersection of multiple shocklines, which is surprisingly common in this application as we show in further sections.)

This challenge is even more pronounced because Algorithm 4.2 yields *an approximate* value of $\boldsymbol{\lambda}^*$, since $\boldsymbol{x}_S$ will now be only *near* a shockline for some perturbed $\boldsymbol{\lambda}_\delta^* = \boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda}$. The resulting single $\boldsymbol{\lambda}_\delta^*$-optimal control will be a reasonable approximate solution for the max–min problem, but can be arbitrarily far from the solution to a min–max problem (where O has a chance to switch to another strategy).

In view of these challenges, we opt for a different approach, where an approximation to $\mathcal{A}(\boldsymbol{\lambda}^*)$ is computed iteratively, by adaptively growing a collection of $\boldsymbol{\lambda}_\delta^*$-optimal controls corresponding to different $\delta\boldsymbol{\lambda}$'s. In some degenerate cases, generating even the first $\boldsymbol{a}_1(\cdot) \in \mathcal{A}(\boldsymbol{\lambda}^*)$ may not be trivial since some $\boldsymbol{\lambda}^*$-optimal control computed by solving the Eikonal will not be necessarily Pareto optimal. For example, in Fig. 5 two control functions are $\boldsymbol{\lambda}^*$-optimal, but only one of them is used in the Nash strategy of E as the blue trajectory violates Item

3. However, both trajectories are indistinguishable from the point of view of the Eikonal solver since the position $\hat{\boldsymbol{x}}_1$ has zero weight in the weighted observability function $K^{\boldsymbol{\lambda}^*}$. To address this issue whenever $s < r$, we set the weight of the pointwise observability of each unused position $i \notin \mathcal{I}$ to some small value ϵ. (Our implementation uses $\epsilon = 10^{-6}$.) This is equivalent to seeking the solution of the weighted cost Eikonal equation for some perturbed $\boldsymbol{\lambda}_\delta^* = (1-\epsilon)\boldsymbol{\lambda}^* + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$, where $I_{\mathcal{I}^c}$ is the characteristic function of the complement of $\mathcal{I}$. We now turn our attention to finding further perturbations needed to generate $\boldsymbol{\lambda}_\delta^*$-optimal trajectories in order to make Item 2 approximately hold. Our goal is to have

$$\sum_{j=1}^{g} \omega_j \mathcal{J}_i(\boldsymbol{a}_j(\cdot)) = G(\boldsymbol{\lambda}^*) \tag{4.8}$$

approximately hold for all $i \in \mathcal{I} = \{i \mid \lambda_i^* > 0\}$. Unless this is already true with $g = 1$ (based on the previously found $\boldsymbol{a}_1(\cdot)$), we will need to find more $\boldsymbol{\lambda}_\delta^*$-optimal controls. Without loss of generality, assume that $\mathcal{I} = \{1, \ldots, s\}$, and suppose we have already generated a set of k $\boldsymbol{\lambda}_\delta^*$-optimal trajectories $\mathcal{A}^k = \{\boldsymbol{a}_1(\cdot), \boldsymbol{a}_2(\cdot), \ldots, \boldsymbol{a}_k(\cdot)\}$, for some $k < g$. In order for (4.8) to approximately hold, we will be increasing k until the norm of residual

$$\boldsymbol{R}(\boldsymbol{\omega}) = \begin{bmatrix} G(\boldsymbol{\lambda}^*) \\ G(\boldsymbol{\lambda}^*) \\ \vdots \\ G(\boldsymbol{\lambda}^*) \end{bmatrix} - \begin{bmatrix} \mathcal{J}_1(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_1(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_1(\boldsymbol{a}_k(\cdot)) \\ \mathcal{J}_2(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_2(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_2(\boldsymbol{a}_k(\cdot)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{J}_s(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_s(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_s(\boldsymbol{a}_k(\cdot)) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_k \end{bmatrix} \tag{4.9}$$

falls under a threshold $\text{tol}_{\boldsymbol{R}}$. Assuming the set of trajectories $\mathcal{A}^k$ has already been computed, the probability distribution $\boldsymbol{\omega}^k \in \Delta_k$ minimizing the norm of this residual $\|\boldsymbol{R}(\boldsymbol{\omega}^k)\|_2$ can be found by quadratic programming. The residual vector $\boldsymbol{R}(\boldsymbol{\omega}^k)$ provides information about which control functions are missing. For example, consider the case where we observe that a single entry of $\boldsymbol{R}(\boldsymbol{\omega}^k)$ is large and positive, i.e., for some $i \in \mathcal{I}$:

$$\sum_{j=1}^{k} \omega_j^k \mathcal{J}_i(\boldsymbol{a}_j(\cdot)) << G(\boldsymbol{\lambda}^*) \, .$$

The characterization in Theorem 4.2 implies that $\mathcal{A}(\boldsymbol{\lambda}^*)$ should include at least one trajectory much more observable from position $\hat{\boldsymbol{x}}_i$. A $\boldsymbol{\lambda}_\delta^*$-optimal trajectory with that property can be found by perturbing $\boldsymbol{\lambda}$ to slightly decrease the role of $\hat{\boldsymbol{x}}_i$ in O's chosen strategy. This is equivalent to resolving the Eikonal with $K^{\boldsymbol{\lambda}_\delta^*}$ corresponding to $\boldsymbol{\lambda}_\delta^* = \Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{e}_i\right)$ where $\delta << 1$ is chosen adaptively (see Algorithm 5.1), $\boldsymbol{e}_i$ is the i-th standard basis vector, and $\Pi_{\mathcal{I}}$ is the orthogonal projection onto the simplex defined only with elements of $\mathcal{I}$. Once a new $\boldsymbol{\lambda}_\delta^*$-optimal control function has been found, we may solve the quadratic program in (4.9) again with an additional column and repeat the process until the norm of the residual is sufficiently small. More generally, a large $\|\boldsymbol{R}(\boldsymbol{\omega})\|$ implies that some control functions in $\mathcal{A}(\boldsymbol{\lambda}^*)$ (or some mix of control functions) not in the current set $\mathcal{A}^k$ have a high observability with respect to the positive entries of $\boldsymbol{R}(\boldsymbol{\omega})$ while having a low observability with respect to the negative entries of $\boldsymbol{R}(\boldsymbol{\omega})$. Thus, we set the perturbation direction to $-\boldsymbol{R}(\boldsymbol{\omega})$ instead of $-\boldsymbol{e}_i$. Throughout this perturbation step, the entries of $\boldsymbol{\lambda}^*$ associated with the complement $\mathcal{I}$ are held fixed. Our full method for computing an approximate Nash equilibrium is summarized in Algorithm 4.3.

This method also has a geometric interpretation in terms of the Pareto front. Whenever $\boldsymbol{\theta}^*$ is not a pure strategy, a hyperplane normal to $\boldsymbol{\lambda}^*$ supports PF at multiple points (corresponding to all controls in $\mathcal{A}(\boldsymbol{\lambda}^*)$). However, any generic perturbation of $\boldsymbol{\lambda}^*$ would result in a hyperplane supporting PF near only one of these points, and the approximation to $\boldsymbol{\lambda}^*$ found by Algorithm 4.2 will correspond to a single optimal trajectory. For example, if we start with $\boldsymbol{a}_1(\cdot)$ corresponding to the yellow point in Fig. 6c (and associated yellow trajectory in Fig. 6b), then a small tilt (decreasing the role of position $\hat{\boldsymbol{x}}_1$ in O's plan) will yield a hyperplane supporting PF near the green point, allowing us to approximate the green trajectory in Fig. 6b by solving the weighted cost Eikonal equation with observability function $K^{\boldsymbol{\lambda}^*_\delta}$.

Algorithm 4.3 Computing an approximate Nash equilibrium of the SEG.

Find $\boldsymbol{\lambda}^*$ using Algorithm 4.2
$\mathcal{I} \leftarrow \{i \mid \lambda_i^* > \mathrm{tol}_\lambda\}$
$\boldsymbol{\lambda}^*_\delta \leftarrow (1-\epsilon)\boldsymbol{\lambda}^* + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
Find $\boldsymbol{\lambda}^*_\delta$-optimal control $\boldsymbol{a}_1(\cdot)$ and compute $\mathcal{J}_i(\boldsymbol{a}_1(\cdot))$ for all $i \in \mathcal{I}$
$k \leftarrow 1$, $\mathcal{A}^1 \leftarrow \{\boldsymbol{a}_1(\cdot)\}$, $\boldsymbol{\omega}^1 \leftarrow 1$
while $\|\boldsymbol{R}(\boldsymbol{\omega}^k)\| > \mathrm{tol}_{\boldsymbol{R}}$ **do**
  $\boldsymbol{\lambda}^*_\delta \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
  Find $\boldsymbol{\lambda}^*_\delta$-optimal control $\boldsymbol{a}_{k+1}(\cdot)$ and compute $\mathcal{J}_i(\boldsymbol{a}_{k+1}(\cdot))$ for all $i \in \mathcal{I}$
  $\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k \cup \{\boldsymbol{a}_k(\cdot)\}$
  $\boldsymbol{\omega}^{k+1} \leftarrow \arg\min_{\boldsymbol{\omega}^{k+1} \in \Delta_{k+1}} \|\boldsymbol{R}(\boldsymbol{\omega}^{k+1})\|_2$
  $k \leftarrow k+1$
end while
return $\boldsymbol{\lambda}^*, \mathcal{A}^k, \boldsymbol{\omega}^k$

5 Numerical Matters

In this section, we detail the implementation of our algorithm and present the additional numerical results. All algorithms were implemented in C++ and compiled with icpc version 16.0 on a MacBook Pro (16 GB RAM and an Intel Core i7 processor with four 2.5 GHz cores). The code is available online at https://github.com/eikonal-equation/Stationary_SEG. Our implementation relies on data structures and methods from Boost, Eigen, and QuadProg++ libraries.

5.1 Functions, Parameters, Methods

All of our examples are posed on the domain $\Omega = [0,1]^2$ with the possible exclusion of obstacles. All figures are based on computations on a uniform Cartesian grid of size $n \times n = 501 \times 501$ (with the grid spacing $h = 1/500$). To simplify the discussion, we always use a constant speed function $f(x) = 1$ though any inhomogeneous speed can be similarly handled by solving the Eikonal equation (2.3).

The pointwise observability functions are defined as

$$K_i(\boldsymbol{x}) = \begin{cases} \sigma, & \text{if } \boldsymbol{x} \text{ is in a shadow zone of } \hat{\boldsymbol{x}}_i; \\ \hat{K}(|\boldsymbol{x} - \hat{\boldsymbol{x}}_i|) + \sigma, & \text{otherwise.} \end{cases}$$

We set $\sigma = 0.1$ and $\hat{K}(r) = (\rho r^2 + 0.1)^{-1}$ with $\rho = 1$ in all examples except in Fig. 5 (where we set $\rho = 30$ simply to improve the visualization). The visibility of each gridpoint with respect to each observer position is precomputed and stored, but the K_i values are computed on the fly as needed.

The shadow zones for each observer are precomputed as follows. For each observer location $\hat{\boldsymbol{x}}_i$, two distance functions are computed: $D_0^i(\boldsymbol{x})$ and $D^i(\boldsymbol{x})$. The first is the distance between $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{x}$ when the obstacles are absent, while the second is that distance when obstacles are present. These distance functions can be computed by imposing the boundary conditions $D_0^i(\hat{\boldsymbol{x}}_i) = D^i(\hat{\boldsymbol{x}}_i) = 0$ and then solving two Eikonal equations [30]:

$$\left|\nabla D_0^i(\boldsymbol{x})\right| = 1, \qquad \left|\nabla D^i(\boldsymbol{x})\right| = \text{Obs}(\boldsymbol{x}), \tag{5.1}$$

with $\text{Obs}(\boldsymbol{x})$ set to ∞ inside the obstacles and 1 otherwise. The shadow zone of $\hat{\boldsymbol{x}}_i$ is characterized by $D^i > D_0^i$. But due to numerical errors in their approximation, we use a threshold value $\tau = 10^{-3}h$ (where h is the grid spacing) and specify that $\boldsymbol{x}$ is in this shadow zone whenever $D^i(\boldsymbol{x}) > D_0^i(\boldsymbol{x}) + \tau$.

The perturbation stepsize δ in Algorithm 4.3 is chosen adaptively using Algorithm 5.1. The goal of the adaptive strategy is to find the smallest perturbation δ necessary to obtain an additional $\boldsymbol{\lambda}_\delta^*$-optimal control function $\boldsymbol{a}_{k+1}(\cdot)$.

Algorithm 5.1 Adaptive strategy for choosing δ to generate $\boldsymbol{a}_{k+1}(\cdot)$

$\delta \leftarrow \delta_0$
$\boldsymbol{\lambda}_\delta^* \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
Compute a $\boldsymbol{\lambda}_\delta^*$-optimal control function $\hat{\boldsymbol{a}}(\cdot)$
while $\|\boldsymbol{\mathcal{J}}(\hat{\boldsymbol{a}}(\cdot)) - \boldsymbol{\mathcal{J}}(\boldsymbol{a}_j(\cdot))\|_2 < \text{tol}_\delta$ for any $j \in \{1, \ldots k\}$ **do**
  $\delta \leftarrow 2\delta$
  $\boldsymbol{\lambda}_\delta^* \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
  Compute $\boldsymbol{\lambda}_\delta^*$-optimal control function $\hat{\boldsymbol{a}}(\cdot)$
end while
$\boldsymbol{a}_{k+1}(\cdot) \leftarrow \hat{\boldsymbol{a}}(\cdot)$

The initialization used in our implementation is $\delta_0 = 10^{-4}$, and the tolerance is set to $\text{tol}_\delta = 10^{-2}\|\boldsymbol{\mathcal{J}}(\hat{\boldsymbol{a}}(\cdot))\|_2$. The stepsize rule used in the supergradient iteration in Algorithm 4.2 is $\alpha_k = 1/(k\|\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}_0}(\cdot))\|)$, the initial guess $\boldsymbol{\lambda}_0$ is a uniform distribution on $\mathcal{A}$, and the tolerance criteria on the residual and the near 0 entries used in Algorithm 4.3 are $\text{tol}_{\boldsymbol{R}} = 10^{-3}G(\boldsymbol{\lambda}^*)$ and $\text{tol}_\lambda = 5 \cdot 10^{-3}$, respectively. The quadratic programming problem in (4.9) is solved using the library QuadProg++.

5.2 Computation of Individual Costs

Running Algorithm 4.2 requires computing the vector of individual observability $\boldsymbol{\mathcal{J}}(\boldsymbol{x}_S, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))$. This problem is exactly the one solved by the scalarization approach described

in Sect. 3.1. Therefore, it can in principle be done by solving the Eikonal equation in (2.3) with cost function $K^{\boldsymbol{\lambda}}$ and associated linear equations in (3.2), i.e., $G(\boldsymbol{\lambda}) = u^{\boldsymbol{\lambda}}(\boldsymbol{x}_S)$ and $\mathcal{J}_i(\boldsymbol{x}_S, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) = v_i^{\boldsymbol{\lambda}}(\boldsymbol{x}_S)$. However, this technique has a severe drawback for this particular application: at the optimal $\boldsymbol{\lambda}^*$, $v_i^{\boldsymbol{\lambda}^*}$ is often discontinuous at $\boldsymbol{x}_S$. For example, in Fig. 8b, the upwind stencil containing the two $\boldsymbol{\lambda}^*$-optimal trajectories contains a point on either side of the discontinuity line of $v_1^{\boldsymbol{\lambda}^*}$ (which is the shockline of $u^{\boldsymbol{\lambda}^*}$). As a result, the value of $v_1^{\boldsymbol{\lambda}^*}(\boldsymbol{x}_S)$ is updated by interpolating the discontinuous function $v_1^{\boldsymbol{\lambda}^*}$ across the line of discontinuity.

This effect happens when multiple trajectories are $\boldsymbol{\lambda}^*$-optimal. Each of these trajectories has the same expected cumulative observability $\mathcal{J}^{\boldsymbol{\lambda}^*} = \sum_i \lambda_i^* \mathcal{J}_i$, but different individual observability $\mathcal{J}_i$. This issue leads to a large numerical error when using $v_i^{\boldsymbol{\lambda}}(\boldsymbol{x}_S)$ to estimate the supergradient in Algorithm 4.2, causing poor convergence of the method. Instead, we use the following process to compute the individual costs: first, we solve the weighted cost Eikonal equation (3.1) to obtain $u^{\boldsymbol{\lambda}}$ for a fixed $\boldsymbol{\lambda}$, and then, we trace the path $\boldsymbol{y}(t)$ using a gradient descent method on the value function $u^{\boldsymbol{\lambda}}$ and numerically estimate the integrals:

$$\mathcal{J}_i(\boldsymbol{x}_S, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) = \int_0^{T_{\boldsymbol{a}^{\boldsymbol{\lambda}}}} K_i(\boldsymbol{y}(t), \boldsymbol{a}^{\boldsymbol{\lambda}}(t))\,\mathrm{d}t, \quad i = 1, \ldots, r.$$

5.3 Additional Experiments and Error Metrics

We present two additional examples that include a higher number of observer plans. In Fig. 9, we show an example where the mixed strategy Nash equilibrium consists of a distribution over three strategies for both the evader and the observer. Figure 9 shows the value function $u^{\boldsymbol{\lambda}^*}$ at the optimal $\boldsymbol{\lambda}^*$. We observe that three shocklines of the value function $u^{\boldsymbol{\lambda}^*}$ meet at the source location $\boldsymbol{x}_S$, which implies that four trajectories are optimal starting from this location. However, the minimax theorem for infinite games assures that only three pure strategies are necessary to form a Nash equilibrium. Using Algorithm 4.3, we find an approximate Nash equilibrium which uses a mix of such three trajectories.

In Fig. 10, we show a maze-like example where the observer may choose among six possible positions. Using Algorithm 4.3, we determine that at the approximate Nash equilibrium, only four positions are used with positive probability by O, and E uses four different trajectories which are displayed in Fig. 10.

In order to test the performance of Algorithm 4.3, we consider three error metrics:

1. *The optimization error* in $G(\boldsymbol{\lambda})$ arises from several effects: the discretization error of the Eikonal solver, the discretization error of the path tracing and path integral evaluation, and the early stopping of the supergradient iterations. To generate the "ground truth", we performed the same computation on a finer grid of size of $n = 2001 \times 2001$ (i.e., we consider a grid with 16 times more unknowns) and run the supergradient iteration until we observe stagnation in the objective function value of the iterates. We approximate the relative error in our computations on a 501×501 grid as:

$$E_{rel}\left[G(\boldsymbol{\lambda}^*)\right] = \left|G_{501}(\boldsymbol{\lambda}^*_{501}) - G_{2001}(\boldsymbol{\lambda}^*_{2001})\right| \,/\, G_{501}(\boldsymbol{\lambda}^*_{501})\,.$$

2. *The observer's regret* estimates how much the observer could improve his payoff by unilaterally deviating from our approximate Nash equilibrium. (Recall that, if the approximate Nash equilibrium were exact, the observer would not be able to increase his payoff

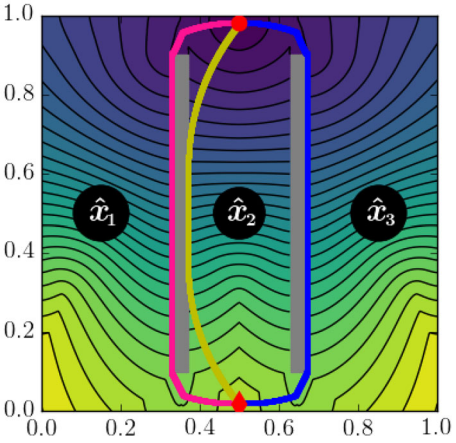

Fig. 9 Computed Nash equilibrium for a situation where a mix of three pure strategies is necessary for each player. The value function $u^{\boldsymbol{\lambda}^*}$ with three near-$\boldsymbol{\lambda}^*$-optimal trajectories in pink, blue, and yellow. Part of the pink is obstructed by the blue and green path. The optimal strategy for O is $\boldsymbol{\lambda}^* = \big[p(\boldsymbol{x}_1), p(\boldsymbol{x}_2), p(\boldsymbol{x}_3)\big] = [0.34, 0.32, 0.34]$, and the optimal strategy for E consists of three trajectories used with probability $\boldsymbol{\omega}^* = \big[p(\text{blue}), p(\text{yellow}), p(\text{pink})\big] = [0.40, 0.20, 0.40]$. In this example, the pink and yellow $\boldsymbol{\lambda}^*$-optimal trajectories initially coincide near $\boldsymbol{x}_S$, and hence, one cannot find both of them by perturbing the initial position $\boldsymbol{x}_S$ (Color figure online)

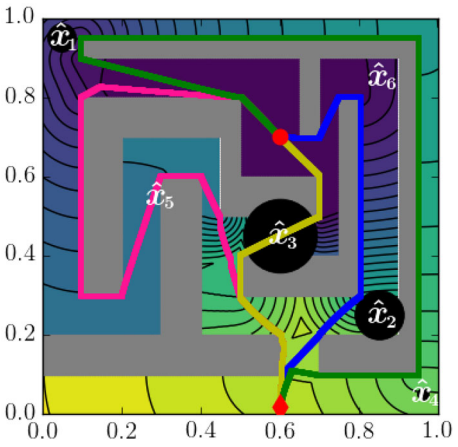

Fig. 10 Computed Nash equilibrium for a maze-like example. The value function $u^{\boldsymbol{\lambda}^*}$ and four near-$\boldsymbol{\lambda}^*$-optimal trajectories in pink, blue, yellow and green. The approximate Nash equilibrium strategy for O is $\boldsymbol{\lambda}^* = \big[p(\hat{x}_i)\big]_{i=1}^{i=6} = [0.174, 0.301, 0.452, 0.073, 0, 0]$. The approximate Nash equilibrium strategy for E uses four trajectories with probability $\boldsymbol{\omega}^* = \big[p(\text{pink}), p(\text{yellow}), p(\text{blue}), p(\text{green})\big] = [0.246, 0.461, 0.144, 0.149]$ (Color figure online)

at all.) We quantify this error using the normalized residual in (4.9), i.e.,

$$\text{Observer's regret} = \|R(\boldsymbol{\omega})\|_2 \;/\; \big(|\mathcal{I}|G(\boldsymbol{\lambda}^*)\big) \;.$$

3. *The evader's regret* estimates how much the evader could improve his payoff by unilaterally deviating from our approximate Nash equilibrium. This corresponds to how far from $\boldsymbol{\lambda}^*$-optimal are the controls produced by Algorithm 4.3. Recall that the control function $\boldsymbol{a}_1(\cdot)$ is (up to numerical errors) $\boldsymbol{\lambda}^*$-optimal, whereas $\boldsymbol{a}_k(\cdot)$ for $k \geq 2$ are $(\boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda})$-optimal. We report the maximum relative error in $\boldsymbol{\lambda}^*$ cumulative observability of the

Table 1 Table of timing and error metrics

	Figure 3	Figure 5	Figure 6	Figure 9	Figure 10
Number of it. of Algorithm 4.2	100	100	100	300	400
Total CPU time (seconds)	61	61	69	198	321
$E_{rel}\left[G(\boldsymbol{\lambda}^*)\right]$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$9 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	$3 \cdot 10^{-4}$
Observer's regret	$1 \cdot 10^{-4}$	0	$3 \cdot 10^{-4}$	$4 \cdot 10^{-6}$	$1 \cdot 10^{-4}$
Evader's regret	0	0	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-2}$

The error metrics are described in the main body of the text

$(\boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda})$-optimal trajectories, that is:

$$\text{Evader's regret} = \max_k \left| \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_1(\cdot)) - \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_k(\cdot)) \right| \Big/ \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_1(\cdot))$$

These error metrics are reported in Table 1 along with timing metrics for each example presented in the paper.

6 Extension to Groups of Evaders

We now consider an extension of the surveillance-evasion game to a game which involves a team of q evaders. Each evader E^l chooses a trajectory leading him from his own source location $\boldsymbol{x}_S^l$ to a target location $\boldsymbol{x}_T^l$, according to his own speed function $f^l(x)$. The pointwise observability function $K^{\boldsymbol{\lambda}}$ is shared for all evaders and depends only on the strategy $\boldsymbol{\lambda}$ of the observer. This induces q different cumulative observability functions $\mathcal{J}^{l,\boldsymbol{\lambda}}(\boldsymbol{x}_S^l, \boldsymbol{a}^l(\cdot))$ defined as in (2.1) and q different value functions $u^{l,\boldsymbol{\lambda}}$ which are solutions of Eikonal equations with q different boundary conditions.

In this version of the game, we assume that a central organizer for evaders faces off against the observer. The goal of that central organizer is to minimize the weighted sum of evaders' cumulative expected observabilities. The weights $\{w_l\}_{l=1}^{l=q}$ in the sum reflect the relative importance of each evader. We further assume that the central organizer and the observer agree on that relative importance, making this a two player zero-sum game with a payoff function defined by:

$$P(\boldsymbol{\lambda}, \{\boldsymbol{\theta}^l\}_{l=1}^q) = \sum_{l=1}^{l=q} w_l \mathbb{E}_{\boldsymbol{\theta}^l}\left[\mathcal{J}^{l,\boldsymbol{\lambda}}(\boldsymbol{x}_S^l, \boldsymbol{a}^l(\cdot))\right]. \tag{6.1}$$

Although we focus on a zero-sum two player game, we note that its Nash equilibrium $\left(\boldsymbol{\lambda}^*, \{\boldsymbol{\theta}^l\}_{l=1}^{l=q}\right)$ must also be among Nash equilibria of a different $(q+1)$-player game: the one, where each of the q evaders is selfishly minimizing their own cumulative observability $\mathcal{J}^{l,\boldsymbol{\lambda}}(\boldsymbol{x}_S^l, \boldsymbol{a}_l(\cdot))$, while the observer still attempts to maximize the crowd-wide observability in (6.1). This property follows from two simple facts:

1. The observer's payoff is the same in both versions of the game and thus cannot be improved unilaterally in a $(q+1)$ player game.
2. In the Nash equilibrium for the two-player game, the central organizer would only ask each evader to assign positive probabilities to their $\boldsymbol{\lambda}^*$-optimal trajectories. (Otherwise,

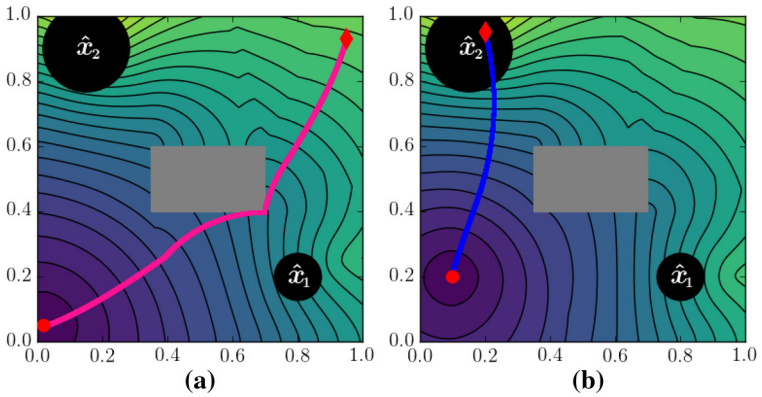

Fig. 11 Computed approximate Nash equilibrium for a group of two evaders. The approximate Nash equilibrium pair of strategies is $\boldsymbol{\lambda}^*$ for O, and a single $\boldsymbol{\lambda}^*$-optimal trajectory for each evader. **a** The value function $u^{1,\boldsymbol{\lambda}^*}$ for $\boldsymbol{\lambda}^* = [0.35, 0.65]$ of evader 1, and the $\boldsymbol{\lambda}^*$-optimal trajectories for evader 1 shown in pink. **b** The value function $u^{2,\boldsymbol{\lambda}^*}$ for the same $\boldsymbol{\lambda}^*$ of evader 2, and his $\boldsymbol{\lambda}^*$-optimal trajectory shown in blue (Color figure online)

the weighted sum in (6.1) could be improved.) Thus, they would also be maximizing their individual payoffs.

In this new setting, Theorem 4.2 holds and the observer's half of the Nash equilibrium may be found by maximizing the concave function:

$$G^q(\boldsymbol{\lambda}) = \min_{\boldsymbol{a}^l(\cdot)} \sum_{l=1}^{l=q} w_l \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{x}_S^l, \boldsymbol{a}^l(\cdot)) \,. \tag{6.2}$$

The function $G^q(\boldsymbol{\lambda})$ and its supergradients may be evaluated in a similar way to Sect. 5.2, but require q solves of the Eikonal equation with different boundary conditions and speed functions, and the numerical evaluation of $q \times r$ path integrals. However, we note that if all evaders have the same speed function and share the same target location (or, alternatively, share the same source location), only a single Eikonal equation solve is in fact required. With minor modifications, Algorithm 4.3 may be also applied to solve this version of the problem. For each perturbation of $\boldsymbol{\lambda}^*$, a set of q control functions is generated on line 8 of Algorithm 4.3, with one control function found for each evader. Although we obtain a new set of q control functions for each perturbation, some of the control functions for specific evaders may be essentially the same as those already obtained from previous perturbations. We address this in post-processing, by pruning the output of modified Algorithm 4.3 to identify distinct trajectories for each evader.

We show the numerical results for two test problems with $q = 2$ equally important evaders (i.e., $w_1 = w_2$) in each of them. An example presented in Fig. 11 uses the same obstacle and the same $r = 2$ possible observer locations already used in Fig. 6. At the approximate Nash equilibrium found using Algorithm 4.3, the observer uses these two locations with probabilities $\boldsymbol{\lambda}^* = (0.35, 0.65)$ and the central controller directs both evaders to use pure policies: deterministically choose pink and blue trajectories to their respective targets. Even though the first evader's starting position and destination are also the same as in Fig. 6, his (and the observer's) optimal strategies are quite different here due to the second evader's participation.

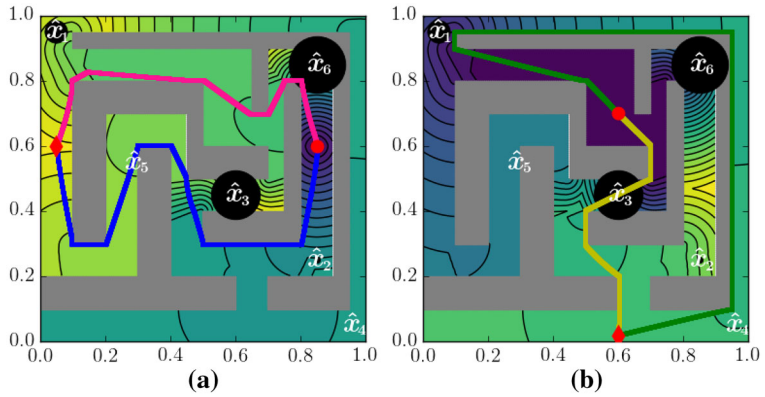

Fig. 12 Computed approximate Nash equilibrium for a maze-like example with two evaders. **a** The value function $u^{1,\boldsymbol{\lambda}^*}$ of evader 1, and two near-$\boldsymbol{\lambda}^*$-optimal trajectories for evader plotted in pink and blue. **b** The value function $u^{2,\boldsymbol{\lambda}^*}$ of evader 2, and two near-$\boldsymbol{\lambda}^*$-optimal trajectories for evader 2 plotted in yellow and green. The approximate Nash equilibrium $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is $\boldsymbol{\lambda}^* = \left[p(\hat{x}_i)\right] = [0.168, 0.0455, 0.364, 0, 0, 0.422]$, and $\boldsymbol{\theta}^*$ consists of a mixed strategy for the group of evaders. The mixed strategy of evader 1 is $\left[p(\text{pink}),\ p(\text{blue})\right] = [0.85, 0.15]$, and the mixed strategy for evader 2 is $\left[p(\text{yellow}),\ p(\text{green})\right] = [0.89, 0.11]$ (Color figure online)

Table 2 Table of running times and errors for examples with multiple evaders

	Figure 11	Figure 12
Number of it. of Algorithm 4.2	353	300
Total CPU time (seconds)	631	594
$E_{rel}\left[G(\boldsymbol{\lambda}^*)\right]$	$5 \cdot 10^{-4}$	$7 \cdot 10^{-3}$
Observer's regret	$5 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
Evader's regret	$5 \cdot 10^{-3}$	$1 \cdot 10^{-2}$

In a maze-like example presented in Fig. 12, O can choose among six possible locations, but his optimal mixed strategy $\boldsymbol{\lambda}^*$ uses only four of them. Algorithm 4.3 yields three sets of two near-$\boldsymbol{\lambda}^*$-optimal trajectories which form an approximate Nash equilibrium, but they only contain two distinct trajectories for each of the evaders. We report timing and error metrics for these two examples in Table 2.

7 Conclusion

We have considered an adversarial path planning problem, where the goal is to minimize the cumulative exposure/observability to a hostile observer. The current position of the latter is unknown, but the full list of possible positions is assumed to be available in advance. The key assumption of our model is that neither the evader (E) nor the enemy observer (O) can adjust their plan in real time based on the opponent's state and actions. Instead, both of them are required to choose their (possibly randomized) strategies in advance. We discussed two versions of this problem; in the first one, a completely risk-averse evader attempts to minimize his worst-case cumulative observability. We showed that this version can be solved using previously developed methods for multiobjective path planning. However, the

solution is prohibitively computationally expensive when O has a large number of surveillance plans to choose from. In the second version, the subject of optimization is the E's expected cumulative observability on its way to the target. We modeled this as a zero-sum surveillance-evasion game (SEG) between two players: E (the minimizer) and O (the maximizer). We then presented an algorithm combining ideas from continuous optimal control, the scalarization approach for multiobjective optimization, and convex optimization which allows us to quickly compute an approximate Nash equilibrium of this semi-infinite strategic game. Finally, we showed that this algorithm extends to solve a similar problem involving a group of multiple evaders controlled by a central planner. The presented algorithm displays at most linear scaling in the number of observation plans, but further speedup techniques would be desirable; the computational bottleneck (numerically solving the Eikonal equation) could be alleviated with domain restriction methods [9] and factoring approaches [27].

Although this paper focused on isotropic problems, the anisotropic observer case could be treated in a similar fashion. (In practice, the pointwise observability might depend on the angle between the evader's direction of motion and the observer's line of sight.) This generalization will have to rely on fast numerical methods developed for anisotropic HJB PDEs, e.g., [1,24,31,36]. In a follow-up paper [6], we show that time-dependent observation plans (e.g., different patrol routes) can be similarly treated by solving λ-parametrized finite-horizon optimal control problems with numerical methods for time-dependent HJB equations, e.g., [16,32].

We note that the computational cost of our algorithm increases quickly with the number of evaders considered. The case involving a large number of selfish evaders could be covered by considering the evolution of a time-dependent density of observers and treating the problem using mean field games [5,17]. Another possible extension would be to consider a group of observers choosing among a larger set of surveillance plans. In that situation, the set of pure strategies of the observers could increase exponentially, but we anticipate that the computational cost will grow much slower since the number of required Eikonal solves would not increase.

Acknowledgements The authors would like to thank Alex Townsend and anonymous reviewers for their helpful suggestions.

Funding This work is supported in part by the National Science Foundation Grant DMS-1738010. The second author's work is also supported by the Simons Foundation Fellowship.

References

1. Alton K, Mitchell IM (2012) An ordered upwind method with precomputed stencil and monotone node acceptance for solving static convex Hamilton–Jacobi equations. J Sci Comput 51:313–348
2. Bardi M, Capuzzo-Dolcetta I (2008) Optimal control and viscosity solutions of Hamilton–Jacobi-Bellman equations. Springer, Berlin
3. Beck A (2017) First-order methods in optimization, vol 25. SIAM, Philadelphia
4. Brucker P (1984) An $O(n)$ algorithm for quadratic knapsack problems. Oper Res Lett 3:163–166
5. Carmona R, Delarue F (2017) Probabilistic theory of mean field games with applications I–II. Springer, Berlin
6. Cartee E, Lai L, Song Q, Vladimirsky A (2019) Time-dependent surveillance-evasion games, preprint arXiv:1903.01332
7. Chacon A, Vladimirsky A (2012) Fast two-scale methods for Eikonal equations. SIAM J Sci Comput 34:A547–A578
8. Chacon A, Vladimirsky A (2015) A parallel two-scale method for Eikonal equations. SIAM J Sci Comput 37:A156–A180

9. Clawson Z, Chacon A, Vladimirsky A (2014) Causal domain restriction for Eikonal equations. SIAM J Sci Comput 36:A2478–A2505
10. Clawson Z, Ding X, Englot B, Frewen TA, Sisson WM, Vladimirsky A (2015) A bi-criteria path planning algorithm for robotics applications, preprint arXiv:1511.01166
11. Crandall MG, Lions P-L (1983) Viscosity solutions of Hamilton–Jacobi equations. Trans Am Math Soc 277:1–42
12. Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct Optim 14:63–69
13. Desilles A, Zidani H (2018) Pareto front characterization for multi-objective optimal control problems using Hamilton-Jacobi approach, preprint
14. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1:269–271
15. Dobbie J (1966) Solution of some surveillance-evasion problems by the methods of differential games. In: Proceedings of the 4th international conference on operational research. MIT, Wiley, New York
16. Falcone M, Ferretti R (2014) Semi-Lagrangian approximation schemes for linear and Hamilton–Jacobi equations, vol 133. SIAM, Philadelphia
17. Guéant O, Lasry J-M, Lions P-L (2010) Mean field games and applications. Paris–Princeton lectures on mathematical finance. Springer, Berlin, pp 205–266
18. Guigue A (2014) Approximation of the pareto optimal set for multiobjective optimal control problems using viability kernels. ESAIM COCV 20:95–115
19. Kumar A, Vladimirsky A (2010) An efficient method for multiobjective optimal control and optimal control subject to integral constraints. J Comput Math 1:517–551
20. Lewin J (1973) Decoy in pursuit-evasion games. PhD thesis, Department of Aeronautics and Astronautics, Stanford University
21. Lewin J, Breakwell JV (1975) The surveillance-evasion game of degree. J Optim Theory Appl 16:339–353
22. Lewin J, Olsder GJ (1979) Conic surveillance evasion. J Optim Theory Appl 27:107–125
23. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26:369–395
24. Mirebeau J-M (2014) Efficient fast marching with Finsler metrics. Numer Math 126:515–557
25. Mitchell IM, Sastry S (2003) Continuous path planning with multiple constraints. In: 2003 42nd IEEE conference on decision and control, vol 5, pp 5502–5507
26. Osborne MJ, Rubinstein A (1994) A course in game theory. MIT press, Cambridge
27. Qi D, Vladimirsky A (2019) Corner cases, singularities, and dynamic factoring. J Sci Comput 79:1456–1476
28. Raghavan T (1994) Zero-sum two-person games. Handb Game Theory Econ Appl 2:735–768
29. Sethian JA (1996) A fast marching level set method for monotonically advancing fronts. In: Proceedings of the National Academy of Sciences, vol 93, pp 1591–1595
30. Sethian JA (1999) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol 3. Cambridge University Press, Cambridge
31. Sethian JA, Vladimirsky A (2003) Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms. SIAM J Numer Anal 41:325–363
32. Shu C-W (2007) High order numerical methods for time dependent Hamilton–Jacobi equations. Mathematics and computation in imaging science and information processing. World Scientific, Singapore, pp 47–91
33. Soner H (1986) Optimal control with state-space constraint. I SIAM J Control Optim 24:552–561
34. Takei R, Chen W, Clawson Z, Kirov S, Vladimirsky A (2015) Optimal control with budget constraints and resets. SIAM J Control Optim 53:712–744
35. Tijs S (1976) Semi-infinite linear programs and semi-infinite matrix games. Katholieke Universiteit Nijmegen, Mathematisch Instituut, Nijmegen
36. Tsai Y-HR, Cheng L-T, Osher S, Zhao H-K (2003) Fast sweeping algorithms for a class of Hamilton–Jacobi equations. SIAM J Numer Anal 41:673–694
37. Tsitsiklis JN (1995) Efficient algorithms for globally optimal trajectories. IEEE Trans Autom Control 40:1528–1538
38. Wang W, Carreira-Perpinán MA (2013) Projection onto the probability simplex: an efficient algorithm with a simple proof, and an application, arXiv preprint arXiv:1309.1541
39. Zhao H (2005) A fast sweeping method for Eikonal equations. Math Comput 74:603–627