

OPTIMALITY AND ROBUSTNESS IN PATH-PLANNING UNDER INITIAL UNCERTAINTY

DONGPING QI^{*}, ADAM DHILLON[†], AND ALEXANDER VLADIMIRSKY[‡]

Abstract. Classical deterministic optimal control problems assume full information about the controlled process. The theory of control for general partially-observable processes is powerful, but the methods are computationally expensive and typically address the problems with stochastic dynamics and continuous (directly unobserved) stochastic perturbations. In this paper we focus on path planning problems which are in between – deterministic, but with an initial uncertainty on either the target or the running cost on parts of the domain. That uncertainty is later removed at some time T , and the goal is to choose the optimal trajectory until then. We address this challenge for three different models of information acquisition: with fixed T , discretely distributed and exponentially distributed random T . We develop models and numerical methods suitable for multiple notions of optimality: based on the average-case performance, the worst-case performance, the average constrained by the worst, the average performance with probabilistic constraints on the bad outcomes, risk-sensitivity, and distributional-robustness. We illustrate our approach using examples of pursuing random targets identified at a (possibly random) later time T .

Key words. optimal control, path-planning, Hamilton-Jacobi PDEs, uncertainty, robustness, delayed information acquisition

AMS subject classifications. 49L20, 49N90, 60J28, 35R35

1. Introduction. A common task in robotic navigation is to find a trajectory that minimizes cumulative cost incurred on the way to a target. If the system state is perfectly known in real time, this can be accomplished in the framework of dynamic programming by solving Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs) [7, 25]. The task of controlling *partially-observable* processes is generally harder due to the added challenge of estimating the process state probabilistically [30, 55]. Whether one aims for on-average optimality or robustness, the methods in this general space are focused on the uncertainty due to frequent stochastic perturbations in between observations. In contrast, we are interested in controlling processes with uncertainty due to structured or delayed information acquisition patterns. Here we focus on the simplest subset of such problem, in which everything is deterministic, but some important aspects of the process are a priori unknown. We start with a probabilistic description of the global environment, dynamics, costs or targets, and then need to control the system until this initial uncertainty is resolved.

In a discrete setting, a classical example of such monotonically non-increasing uncertainty is the “Canadian Traveller Problem” [38], in which one aims to find a cost-minimizing path to a target on a partially known graph: some of the graph edges might be absent with pre-specified/known probabilities, but whether they actually are present is only learned later, as we enter nodes adjacent to them. A generalization allowing random edge costs and more complicated information-gathering on a graph is known as a “Stochastic Shortest Path with Recourse” [40]. While dynamic programming algorithms are well known for these problems, they operate on a much larger information-enriched state space, and their computational cost is typically pro-

^{*}Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 (dq48@cornell.edu).

[†]Department of Mathematics, University of California, Berkeley, CA 94720 (adamdhillon@berkeley.edu).

[‡]Department of Mathematics and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 (vladimirsky@cornell.edu).

hibitive.

In this paper we examine path-planning problems in a continuous setting but with a fairly simple information acquisition pattern: the full information about the system and the environment is instantaneously gained at some “certainty time” T , and the problem remains deterministic thereafter. This simplification allows us to develop efficient algorithms for optimizing the pre-certainty control and also helps in comparing the merits of several notions of robustness. For motivation, consider a semi-autonomous rover exploring a sequence of locations on Mars with infrequent guidance from a command center on Earth. The center might identify a set of possible future targets (with a probability specified for each of them a priori), but the final choice will be identified at a later time T . While the rover could wait in place until then, its cumulative cost-to-target can be reduced by choosing a suitable waypoint and starting to move to it already now. To simplify the exposition, we describe all algorithms in the context of such *initial target uncertainty* problems in 2D and with isotropic cost/dynamics. But our general approach is broader and also covers other types of initial uncertainty. In Appendix A, we show how it can be applied to a simple environment-uncertainty problem: planning a path for an airplane to avoid a storm front whose position remains uncertain until the time T .

In section 2, we start by reviewing the standard deterministic setting: planning an optimal path to a known target by solving a single HJB equation. We then introduce the target uncertainty in section 3 and develop methods for optimizing the *average-case performance*. Interestingly, the methods for doing this depend entirely on what is known about the certainty time T . Once the min-cost to each potential target is known, we show that, for a fixed/constant T , the expected-cost-minimizing waypoint is found by optimizing over a constraint set determined by solving a single stationary HJB equation. If T is random and has a known discrete distribution, we show that optimal waypoints are found by solving a sequence of time-dependent HJB equations. For simplicity, our method is described with the likelihood of individual targets viewed as independent of the realized value of T , but we later show in Appendix B that this assumption is not essential. If T is an exponentially distributed random variable, finding an optimal waypoint is not enough – we also need to find an optimal path toward that waypoint since the target might be identified before we reach it. We explain how this can be accomplished by solving a quasi-variational HJB inequality of obstacle type.

The issue of robustness under initial uncertainty is the subject of section 4. Robust control is an important and well-developed research area, with H^∞ methods [9] particularly popular in stochastic systems subject to frequent or continuous random disturbances. But in our setting, this approach is not directly applicable, though one can still mitigate the risk of “unlucky scenarios” and guard against modeling errors. Focusing primarily on the “fixed T ” case, we develop methods for several competing notions of robustness: optimizing the worst-case scenario, risk-sensitive optimization, and optimizing the average case performance subject to constraints (either hard or probabilistic) on bad outcomes. The case of probabilistic constraints is particularly interesting since the optimal policy for selecting waypoints turns out to be non-deterministic. We find a geometric interpretation of this challenge and leverage it to construct an efficient algorithm in subsection 4.4. We further examine the Distributionally Robust Optimization (DRO) approach to guard against mistakes in the perceived relative likelihood of potential targets. In addition, we also provide a sub-optimality bound for the case when the set of targets is either misidentified or intentionally subsampled.

Many of the problems described above are reminiscent of those handled in discrete settings by the techniques of *multistage stochastic programming* [48]. However, their naive use would be prohibitive here since our state and action spaces are infinite/continuous. In addition, our approach based on dynamic programming allows solving these multi-stage optimization problems for all initial configurations simultaneously. These distinctions (and the computational efficiency gained by exploiting the problem structure) are highlighted throughout the paper. We conclude by discussing possible generalizations of our approach and directions for future work in section 5.

2. Deterministic setting overview. We first consider a fully deterministic controlled process with a time-dependent process state $\mathbf{y} \in \mathbb{R}^d$ and a terminal time T . This is the setting of the classical optimal control theory and we include only a brief overview, referring to [7] for technical details.

We will assume that the process dynamics is defined by an ODE

$$\begin{cases} \mathbf{y}'(s) = \mathbf{f}(\mathbf{y}(s), \mathbf{a}(s), s), & s \in [t, T]; \\ \mathbf{y}(t) = \mathbf{x} \in \mathbb{R}^d. \end{cases} \quad (2.1)$$

The controller has a compact set A of available control values and chooses a measurable control function $\mathbf{a} : \mathbb{R} \mapsto A$ to guide the evolution of the process state. The “velocity function” $\mathbf{f} : (\mathbb{R}^d \times A \times \mathbb{R}) \mapsto \mathbb{R}^d$, the initial state and time $(\mathbf{x}, t)$, and the terminal time T are all explicitly known ahead of time. Given a running cost $K : \mathbb{R}^d \times A \times \mathbb{R} \mapsto \mathbb{R}$ and a terminal cost $q : \mathbb{R}^d \mapsto \mathbb{R}$, one can define the cost corresponding to a control $\mathbf{a}(\cdot)$ starting from $(\mathbf{x}, t)$ as

$$\mathcal{J}(\mathbf{x}, t, \mathbf{a}(\cdot)) = \int_t^T K(\mathbf{y}(s), \mathbf{a}(s), s) ds + q(\mathbf{y}(T)). \quad (2.2)$$

A *value function* is then defined as the minimal cost-to-termination from a given location; i.e., $u(\mathbf{x}, t) = \inf_{\mathbf{a}(\cdot)} \mathcal{J}(\mathbf{x}, t, \mathbf{a}(\cdot))$. The principle of dynamic programming is based on a *tail optimality* property of optimal controls; i.e.,

$$u(\mathbf{x}, t) = \min_{\mathbf{a}(\cdot)} \left\{ \int_t^{t+\tau} K(\mathbf{y}(s), \mathbf{a}(s), s) ds + u(\mathbf{y}(t+\tau), t+\tau) \right\}$$

should hold for any small $\tau > 0$. Assuming that the value function is smooth, one can use a Taylor series expansion and then let $\tau \rightarrow 0$ to derive a time-dependent Hamilton-Jacobi PDE that u must satisfy:

$$u_t(\mathbf{x}, t) + \min_{\mathbf{a} \in A} \{K(\mathbf{x}, \mathbf{a}, t) + \nabla u(\mathbf{x}, t) \cdot \mathbf{f}(\mathbf{x}, \mathbf{a}, t)\} = 0, \quad \forall (\mathbf{x}, t) \in \mathbb{R}^d \times [0, T] \quad (2.3)$$

subject to a *terminal condition* $u(\mathbf{x}, T) = q(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{R}^d$. Since the value function is generally not smooth, this PDE usually does not have a classical solution. The non-uniqueness of weak (locally Lipschitz) solutions made it necessary to introduce additional test conditions [17] that pick out a *viscosity solution* of this PDE – the unique weak solution that coincides with the value function of the original control problem.

Even though the approach presented in this paper for treating the initial uncertainty is general, all of our examples will be based on *isotropic* problems, in which the cost and speed of motion depend only on $\mathbf{x}$. We let $A = \{\mathbf{a} \in \mathbb{R}^d \mid |\mathbf{a}| = 1\}$ and interpret $\mathbf{a}$ as the chosen direction of motion. Then $K(\mathbf{x}, \mathbf{a}, t) = K(\mathbf{x}, t)$ and

$\mathbf{f}(\mathbf{x}, \mathbf{a}, t) = f(\mathbf{x}, t)\mathbf{a}$, with f encoding the speed of motion through the point $\mathbf{x}$. In this case, the optimal direction is known analytically: $\mathbf{a}^* = -\nabla u / |\nabla u|$ and (2.3) reduces to a *time-dependent Eikonal equation*

$$u_t - |\nabla u|f(\mathbf{x}, t) + K(\mathbf{x}, t) = 0; \quad \forall \mathbf{x} \in \mathbb{R}^d, t \in [0, T]. \quad (2.4)$$

Both the isotropic and the general problems can be similarly posed on a compact domain $\Omega \subset \mathbb{R}^d$, with additional boundary conditions $u = g$ posed on $\partial\Omega \times [0, T]$, which can be interpreted as a cost of stopping the process prematurely as soon as it exits Ω at some $t < T$. Setting $g = +\infty$ can be used to restrict the process to this domain¹ and force the trajectories to avoid any obstacles (by simply excluding them from Ω). Setting $g(\mathbf{x}, t) = 0$ on any closed *target set* $\Gamma \subset \partial\Omega$ makes $u(\mathbf{x}, t)$ encode the minimal cost to target Γ , with q specifying a penalty in case we cannot (or choose not to) reach Γ by the time T . The latter case is further simplified when K and f do not depend on t and $T = +\infty$ (i.e., there is no time-restriction for reaching the target). The corresponding value function satisfies a (stationary) *Eikonal equation*

$$|\nabla u|f(\mathbf{x}) = K(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega; \quad u(\Gamma) = 0; \quad u(\partial\Omega \setminus \Gamma) = +\infty. \quad (2.5)$$

If $K \equiv 1$, $u(\mathbf{x})$ simply encodes the minimum time to the target Γ (while staying inside Ω) and the gradient descent yields time-optimal trajectories. When Γ consists of only a single point $\mathbf{x}_0$, we denote the value function of this min-cost-from- $\mathbf{x}_0$ -to-a-target-at- $\mathbf{x}$ problem as $u(\mathbf{x}; \mathbf{x}_0)$.

Another convenient property of the Eikonal equation is that an optimal trajectory from a starting position $\mathbf{x}_0$ to a target is the same (up to a time-reversal) as an optimal trajectory from that target to $\mathbf{x}_0$. The optimal trajectory from $\mathbf{x}_0$ to any target $\mathbf{x}$ can be found by following $(-\nabla u(\mathbf{x}; \mathbf{x}_0))$ from $\mathbf{x}$ to $\mathbf{x}_0$ and then tracing it backward. In Figure 1 we present an example of this on a 2D domain with a single rectangular obstacle. Since $K = 1$, we are finding the time-optimal trajectories from $\mathbf{x}_0 = (0.3, 0.2)$ under the speed function $f = 1.4 + 0.6 \cos(2\pi x) \sin(2\pi y)$ shown in Figure 1A. In Figure 1B we show the optimal trajectories from $\mathbf{x}_0$ to 4 different targets as well as the level sets of the function $u(\mathbf{x}; \mathbf{x}_0)$. In subsequent sections, this general setup will be used as our representative example to explore the initial target-uncertainty.

The viscosity solutions to the Hamilton-Jacobi-Bellman (HJB) equations are usually unavailable in analytic form and numerical approximations are thus unavoidable. Numerical methods for time-dependent and stationary HJB equations have been an active area of research in the last 25 years, with particularly many efficient techniques developed for the stationary Eikonal equation (2.5) and its anisotropic generalizations. The first challenge is to choose a discretization converging to the viscosity solution. The simplest grid discretization approach (employed here) is to use the first-order *upwind divided differences* to approximate the partial derivatives of u ; e.g., with $d = 2$ and a space-time grid based on $(x_i, y_j, t_n) = (ih, jh, n\Delta t)$, we can define the standard one-sided divided differences as

$$D^{\pm x} U_{i,j}^n = \frac{U_{i\pm 1,j}^n - U_{i,j}^n}{\pm h} \quad \text{and} \quad D^{\pm y} U_{i,j}^n = \frac{U_{i,j\pm 1}^n - U_{i,j}^n}{\pm h},$$

¹To define such *domain-constrained viscosity solutions*, one has to treat the boundary conditions “in the viscosity sense” [7] and we adopt this interpretation throughout the paper. Intuitively, this means that the infinite exit cost g is charged not for touching $\partial\Omega$ but for attempting to leave Ω . This allows traveling along $\partial\Omega$ (including the obstacle boundaries) and, combined with our isotropy assumption, ensures the existence of optimal controls for every starting position.

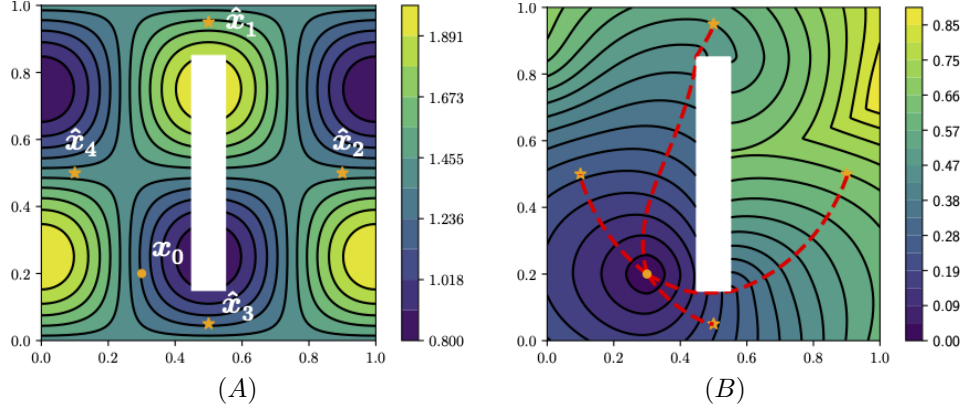


FIG. 1. Time-optimal path planning. (A): a contour plot of the speed $f = 1.4 + 0.6 \cos(2\pi x) \sin(2\pi y)$ on a unit square domain with one rectangular obstacle (shown in white) at $(0.45, 0.55) \times (0.15, 0.85)$. The starting point $\mathbf{x}_0 = (0.3, 0.2)$ is shown by an orange dot. The approach finds optimal trajectories to all target locations, but we highlight four potential targets (shown by orange stars and numbered clockwise, starting from the top): $\hat{\mathbf{x}}_1 = (0.5, 0.95)$, $\hat{\mathbf{x}}_2 = (0.9, 0.5)$, $\hat{\mathbf{x}}_3 = (0.5, 0.05)$, and $\hat{\mathbf{x}}_4 = (0.1, 0.5)$. (B): a contour plot of $u(\mathbf{x}; \mathbf{x}_0)$ and the optimal trajectories (shown by dashed red lines) to these four targets.

then plug

$$\begin{aligned}
 u_t(x_i, y_j, t_n) &\approx D^{-t} U_{i,j}^n = \frac{U_{i,j}^n - U_{i,j}^{n-1}}{\Delta t}, \\
 [u_x(x_i, y_j, t_n)]^2 &\approx \left[\max \left\{ D^{-x} U_{i,j}^n, -D^{+x} U_{i,j}^n, 0 \right\} \right]^2, \\
 [u_y(x_i, y_j, t_n)]^2 &\approx \left[\max \left\{ D^{-y} U_{i,j}^n, -D^{+y} U_{i,j}^n, 0 \right\} \right]^2
 \end{aligned} \tag{2.6}$$

into equation (2.4) and solve for $U_{i,j}^{n-1}$. (The state constraints are handled by setting U values to $+\infty$ outside of Ω , including inside the obstacles.)

This discretization guarantees a monotone dependence on the neighboring grid values, and this monotonicity yields convergence to the viscosity solution [8]. For the time-dependent PDE (2.4), a time-explicit discretization makes the issues of efficiency trivial: the solution is computed by time-marching, computing one time-slice at the time, from the terminal time $t = T$ to the initial time $t = 0$, with the $O(NM)$ computational cost². But in the stationary case (2.5), the resulting system of discretized equations for $U_{i,j}$ s is non-linear, non-smooth, and coupled, adding the challenge of solving that discretized system efficiently. A number of methods are based on competing ideas for handling this task: from Fast Marching (e.g., [50], [44]), to Fast Sweeping (e.g., [10], [57], [49]), to their hybrid versions [12, 13] and methods mirroring the logic of label-correcting algorithms on graphs (e.g., [6, 29]). When optimal paths are needed from a small number of starting positions only, additional gains

²Throughout the paper, we will assume that PDEs are discretized on a spatial grid with a total of M gridpoints and, whenever these PDEs are time-dependent, we will assume that there is a total of N time-slices. Since our time discretizations are explicit, these N and M are not really independent: e.g., with $d = 2$, Courant-Friedrichs-Lewy stability condition yields $N = O(M^{1/2})$. But we will not directly use this expression in computational complexity estimates since our goal is to emphasize the general space-time structure and other (time-implicit [52] or semi-Lagrangian [24]) discretizations could be used to avoid this dependence.

in efficiency can be also obtained through a dynamic restriction of the computational domain [16, 54]. Many of these methods have been generalized to simplicial meshes, problems on manifolds and higher order accurate schemes. Other approaches have also been developed using semi-Lagrangian [18, 23, 24, 41] and discontinuous Galerkin discretizations [32, 56].

For the purposes of the current paper, we are indifferent to the choice of a specific numerical method. For the sake of simplicity, our actual implementation uses the first-order upwind discretization based on (2.6) and the Fast Marching solver (in stationary cases), which can find a discretized solution of (2.5) in $O(M \log M)$ operations on a spatial grid with M gridpoints. But other numerical techniques could be used as well whenever we need to solve a fully deterministic problem. Our real focus is on treating initial uncertainties – either in the target Γ or in the running cost K on parts of Ω .

3. Average case optimality under initial uncertainty. We begin by introducing our main sample problem of “optimally pursuing random targets”. We will assume that the dynamics and the running cost are isotropic and time-independent (i.e., $K = K(\mathbf{x})$ and $\mathbf{f}(\mathbf{x}, \mathbf{a}) = f(\mathbf{x})\mathbf{a}$, with $|\mathbf{a}| = 1$), and the goal is to minimize the total cost of reaching a target $\hat{\mathbf{x}} \in \Omega \subset \mathbb{R}^2$. When $\hat{\mathbf{x}}$ is known and there is no deadline, one can do this by solving (2.5) with $\Gamma = \{\hat{\mathbf{x}}\}$.

But what if we only know a set of possible targets $\hat{X} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_m\}$ and the corresponding probability distribution $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_m)$? We can compute each “min cost to the i -th target” $u(\mathbf{x}; \hat{\mathbf{x}}_i)$ by solving (2.5) with $\Gamma = \{\hat{\mathbf{x}}_i\}$, and from now on we will use $u_i(\mathbf{x})$ to denote $u(\mathbf{x}; \hat{\mathbf{x}}_i)$ as long as there is no ambiguity. Then we can define the “expected min cost to target”

$$q(\mathbf{x}) = \mathbb{E}_{\hat{\mathbf{x}}} u(\mathbf{x}) = \sum_{i=1}^m \hat{p}_i u_i(\mathbf{x}). \quad (3.1)$$

Here $u(\mathbf{x})$ is a random variable satisfying $P(u(\mathbf{x}) = u_i(\mathbf{x})) = \hat{p}_i$. The global minima of q suggest the best starting point(s) if the target is immediately revealed. But what if we start elsewhere and what if the actual $\hat{\mathbf{x}}$ is only learned at a later time T ? The real question is what to do until then since the gradient descent in “correct” u_i will define the trajectory after $\hat{\mathbf{x}} = \mathbf{x}_i$ is known.

3.1. Fixed certainty time T . For a fixed and known certainty time T , the answer is encoded by the solution of (2.4) with the terminal condition $u = q$ on $\Omega \times \{T\}$ and the boundary condition $u = +\infty$ on $\partial\Omega \times [0, T)$.

The case $K = 1$ is particularly simple since the total cost incurred until the target identification is T regardless of the chosen control/trajectory. Thus, the goal then becomes to minimize the “terminal cost” q among the points reachable from our starting position $\mathbf{x}_0$. Solving (2.4) has the advantage of encoding the answer for all starting positions simultaneously, but for any specific $\mathbf{x}_0$ we can further simplify the computation. In this case, we solve (2.5) with $\Gamma = \{\mathbf{x}_0\}$ to find the min-time-from- $\mathbf{x}_0$ (denoted $u(\mathbf{x}; \mathbf{x}_0)$; see the example in Figure 1B) and define the reachable set³

$$\Omega_T(\mathbf{x}_0) = \{\mathbf{x} \in \Omega \mid u(\mathbf{x}; \mathbf{x}_0) \leq T\}.$$

To simplify the notation, we will use Ω_T whenever $\mathbf{x}_0$ is clear from the context. The

³With $K = 1$ and fixed T , if we reach $\mathbf{x}$ too early, we can simply wait there without affecting the cumulative cost.

optimal expected time-to-target is then obtained by traveling from $\mathbf{x}_0$ to any waypoint

$$\mathbf{s} \in \arg \min_{\mathbf{x} \in \Omega_T(\mathbf{x}_0)} q(\mathbf{x}).$$

Since there are a total of m targets (each with its own value function u_i), q can be found on the entire Ω in $O(mM \log M)$ operations. After that, for any specific starting position $\mathbf{x}_0$ the optimal waypoint $\mathbf{s}$ can be found in $O(\tilde{M} \log \tilde{M})$ operations needed to determine the $\tilde{M} \leq M$ gridpoints falling in the set $\Omega_T(\mathbf{x}_0)$.

In Figure 2 we show two examples of such delayed-target-identification planning for $T = 0.08$ and $T = 0.4$ and the set of four potential targets already specified in Figure 1. Since q has multiple local minima, the value of T strongly influences the optimal direction of motion starting from $\mathbf{x}_0$. When T is large enough (e.g., above ≈ 0.4528), the optimal waypoint $\mathbf{s}$ is a global minimum of q on the entire Ω .

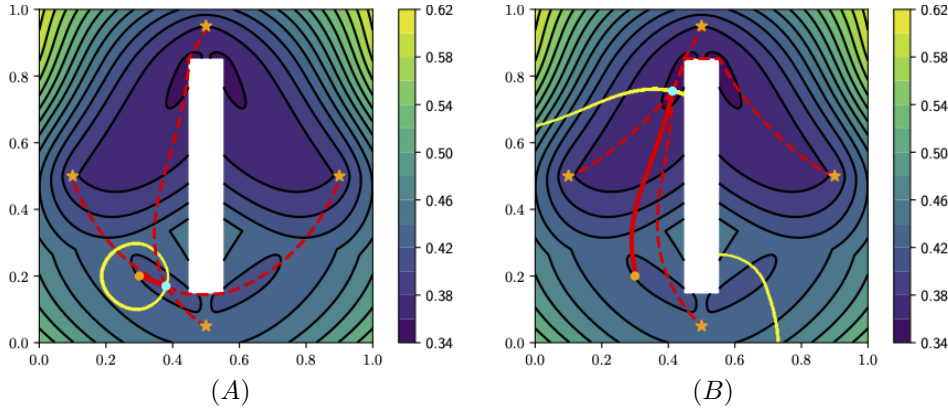


FIG. 2. Fixed certainty time. (A): $T = 0.08$. (B): $T = 0.4$. The speed f , domain geometry, starting position $\mathbf{x}_0$ (orange dot) and four targets $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_4$ (orange stars) are the same as in Figure 1. The corresponding target probabilities (clockwise, starting at the top) are $\hat{\mathbf{p}} = (0.2, 0.3, 0.2, 0.3)$. Boundaries of Ω_T (in yellow) are superimposed on a contour plot of q . The time optimal path from $\mathbf{x}_0$ to the optimal waypoint $\mathbf{s} = \arg \min_{\Omega_T} q(\mathbf{x})$ is shown in red, with dotted red lines showing time-optimal trajectories from $\mathbf{s}$ to each $\hat{\mathbf{x}}_i$. In the left subfigure, the $\mathbf{s}$ -to- $\hat{\mathbf{x}}_4$ trajectory overlaps the already traversed $\mathbf{x}_0$ -to- $\mathbf{s}$ trajectory.

Remark 3.1. We note that our approach in the fixed- T case can be interpreted as an infinite-dimensional version of the standard Two-Stage Stochastic Programming (TSSP) [48]. In the first stage of a *general TSSP* problem, some decision variable $\zeta_1 \in \mathcal{G}_1 \subset \mathbb{R}^{d_1}$ has to be chosen at a known cost $Q_1(\zeta_1)$. A random variable ω is then drawn from a known distribution and the planner chooses their second decision variable $\zeta_2 \in \mathbb{R}^{d_2}$, satisfying a known nonlinear constraint $\zeta_2 \in \mathcal{G}_2(\zeta_1, \omega)$ and incurring the cost $Q_2(\zeta_1, \omega, \zeta_2)$. The goal is to minimize the expected sum of these two costs; i.e.,

$$\min_{\zeta_1 \in \mathcal{G}_1} \left\{ Q_1(\zeta_1) + \mathbb{E}_\omega \left[\min_{\zeta_2 \in \mathcal{G}_2(\zeta_1, \omega)} Q_2(\zeta_2, \omega) \right] \right\}.$$

In our setting, (ζ_1, Q_1) would describe the decisions and cost prior to the target-identification time T , while (ζ_2, Q_2) would describe the decisions and cost after T . If one needs to handle a general running cost K , ζ_1 becomes a choice of a path from our starting point $\mathbf{x}_0$, ω becomes the soon-to-be-identified target, ζ_2 is a chosen path to that target from our position $\mathbf{x}$ at the time T , while $q(\mathbf{x})$ encodes the expected

cumulative cost for the second stage. By using the time-dependent PDE (2.4) with the terminal condition $u = q$ on $\Omega \times \{T\}$, we manage to solve this two-stage problem efficiently despite the fact that the spaces of paths ζ_1 and ζ_2 are infinite-dimensional. Moreover, the value function provides the solution to *an infinite collection* of TSSP problems – for all starting positions $\mathbf{x}_0$ simultaneously. For a single source $\mathbf{x}_0$ and a constant running cost $K = 1$, one can also use the more efficient approach based on a stationary PDE (2.5), with $\zeta_1 \in G_1 = \Omega_T(\mathbf{x}_0)$ interpreted as a chosen reachable waypoint and $Q_1 = T$.

It is also important to note that the source of initial uncertainty needs not be the random nature of the target. Indeed, the element of initial uncertainty could have resulted from one-time changes in the global environment – e.g., changes in the geometry of the domain or in the running cost K . For instance, suppose that the target was known, but we had a list of possible “obstacle” locations (subregions of Ω that would be impossible or highly risky to traverse through). If the actual obstacles were to be revealed at a later time T , we could use the above approach to plan the optimal path until then. In Appendix A we illustrate this with an example of airplane flight-path planning under weather uncertainty.

3.2. Random discrete certainty time T . We now consider a problem in which the time of target discovery T is a discretely distributed random variable. That is, we have a set of times $\{T_1, \dots, T_r\}$ with $0 < T_1 < \dots < T_r$, the probability $\mathbb{P}(T = T_j) = p_j > 0$ is known for each $j = 1, \dots, r$, and $\sum_{j=1}^r p_j = 1$. We will assume that p_j ’s can be also naturally used to define conditional probabilities. I.e., if we know that the target has not been identified by the time $T_j < T_r$, we assume that

$$\mathbb{P}(T = T_{j+k} \mid T > T_j) = p_{j+k} / \sum_{l=j+1}^r p_l, \quad \forall k = 1, \dots, r - j. \quad (3.2)$$

In this setting, the problem can be handled by solving a sequence of HJB equations. To avoid confusion, in this section we will always use v for time-dependent value functions, saving u for solutions of static HJB PDEs. Let $v_j(\mathbf{x}, t)$ denote the expected min-cost to target conditional on that target being not yet identified by the time $T_{j-1} \leq t$. For $j = r$ this means that $\mathbb{P}(T = T_r \mid T > t) = 1$, and we are back to the case already considered in the previous section: v_r satisfies the equation (2.4) on $\Omega \times [T_{r-1}, T_r)$ with the terminal condition $v_r(\mathbf{x}, T_r) = q(\mathbf{x})$.

If $j = r - 1$ and $t \in [T_{r-2}, T_{r-1})$, the target might be identified at the time T_{r-1} (in which case the optimal remaining cost is encoded in q) or postponed until the time T_r (in which case the optimal remaining cost is encoded in $v_r(\cdot, T_{r-1})$ already found above). Thus, on this time interval v_{r-1} satisfies the same PDE (2.4), but with the terminal condition

$$v_{r-1}(\mathbf{x}, T_{r-1}) = \frac{p_r}{p_{r-1} + p_r} v_r(\mathbf{x}, T_{r-1}) + \frac{p_{r-1}}{p_{r-1} + p_r} q(\mathbf{x}). \quad (3.3)$$

Iterating this process (from $j = r - 1$ to $j = 1$) and using $T_0 = 0$ to simplify the notation, we can derive the general case. The value function v_j satisfies the PDE (2.4) on $\Omega \times [T_{j-1}, T_j)$ with the terminal condition

$$v_j(\mathbf{x}, T_j) = \left(1 - \frac{p_j}{\sum_{l=j}^r p_l} \right) v_{j+1}(\mathbf{x}, T_j) + \left(\frac{p_j}{\sum_{l=j}^r p_l} \right) q(\mathbf{x}). \quad (3.4)$$

As before, the computational cost of obtaining q is $O(mM \log M)$, where M is the number of gridpoints in the spatial grid and m is the number of potential targets. If we assume that there are N_j time slices used in discretizing $[T_{j-1}, T_j]$, then the additional computational cost of solving the above r -stage problem is $O(M \sum_{j=1}^r N_j)$.

We now consider an example using the same setup already presented in Figure 2, but with two possible times of target discovery: $T_1 = 0.08$ and $T_2 = 0.4$. Note that, even though $K = 1$ and we are optimizing the expected time to target, it is still necessary to solve two time-dependent PDEs. As an added benefit, we recover the optimal control from every starting position, but for the sake of comparison we focus on the same specific $\mathbf{x}_0$ (shown by an orange dot). In Figure 3 the inner yellow ring illustrates $\partial\Omega_{T_1}(\mathbf{x}_0)$, the boundary of the set reachable from $\mathbf{x}_0$ by the time T_1 . The cyan dot indicates the optimal location $\mathbf{s}_1 \in \Omega_{T_1}(\mathbf{x}_0)$ to reach by that time. The outer yellow boundary shows $\partial\Omega_{(T_2-T_1)}(\mathbf{s}_1)$, the boundary of the set we will have to consider if the target is not revealed at T_1 , and the second cyan dot $\mathbf{s}_2$ is the optimal point to reach in that case. Note that the dependence of $\Omega_{(T_2-T_1)}$ on $\mathbf{s}_1$ is precisely the reason why the stationary formulation of the previous section is not usable. Again, the background shows the level sets of q (the expected time till arrival if the true target is revealed immediately). Figure 3A shows that for $p_1 = 0.9$,

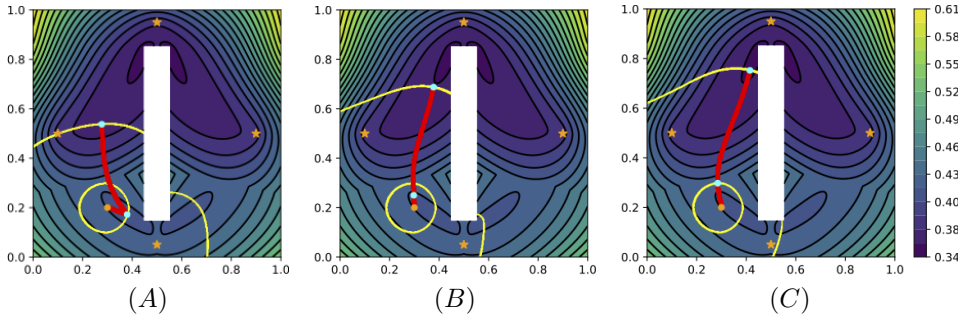


FIG. 3. Random T with possible values $T_1 = 0.08$ and $T_2 = 0.4$. The basic setup is the same as in Figure 2, but the optimal behavior is heavily dependent on T 's probability distribution. (A): $p_1 = 0.9$, $p_2 = 0.1$. (B): $p_1 = 0.55$, $p_2 = 0.45$. (C): $p_1 = 0.1$, $p_2 = 0.9$.

the first optimal waypoint $\mathbf{s}_1$ is close to $\arg \min_{\mathbf{x} \in \Omega_{T_1}(\mathbf{x}_0)} q(\mathbf{x})$, the optimal solution when $p_1 = 1$ (and $T = T_1$ is fixed). There is a somewhat opposite effect in the case of $p_1 = 0.1$ shown in Figure 3C. Since $T = T_2$ is far more likely, we see that $\mathbf{s}_2$ is close to $\arg \min_{\mathbf{x} \in \Omega_{T_2}(\mathbf{x}_0)} q(\mathbf{x})$ and $\mathbf{s}_1$ is selected (despite its relatively high q value) to make this possible. The example with $p_1 = 0.55$ presented in Figure 3B is more of a compromise. It is also interesting that in this case the optimal $\mathbf{s}_1$ is in the interior of $\Omega_{T_1}(\mathbf{x}_0)$.

The above modeling framework is flexible enough to treat possible temporal changes in the target probabilities $\hat{\mathbf{p}}$ and even $\hat{\mathbf{p}}$'s possible dependence on the certainty-time probabilities $\mathbf{p}$. We illustrate this with an additional “emergency rescue” example in Appendix B.

Remark 3.2. We note that the discrete-random- T case can be viewed as a “multistage” (MSSP) generalization of TSSP [48] mentioned in Remark 3.1. A general r -stage stochastic programming problem can be written in a nested form:

$$\min_{\zeta_1 \in \mathcal{G}_1} Q_1(\zeta_1) + \mathbb{E}_{\omega_1} \left[\min_{\zeta_2 \in \mathcal{G}_2(\zeta_1, \omega_1)} Q_2(\zeta_2, \omega_1) + \mathbb{E}_{\omega_2} \left[\cdots + \mathbb{E}_{\omega_{r-1}} \left[\min_{\zeta_r \in \mathcal{G}_r(\zeta_{r-1}, \omega_{r-1})} Q_r(\zeta_r, \omega_{r-1}) \right] \cdots \right] \right],$$

with $\omega_1, \omega_2, \dots, \omega_{r-1}$ being a stochastic process. The expectation $\mathbb{E}_{\omega_j}[\cdot]$ is conditioned on an observed realization of $\omega_1, \omega_2, \dots, \omega_{j-1}$.

For our model, the random outcomes of whether the true target is revealed at each T_j correspond to the stochastic process of MSSP. The terminal conditions defined in (3.3) and (3.4) are conditional expectations of the two possibilities at T_j , provided that $T > T_{j-1}$. The HJB solution $v_j(\mathbf{x}, t)$ in the j th stage encodes the optimal remaining cost for any starting state $\mathbf{x}$, which is naturally used to determine the terminal conditions for the previous stage. Since we are obtaining the optimal control for all initial $(\mathbf{x}, t)$ simultaneously, this is again equivalent to solving an infinite family of infinite-dimensional MSSP problems.

3.3. Exponentially distributed certainty time T . We now move beyond problems that can be solved stage-by-stage in a typical MSSP fashion. Section 3.2 assumes the certainty time to be a discretely distributed random variable. A natural extension is to consider a continuously distributed T . The total cost can be again split into two parts (before and after certainty is achieved), with the expected cost of the latter part still encoded by $q(\mathbf{x})$ defined in section 3.1. The target identification is the termination of the first part, and the optimal control of such *randomly-terminated* processes has already been considered in [5]. We will follow the same approach here, taking T to be exponentially distributed with a known rate $\lambda > 0$. I.e., $\mathbb{E}(T) = 1/\lambda$ and $\mathbb{P}(T > t + \tau \mid T \geq t) = e^{-\lambda\tau}$ for all $t, \tau > 0$.

Under this assumption, the expected cost can be computed as

$$\begin{aligned} \mathcal{J}(\mathbf{x}, \mathbf{a}(\cdot)) &= \int_0^\infty \lambda e^{-\lambda T} \left[\int_0^T K(\mathbf{y}(t)) dt + q(\mathbf{y}(T)) \right] dT \\ &= \int_0^\infty e^{-\lambda t} [K(\mathbf{y}(t)) + \lambda q(\mathbf{y}(t))] dt. \end{aligned}$$

Thus, this problem can be also viewed as a discounted infinite-horizon problem with λ interpreted as a discounted factor and $(K(\mathbf{x}) + \lambda q(\mathbf{x}))$ as a new running cost. The value function $u^\lambda(\mathbf{x}) = \inf_{\mathbf{a}(\cdot)} \mathcal{J}(\mathbf{x}, \mathbf{a}(\cdot))$ can be recovered as a *viscosity solution* of a Hamilton-Jacobi PDE

$$\lambda (u^\lambda(\mathbf{x}) - q(\mathbf{x})) + |\nabla u^\lambda(\mathbf{x})| f(\mathbf{x}) = K(\mathbf{x}). \quad (3.5)$$

Focusing on time optimality, we can use $K = 1$ to have $u^\lambda(\mathbf{x})$ encode the (minimized) full expected-time-to-target from $\mathbf{x}$ to uncertain $\hat{\mathbf{x}}$. Alternatively, we can use the fact that $\mathbb{E}(T) = 1/\lambda$ is trajectory independent and take $K = 0$ to have $u^\lambda(\mathbf{x})$ encode the (minimized) expected-time-to-target-after-it-is-identified. We follow [5] and use the latter, which yields $u^\lambda \leq q$ since one of the options is to stay in place until the target is identified. This allows recovering the value function from a quasi-variational inequality of obstacle type

$$\max \left(u^\lambda - q, \quad \lambda (u^\lambda - q) + f|\nabla u^\lambda| \right) = 0. \quad (3.6)$$

We use $\mathcal{M}^\lambda$ to denote the *motionless set* of starting positions at which $u^\lambda = q$, so that $\lambda (u^\lambda - q) + f|\nabla u^\lambda| = 0$ is solved on $\Omega \setminus \mathcal{M}^\lambda$. For any starting position $\mathbf{x}_0 \notin \mathcal{M}^\lambda$, the optimal trajectory is found by gradient descent in u^λ and leads to some motionless waypoint $\mathbf{s} \in \mathcal{M}^\lambda$. If the target is not identified by the time we reach $\mathbf{s}$, it is optimal

to stay there until $\hat{\mathbf{x}}$ is revealed. This is a problem with *free boundary* since $\mathcal{M}^\lambda$ is not known in advance. (With $K = 0$, it is easy to show that every point in $\mathcal{M}^\lambda$ is a local minimum of q . The converse is true for global minima of q , but its local minima may or may not be motionless depending on λ and the global properties of q .) Nevertheless, a modified version of Fast Marching Method can be still used to compute the numerical approximation efficiently (see [5, Section 3.3 and Appendix B] for implementation details). Since q has to be computed first before solving (3.5), the overall cost is $O((m+1)M \log M)$.

The value of parameter λ determines whether one should make a quick transition to a nearby local minimum or spend more time on a path to the global minimum (risking that the target is identified while we are still traveling through high values of q). The example in Figure 4 illustrates this, with $\mathbb{E}[T] = 1/2.5 = 0.4$ in the left subfigure producing a trajectory similar to the one in Figure 2B, and $\mathbb{E}[T] = 1/30$ in the right subfigure resulting in a completely different path to a local minimum. We note that with an intermediate $\mathbb{E}[T] = 1/20$ that nearby local minimum is already motionless, but the optimal trajectory from our chosen $\mathbf{x}_0$ still leads to a much farther global minimum.

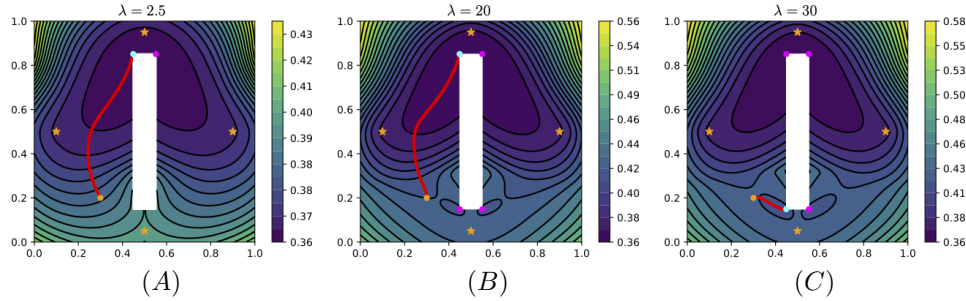


FIG. 4. Exponentially distributed T . Each subfigure shows the level sets of u^λ , starting position $\mathbf{x}_0$ (in orange), optimal waiting position $\mathbf{s}$ (in cyan), motionless local minima of u^λ (in magenta) and the optimal trajectory (in red). (A): with $\lambda = 2.5$ an early target identification is not very likely, and the optimal trajectory leads to the global minimum of q . (B): with $\lambda = 20$ still heading to the global minimum although the nearby local minimum is already motionless. (C): with $\lambda = 30$ reaching the global minimum before the target identification is less unlikely, and it is optimal to head toward the local minimum nearby.

4. Robust path planning. We now turn our attention to robust control under initial uncertainty, using the same example previously considered in section 3.1 (i.e., minimizing time-to-target that will be revealed at a known time T). Our goal is to compare several different notions of robustness: the worst-case optimization (§4.1), the risk-sensitive average optimization (§4.2), optimizing the average with a hard constraint on the worst case (§4.3), and optimizing the average with a “chance constraint” on bad outcomes (§4.4). As we will show, the corresponding optimal waypoints and trajectories are quite different in each case, including the probabilistic definition of optimal waypoints in §4.4. Along the way, we will also provide a brief discussion of how these robustness models relate to each other. Two of these (in §4.1 and §4.3) will be also extended to the case of an exponentially distributed random certainty time T (introduced in §3.3).

All of the above describe robustness to *aleatoric uncertainty*, i.e., uncertainty due to a random choice based on a known probability distribution $\hat{\mathbf{p}}$ over a known set

of potential targets $\hat{X}$. We end this section by exploring robustness with respect to *epistemic uncertainty* (§4.5); i.e., possible uncertainties in $\hat{\mathbf{p}}$ or $\hat{X}$.

4.1. Worst-case scenario. Perhaps the most obvious robust approach is to consider a Stackelberg-type “game against nature”, where the worst target is always chosen (by a supposed opponent) based on our choice of a waypoint. In this case, the time to target after identification is

$$\bar{q}(\mathbf{x}) = \max\{u_1(\mathbf{x}), \dots, u_m(\mathbf{x})\}. \quad (4.1)$$

and the worst-case optimal solution is to go from $\mathbf{x}_0$ to any waypoint

$$\bar{\mathbf{s}} \in \arg \min_{\mathbf{x} \in \Omega_T(\mathbf{x}_0)} \bar{q}(\mathbf{x}). \quad (4.2)$$

This approach (illustrated in Figure 5A) does not use any information from $\hat{\mathbf{p}}$ and is thus very conservative. E.g., the worst-case scenario could be very unlikely and $q(\bar{\mathbf{s}})$ can be quite far from the average-case optimal $q(\mathbf{s})$. Similarly, the worst-case scenario $\bar{q}(\mathbf{s})$ can be much costlier than $\bar{q}(\bar{\mathbf{s}})$. While these quantities can be compared directly, a more nuanced discussion of the “worst versus average” trade-offs will be covered in §4.3. The computational cost of obtaining $\bar{\mathbf{s}}$ through our Fast Marching implementation is still $O(mM \log M)$.

As we show in Figure 5B, the same approach also works for the exponentially distributed random certainty time discussed in section 3.3. For the specified rate $\lambda > 0$, the worst-case optimal trajectory is found by using $\bar{q}$ instead of q in PDE (3.5) and then following $(-\nabla u^\lambda)$ from $\mathbf{x}_0$ to the corresponding motionless point. This waypoint $\bar{\mathbf{s}}$ is one of the local minima of $\bar{q}$, but we may not reach it if the target is identified earlier. (Note that this is the worst case with respect to $\hat{\mathbf{x}}$ only while T is still assumed to be random. If we allowed T to be chosen by the opponent/nature, it would always result in $T = 0$, yielding $\bar{q}(\mathbf{x}_0)$ as our time to the worst-case target.)

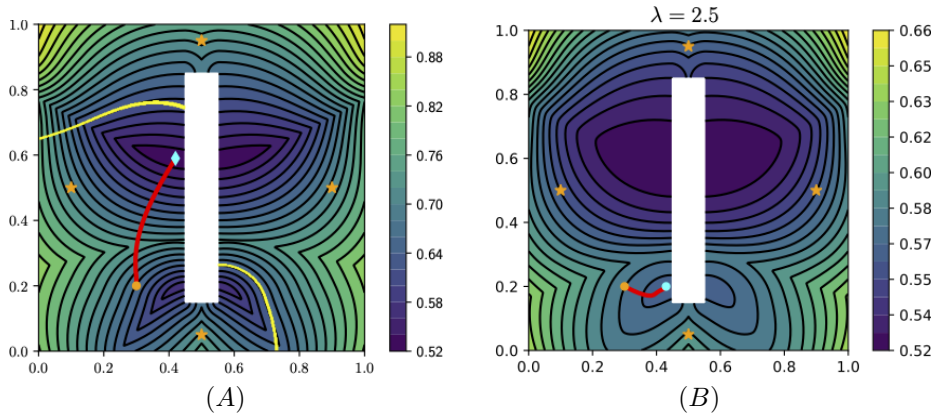


FIG. 5. Worst-case optimal planning. (A): the same (“Fixed $T = 0.4$ ”) setting as in Figure 2B. The cyan diamond represents a $\bar{\mathbf{s}}$ and the red curve is the time-optimal trajectory, with level sets of $\bar{q}$ in the background. (B): the same (“Exponentially distributed T with $\mathbb{E}[T] = 0.4$ ”) setting as in Figure 4A except that q is replaced by $\bar{q}$. Level sets of u^λ are shown in the background.

4.2. Risk-sensitive optimization. The notion of risk sensitivity most commonly used in optimal control and Markov Decision Processes is based on minimizing

the expectation of an exponential function of the outcome [25, 28]. Given a risk-sensitivity parameter $\beta > 0$, it is natural to define

$$q_\beta(\mathbf{x}) = \mathbb{E}_{\hat{\mathbf{x}}}[e^{\beta u(\mathbf{x})}] = \sum_{i=1}^m \hat{p}_i e^{\beta u_i(\mathbf{x})}. \quad (4.3)$$

The waypoint is then chosen as

$$\mathbf{s}_\beta \in \arg \min_{\mathbf{x} \in \Omega_T} q_\beta(\mathbf{x}).$$

Figure 6 illustrates this approach for $T = 0.4$ and two different β values. Higher β corresponds to more risk-averse behavior. As β increases, the exponential of the worst outcome starts to dominate the expectation; so, $\mathbf{s}_\beta \rightarrow \bar{\mathbf{s}}$ as $\beta \rightarrow +\infty$. On the other hand, when β is small, the Taylor expansion yields $q_\beta \approx 1 + \beta q$; so, $\mathbf{s}_\beta \rightarrow \mathbf{s}$ as $\beta \rightarrow 0$.

Another approximation valid for small β 's establishes the connection with the mean-variance trade-offs:

$$\log(q_\beta)/\beta \approx \mathbb{E}_{\hat{\mathbf{x}}} u(\mathbf{x}) + \frac{\beta}{2} \mathbb{E}_{\hat{\mathbf{x}}} [u(\mathbf{x}) - \mathbb{E}_{\hat{\mathbf{x}}} u(\mathbf{x})]^2. \quad (4.4)$$

Unfortunately, this holds only asymptotically and is not very useful in estimating the worst-average case trade-offs for positive β values; see Figure 7 and Remark 4.1.

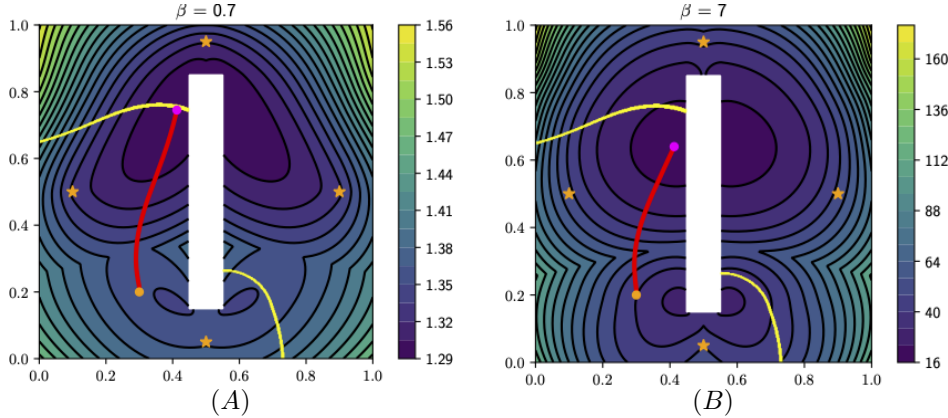


FIG. 6. Risk-sensitive planning. The red curve is a time-optimal trajectory toward $\mathbf{s}_\beta$ (magenta dot). Level sets of q_β are shown in the background. The position of $\mathbf{s}_\beta$ depends on the magnitude of β . (A): A small β leads to $\mathbf{s}_\beta$ being close to $\mathbf{s}$ (compare with Figure 2B). (B): β is large and $\mathbf{s}_\beta$ moves closer to $\bar{\mathbf{s}}$ (compare with Figure 5A).

4.3. Optimizing the average with a hard constraint on the worst case.

As usual, when evaluating more than one performance criterion, we have to employ the notion of *Pareto-optimality*. Focusing on $K = 1$ and deterministic T , we will say that a waypoint $\mathbf{z}_1$ is dominated by another waypoint $\mathbf{z}_2$ if both $\bar{q}(\mathbf{z}_2) \leq \bar{q}(\mathbf{z}_1)$, $q(\mathbf{z}_2) \leq q(\mathbf{z}_1)$, and at least one of these inequalities is strict. We will say that a waypoint $\mathbf{z} \in \Omega_T(\mathbf{x}_0)$ is Pareto-optimal if it is not dominated by any other waypoint in $\Omega_T(\mathbf{x}_0)$. Collectively, these waypoints encode all rational “average versus worst” trade-offs since none of them can be improved with regard to both criteria simultaneously. These trade-offs are well represented visually by a Pareto Front

$PF = \{(\bar{q}(\mathbf{z}), q(\mathbf{z})) \mid \mathbf{z} \text{ is Pareto-optimal}\}$, which a practitioner could use to find the desired compromise. Every such waypoint can be viewed as an answer to a constrained optimization problem: minimize q subject to $\bar{q} \leq C$ (or, equivalently, ensuring that the total time to target is below $T + C$). For a fixed starting position $\mathbf{x}_0$, this can be done by restricting the domain to

$$\Omega_{T,C}(\mathbf{x}_0) = \Omega_T \cap \Omega_C, \quad \text{where } \Omega_C = \{\mathbf{x} \in \Omega \mid \bar{q}(\mathbf{x}) \leq C\},$$

and then selecting a waypoint

$$\mathbf{s}_c \in \arg \min_{\mathbf{x} \in \Omega_{T,C}(\mathbf{x}_0)} q(\mathbf{x}).$$

Note that the problem has no solution if C is too small (resulting in $\Omega_{T,C}(\mathbf{x}_0) = \emptyset$), while for large enough C the problem becomes unconstrained, yielding the same waypoints obtained in section 3.1. A more efficient approximation of the entire PF is obtained by evaluating $(\bar{q}, q)$ at every gridpoint in $\Omega_T(\mathbf{x}_0)$ and then removing the dominated ones. The result of this approach for $T = 0.4$ and $C = 0.56$ is shown in Figure 7.

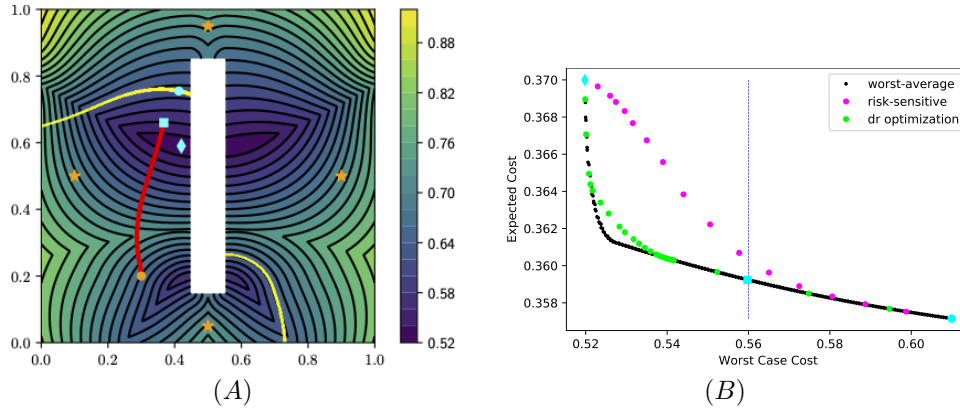


FIG. 7. Minimizing averaged cost with a hard constraint $C = 0.56$. (A): Three different optimal waypoints $(\mathbf{s}, \bar{\mathbf{s}}, \mathbf{s}_c)$ corresponding to sections 3.1, 4.1 and 4.3 are shown by cyan markers (the circle, diamond and square respectively). The red curve is the time optimal trajectory from $\mathbf{x}_0$ to the latter, with the level sets of $\bar{q}$ in the background. (B): The Pareto Front approximation is shown in black, with the cyan markers corresponding to the 3 waypoints shown on the left. The dashed line is $\bar{q} = 0.56$ and the square marker is the lowest one on the Pareto front satisfying $\bar{q} \leq 0.56$. The scattered magenta and green dots correspond to the risk-sensitive and distributionally-robust waypoints discussed in sections 4.2 and 4.5.

Remark 4.1. By the optimality of $\mathbf{s}_\beta$ and Jensen's inequality, we have

$$\mathbb{E}_{\hat{\mathbf{x}}}[e^{\beta u(\mathbf{s})}] = q_\beta(\mathbf{s}) \geq q_\beta(\mathbf{s}_\beta) = \mathbb{E}_{\hat{\mathbf{x}}}[e^{\beta u(\mathbf{s}_\beta)}] \geq e^{\beta \mathbb{E}_{\hat{\mathbf{x}}}[u(\mathbf{s}_\beta)]} = e^{\beta q(\mathbf{s}_\beta)}.$$

After rearranging this inequality and applying (4.4) we obtain

$$q(\mathbf{s}_\beta) \leq \log \left(\mathbb{E}_{\hat{\mathbf{x}}}[e^{\beta u(\mathbf{s})}] \right) / \beta \approx q(\mathbf{s}) + \frac{\beta}{2} \mathbb{E}_{\hat{\mathbf{x}}}[u(\mathbf{s}) - q(\mathbf{s})]^2.$$

which means the vertical gap between $q(\mathbf{s}_\beta)$ and $q(\mathbf{s})$ is approximately bounded by $\frac{\beta}{2} \text{Var}[u(\mathbf{s})]$ when β is relatively small.

If C and T are fixed, the solution can be obtained for all starting positions at once by solving the time-dependent PDE (2.4) with $K = 1$ and the terminal condition $u(\mathbf{x}, T) = q(\mathbf{x})$ on Ω_C and $u(\mathbf{x}, T) = +\infty$ on $\Omega \setminus \Omega_C$. A constraint on the *remaining time-to-target after $\hat{\mathbf{x}}$ is identified* can be similarly handled even with the exponentially distributed random T discussed in section 3.3. We define $u^{\lambda, C}$ to be the solution of (3.5) on the restricted domain Ω_C (see Figures 8A, 8B for two examples with different C values). By varying C we can also approximate the Pareto Front as shown in Figure 8C. The average case performance is encoded by $u^{\lambda, C}(\mathbf{x}_0)$, while the worst case corresponds to the largest value of $\bar{q}$ encountered on the optimal trajectory from $\mathbf{x}_0$ (found by following $-\nabla u^{\lambda, C}$). This worst-case result is equal to C if that trajectory touches $\partial\Omega_C \setminus \partial\Omega$, but as Figure 8B shows, this is not always the case.

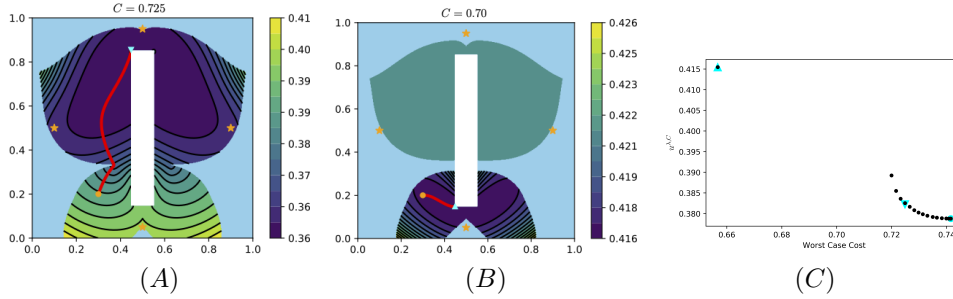


FIG. 8. Worst-case-constrained optimization for the exponentially distributed T with $\lambda = 2.5$. (A): a binding constraint $\bar{q} \leq 0.725$. The cyan dot shows the same waypoint as in Figure 4A; i.e., $\mathbf{s}_c = \mathbf{s}$ is the global minimum of q . But the optimal trajectory bends to stay inside Ω_C and $u^{\lambda, C}(\mathbf{s}) > u^\lambda(\mathbf{s})$. (B): a non-binding constraint $\bar{q} \leq 0.7$. As C decreases, Ω_C becomes disconnected and the optimal trajectory leads to a closer local minimum. The dark cyan region is another connected component of Ω_C . (C): the Pareto Front (worst vs average case performance). There are no solutions for $C < \bar{q}(\mathbf{s}) \approx 0.6566$ and we use C values between 0.66 and 0.81 to approximate PF. The cyan upward and downward triangles correspond to the the left and middle figures respectively. The cyan dot corresponds to the unconstrained λ -optimal trajectory in Figure 4A.

Before moving on to other types of robust path planning, we note that this “constrain the worst, optimize the average” strategy can be also computationally efficient even in a more general setting, where the uncertainty is not monotonically decreasing. E.g., it was successfully used in robust routing on stochastic networks [22], where the target was pre-specified, but each edge transition incurred a random penalty.

Remark 4.2. The computational cost of producing q , $\bar{q}$, and q_β is the same. So, finding $\bar{\mathbf{s}}$ or $\mathbf{s}_\beta$ for a specific β costs exactly as much as obtaining the risk-neutral $\mathbf{s}$. The same is also true for $\mathbf{s}_c$ corresponding to any specific C . (In fact, the computational cost is usually reduced since Ω_C is smaller than Ω .) To approximate the Pareto front, we can first sort all $(\bar{q}, q)$ pairs in $\Omega_T(\mathbf{x}_0)$ by the $\bar{q}$ values, then go through all of them in increasing order and prune all dominated pairs. The computational cost of this process is $O(\tilde{M} \log \tilde{M})$, where $\tilde{M} \leq M$ is the number of gridpoints in $\Omega_T(\mathbf{x}_0)$.

4.4. Probabilistic constraints on bad outcomes. Probabilistic constraints (also frequently known as *chance constraints*) provide a fairly general version of robustness when optimizing the expected performance. Unfortunately, they often result in much tougher optimization problems [37, 39, 53], but we will show that in our simple setting the additional computational cost is only moderate.

Focusing on each potential target $\hat{\mathbf{x}}_i$, it is natural to define its “ C -unreachable

set" $\hat{\Omega}_i = \{\mathbf{x} \in \Omega_T(\mathbf{x}_0) \mid u_i(\mathbf{x}) > C\}$ and the corresponding indicator function $\chi_i(\mathbf{x}) = \mathbf{1}_{\hat{\Omega}_i}(\mathbf{x})$. The risk function can be then defined as the probability of not meeting this constraint on the remaining time to target if we are at the point $\mathbf{x}$ when that target is finally identified. I.e.,

$$r(\mathbf{x}) = \sum_{i=1}^m \hat{p}_i \chi_i(\mathbf{x}). \quad (4.5)$$

We note that the worst-case-constrained expectation optimization of section 4.3 can be viewed as minimizing $q(\mathbf{x})$ while guaranteeing that $r(\mathbf{x}) = 0$. This approach is highly conservative since it disqualifies many waypoints $\mathbf{x}$ with small q simply because some $u_i(\mathbf{x}) > C$ even if the corresponding $\hat{p}_i$ is quite small. It might seem natural to relax this hard constraint by minimizing q over the set $\{\mathbf{x} \in \Omega_T(\mathbf{x}_0) \mid r(\mathbf{x}) \leq \epsilon\}$. But since our goal is to limit the overall risk, better results are obtained by allowing the planner to use probabilistic/mixed strategies. We can occasionally use waypoints with $r > \epsilon$ which are attractive because of their small q , and still limit the overall risk by also using waypoints with $r < \epsilon$ whose q values are not as good. An intermittent use of both $\mathbf{s}$ and $\mathbf{s}_c$ is a simple example of this kind of strategy. With $C = 0.56$ and $T = 0.4$ corresponding to Figure 7, this actually happens to be optimal if we use $\mathbf{s}$ with probability $\theta_s = \min(1, \epsilon/r(\mathbf{s}))$ and head to $\mathbf{s}_c$ with probability $(1 - \theta_s)$. However, we will show that this use of $\mathbf{s}$ and $\mathbf{s}_c$ is not generically optimal. In many cases, the set $\Omega_{T,C}(\mathbf{x}_0)$ may be even empty, making $\mathbf{s}_c$ undefined. Below we consider a more interesting example with a lower C value resulting in a positive risk r on the entire $\Omega_T(\mathbf{x}_0)$.

For full generality, we could let a planner select a probability measure over all possible waypoints in $\Omega_T(\mathbf{x}_0)$, minimizing the expected q and restricting the expected r with respect to that measure. But in practice we will assume that the planner selects a discrete pdf $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$ over the set of gridpoints $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \Omega_T(\mathbf{x}_0)$. I.e., we will assume that $\theta_j \geq 0$ is the probability of choosing a gridpoint $\mathbf{x}_j$ as our waypoint. In this setting, the probabilistically-constrained optimization can be performed by solving a finite-dimensional *linear program* with θ_j 's as decision variables:

$$\begin{aligned} & \text{minimize} && \sum_{j=1}^n \theta_j q(\mathbf{x}_j) \\ & \text{subject to} && \sum_{j=1}^n r(\mathbf{x}_j) \theta_j \leq \epsilon, \\ & && \sum_{j=1}^n \theta_j = 1, \quad \theta_j \geq 0, \quad j = 1, \dots, n. \end{aligned} \quad (4.6)$$

As long as the feasible set is non-empty, it will be an $(n - 1)$ -dimensional polytope with a minimum attained at one of its vertices (i.e., where $(n - 1)$ constraints are active). Since there is only one constraint based on the risk, at least $(n - 2)$ values of θ_j 's are actually zero at any vertex. Thus, regardless of m and n , there always exists an optimal solution assigning non-zero probability to at most 2 waypoints as long as the feasible set is non-empty. If one wished to impose κ probabilistic constraints

$$\sum_{j=1}^n \mathbb{P}(u > C_k) \leq \epsilon_k, \quad k = 1, \dots, \kappa,$$

a similar argument shows that an optimal mixed strategy could be found based on at most $(\kappa + 1)$ waypoints⁴.

While the general linear programming algorithms are certainly suitable for solving (4.6) based on a 2D grid, we also describe a better (more geometric and efficient) approach based on (r, q) -Pareto optimality. The set of possible waypoints $\mathcal{X}$ is naturally mapped to $\mathcal{X}^{r,q} = \{(r(\mathbf{x}), q(\mathbf{x})) \mid \mathbf{x} \in \mathcal{X}\}$. Any point within its convex hull $\Upsilon = \text{co}(\mathcal{X}^{r,q})$ is attainable by selecting an appropriate $\boldsymbol{\theta}$. The solution of (4.6) corresponds to $\min \{q \mid (r, q) \in \Upsilon, r \leq \epsilon\}$, which is actually attained on Υ 's "southwestern" boundary

$$\mathcal{V} = \{(r, q) \in \partial\Upsilon \mid \text{with } r < r' \text{ or } q < q', \forall (r', q') \in \Upsilon \setminus \{(r, q)\}\}.$$

This $\mathcal{V}$ is a simple polygonal chain whose vertices are among the Pareto-optimal points of $\mathcal{X}^{r,q}$, stretching from a "first- r -then- q " minimizing waypoint on the left to a "first- q -then- r " minimizing waypoint on the right; see Figure 9B. Any of the planar convex hull methods can be used to compute $\mathcal{V}$ efficiently; e.g., *Graham scan* [27] or *monotone chain* [4] algorithms can compute it in $\mathcal{O}(n \log n)$ operations. If $\partial\Upsilon$ has only $k \ll n$ vertices on it, *Chan's algorithm* [14] yields an even better worst-case complexity of $\mathcal{O}(n \log k)$.

Once $\mathcal{V}$ is obtained as an r -sorted list of its k vertices, this essentially solves (4.6) for all possible ϵ 's at once. If $\epsilon < \min_{(r,q) \in \mathcal{V}} r$, the problem has no solution. If $\epsilon \geq \max_{(r,q) \in \mathcal{V}} r$, the problem is unconstrained. In all other cases, the line $r = \epsilon$ intersects some segment of $\mathcal{V}$, which can be found by a binary search in $\mathcal{O}(\log k)$ steps. This intersection point yields the corresponding probabilities with which the waypoints at the segment vertices should be selected; see Figure 9B. If the line $r = \epsilon$ passes through a vertex of $\mathcal{V}$, it is optimal to use the corresponding single waypoint deterministically.

Remark 4.3. For the case $m \ll \log n$, additional speed up can be achieved by ignoring the dominated (i.e., the non-Pareto-optimal) points in $\mathcal{X}^{r,q}$. Note that in our setting $r(\mathbf{x})$ is a piecewise constant function with at most 2^m possible values (based on whether or not $\mathbf{x}$ belongs to each $\hat{\Omega}_i$). When computing $r(\mathbf{x})$, we can check which $\hat{\Omega}_i$'s $\mathbf{x}$ belongs to and compare its $q(\mathbf{x})$ with the smallest q already computed within the same- r -value group. Thus, maintaining a list of minimal q values for all groups takes $\mathcal{O}(nm)$ steps. Notice that all the 2^m r values are generated by taking sums of all the subsets of $\{\hat{p}_1, \dots, \hat{p}_m\}$. If all the $\hat{p}_i$'s are sorted (which can be done with $\mathcal{O}(m \log m)$ steps), using the merge sort algorithm we can sort all the 2^m values in $\mathcal{O}(2^m)$ steps. Once groups are already r -sorted, discarding the dominated group minimals takes $\mathcal{O}(2^m)$ steps and Graham scan will now compute $\mathcal{V}$ in only $\mathcal{O}(2^m)$ steps. Each look-up for a specific ϵ will now cost $\mathcal{O}(m)$.

4.5. Robustness to modeling errors. Up till now, we have only considered aleatoric uncertainty, taking on faith the correctness of input data: the probability distribution over the set of possible targets and their exact locations were always assumed to be known. We next discuss the robustness with respect to epistemic (or "systemic") uncertainty caused by modeling errors.

4.5.1. Robustness to errors in $\hat{\mathbf{p}}$. In practice, the probability distribution $\hat{\mathbf{p}}$ over the set of targets will be usually defined using historical data. So, the true

⁴The authors are grateful to Dmitriy Drusvyatskiy for pointing this out.

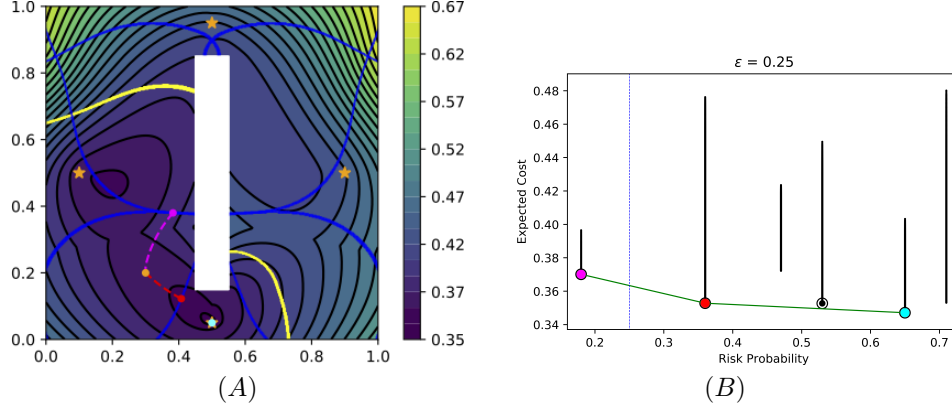


FIG. 9. Probabilistic constraints ($C = 0.365$; $\epsilon = 0.25$). (A): Level sets of $q(\mathbf{x})$ corresponding to $\hat{p}_1 = \hat{p}_2 = 0.18$, $\hat{p}_3 = 0.35$, and $\hat{p}_4 = 0.29$ with the same target locations as in Figure 2. The boundary of $\Omega_T(\mathbf{x}_0)$ is yellow and the boundaries of Ω_i 's are blue. The “unconstrained optimal” waypoint $\mathbf{s} = \arg \min_{\mathbf{x} \in \Omega_T(\mathbf{x}_0)} q(\mathbf{x})$ is located exactly at the target $\hat{\mathbf{x}}_3$ (in cyan). The pair of (C, ϵ) -optimal waypoints are shown in red and magenta, and the optimal mixed strategy selects them with the respective probabilities of 0.3889 and 0.6111. (B): The set $\mathcal{X}^{r,q}$ is shown in black; its non-dominated points are shown by circular markers, while $\mathcal{V}$ (the “southwestern boundary” of its convex hull Υ) is shown in green. The line $r = \epsilon$ is shown in blue and the ratio in which it divides the segment of $\mathcal{V}$ determines the probabilities of red and magenta waypoints in the optimal mixed strategy. The magenta waypoint can be also viewed as maximizing the probability of a desirable event (i.e., $u < C$). Even though $m = 4$, there are only 6 values of r in this example since many of the 2^m groups of gridpoints are empty and a few groups give the same values (since $\hat{p}_1 = \hat{p}_2$).

distribution would typically be different from such nominal/assumed $\hat{\mathbf{p}}$. *Distributionally robust* (DR) optimization deals with this uncertainty by defining an *ambiguity set* of possible distributions, and letting the opponent/Nature select the worst among them while we are minimizing the expected cost [20]. We will use the standard *total variation distance*, which for probability distributions $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_m)$ and $\tilde{\mathbf{p}} = (\tilde{p}_1, \dots, \tilde{p}_m)$ over a finite set $\hat{X}$ can be conveniently computed as

$$W(\hat{\mathbf{p}}, \tilde{\mathbf{p}}) = 1 - \sum_{i=1}^m \min(\hat{p}_i, \tilde{p}_i) = \frac{1}{2} \|\hat{\mathbf{p}} - \tilde{\mathbf{p}}\|_1, \quad (4.7)$$

coinciding with a Wasserstein-1 distance if we use a discrete metric $d(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) = 1 - \delta_{ij}$ on the set of targets [51].

We will refer to our assumed pdf $\hat{\mathbf{p}}$ as the *nominal distribution* and will define the ambiguity set to be a closed ball $B_\delta(\hat{\mathbf{p}}) = \{\tilde{\mathbf{p}} : W(\hat{\mathbf{p}}, \tilde{\mathbf{p}}) \leq \delta\}$. If $\tilde{q}(\mathbf{x}, \tilde{\mathbf{p}})$ denotes the expected time-to-target corresponding to $\tilde{\mathbf{p}}$ and $\tilde{q}_\delta(\mathbf{x})$ is the worst $\tilde{q}(\mathbf{x}, \tilde{\mathbf{p}})$ among all $\tilde{\mathbf{p}}$'s in $B_\delta(\hat{\mathbf{p}})$, the DR approach prescribes traveling to a waypoint $\tilde{\mathbf{s}}_\delta$ that minimizes that worst outcome. In other words,

$$\tilde{q}(\mathbf{x}, \tilde{\mathbf{p}}) = \sum_{i=1}^m \tilde{p}_i u_i(\mathbf{x}), \quad \tilde{q}_\delta(\mathbf{x}) = \max_{\tilde{\mathbf{p}} \in B_\delta(\hat{\mathbf{p}})} \tilde{q}(\mathbf{x}, \tilde{\mathbf{p}}), \quad \tilde{\mathbf{s}}_\delta \in \arg \min_{\mathbf{x} \in \Omega_T} \tilde{q}_\delta(\mathbf{x}).$$

Our explicit representation of $W(\hat{\mathbf{p}}, \tilde{\mathbf{p}})$ makes computing $\tilde{q}_\delta(\mathbf{x})$ particularly easy. Suppose that $u_i(\mathbf{x})$'s are sorted in ascending order; i.e., $u_1(\mathbf{x}) \leq u_2(\mathbf{x}) \leq \dots \leq u_m(\mathbf{x})$. The maximizer $\tilde{\mathbf{p}}_\delta = (\tilde{p}_{\delta,1}, \dots, \tilde{p}_{\delta,m})$ will increase the probability of the “hardest target” $\hat{\mathbf{x}}_m$ while compensating with a corresponding decrease in probabilities of the

“easiest targets” ($\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots$). The first half of this plan can be implemented by setting $\tilde{p}_{\delta,m} = \min\{1, \hat{p}_m + \delta\}$. If $\tilde{p}_{\delta,m} = 1$ this immediately implies $\tilde{p}_{\delta,i} = 0$ for $i < m$. Otherwise, this increase is compensated by starting with $\tilde{p}_{\delta,1} = \max\{0, \hat{p}_1 - \delta\}$ and continuing (e.g., with $\tilde{p}_{\delta,2} = \max\{0, \hat{p}_2 - \delta + (\hat{p}_1 - \tilde{p}_{\delta,1})\}$) until we reach a total reduction of δ . Since the difference between $u_i(\mathbf{x})$ ’s is at most $[u_m(\mathbf{x}) - u_1(\mathbf{x})]$, the DR loss can simply be bounded as $\tilde{q}_\delta(\mathbf{x}) \leq q(\mathbf{x}) + \delta[u_m(\mathbf{x}) - u_1(\mathbf{x})]$. This reveals that, for small δ , minimizing $\tilde{q}_\delta$ is approximately equivalent to minimizing $q(\mathbf{x})$. Obtaining $\tilde{\mathbf{s}}_\delta$ mainly involves computing $u_1, \dots, u_m$, which takes $O(mM \log M)$ operations, and sorting $\hat{p}_1, \dots, \hat{p}_m$, which takes $O(m \log m)$ operations.

In Figure 10 we show the results of the DR approach for the fixed $T = 0.4$ example of section 3.1. For $\delta = 0$, there is no ambiguity in distribution and we are left with $\tilde{q}_\delta = q$, the case presented in Figure 2. For $\delta \geq (1 - \min_i \hat{p}_i)$, we know that $\tilde{\mathbf{p}}_\delta$ will assign probability 1 to the worst target, and $\tilde{q}_\delta = \bar{q}$. (Note that this can happen even for smaller δ ’s; e.g., when the probability of the most difficult target is higher. See Figure 10C.) But for the intermediate δ values, $\tilde{q}_\delta$ continues to change and the corresponding waypoint $\tilde{\mathbf{s}}_\delta$ will be different from $\mathbf{s}$ and $\bar{\mathbf{s}}$.

It is also worth asking how good is the DR approach for balancing the worst-average case performance with respect to our nominal distribution $\hat{\mathbf{p}}$. The green dots in Figure 7B correspond to the same waypoints used in Figure 10. They clearly show that the DR waypoints are better in this sense than the risk-sensitive waypoints (defined in section 4.2), but certainly are not Pareto optimal. An additional example in Appendix C shows that the DR approximation can also miss most of the Pareto frontier.

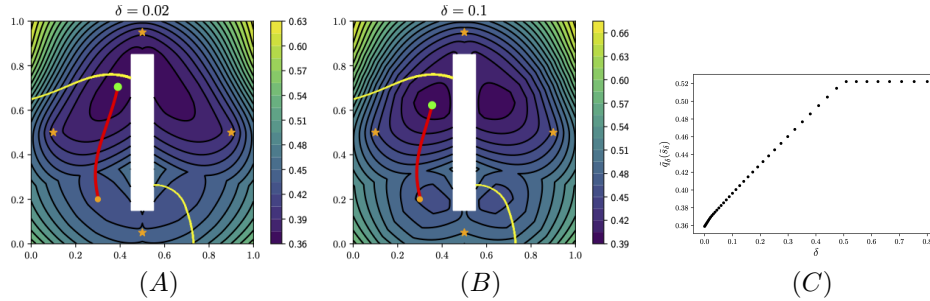


FIG. 10. DR optimization. (A) & (B): $\delta = 0.02, 0.1$ respectively. The red curve is time-optimal to a waypoint $\tilde{\mathbf{s}}_\delta$ (lime dot). Level sets of $\tilde{q}_\delta$ are shown in the background. (C): $\tilde{q}_\delta(\tilde{\mathbf{s}}_\delta)$ computed using 40 different δ values between 0 and 0.8.

4.5.2. Coarsening the target set. All problems considered so far assumed that the target set is finite. We now suppose that the target $\hat{\mathbf{x}}$ is a random variable taking values in Ω , with a general probability measure $\hat{\mu}$. The expected minimum time-to-target is

$$\xi(\mathbf{x}) = \mathbb{E}_{\hat{\mathbf{x}}} u(\mathbf{x}; \hat{\mathbf{x}}) = \int_{\Omega} u(\mathbf{x}; \mathbf{y}) d\hat{\mu}(\mathbf{y}), \quad (4.8)$$

where $u(\mathbf{x}; \mathbf{y})$ is obtained by solving (2.5) with $\Gamma = \{\mathbf{y}\}$. If our goal is to achieve the average-case optimality with a fixed target identification time T , we should select a waypoint

$$\mathbf{s}_{\hat{\mu}} \in \arg \min_{\mathbf{x} \in \Omega_T(\mathbf{x}_0)} \xi(\mathbf{x}).$$

However, for the sake of computational efficiency, we could instead use a domain decomposition $\Omega = \bigcup_{i=1}^m \Upsilon_i$, with m non-overlapping subdomains Υ_i and a single representative target $\hat{\mathbf{x}}_i \in \Upsilon_i$ selected from each of them. The approach from section 3.1 can be now viewed as finding an approximately optimal waypoint for a *coarsened target set* $\hat{X}$ with

$$q(\mathbf{x}) = \sum_{i=1}^m \int_{\Upsilon_i} u(\mathbf{x}; \hat{\mathbf{x}}_i) d\hat{\mu}(\mathbf{y}) = \sum_{i=1}^m u(\mathbf{x}; \hat{\mathbf{x}}_i) \hat{p}_i, \quad \hat{p}_i = \int_{\Upsilon_i} d\hat{\mu}(\mathbf{y}). \quad (4.9)$$

Once we reach the waypoint $\mathbf{s} \in \arg \min_{\Omega_T} q(\mathbf{x})$ and the true target $\mathbf{y} \in \Omega$ is revealed, we will proceed there directly. This strategy yields the average time to target $\xi(\mathbf{s})$, and the natural question is how much worse it is than the optimal $\xi(\mathbf{s}_{\hat{\mu}})$.

Suppose our speed of motion is bounded below by $f_l > 0$ and all subdomains are small enough so that $\forall \mathbf{y} \in \Upsilon_i$ there is a path of length at most h connecting $\mathbf{y}$ to $\hat{\mathbf{x}}_i$ without leaving Υ_i . Then the triangle inequality yields

$$\begin{aligned} |\xi(\mathbf{x}) - q(\mathbf{x})| &\leq \sum_{i=1}^m \int_{\Upsilon_i} |u(\mathbf{x}; \mathbf{y}) - u(\mathbf{x}; \hat{\mathbf{x}}_i)| d\hat{\mu}(\mathbf{y}) \leq \sum_{i=1}^m \int_{\Upsilon_i} u(\mathbf{y}; \hat{\mathbf{x}}_i) d\hat{\mu}(\mathbf{y}) \\ &\leq \sum_{i=1}^m \int_{\Upsilon_i} (h/f_l) d\hat{\mu}(\mathbf{y}) = h/f_l. \end{aligned} \quad (4.10)$$

Recalling the q -optimality of $\mathbf{s}$ and using (4.10) twice (with $\mathbf{x} = \mathbf{s}_{\hat{\mu}}$ and then with $\mathbf{x} = \mathbf{s}$), we see that

$$\xi(\mathbf{s}_{\hat{\mu}}) \geq q(\mathbf{s}_{\hat{\mu}}) - h/f_l \geq q(\mathbf{s}) - h/f_l \geq \xi(\mathbf{s}) - 2h/f_l, \quad (4.11)$$

i.e., the suboptimality of $\mathbf{s}$ is bound by $2h/f_l$.

5. Conclusions. Our focus has been on the challenge of optimal and robust path-planning under initial uncertainty. We have considered three different models for the certainty time T and derived efficient numerical methods for optimizing the average case performance in each case. For the fixed T and exponentially distributed T we have also introduced a number of robust path-planning techniques, encoding different approaches to balancing the average case performance against the particularly bad/unlucky scenarios. Under the assumption that the running cost K and the certainty time T are constant, the problem reduces to a choice among T -reachable waypoints, which allowed us to evaluate several robust planning techniques that are often prohibitively expensive in the general setting. E.g., we have introduced a polynomial time algorithm for optimizing the expectation under probabilistic constraints on bad outcomes. Risk-sensitive and distributionally-robust optimization are often viewed as closely approximating the optimal tradeoffs between the average and worst-case performance. But we have shown that in our setting, the resulting waypoints can be far from Pareto optimal. Extending these robust planning algorithms to the case of general random T is a challenge that we hope to address in the future.

While most of our presentation is focused on target uncertainty, a similar approach can be employed with initial uncertainty on cost or dynamics (on yet to be explored part of the domain); see Appendix A for a representative example. For the sake of

simplicity, our examples were always isotropic and relied on Fast Marching Method with first-order accurate upwind finite difference schemes. But similar techniques would be easy to implement with Fast Marching-style methods developed for higher-order accurate discretizations (e.g., [2, 45, 46]) and more general anisotropic problems (e.g., [3, 19, 34, 35, 47]).

Beyond path planning, we believe that our approach is also promising for treating structured uncertainty in general control problems and Markov Decision Processes [25]. For instance, techniques similar to those described in our section 3.2 have recently been applied to determining optimal car braking policy under traffic signal duration uncertainty [26].

Another useful future direction is to adapt our techniques to the setting of Mean Field Games (MFG), where a large number of interacting agents plan their actions selfishly, and the initial uncertainty can be viewed as a particularly simple model of (instantaneous) “common noise” affecting all agents. E.g., evacuating a building under uncertainty on which exit will be open at a later/known time T is an example of this type considered in [1, 11]. If the agents are cooperative and centrally controlled, this can be handled in the framework of Mean Field Control [31] or, for a smaller number of agents, by solving a high-dimensional HJB equation (e.g., using neural networks [33, 36] or tensor decomposition techniques [21]). As an alternative to the mean field approximations, sequential path-planning [15, 42] can be also used to alleviate the curse of dimensionality if the agents are cooperating and can be assigned priority. We believe that our models of initial uncertainty would be beneficial in each of these settings.

Throughout this paper, we have assumed that the uncertainty disappears instantaneously. It will be useful and challenging to extend our methods to general problems with delayed information acquisition. Example in Appendix B is actually a small step in this direction, with target-uncertainty reduced gradually at several distinct times T_i . A different generalization of this type will be to consider problems where the transition to certainty is still instantaneous, but our choice of pre-certainty control affects the probability distribution of T . This situation arises naturally in many “stochastic shortest path with recourse” problems [40].

Acknowledgements: The authors would like to thank Antonio Farah, Qianli Song, and Tristan Reynoso for their help in initial testing of some of these methods in Summer REU program at Cornell University. The authors are also grateful to Dmitriy Drusvyatskiy for sharing his insight on the number of waypoints needed in non-deterministic optimal policies when optimizing with probabilistic constraints.

Author Contributions All three authors contributed to the development of the paper and the underlying research.

Funding This work was supported in part by the National Science Foundation (grants DMS-1738010 and DMS-1645643), Air Force Office of Scientific Research (FA9550-22-1-0528), and the last author’s Simons Foundation Fellowship.

Data availability No data associated in the manuscript.

Declarations:

Conflict of interest The authors declare no competing interests.

Ethical Approval Not applicable.

Appendices

Appendix A. Uncertainty in running cost: the storm fronts example.

Returning to the setting of a fixed certainty time T in section 3.1, we now show how our approach can similarly handle initial uncertainty arising due to anticipated changes in a global environment even if the target is perfectly known in advance.

Consider a challenge of risk-hedged air traffic routing under uncertain weather conditions. Given a forecast of probable storm front trajectories, one option is to devise a good flight path and commit to it in advance all the way to the target. This is the key idea of the approach developed in [43], but one can do much better by taking into account the future re-routing once a more precise forecast becomes available. What follows is an illustration of how this can be done in our framework under a simplifying assumption that the true position of the storm front will be revealed at a known time T .

First, consider the deterministic case with an a priori known location of an elliptic storm region. Our running cost K is assumed to be identically one on most of the domain, but is higher inside the storm region. In particular, if the storm is centered at $\tilde{\mathbf{x}}$ and the canonical equation for the storm region boundary is $(\mathbf{x} - \tilde{\mathbf{x}})^T A (\mathbf{x} - \tilde{\mathbf{x}}) = 1$, we will assume that inside that region $K(\mathbf{x}) = 1 + \alpha(1 - (\mathbf{x} - \tilde{\mathbf{x}})^T A (\mathbf{x} - \tilde{\mathbf{x}}))^\gamma$ with $\alpha = 2$ and $\gamma = 2.5$. (Note that this yields $K = 1$ on the storm boundary, ensuring the continuity.) For any fixed/known $\tilde{\mathbf{x}}$ and A , finding an optimal path to the target $\hat{\mathbf{x}}$ is simply a matter of solving an Eikonal equation and using gradient descent on the value function. In Figure 11 we illustrate this for three different storm locations (encoded by $(\tilde{\mathbf{x}}, A)$). Unsurprisingly, optimal trajectories deviate from the quickest (straight line) path to the target to decrease the amount of time spent inside the storm.

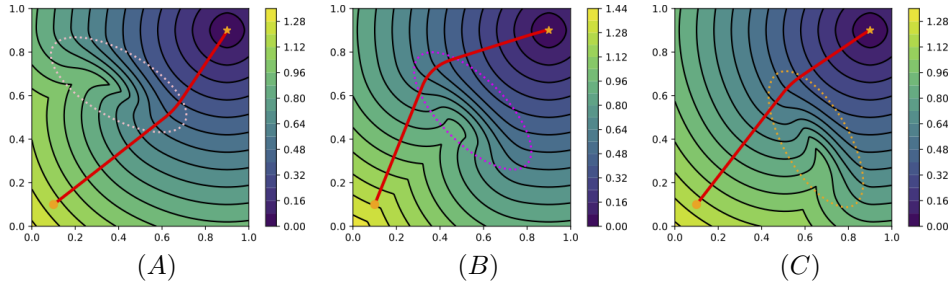


FIG. 11. A deterministic case of different storms with elliptic shapes. $f = 1, K = 1$ outside storm region, while $f = 1, K > 1$ inside. The source $\mathbf{x}_0$ is at $(0.1, 0.1)$ while the target $\hat{\mathbf{x}}$ is at $(0.9, 0.9)$. Each storm location results in a different running cost $K_i(\mathbf{x})$ and value function $u_i(\mathbf{x})$. The red lines are cost-minimal trajectories and in the background are level sets of u_1, u_2 and u_3 .

Suppose that initially we only have a probability distribution $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$ over these three storm regions (enumerated in the same order as in Figure 11) and the true position of the storm will be revealed at a later time T . We will further assume that T is small enough so that all three possible storm regions have no overlap with Ω_T ; i.e., the plane can never reach any storm region before the “storm-revealing time”. The optimal waypoint is found in the same way as in section 3.1. I.e., the plane should

follow the time-optimal path to

$$\mathbf{s} \in \arg \min_{\mathbf{x} \in \Omega_T(\mathbf{x}_0)} q(\mathbf{x}), \quad \text{where } q(\mathbf{x}) = \mathbb{E}_{\hat{\mathbf{x}}, A} [u(\mathbf{x})] = \sum_{i=1,2,3} \hat{p}_i u_i(\mathbf{x}).$$

In general, the time complexity of obtaining $\mathbf{s}$ is $O(m_s M \log M)$ with m_s being the total number of potential storm regions and M the number of gridpoints.

As we can see from Figure 12, the optimal waypoint is heavily dependent on $\hat{\mathbf{p}}$. Of course, the above setting is highly simplified since in reality the weather forecasts

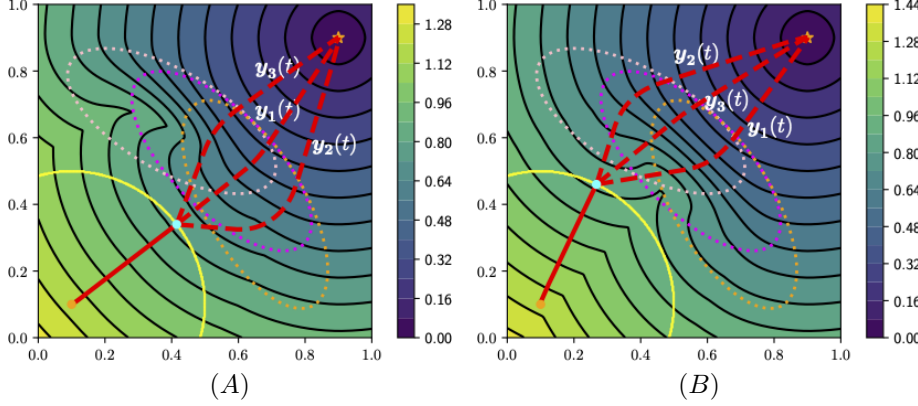


FIG. 12. Uncertain elliptic storms with different $\hat{\mathbf{p}}$'s. Level sets of $q(\mathbf{x})$ are shown in the background. The cyan dot is $\mathbf{s}$, attaining minimal q in Ω_T , with $T = 0.4$. The dashed red lines labeled $\mathbf{y}_i(t)$ ($i = 1, 2, 3$) are the optimal trajectories from $\mathbf{s}$ to the target – to be used once the actual storm location is discovered. (A): $\hat{\mathbf{p}} = (0.8, 0.1, 0.1)$. (B): $\hat{\mathbf{p}} = (0.1, 0.8, 0.1)$.

would be updated numerous times or even continuously. But the amount of uncertainty would normally still decrease with time and the main ideas of sections 3.1-3.3 would be still applicable.

Appendix B. Temporal changes in target probabilities.

The following is an “emergency rescue” example where p_j 's are not independent of $\hat{p}_i$'s. The rescue vehicle starts at $\mathbf{x}_0$ and moves with isotropic speed f in the domain with a rectangular obstacle, exactly as shown in Figure 1. The goal is to minimize the expected time until we rescue a subject known to be located at one of the 4 possible sites $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_4$ with the a priori likelihood reflected by $\hat{p}_1, \dots, \hat{p}_4$. But unlike the previous examples, here we assume the availability of 3 lightweight aerial drones also launched from $\mathbf{x}_0$ and flying to the closest 3 sites ($\hat{\mathbf{x}}_3, \hat{\mathbf{x}}_4$, and $\hat{\mathbf{x}}_2$) along the straight lines (unconstrained by the ground rectangular obstacle). For simplicity, we will assume that drones move with speed $f_d = 5/3$ and notify the main vehicle whether the rescue subject is found as soon as they reach their respective destinations. For the locations specified in Figure 1, these drones' arrival times will be $T_1 = |\hat{\mathbf{x}}_3 - \mathbf{x}_0|/f_d < T_2 = |\hat{\mathbf{x}}_4 - \mathbf{x}_0|/f_d < T_3 = |\hat{\mathbf{x}}_2 - \mathbf{x}_0|/f_d$. If the first drone discovers that subject at $\hat{\mathbf{x}}_3$, the rescue vehicle follows the quickest path to it. Otherwise, the probabilities of other sites is adjusted accordingly: $\hat{p}_j^{new} = \hat{p}_j/(1 - \hat{p}_3)$ for all $j \neq 3$ and $\hat{p}_3^{new} = 0$. The probabilities are similarly adjusted at T_2 if $\hat{\mathbf{x}} \neq \hat{\mathbf{x}}_4$. If the subject has not been located earlier, at the time T_3 we will discover whether he is at the last drone-visited site $\hat{\mathbf{x}}_2$ or at the remaining site $\hat{\mathbf{x}}_1$. The question is how to optimally plan the rescue vehicle's path until the discovery time $T \in \{T_1, T_2, T_3\}$.

This can be handled by solving the same sequence of time-dependent PDEs (2.4). The value functions $v_i(\mathbf{x}, t)$ will satisfy (2.4) on $[T_{i-1}, T_i], i = 1, 2, 3$ with $T_0 = 0$ and the terminal conditions

$$\begin{aligned} v_3(\mathbf{x}, T_3) &= \frac{\hat{p}_2}{\hat{p}_2 + \hat{p}_1} u_2(\mathbf{x}) + \frac{\hat{p}_1}{\hat{p}_2 + \hat{p}_1} u_1(\mathbf{x}), \\ v_2(\mathbf{x}, T_2) &= \frac{\hat{p}_4}{\hat{p}_1 + \hat{p}_2 + \hat{p}_4} u_4(\mathbf{x}) + \frac{\hat{p}_1 + \hat{p}_2}{\hat{p}_1 + \hat{p}_2 + \hat{p}_4} v_3(\mathbf{x}, T_2), \\ v_1(\mathbf{x}, T_1) &= \hat{p}_3 u_3(\mathbf{x}) + (\hat{p}_1 + \hat{p}_2 + \hat{p}_4) v_2(\mathbf{x}, T_1). \end{aligned}$$

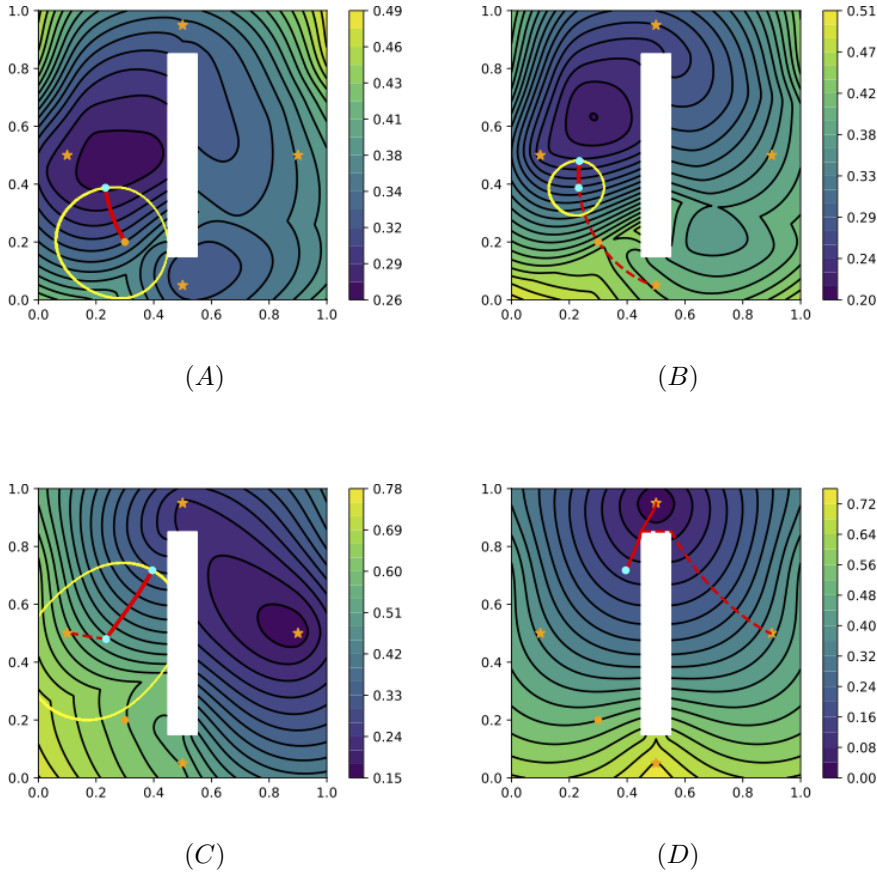


FIG. 13. The drone-assisted emergency rescue example with the same initial $\hat{p}_i$'s as in Figure 2. (A): The optimal waypoint $\mathbf{s}_1$ for the time T_1 and the $\mathbf{x}_0 \rightarrow \mathbf{s}_1$ trajectory, superimposed on the level sets of $v_1(\mathbf{x}, T_1)$. The yellow curve shows the boundary of the reachable region $\Omega_{T_1}(\mathbf{x}_0)$. (B): If the first drone discovers the subject, the vehicle follows a dashed red $\mathbf{s}_1 \rightarrow \hat{\mathbf{x}}_3$ trajectory. Otherwise, it follows a solid red trajectory to the T_2 -optimal waypoint $\mathbf{s}_2$, shown on top of the level sets of $v_2(\mathbf{x}, T_2)$. The yellow curve is the boundary of $\Omega_{T_2-T_1}(\mathbf{s}_1)$. (C): If the subject is not discovered by the first drone but the second, the vehicle follows a dashed red $\mathbf{s}_2 \rightarrow \hat{\mathbf{x}}_4$ trajectory. Otherwise, it follows a solid red trajectory to the T_3 -optimal waypoint $\mathbf{s}_3$, shown on top of the level sets of $v_3(\mathbf{x}, T_3)$. The yellow curve is the boundary of $\Omega_{T_3-T_2}(\mathbf{s}_2)$. (D): By the time T_3 , the drone either finds the subject at $\hat{\mathbf{x}}_2$ and the vehicle follows the dashed red trajectory, or we conclude that the subject is at $\hat{\mathbf{x}}_1$ and the vehicle follows the solid red path shown over the level sets of $u_1(\mathbf{x})$.

Appendix C. DR optimization missing a large part of $(q, \bar{q})$ Pareto front.

Figure 7B showed that risk-sensitive waypoints can remarkably deviate from $q - \bar{q}$ Pareto front. The distributionally-robust waypoints are much closer to Pareto optimality in that same figure. However, the following example shows that often there is no DR-optimal $\bar{\mathbf{s}}_\delta$ satisfying $\bar{q}(\bar{\mathbf{s}}_\delta) = \bar{q}(\mathbf{s}_c)$ for many $\mathbf{s}_c$ values. Thus, the DR results cannot approximate the corresponding parts of the Pareto front, which in this case is highly non-convex and even discontinuous. The DR optimal waypoints cluster into three groups, with large gaps between clusters.

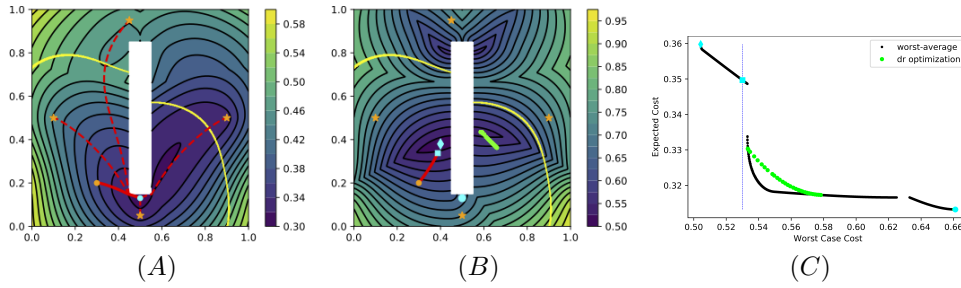


FIG. 14. (A): Expectation-optimal waypoint and post-certainty paths to each target. The corresponding target probabilities (clockwise, starting at the top) are $\hat{p} = (0.17, 0.35, 0.3, 0.18)$. The symmetry of target locations is broken as the top target ($\hat{\mathbf{x}}_1$) is moved slightly left to $(0.45, 0.95)$. The speed function is also slightly different: $f = 1.4 - 0.6\cos(2\pi x)\sin(2\pi y)$. Level sets of $q(\mathbf{x})$ are shown in the background. (B): Three different optimal waypoints ($\mathbf{s}, \bar{\mathbf{s}}, \mathbf{s}_c$) (cyan dot, diamond and square) corresponding to average, worst-case optimality and minimizing average with a hard constraint $C = 0.53$. A hundred distributionally-robust waypoints $\bar{\mathbf{s}}_\delta$'s are also shown as green dots with equally spaced δ s between 0 and 1. Level sets of $\bar{q}(\mathbf{x})$ are shown in the background. (C): $q - \bar{q}$ Pareto front and all the waypoints from (B).

REFERENCES

- [1] Y. ACHDOU AND J.-M. LASRY, *Mean field games for modeling crowd motion*, in Contributions to partial differential equations and applications, Springer, 2019, pp. 17–42.
- [2] S. AHMED, S. BAK, J. McLAUGHLIN, AND D. RENZI, *A third order accurate fast marching method for the eikonal equation in two dimensions*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2402–2420.
- [3] K. ALTON AND I. M. MITCHELL, *An ordered upwind method with precomputed stencil and monotone node acceptance for solving static convex Hamilton-Jacobi equations*, Journal of Scientific Computing, 51 (2012), pp. 313–348.
- [4] A. M. ANDREW, *Another efficient algorithm for convex hulls in two dimensions*, Information Processing Letters, 9 (1979), pp. 216–219.
- [5] J. ANDREWS AND A. VLADIMIRSKY, *Deterministic control of randomly-terminated processes*, Interfaces and Free Boundaries, 16 (2014), pp. 1–40.
- [6] S. BAK, J. McLAUGHLIN, AND D. RENZI, *Some improvements for the fast sweeping method*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2853–2874.
- [7] M. BARDI AND I. CAPUZZO-DOLCETTA, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Birkhauser, 1997.
- [8] G. BARLES AND P. E. SOUGANIDIS, *Convergence of approximation schemes for fully nonlinear second order equations*, in Asymptotic analysis, vol. 4, 1991, pp. 271–283.
- [9] T. BAŞAR AND P. BERNHARD, *H^∞ -optimal control and related minimax design problems*, Birkhauser, 1995.

- [10] M. BOUÉ AND P. DUPUIS, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, SIAM Journal on Numerical Analysis, 36 (1999), pp. 667–695.
- [11] R. CARMONA AND M. LAURIÈRE, *Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II—the finite horizon case*, The Annals of Applied Probability, 32 (2022), pp. 4065–4105.
- [12] A. CHACON AND A. VLADIMIRSKY, *Fast two-scale methods for eikonal equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. A547–A578.
- [13] A. CHACON AND A. VLADIMIRSKY, *A parallel two-scale method for eikonal equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. A156–A180.
- [14] T. M. CHAN, *Optimal output-sensitive convex hull algorithms in two and three dimensions*, Discrete & Computational Geometry, 16 (1996), pp. 361–368.
- [15] M. CHEN, J. F. FISAC, S. SASTRY, AND C. J. TOMLIN, *Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality*, in 2015 European Control Conference (ECC), IEEE, 2015, pp. 3304–3309.
- [16] Z. CLAWSON, A. CHACON, AND A. VLADIMIRSKY, *Causal domain restriction for eikonal equations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2478–A2505.
- [17] M. G. CRANDALL AND P.-L. LIONS, *Viscosity solutions of hamilton-jacobi equations*, Transactions of the American mathematical society, 277 (1983), pp. 1–42.
- [18] E. CRISTIANI AND M. FALCONE, *Fast semi-Lagrangian schemes for the eikonal equation and applications*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1979–2011.
- [19] D. DAHIYA AND M. CAMERON, *Ordered line integral methods for computing the quasi-potential*, Journal of Scientific Computing, 75 (2018), pp. 1351–1384.
- [20] E. DELAGE AND Y. YE, *Distributionally robust optimization under moment uncertainty with application to data-driven problems*, Operations research, 58 (2010), pp. 595–612.
- [21] S. DOLGOV, D. KALISE, AND K. K. KUNISCH, *Tensor decomposition methods for high-dimensional Hamilton-Jacobi-Bellman equations*, SIAM Journal on Scientific Computing, 43 (2021), pp. A1625–A1650.
- [22] S. ERMON, C. GOMES, B. SELMAN, AND A. VLADIMIRSKY, *Probabilistic planning with non-linear utility functions and worst-case guarantees*, in Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 965–972.
- [23] M. FALCONE, *The minimum time problem and its applications to front propagation*, Motion by mean curvature and related topics (Trento, 1992), de Gruyter, Berlin, (1994), pp. 70–88.
- [24] M. FALCONE AND R. FERRETTI, *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*, SIAM, 2013.
- [25] W. H. FLEMING AND H. M. SONER, *Controlled Markov processes and viscosity solutions*, vol. 25, Springer Science & Business Media, 2006.
- [26] M. E. GASPARD AND A. VLADIMIRSKY, *Optimal driving under traffic signal uncertainty*, IFAC-PapersOnLine, 55 (2022), pp. 25–31.
- [27] R. L. GRAHAM, *An efficient algorithm for determining the convex hull of a finite planar set*, Info. Pro. Lett., 1 (1972), pp. 132–133.
- [28] R. A. HOWARD AND J. E. MATHESON, *Risk-Sensitive Markov Decision Processes*, Management Science, 18 (1972), pp. 356–369.
- [29] W.-K. JEONG, P. T. FLETCHER, R. TAO, AND R. T. WHITAKER, *Interactive visualization of volumetric white matter connectivity in DT-MRI using a parallel-hardware Hamilton-Jacobi solver*, Visualization and Computer Graphics, IEEE Transactions on, 13 (2007), pp. 1480–1487.
- [30] L. P. KAEHLING, M. L. LITTMAN, AND A. R. CASSANDRA, *Planning and acting in partially observable stochastic domains*, Artificial intelligence, 101 (1998), pp. 99–134.
- [31] M. LAURIÈRE, *Numerical methods for mean field games and mean field type control*, Proceedings of Symposia in Applied Mathematics, 78 (2021).
- [32] F. LI, C.-W. SHU, Y.-T. ZHANG, AND H. ZHAO, *A second order discontinuous Galerkin fast sweeping method for eikonal equations*, Journal of Computational Physics, 227 (2008), pp. 8191–8208.
- [33] T. MENG, Z. ZHANG, J. DARBON, AND G. KARNIADAKIS, *Sympocnet: Solving optimal control problems with applications to high-dimensional multiagent path planning problems*, SIAM Journal on Scientific Computing, 44 (2022), pp. B1341–B1368.
- [34] J.-M. MIREBEAU, *Efficient fast marching with Finsler metrics*, Numerische Mathematik, 126 (2014), pp. 515–557.
- [35] ———, *Riemannian fast-marching on cartesian grids, using Voronoi’s first reduction of quadratic forms*, SIAM Journal on Numerical Analysis, 57 (2019), pp. 2608–2655.

- [36] T. NAKAMURA-ZIMMERER, Q. GONG, AND W. KANG, *Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations*, SIAM Journal on Scientific Computing, 43 (2021), pp. A1221–A1247.
- [37] A. Z. PALMER AND A. VLADIMIRSKY, *Optimal stopping with a probabilistic constraint*, Journal of Optimization Theory and Applications, 175 (2017), pp. 795–817.
- [38] C. H. PAPADIMITRIOU AND M. YANNAKAKIS, *Shortest paths without a map*, Theoretical Computer Science, 84 (1991), pp. 127–150.
- [39] L. PFEIFFER, *Two approaches to stochastic optimal control problems with a final-time expectation constraint*, Applied Mathematics & Optimization, 77 (2018), pp. 377–404.
- [40] G. H. POLYCHRONOPOULOS AND J. N. TSITSIKLIS, *Stochastic shortest path problems with recourse*, Networks: An International Journal, 27 (1996), pp. 133–143.
- [41] S. F. POTTER AND M. K. CAMERON, *Ordered line integral methods for solving the eikonal equation*, Journal of Scientific Computing, 81 (2019), pp. 2010–2050.
- [42] D. R. ROBINSON, R. T. MAR, K. ESTABRIDIS, AND G. HEWER, *An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments*, IEEE Robotics and Automation Letters, 3 (2018), pp. 1215–1222.
- [43] A. V. SADOVSKY AND K. D. BILIMORIA, *Risk-hedged approach for re-routing air traffic under weather uncertainty*, in 16th AIAA Aviation Technology, Integration, and Operations Conference, 2016, p. 3601.
- [44] J. A. SETHIAN, *A fast marching level set method for monotonically advancing fronts*, Proceedings of the National Academy of Sciences, 93 (1996), pp. 1591–1595.
- [45] J. A. SETHIAN, *Fast marching methods*, SIAM review, 41 (1999), pp. 199–235.
- [46] J. A. SETHIAN AND A. VLADIMIRSKY, *Fast methods for the eikonal and related Hamilton–Jacobi equations on unstructured meshes*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 5699–5703.
- [47] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton–Jacobi equations*, Proceedings of the National Academy of Sciences, 98 (2001), pp. 11069–11074.
- [48] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYNSKI, *Lectures on stochastic programming: modeling and theory*, SIAM, 2021.
- [49] Y.-H. R. TSAI, L.-T. CHENG, S. OSHER, AND H.-K. ZHAO, *Fast sweeping algorithms for a class of Hamilton–Jacobi equations*, SIAM journal on numerical analysis, 41 (2003), pp. 673–694.
- [50] J. N. TSITSIKLIS, *Efficient algorithms for globally optimal trajectories*, IEEE Transactions on Automatic Control, 40 (1995), pp. 1528–1538.
- [51] C. VILLANI, *Optimal transport: old and new*, vol. 338, Springer Science & Business Media, 2008.
- [52] A. VLADIMIRSKY AND C. ZHENG, *A fast implicit method for time-dependent Hamilton–Jacobi PDEs*, Unpublished Manuscript, (2015). Available at <http://arxiv.org/abs/1512.08734>.
- [53] D. WHITE, *Dynamic programming and probabilistic constraints*, Operations Research, 22 (1974), pp. 654–664.
- [54] D. S. YERSHOV AND S. M. LAVALLE, *Simplicial label correcting algorithms for continuous stochastic shortest path problems*, in Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, 2013, pp. 5062–5067.
- [55] P. ZARCHAN AND H. MUSOFF, *Fundamentals of Kalman filtering: a practical approach*, American Institute of Aeronautics and Astronautics, Inc., 2013.
- [56] Y.-T. ZHANG, S. CHEN, F. LI, H. ZHAO, AND C.-W. SHU, *Uniformly accurate discontinuous Galerkin fast sweeping methods for eikonal equations*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1873–1896.
- [57] H. ZHAO, *A fast sweeping method for eikonal equations*, Mathematics of computation, 74 (2005), pp. 603–627.