Interfaces and Free Boundaries **16** (2014), 1–40
DOI 10.4171/IFB/312

Deterministic control of randomly-terminated processes

JUNE ANDREWS
Department of Mathematics and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853
E-mail: ja378@cornell.edu

ALEXANDER VLADIMIRSKY
Department of Mathematics and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853
E-mail: vlad@math.cornell.edu

[Received 9 August 2010 and in revised form 1 October 2013]

We consider both discrete and continuous "uncertain horizon" deterministic control processes, for which the termination time is a random variable. We examine the dynamic programming equations for the value function of such processes, explore their connections to infinite-horizon and optimal-stopping problems, and derive sufficient conditions for the applicability of non-iterative (label-setting) methods. In the continuous case, the resulting PDE has a free boundary, on which all characteristic curves originate. The causal properties of "uncertain horizon" problems can be exploited to design efficient numerical algorithms: we derive causal semi-Lagrangian and Eulerian discretizations for the isotropic randomly-terminated problems, and use them to build a modified version of the Fast Marching Method. We illustrate our approach using numerical examples from optimal idle-time processing and expected response-time minimization.

2010 Mathematics Subject Classification: Primary 49L20, 49L25, 35R35, 65N22, 90C39, 05C85.

Keywords: Random termination; optimal stopping; free boundary; expected response time; label-setting; Hamilton–Jacobi; fast marching.

1. Introduction

Deterministic and stochastic optimal control problems arise naturally in most engineering disciplines and in fields as diverse as economics, geometric optics, robotic navigation, and computational geometry. Dynamic programming [6] is the key technique for solving such problems by formulating equations satisfied by the corresponding *value function*. Fast algorithms for recovering the value function are thus of interest to many practitioners, but the efficiency challenges in building such methods can be quite different depending on the exact type of optimal control problems. One basic taxonomy is based on the time-horizon of optimization in the problem. Is the process stopped at some explicitly specified terminal time (*finite-horizon problems*) or continues forever (*infinite-horizon problems*)? If the terminal time is not specified, but the process eventually stops, does this happen upon entering some pre-specified set (*exit-time problems*), or at any point when the controller chooses to do so (*optimal-stopping problems*)? For finite-horizon problems, the time-dependence of the value function results in availability of simple and non-iterative (time-marching) numerical methods. In contrast, all other scenarios typically yield large systems of coupled non-linear equations, and finding suitable non-iterative methods for them can be

challenging. The above problem types are classical and the resulting equations are well understood (for readers' convenience we review them in Section 2.1 and Section 3.1 for discrete and continuous problems respectively). In this paper we are interested in a less studied class of *randomly-terminated* or *"uncertain-horizon"* problems, where the termination is assumed to be the result of a Poisson random process. As we show here, such problems inherit some properties of both finite-horizon and optimal-stopping cases. Efficient numerical methods for them are not as simple as time-marching, but we show that generalizations of non-iterative algorithms previously derived for exit-time problems are applicable.

Another natural classification approach is to draw a distinction between deterministic and stochastic control processes. Not surprisingly, the former are usually simpler and can be treated by more efficient numerical methods. This observation holds for dynamic programming equations both in discrete and continuous settings.

In the discrete cases, the deterministic Shortest Path (SP) problems on graphs are often solved using fast-iterative (*label-correcting*) and non-iterative (*label-setting*) methods, including the well-known Dijkstra's method [18] and its parallelizable Dial's variant [17]. We refer the readers to [1, 7, 8] for a detailed discussion of these efficient methods on graphs. The more general Stochastic Shortest Path (SSP) problems typically require iterative methods; a good discussion of SSPs can be found in [8]. Except for a few structurally simple examples, the exact (a priori verifiable) conditions under which non-iterative algorithms are applicable to SSPs are still not known. The second author has previously derived such sufficient conditions for a narrower class of *Multimode* SSPs [38], but the problems considered in the current paper lie outside of this class.

In the case of continuous state spaces, the deterministic problems lead to first-order non-linear Hamilton–Jacobi-Bellman partial differential equations; a comprehensive description can be found in [4]. For exit-time problems, the resulting PDE is typically static since the starting time does not affect the optimality of any given control. Efficient numerical methods for such static PDEs formed an active area of research in the last fifteen years. For example, Dijkstra-like non-iterative numerical methods for isotropic problems were independently introduced by Tsitsiklis [35, 36] and Sethian [29, 30]. Later generalizations lead to Ordered Upwind Methods [2, 31, 32] applicable to anisotropic problems. All these methods rely on a careful use of Lagrangian information to efficiently solve the Eulerian discretized equations, yielding *space-marching* algorithms, in which the numerical solution is marched on the grid inward from the boundary of the domain.

On the other hand, most commonly considered stochastic control problems on continuous state space assume that the controlled dynamics is affected by some time-continuous stochastic process (usually, by a scaled Brownian motion). This yields second-order semi-linear PDEs and, in the static case, non-iterative numerical methods are not applicable.

The stochasticity considered in this paper is of a different kind – the process dynamics and the running cost are assumed to be fully deterministic up to the time of termination, but that terminal time is itself a random variable. Such uncertain-horizon problems have applications in production/maintenance planning [12], economic growth and global climate change modeling [21], and multi-generational games [20]. We first show that the optimal control of randomly-terminated problem can be always re-stated as a *time-discounted infinite-horizon* problem on the same state space. (In the continuous case, this reformulation is well-known; e.g., [12]). We then prove that efficient non-iterative methods are applicable for a wide subclass of such problems. Our exact technical assumptions are specified in Sections 2.2 and 3.2, but this class generally includes all problems where maintaining the "status quo" is always possible and incurs a smaller immediate cost

than any attempt to change the system state. That assumption allows re-stating this as a deterministic *optimal stopping* problem. We define a *motionless set* M of all states, starting from which the optimal behavior is to remain in place (awaiting the termination). Unfortunately, the set M is not a priori known, which presents a challenge in using label-setting methods, since $B = \partial M$ forms a *free boundary* of this problem, from which the numerical solution should be marched into the rest of the domain.

Our exposition proceeds on two parallel tracks: the discrete and continuous settings are handled in Sections 2 and 3 respectively. Some of the technical results are included in the Appendixes. Throughout the paper the letters v and V are reserved for the value functions of randomly-terminated problems, while letters u and U are used for all other optimal control examples. We will also use the expressions "deterministic randomly-terminated processes" and "uncertain-horizon processes" interchangeably. In Section 2.3 and Section 3.2 we prove the properties of value functions and motionless sets, including their asymptotic behavior depending on the probability distribution for the terminal time. We review the label-setting for the fully deterministic problems on graphs in Section 2.4.1 and then prove the applicability of a Dijkstra-like method to uncertain-horizon problems in Section 2.4.2. The corresponding generalization of the Fast Marching Method is developed in Section 3.3. Numerical experiments illustrating the properties of the latter are included in Section 4; several possible generalizations are discussed in Section 5.

2. Uncertain-horizon problems on a finite state space

Optimal control problems on a finite state space are usually presented as problems of finding optimal paths in a finite directed graph. The notion of "optimality" here depends not only on edge costs, but also on termination conditions for the process.

We begin by briefly reviewing several common types of such problems to put uncertain-horizon optimization in context.

2.1 *Common optimal control problems on graphs*

We will assume that all paths are considered on a directed graph with M nodes $X = \{x_1, \ldots, x_M\}$. Let $N(x_i) \subset X$ denote the set of nodes to which a direct transition from x_i is possible, and assume that $\kappa << M$ is an upper bound on the outdegrees of all nodes (i.e., $|N(x_i)| \leqslant \kappa$ for $\forall x_i \in X$).

Suppose $y_k \in X$ is the position after k steps. A sequence $\boldsymbol{y} = (y_0, \ldots, y_k, \ldots)$ is a *path* on this graph if $y_{k+1} \in N(y_k)$ for each $k \geqslant 0$. Each transition from x_i to x_j incurs a cost $K_{ij} = K(x_i, x_j)$ (assumed to be $+\infty$ if $x_j \notin N(x_i)$). If the termination occurs at x_j, this results in an additional terminal-penalty $q_j = q(x_j)$.

- *Finite-horizon problems:*
  The process runs for exactly t steps (where t is a constant specified as a part of problem description). The total cost of a path $(y_0, \ldots, y_t)$ is

$$Cost(y_0, \ldots, y_t) = \sum_{k=0}^{t-1} K(y_k, y_{k+1}) \; + \; q(y_t). \tag{2.1}$$

  The central idea of *dynamic programming* is to define the *value function* $U(x_i, k) = U_i^k$ as the minimum cost to spend starting from x_i with $(t - k)$ steps remaining. Bellman's optimality

principle [6] allows to derive equations for U_i^k using the values in $\{U_j^{k+1} \mid x_j \in N(x_i)\}$ only. In particular,

$$\begin{cases} U_i^k = \min\limits_{x_j \in N(x_i)} \left\{ K_{ij} + U_j^{k+1} \right\}, & \text{for } k = 0, \dots, (t-1) \text{ and } i = 1, \dots, M; \\ U_i^t = q_i & \text{for } i = 1, \dots, M. \end{cases} \tag{2.2}$$

It is easy to generalize this to use time-dependent $K(x_i, x_j, k)$ and the system (2.2) can be efficiently solved in a single backward-sweep (from $k = t$ to $k = 0$), regardless of the sign of transition cost K.

- *Exit-time a.k.a. shortest path (SP) problems:*
  Here the termination occurs immediately upon reaching the exit-set $Q \subset X$; i.e., $t_{\boldsymbol{y}} = min\{k | y_k \in Q\}$. As a result, $U_i = U(x_i)$ (the min-cost-to-exit-starting-from-x_i) no longer depends on the number of steps already used to reach x_i. The value function is defined only in the absence of "negative cost cycles" and, by optimality principle, must satisfy

$$\begin{cases} U_i = \min\limits_{x_j \in N(x_i)} \left\{ K_{ij} + U_j \right\}, & \text{for } \forall x_i \in X \backslash Q; \\ U_i = q_i, & \text{for } \forall x_i \in Q. \end{cases} \tag{2.3}$$

  This is a system of M coupled non-linear equations, and, unless the graph is acyclic, this system might be expensive to solve iteratively. If all K_{ij}'s are non-negative, label-setting methods (further discussed in section 2.4.1) provide an efficient way of solving it.
- *Infinite-horizon time-discounted problems:*
  Here the process never terminates – the paths are infinite, but the cost of each subsequent step is discounted, i.e.,

$$Cost(y_0, \dots) = \sum_{k=0}^{+\infty} \alpha^k \, K(y_k, y_{k+1}), \tag{2.4}$$

  where $\alpha \in (0, 1)$ is the discounting factor. The value function $U_i = U(x_i)$ is well-defined for all nodes regardless of sign of K_{ij}'s and satisfies

$$U_i = \min_{x_j \in N(x_i)} \left\{ K_{ij} + \alpha U_j \right\} \qquad \text{for } \forall x_i \in X. \tag{2.5}$$

  Unlike the SP, in this case optimal paths might include cycles.
- *Optimal stopping problems:*
  Same as the infinite-horizon problem, but with an option of deciding to terminate the process at any node x by paying an exit time-penalty $q(x)$. The value function then satisfies

$$U_i = \min \left(q_i, \min_{x_j \in N(x_i)} \left\{ K_{ij} + \alpha U_j \right\} \right) \qquad \text{for } \forall x_i \in X. \tag{2.6}$$

  We note that, unlike the previous case, the value function is well-defined even with $\alpha = 1$ (i.e., without time-discounting).

2.2 *The value function for uncertain-horizon processes on graphs*

In this paper we concentrate on *uncertain-horizon problems*, where the process does not terminate deterministically after a pre-specified number of steps nor upon reaching some special exit set Q, but instead might terminate with probability $p \in (0, 1)$ after each transition[1]. Two simple examples of such randomly-terminated problems are provided in Figure 1.

We will use Y to denote the set of all infinite paths on X and $Y(x) = \{\boldsymbol{y} \in Y \mid y_0 = x\}$ for the set of all paths starting from x. If the termination occurs after t steps, the total cost of the path is the same as in (2.1). The a priori probability of termination after exactly $t \geqslant 1$ steps is clearly $\hat{P}_t = (1-p)^{t-1}p$. The cost of an infinite path $\boldsymbol{y} = (y_0, \dots)$ is now a random variable whose expected value is

$$J(\boldsymbol{y}) = E[Cost(\boldsymbol{y})] = \sum_{t=1}^{\infty} \hat{P}_t \, Cost(y_0, \dots, y_t). \tag{2.7}$$

Since the graph is finite, functions K and q are bounded, the above series is absolutely convergent, and $|J(\boldsymbol{y})|$ is uniformly bounded for all $\boldsymbol{y} \in Y$.

Starting the process from any node $x \in X$, the goal is to minimize the expected total cost up to the termination. The value function $V(x)$ can be defined as

$$V(x) = \min_{\boldsymbol{y} \in Y(x)} J(\boldsymbol{y}), \tag{2.8}$$

where the existence of a minimizer follows from the compactness of $Y(x)$ and continuity of J; see Lemma A.2 in Appendix A.

We will say that $\boldsymbol{y} = (y_0, y_1, \dots) \in Y$ is a *simple path* if $y_k = y_m$ implies $y_{k+1} = y_{m+1}$. A *simple loop* is a periodic simple path. Since the set X is finite, any simple path leads to a simple loop within the first M steps. We will use Y^s to denote the set of all simple paths and $Y^s(x) \subset Y^s$ for the set of all such paths starting from x. Theorem A.5 in Appendix A proves the existence of a *minimizing simple path* for every $x \in X$.

We note that on any path at least one transition happens before the termination with probability one. As a result we can rewrite

$$\begin{aligned} J(\boldsymbol{y}) &= K(y_0, y_1) + pq(y_1) + \sum_{t=2}^{\infty} \hat{P}_t \, Cost(y_1, \dots, y_t) \\ &= K(y_0, y_1) + pq(y_1) + (1-p)J(y_1, \dots). \end{aligned} \tag{2.9}$$

This yields the Optimality Principle:

$$V_i = V(x_i) = \min_{x_j \in N(x_i)} \left\{ K_{ij} + pq_j + (1-p)V_j \right\}, \quad i = 1, \dots, M. \tag{2.10}$$

REMARK 2.1 This problem can also be restated as an infinite-horizon deterministic control by setting the costs of transition $\tilde{K}_{ij} = K_{ij} + pq_j$ and the discounting factor $\alpha = (1-p)$. Indeed, the

[1] More general problems with (p dependent on the current state or even on the last transition) can be handled very similarly; in this section we use the same constant p at all nodes for the sake of notational simplicity.

expected cost of any uncertain-horizon path can be re-written as

$$\begin{aligned} E[Cost(\mathbf{y})] &= \sum_{k=0}^{\infty} K(y_k, y_{k+1}) \Big(\sum_{t=k+1}^{\infty} \hat{P}_t \Big) + \sum_{t=1}^{\infty} q(y_t) \hat{P}_t \\ &= \sum_{k=0}^{\infty} K(y_k, y_{k+1})(1-p)^k + \sum_{t=1}^{\infty} q(y_t) \hat{P}_t \\ &= \sum_{k=0}^{\infty} \left(K(y_k, y_{k+1}) + pq(y_{k+1}) \right) (1-p)^k = \sum_{k=0}^{\infty} \alpha^k \tilde{K}(y_k, y_{k+1}). \end{aligned} \tag{2.11}$$

Conversely, consider an infinite-horizon deterministic control problem on a graph specified by a matrix of $\tilde{K}_{ij}$'s and by a discounting factor $\alpha \in (0,1)$. Assume that self-transitions are allowed at every node, i.e.,

$$x_i \in N(x_i), \quad \text{for } \forall x_i \in X. \tag{A1}$$

(Whenever we need to refer to all *other* transitions possible from x_i, we will use the set $\hat{N}(x_i) = N(x_i) \backslash \{x_i\}$.) Subject to assumption A1, each infinite-horizon problem can also be restated as a randomly-terminated problem by setting:

$$p = (1-\alpha); \quad q_i = \tilde{K}_{ii}/p; \quad K_{ij} = \tilde{K}_{ij} - pq_j; \quad K_{ii} = 0.$$

REMARK 2.2 Similarly, if the assumption A1 holds in an uncertain-horizon problem, we can always assume that self-transitions incur zero cost by setting

$$q_i^{new} = q_i + K_{ii}/p; \qquad K_{ij}^{new} = K_{ij} + p\,(q_j - q_j^{new}) = K_{ij} - K_{jj}; \qquad K_{ii}^{new} = 0.$$

Thus, for the rest of this paper we will assume without loss of generality that

$$K_{ii} = 0 \quad \text{for all} \quad i = 1, \ldots, M. \tag{A2}$$

We will also make an additional assumption, the computational consequences of which will become clear in Section 2.4.2:

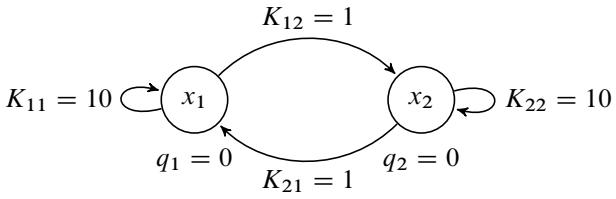
$$K_{ij} \geqslant \Delta \geqslant 0 \quad \text{for all } i = 1, \ldots, M \text{ and for all } j \neq i. \tag{A3}$$

More generally, we can similarly treat any randomly-terminated problem, where a transition from another node costs at least as much as staying in place. (Indeed, if A1 holds and $K_{ij} \geqslant K_{jj}$ for all i and j, then the procedure described in Remark 2.2 will ensure that K^{new} will satisfy A2 and A3.)

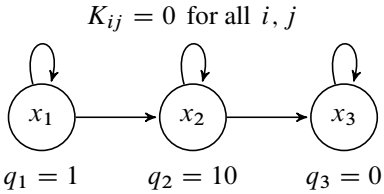
In view of (2.7), if a constant is added to all q_i's, then the same constant is added to all V_i's. As a result, we can also assume without loss of generality that $q_i > 0$ for $\forall x_i \in X$.

2.3 *Properties of the value function*

Figure 1A shows a simple example, where every optimal path is a loop. As we'll see later, such loops make it impossible to solve the system (2.10) efficiently. Fortunately, conditions A1–A3 preclude this scenario and also yield useful bounds on V.



(A)



(B)

FIG. 1. Top: A simple example violating assumption A2. The optimal path clearly alternates between x_1 and x_2. By the symmetry, $V_1 = V_2 = 1/p$. We note that the value iterations process described in Section 2.4 would generally require infinitely many iterations here (except for a lucky initial guess $W^0 = V$).
Bottom: A simple example satisfying assumptions A1–A3. We note that $V_2 = V_3 = q_3 = 0$ and, starting from x_2 the path $(x_2, x_3, x_3, \ldots)$ is always optimal regardless of p. However, starting from x_1, the path $(x_1, x_2, x_3, x_3, \ldots)$ is optimal only for $p \leqslant 0.1$, and $V_1 = \min(1,\ 10p)$.

We will refer to a path $\boldsymbol{y} = (y_0, y_1, \ldots)$ as *eventually motionless* (or *e.m.*) if there exists some $m \geqslant 0$ such that $y_k = y_m$ for all $k \geqslant m$. We note that for any such e.m. path the formula (2.7) can be rewritten as

$$J(\boldsymbol{y}) \;=\; \sum_{t=1}^{m-1} (1-p)^{t-1}\, p\, Cost(y_0, \ldots, y_t) \;+\; (1-p)^{m-1} Cost(y_0, \ldots, y_m). \tag{2.12}$$

We will refer to a node $x \in X$ as *motionless* if $\boldsymbol{y} = (x, x, x, \ldots)$ is a valid path and $V(x) = J(\boldsymbol{y}) = q(x)$. We will also use $M \subset X$ to denote a set of all such motionless nodes. To illustrate, in the example of Figure 1A $M = \emptyset$ regardless of p. In the example of Figure 1B, $\boldsymbol{x}_3 \in M$, $\boldsymbol{x}_2 \notin M$, but $\boldsymbol{x}_1 \in M$ only if $p \geqslant 0.1$.

THEOREM 2.3 Suppose an uncertain-horizon problem is such that A1, A2, and A3 hold. Then
1. $V_i \leqslant q_i$ for all $x_i \in X$.
2. If $\boldsymbol{y} \in Y^s(x_i)$ is optimal and $y_k = x_j$ then $V_i \geqslant k\Delta + V_j$.
3. If a loop $(y_0, \ldots, y_m)$ is an optimal path, then $V(y_k) = q(y_k) = q(y_0)$ for all $k = 0, \ldots, m$.
4. For every $x \in X$ there exists an e.m. optimal path $\boldsymbol{y} \in Y^s(x)$.
5. If an e.m. optimal path from x leads through $\tilde{x}$, then there exists an e.m. optimal path from $\tilde{x}$ avoiding x.

Proof. 1. Assumption A2 and (2.10) yield

$$V_i \leqslant pq_i + (1-p)V_i \quad \Longrightarrow \quad V_i \leqslant q_i = J(x_i, x_i, x_i, \ldots).$$

2. First, suppose $y_1 = x_j$; i.e., $x_j \in N(x_i)$ is a minimizer in (2.10). Then, using the above, $V_i = K_{ij} + pq_j + (1-p)V_j \geqslant K_{ij} + V_j \geqslant \Delta + V_j$. Applying this argument recursively yields $V_i \geqslant k\Delta + V_j$.
3. An optimal path is optimal for every node on it. Suppose $q(y_n) \leqslant q(y_k)$ for $k = 0, \ldots, m$. Then $\tilde{x} = y_n \in M$: given A3, the path $\tilde{\boldsymbol{y}} = (\tilde{x}, \tilde{x}, \tilde{x}, \ldots)$ is at least as good for $\tilde{x}$ as the original loop $\boldsymbol{y}$. Also $V(\tilde{x}) = J(\tilde{\boldsymbol{y}}) = q(\tilde{x})$. The equality $J(\boldsymbol{y}) = J(\tilde{\boldsymbol{y}})$ is only possible if $K(y_k, y_{k+1}) = 0$ and $q(y_k) = q(\tilde{x})$ for $k = 0, \ldots, m$. As a result, every $y_k \in M$ and $V(y_0) = \ldots = V(y_m)$. (This also shows that a non-trivial loop cannot be optimal if $\Delta > 0$.)
4. Theorem A.5 shows the existence of an optimal simple path $\boldsymbol{y}$ (eventually leading to a simple loop). The above also shows that the entry point of that simple loop y_k is a motionless node. Thus, the loop can be replaced by $(y_k, y_k, \ldots)$ without changing the cost of $\boldsymbol{y}$.
5. For $\Delta > 0$ this is obvious (if an optimal path from x passes from $\tilde{x}$, then $V(x) > V(\tilde{x})$). If $\Delta = 0$, then combining segments of both optimal paths we can form an optimal loop passing through both x and $\tilde{x}$. As shown above, this implies that $\tilde{x}$ is a motionless node.

□

In the rest of this section we will use superscripts to indicate the dependence on p of the expected path-cost $J^p(\boldsymbol{y})$, the value function $V^p(x)$ and the motionless set $M^p = \{x \mid V^p(x) = q(x)\}$. Wherever this superscript is omitted, the properties hold for any fixed $p \in (0,1)$. To address two extreme cases, we introduce V^0 and V^1 as respective solutions of two systems:

$$V_i^0 = V^0(x_i) = \min\Big\{ \min_{x_j \in \hat{N}} \{K_{ij} + V_j^0\},\, q_i \Big\}, \quad i = 1, \ldots, M; \tag{2.13}$$

$$V_i^1 = V^1(x_i) = \min_{x_j \in N(x_i)} \big\{K_{ij} + q_j\big\}, \quad i = 1, \ldots, M. \tag{2.14}$$

We see that (2.13) is a variant of (2.6) with $\alpha = 1$. In other words, V^0 is the value function for a deterministic optimal stopping problem with no time-discounting. As a result, the label-setting methods of section 2.4.1 are applicable, and for every starting node x there exists some optimal finite path $(y_0, \ldots, y_m)$; i.e., $y_0 = x$ and $V^0(x) = Cost(y_0, \ldots, y_m)$. In the framework of randomly-terminated problems, V^0 can be intuitively interpreted as the limiting case, in which the termination is so unlikely that we are guaranteed to reach any node of our choice before it occurs. (This interpretation is justified in Theorem 2.4 below).

V^1 corresponds to the opposite case, where the termination always happens after the very first transition. We note that since $x_i \in N(x_i)$, we have $V_i^1 \leqslant q_i$ for all $x_i \in X$. By the above definition, if $x_i \in M^1$, we have $q_i \leqslant K_{ij} + q_j$ for all $x_j \in \hat{N}(x_i)$. An important subset is obtained when the self-transition is *the only* optimal strategy:

$$M_0^1 = \big\{x_i \in X \mid q_i < K_{ij} + q_j,\ \forall x_j \in \hat{N}(x_i)\big\}.$$

Below we show that $M^p \subset M^1, \forall p \in (0,1)$. However the example in Figure 2 shows that some nodes in $M^1 \backslash M_0^1$ might become motionless only in the limit (as $p \to 1$). On the other hand, every node in M_0^1 is already motionless for some $p < 1$.

To simplify the notation, we will refer to a path $\boldsymbol{y} \in Y(x)$ as *p-optimal* if $J^p(\boldsymbol{y}) = V^p(x)$.

THEOREM 2.4 Suppose an uncertain-horizon problem is such that A1, A2, and A3 hold. Then

1. Suppose $\boldsymbol{y} = (y_0, \ldots, y_m, y_m, \ldots)$ is a p-optimal e.m. path leading from $x = y_0$ to y_m. Define the cost of each finite subpath as in (2.1). Then $Cost(y_0, \ldots y_m) \leqslant Cost(y_0, \ldots y_n)$ and $q(y_m) \leqslant q(y_n)$ for all $n < m$ (and the second inequality becomes strict if $\Delta > 0$).

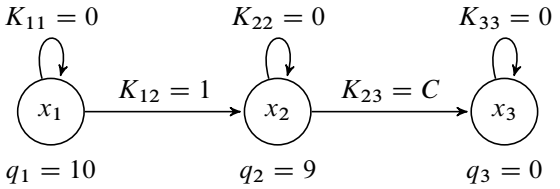


FIG. 2. A simple example with subtle motionless set properties. First, note that $x_1 \in M^1 \backslash M_0^1$ for all $C > 0$. Second, assuming that $C = 1$, we have $V_3^p = 0$, $V_2^p = 1$, $V_1^p = 2 + 8p$, and $x_1 \notin M^p$ for any $p < 1$. On the other hand, if $C > 9$ then $x_1 \in M^p$ for all $p \in [0, 1]$.

2. $V^p(x) \in [V^0(x),\ V^1(x)]$ for all $p \in (0, 1)$ and all $x \in X$.
3. $0 < p_1 \leqslant p_2 < 1 \quad \Longrightarrow \quad V^{p_1}(x) \leqslant V^{p_2}(x), \quad \forall x \in X.$
4. $V^p(x_i) \to V^1(x_i)$ for all $x_i \in X$ as $p \to 1$.
5. $V^p(x_i) \to V^0(x_i)$ for all $x_i \in X$ as $p \to 0$.
6. If $p_1 \leqslant p_2$, then $M^{p_1} \subset M^{p_2}$. In particular, $M^0 \subset M^p \subset M^1$ for $\forall\, p \in (0, 1)$.
7. For every $x_i \in M_0^1$, there exists a sufficiently large $p < 1$, such that $x_i \in M^p$.
8. For every $x_i \notin M^0$, there exists a sufficiently small $p > 0$, such that $x_i \notin M^p$.
9. Define the sets of local and global minima of q:

$$Q_l = \{x_i \mid q(x_i) \leqslant q(x_j),\ \forall x_j \in N(x_i)\}; \qquad Q_g = \{x_i \mid q(x_i) \leqslant q(x_j),\ \forall x_j \in X\}.$$

Then $Q_g \subset M^0$, $\quad Q_l \subset M^1$, $\quad$ and $\quad \forall K_{ij} = 0 \Longrightarrow Q_l = M^1$.

Proof. 1. Let $l = \operatorname{argmin}_{n \leqslant m} Cost(y_0, \ldots, y_n)$ and define $\tilde{\boldsymbol{y}} = (y_0, \ldots, y_l, y_l, \ldots)$. If $Cost(y_0, \ldots, y_l) < Cost(y_0, \ldots, y_m)$, then the formula (2.7) implies that $J^p(\tilde{\boldsymbol{y}}) < J^p(\boldsymbol{y})$, which contradicts the p-optimality of $\boldsymbol{y}$. (We emphasize that the above argument *does not* imply that $Cost(y_0, \ldots, y_n)$ is monotone non-increasing in n along every p-optimal path; see Figure 1B for a counter-example.) We note that

$$\begin{aligned} Cost(y_0, \ldots, y_n) &= \sum_{k=0}^{n-1} K(y_k, y_{k+1}) \ + \ q(y_n) \\ &\geqslant Cost(y_0, \ldots, y_m) \geqslant \sum_{k=0}^{n-1} K(y_k, y_{k+1}) \ + \ q(y_m), \end{aligned}$$

where the second inequality uses A3 and becomes strict if $\Delta > 0$. Thus, $q(y_m) \leqslant q(y_n)$.
2. Let $\boldsymbol{y} = (y_0, \ldots, y_m, y_m, \ldots)$ be a p-optimal e.m. path leading from x to y_m. Note that (2.7) defines $J^p(\boldsymbol{y})$ as a weighted average of $Cost(y_0, \ldots, y_n)$ terms. Then, from the previous part, $J^p(\boldsymbol{y}) \geqslant Cost(y_0, \ldots, y_m) \geqslant V^0(x)$. On the other hand, V^1 can also be defined as $V^1(x) = \min J^p(\boldsymbol{y})$, minimizing over all "infinite" paths $\boldsymbol{y} \in Y^s(x)$ of the form $\boldsymbol{y} = (x, y_1, y_1, \ldots)$; hence, $V^p(x) \leqslant V^1(x)$.
3. Fixing p_2, we define $m(x)$ to be the minimum number of transitions before reaching the motionless node along the p_2-optimal paths from x. (E.g., $x \in M^{p_2} \iff m(x) = 0$.) We prove this statement by induction on $m(x)$. First, for $m(x) = 0$, we have $V^{p_2}(x) = q(x) \geqslant V^{p_1}(x)$. Now suppose the statement is true for all x_j such that $m(x_j) \leqslant k$ and consider x_i such that $m(x_i) = k+1$.

Then there exists $x_j^* \in N(x_i)$, a minimizer in formula (2.10) such that $m(x_j^*) = k$. As a result,

$$\begin{aligned} V^{p_2}(x_i) &= K_{ij^*} + p_2 q(x_j^*) + (1-p_2)V^{p_2}(x_j^*) \;\geqslant\; K_{ij^*} + p_2 q(x_j^*) + (1-p_2)V^{p_1}(x_j^*) \\ &= K_{ij^*} + V^{p_1}(x_j^*) + p_2\left(q(x_j^*) - V^{p_1}(x_j^*)\right) \\ &\geqslant K_{ij^*} + V^{p_1}(x_j^*) + p_1\left(q(x_j^*) - V^{p_1}(x_j^*)\right) \\ &\geqslant V^{p_1}(x_i). \end{aligned}$$

4. For any $\boldsymbol{y} \in Y(x)$, formulas (2.1) and (2.7) can be combined to show that,

$$J^p(\boldsymbol{y}) = p\left[K(y_0, y_1) + q(y_1)\right] + \sum_{t=2}^{\infty}(1-p)^{t-1} p\, Cost(y_0, \ldots, y_t)$$
$$\to \left[K(y_0, y_1) + q(y_1)\right], \quad \text{as } p \to 1.$$

5. For any e.m. $\boldsymbol{y} = (y_0, \ldots, y_m, y_m, \ldots) \in Y(x_i)$, the formula (2.12) implies that $J^p(\boldsymbol{y}) \to Cost(y_0, \ldots, y_m)$ as $p \to 0$. If the finite path $(y_0, \ldots, y_m)$ is optimal for $V^0(x_i)$, then $Cost(y_0, \ldots, y_m) = V^0(x_i) \leqslant V^p(x_i) \leqslant J^p(\boldsymbol{y})$, implying $V^p(x_i) \to V^0(x_i)$.
6. If $x \in M^{p_1}$, then $q(x) = V^{p_1}(x) \leqslant V^{p_2}(x) \leqslant q(x)$. Thus, $x \in M^{p_2}$.
7. Since $x_i \in M_0^1$, we can define

$$p = q_i \;/\; \min_{x_j \in \hat{N}(x_i)} \{K_{ij} + q_j\} \;<\; 1.$$

As a result, for any simple path $\boldsymbol{y} \in Y^s(x_i)$ involving at least one non-trivial transition (i.e., with $y_1 \neq y_0 = x_i$), we have $J^p(\boldsymbol{y}) \geqslant p\left[K(y_0, y_1) + q(y_1)\right] \geqslant q_i$. Thus, $x_i \in M^p$.
8. Part 5 of this Theorem implies $\lim_{p\to 0} V^p(x_i) = V^0(x_i) < q_i$. Thus, for sufficiently small p, $V^p(x_i) < q_i$, and $x_i \notin M^p$.
9. If $x_i \in Q_g$, then for any path $(y_0, \ldots, y_m)$ starting from $y_0 = x_i$, we have $Cost(y_0, \ldots, y_m) \geqslant q(y_m) \geqslant q_i \implies x_i \in M^0$. If $x_i \in Q_l$, then $q_i \leqslant q_j \leqslant K_{ij} + q_j$ for all $x_j \in N(x_i) \implies x_i \in M^1$.
If $\forall K_{ij} = 0$, the definitions of M^1 and Q_l coincide. In this case every p-optimal trajectory becomes motionless at some local minimum of q.

□

REMARK 2.5 For the general infinite-horizon and/or optimal stopping problems reviewed in Section 2.1, the cost of any trajectory is obviously a non-decreasing function of the discounting factor α. We note that Part 3 of the above theorem does not contradict this monotonicity, since the procedure described in Remark 2.1 also makes $\tilde{K}_{ij}$'s dependent on p.

2.4 *Solving the Dynamic Programming equations*

The Optimality Principle (2.10) yields a system of M coupled non-linear equations, which can be challenging to solve efficiently.

An operator T can be defined on $\boldsymbol{R}^M$ component-wise by applying the right hand side of equation (2.10); i.e., for any $W \in \boldsymbol{R}^M$,

$$(TW)_i = \min_{x_j \in N(x_i)} \left\{K_{ij} + pq_j + (1-p)W_j\right\}. \tag{2.15}$$

Clearly, $V = \begin{bmatrix} V_1 \\ \vdots \\ V_M \end{bmatrix}$ is a fixed point of T and one hopes to recover V by *value iteration*:

$$W^{n+1} := T\,W^n \quad \text{starting from an initial guess } W^0 \in \boldsymbol{R}^M. \tag{2.16}$$

The value iterations will converge (i.e., $W^n \to V$) for any initial guess W^0 [9], but this approach is not very efficient. It is generally possible that the value iterations will not converge in a finite number of steps (e.g., see the example in Figure 1A). If the problem is such that the convergence is attained in finitely many steps for every W^0, it is easy to show that $W^M = V$ (i.e., at most M value iterations will be needed). Since the computational cost of each value iteration is $O(M)$, this results in the overall cost of $O(M^2)$. A standard Gauss–Seidel relaxation can be used to speed up the convergence, but then the efficiency becomes strongly dependent on the chosen node-ordering and the worst case computational cost remains $O(M^2)$.

Label-setting methods provide an attractive alternative to value iterations for problems with intrinsic *causality*. These methods reorder the iterations over the nodes to guarantee that each V_i is recomputed at most $\kappa << M$ times. This can also be interpreted as finding the optimal Gauss–Seidel ordering *dynamically* – based on the value function already correctly computed on a part of the graph.

2.4.1 *Label-setting for SP problems.* For reader's convenience, we first provide a brief overview of standard label-setting methods for deterministic shortest path problems (2.3). If we assume $K_{ij} \geqslant \Delta \geqslant 0$, then $U_i = K_{ij} + U_j$ implies $U_i \geqslant \Delta + U_j \geqslant U_j$. This induces a causal ordering: each U_i depends only on the smaller values in the adjacent nodes. I.e., if $x_j \in N(x_i)$ and $U_j > U_i$ then replacing U_j with $+\infty$ will not change the value of U_i as computed by formula (2.3). This is the basis of Dijkstra's classic method [18].

The method subdivides X into three classes: *Accepted* (or "permanently labeled") nodes, *Considered* (or "tentatively labeled") nodes that have *Accepted* nodes among their neighbors, and *Far* (or "unlabeled") nodes. The values for *Considered* x_i's are successively re-evaluated using only the previously *Accepted* adjacent values:

$$U(x_i) := \min_{x_j \in \tilde{N}(x_i)} \left\{K_{ij} + U_j\right\}, \tag{2.17}$$

where $\tilde{N}(x_i) = N(x_i) \cap$ *Accepted*. The algorithm is initialized by designating all exit-nodes as *Considered* (with tentative labels $U(x_i) = q_i$ for all $x_i \in Q$) and all other nodes as *Far* (with tentative labels $U(x_i) = +\infty$ for all $x_i \in X \backslash Q$). At each stage the algorithm chooses the smallest of tentative labels $U(\bar{x})$, designates $\bar{x}$ as *Accepted* (making this label permanent and removing $\bar{x}$ from the list of *Considered*), and re-evaluates U_i for each not-yet-*Accepted* x_i such that $\bar{x} \in N(x_i)$. Since $\bar{x}$ is the only new element in $\tilde{N}(x_i)$, that re-evaluation can be more efficiently performed by setting

$$U(x_i) := \min\left\{U(x_i), (K(x_i, \bar{x}) + U(\bar{x}))\right\}. \tag{2.18}$$

In addition, all previously *Far* x_i's updated at this step are designated *Considered*.

The algorithm terminates once the list of *Considered* nodes is empty, at which point the vector $U \in \boldsymbol{R}^M$ satisfies the system of equations (2.3). The necessity to sort all tentative labels suggests

the use of heap-sort data structures for *Considered* nodes [1], resulting in the overall computational complexity of $O(M \log M)$.

If $\Delta > 0$, then Dial's label-setting method is also applicable [17]. The idea is to avoid sorting *Considered* nodes and instead place them into "buckets" of width Δ based on their tentative values. If $U(\bar{x})$ is the "smallest" of tentative labels and $U(\boldsymbol{x})$ is currently in the same bucket, then even after $\bar{x}$ is permanently labeled, it cannot affect $U(\boldsymbol{x})$ since

$$U(x) < U(\bar{x}) + \Delta \leqslant U(\bar{x}) + K(x, \bar{x}).$$

Thus, a typical stage of Dial's method consists of *Accepting* everything in the current bucket, recomputing all not-yet-*Accepted* nodes adjacent to those newly *Accepted*, switching them to other buckets if warranted by the new tentative labels, and then moving on to the next bucket. Since inserting to and deleting from a bucket can be performed in $O(1)$ time, the overall computational complexity of Dial's method becomes $O(M)$. In addition, while Dijkstra's approach is inherently sequential, Dial's method is naturally parallelizable. However, in practice, Dial's method is often less efficient then Dijkstra's because the constant factor hidden in the former's $O(M)$ asymptotic complexity depends on Δ and on average K_{ij} values encountered along optimal paths. Many other enhancements of the above label-setting methods are available in the literature (e.g., see [7], [1] and references therein).

2.4.2 *Label-setting for uncertain-horizon problems.* We now show that a variation of Dijkstra's method is applicable in the uncertain-horizon case provided A1–A3 hold.

Part 2 of Theorem 2.3 shows the causality necessary for a variant of Dijkstra's method to correctly compute the value function. Some modifications are needed, however. First of all, the tentative labels should be initially set to $V_i = q_i$. Secondly, the update formula (2.18) has to be replaced by

$$V(x_i) := \min\Big\{V(x_i), \big(K(x_i, \bar{x}) + pq(\bar{x}) + (1-p)V(\bar{x})\big)\Big\}. \tag{2.19}$$

Thirdly, there is a question of which nodes should be initially placed on the *Considered* list. (Since this list is maintained using a heap-sort data structure, this decision will directly impact the efficiency of the method.) The only way for Dijkstra's to produce a wrong result is to have some node $\bar{x}$ *Accepted* before its optimal neighbor $\hat{x} \in N(\bar{x})$. Suppose $\boldsymbol{y}$ is $\bar{x}$'s optimal path becoming motionless at some node $\tilde{x}$; i.e., $\boldsymbol{y} = (y_0, \ldots, y_{m-1}, y_m, y_m, y_m, \ldots) \in Y^s(\bar{x})$, where $y_0 = \bar{x}$, $y_1 = \hat{x}$, and $y_m = \tilde{x}$. By Part 2 of Theorem 2.3, $V(\bar{x}) \geqslant V(y_1) \geqslant \ldots V(y_m) = q(y_m)$. As long as, y_m is initially marked as *Considered*, it is easy to show that $\bar{x}$ will not be *Accepted* prematurely. So, it is sufficient to initially mark as *Considered* all motionless nodes. Unfortunately, the set M is not known in advance, but Part 6 of Theorem 2.4 can be used to over-estimate it with $M^1 = \{x_i \mid q_i \leqslant K_{ij} + q_j, \forall x_j \in \hat{N}(x_i)\}$. However, the following Lemma shows why it is already sufficient to mark as *Considered* a smaller set Q_1:

LEMMA 2.6 For every $\bar{x} \in X$ there exists a path $\boldsymbol{y} = (y_0, \ldots, y_r)$ such that $y_r \in Q_1$, $y_0 = \bar{x}$, and, for all $k = 0, \ldots, (r-1)$,
either y_k is a motionless node and $V(y_k) = q(y_k) > V(y_{k+1})$;
or y_{k+1} is an optimal transition from y_k
(i.e., $V(y_k) = K(y_k, y_{k+1}) + pq(y_{k+1}) + (1-p)V(y_{k+1})$).

Proof. Start with an optimal e.m. path for $\bar{x}$ leading to some motionless node $\tilde{x}$. If $\tilde{x} \notin Q_1$, then

there exists $x \in N(\tilde{x})$ such that $V(\tilde{x}) = q(\tilde{x}) > q(x) \geqslant V(x)$. Now concatenate the x's optimal e.m. path and repeat the process until reaching Q_l. □

Using concatenated paths described in Lemma 2.6, it is easy to prove by induction that, starting with Q_l marked as *Considered*, the order of acceptance will be also causal (i.e., $\hat{x}$ will be *Accepted* before $\bar{x}$). We summarize the resulting Dijkstra-like method for uncertain-horizon processes on graphs in Algorithm 1 (allowing for edge-dependent probabilities of termination p_{ij}'s).

Algorithm 1 A Dijkstra-like method for randomly-terminated processes on graphs

```
start with all nodes marked as Far;
set V_i := q_i for all x_i ∈ X;
mark all local minima of q as Considered;
while (Considered list is not empty)   {
   let V_j = V(x_j) be the smallest Considered value;
   mark x_j Accepted;
   for each not-yet-Accepted x_i such that x_j ∈ N(x_i)   {
      set V_i := min(V_i, K_ij + p_ij q_j + (1 − p_ij) V_j);
      if x_i is Far, mark it Considered;
   }
}
```

The direct procedure of identifying all local minima takes $O(\kappa M)$ operations and may or may not be advantageous (after all, the algorithm will also work if the entire X is initially labeled as *Considered*). However, this pre-processing becomes clearly worthwhile if several different uncertain-horizon problems are to be solved on the same graph (with the same terminal cost function $q : X \mapsto \boldsymbol{R}$) but with different transition cost functions $K : X \times X \mapsto \boldsymbol{R}_{+,0}$.

Finally, we note that if $\Delta > 0$, then a similarly modified Dial's method will also be applicable with buckets of width Δ.

2.5 *Example: Optimal idle-time processing*

As a sample application for uncertain-horizon processes, we consider the task of optimally using the idle-time to minimize the expected waiting time of the first client. We model arrival of client requests as a Poisson process with λ requests arriving on average per unit time; i.e., if T_1 is the time of the first request, then $P(T_1 \geqslant t) = e^{-\lambda t}$ and $P(T_1 \in [t, t + dt]) = \lambda e^{-\lambda t}\, dt$. If $x_i \in X$ is the current state of the system when the first request arrives, then $q(x_i)$ is the expected time for completing that request. At every stage of the process we can either decide to stay in the current state or start a transition to any state $x_j \in N(x_i)$. Such a transition will take some time $\tau_{ij} > 0$ and the system will not be capable of responding to any incoming requests until that transition is completed. Thus, even if $q(x_j) < q(x_i)$, this has to be balanced against the expected additional waiting time $(\tau_{ij} - T_1)$ provided $T_1 < \tau_{ij}$, i.e.,

$$K_{ij} = \int_0^{\tau_{ij}} (\tau_{ij} - t)\lambda e^{-\lambda t}\, dt = \frac{e^{-\lambda\tau_{ij}} - (1 - \lambda\tau_{ij})}{\lambda} \geqslant 0, \quad \forall i \neq j.$$

If $V_i = V(x_i)$ is the minimal expected wait time starting from the state x_i, then

$$V_i = \min\left\{q_i, \min_{x_j \in \hat{N}(x_i)} \left\{K_{ij} + p_{ij} q_j + (1 - p_{ij}) V_j\right\}\right\}, \tag{2.20}$$

where $p_{ij} = P(T_1 < \tau_{ij}) = 1 - e^{-\lambda \tau_{ij}}$. This system clearly satisfies the assumptions A1, A2, and A3 and the label-setting methods of Section 2.4.2 are therefore applicable.

One interesting application of this problem is to minimize the expected wait time of the first emergency caller by moving an ambulance during the idle-time. In that case, x is the current position of an ambulance and $q(x)$ is the expected travel time from x to the caller's location. Suppose $d(x, \tilde{x})$ is the minimum travel time through the graph from x to $\tilde{x}$ and $\tilde{P}(\tilde{x})$ is the probability that the next call originates from $\tilde{x}$. Then

$$q(x) = \sum_{\tilde{x} \in X} \tilde{P}(\tilde{x})\, d(x, \tilde{x}).$$

If $\tilde{P}$ is non-zero for a small fraction of nodes in X only, then the distances $d(x, \tilde{x})$ are best evaluated through a repeated application of the standard Dijkstra's method. Otherwise, Floyd–Warshall algorithm will likely be more efficient [1].

Any global minimum of q is obviously the optimal location to "park" an ambulance (or to build an ambulance depot). But if the ambulance starts at any other location x_i, the optimal strategy for moving it while expecting the call is recovered by using the minimizer in formula (2.20).

REMARK 2.7 After this paper was already submitted, we have found a somewhat different single-ambulance model introduced in a recent technical report [39], which also contains an independently developed label-setting algorithm for the "no running cost" subcase (i.e., all $K_{ij} = 0$).

Of course, realistic dynamic fleet management problems require efficient control of multiple vehicles simultaneously. The expected response time would then depend on the current position and status of all vehicles, resulting in the exponential growth of the state space. This curse of dimensionality typically precludes the direct use of the above approach, but a computationally feasible alternative is provided by the methods of *approximate dynamic programming* [27]. A recent example of the latter applied to the ambulance redeployment problem can be found in [24].

Non-iterative algorithms for approximate dynamic programming would be clearly very useful, but, to the best of our knowledge, no such methods are currently available.

3. Uncertain-horizon processes in continuous state-space

We now derive the results parallel to those in Section 2 but in continuous state-space. We start with the taxonomy of "common" deterministic optimal control problems on $\Omega \subset \boldsymbol{R}^n$. We then show the relationship between the uncertain-horizon problems and time-discounted optimal-stopping problems. A modified version of Fast Marching Method is then developed for the latter and illustrated by a number of numerical examples in Section 4.

3.1 *Common types of optimal control problems in continuous state-space*

We will assume that Ω is an open bounded subset of $\boldsymbol{R}^n$. Suppose $A \in \boldsymbol{R}^m$ is a compact set of control values, and the set of admissible controls $\mathfrak{A}$ consists of all measurable functions $\boldsymbol{a} : \boldsymbol{R} \mapsto A$.

The dynamics of the system is defined by

$$\begin{aligned} \boldsymbol{y}'(s) &= \boldsymbol{f}\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big), \\ \boldsymbol{y}(0) &= \boldsymbol{x} \in \Omega, \end{aligned} \tag{3.1}$$

where $\boldsymbol{y}(s)$ is the system state at the time s, $\boldsymbol{x}$ is the initial system state, and $\boldsymbol{f} : \overline{\Omega} \times A \mapsto \boldsymbol{R}^n$ is the velocity.

We will also assume that the running cost $K : \overline{\Omega} \times A \mapsto [K_1, K_2]$ and the discount rate $\beta \geqslant 0$ are known. (The case $\beta = 0$ corresponds to no discounting.)

We note that the partial differential equations derived for problems in this subsection typically do not have classical (smooth) solutions on the entire domain and the weak solutions are generally not unique. The theory of *viscosity solutions* introduced by Crandall and Lions [15] is used to overcome this difficulty, picking out the unique weak solution coinciding with the value function of the corresponding control problem. Here we only provide a formal derivation and classification of PDEs; rigorous proofs and many relevant references can be found in [4].

- *Finite-horizon problems:*
  Suppose the process runs until the specified time $\mathcal{T}$. This allows us to define the total cost of using a control $\boldsymbol{a}(\cdot)$ starting from $\boldsymbol{x}$ with $T = \mathcal{T} - t$ seconds left before the termination:

$$J(\boldsymbol{x}, t, \boldsymbol{a}(\cdot)) = \int_0^T e^{-\beta s} K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big)\, ds + e^{-\beta T} q\big(\boldsymbol{y}(T)\big),$$

  where t is the starting time and $q : \Omega \mapsto R$ is the terminal cost. The value function $u : \Omega \times [0, \mathcal{T}]$ can be defined as usual: $u(\boldsymbol{x}, t) = \inf J(\boldsymbol{x}, t, \boldsymbol{a}(\cdot))$, where the inf is taken over

$$\mathcal{A}_0(\boldsymbol{x}) = \big\{\boldsymbol{a}(\cdot) \in \mathcal{A} \mid \boldsymbol{y}(s) \in \overline{\Omega} \text{ for all } s \in [0, T]\big\}.$$

  The optimality condition shows that, for any $\tau \in (0, T)$,

$$u(\boldsymbol{x}, t) = \inf_{\boldsymbol{a}(\cdot)} \left\{ \int_0^\tau e^{-\beta s} K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big)\, ds \;+\; e^{-\beta \tau} u\big(y(\tau), t + \tau\big) \right\}.$$

  Assuming that $u \in C^2$, Taylor-expanding and letting $\tau \to 0$, it is easy to show that u is a solution of a Hamilton–Jacobi–Bellman PDE:

$$\begin{aligned} -u_t + \beta u - \min_{\boldsymbol{a} \in A} \big\{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla u \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\big\} = 0, \ \text{for } \boldsymbol{x} \in \Omega, t \in [0, \mathcal{T}); \\ u = q, \ \text{for } \boldsymbol{x} \in \Omega, t = \mathcal{T}. \end{aligned} \tag{3.2}$$

  Here u_t is the derivative with respect to time, and ∇u is a vector of spatial derivatives. This is a *terminal value problem* for a hyperbolic non-linear PDE. The optimal trajectories coincide with the characteristics of this equation. Since it is time-dependent, the numerical approximation can be computed very efficiently by *time-marching* (from $t = \mathcal{T}$ into the past).
- *Undiscounted exit-time problems:*
  Here, $\beta = 0$ and the process terminates upon exiting from $\overline{\Omega}$; i.e.,

$$T = T_{\boldsymbol{x}, \boldsymbol{a}} = \inf\big\{t \in \boldsymbol{R}_{+,0} | \boldsymbol{y}(t) \notin \overline{\Omega}\big\}$$

and the terminal cost is defined only on the boundary ($q : \partial\Omega \to \boldsymbol{R}$). Since both the cost and dynamics are not directly dependent on time, the value function does not depend on t either and $u(\boldsymbol{x})$ is defined by taking inf over $\mathcal{A}$. Similar formal reasoning shows that $u(\boldsymbol{x})$ must satisfy a static Hamilton–Jacobi-Bellman PDE with Dirichlet boundary conditions

$$\begin{aligned} -\min_{\boldsymbol{a}\in A}\left\{K(\boldsymbol{x},\boldsymbol{a}) + \nabla u\cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\right\} = 0, \text{ for } \boldsymbol{x}\in\Omega;\\ u = q, \text{ for } \boldsymbol{x}\in\partial\Omega. \end{aligned} \tag{3.3}$$

If we assume $K_1 > 0$, this guarantees that wherever $u(\boldsymbol{x}) < \infty$, all optimal paths lead to $\partial\Omega$ in finite time. Thus, all characteristics originate from $\partial\Omega$ and $K_1 > 0$ implies the monotone growth of u along each characteristic. This gives one reasons to hope for causal (label-setting) numerical methods after the PDE is discretized on a grid. Such efficient methods were, indeed, constructed for a number of special cases. In the fully *isotropic* case, $K = K(\boldsymbol{x})$, $A = S_{n-1}$, and $\boldsymbol{f} = f(\boldsymbol{x})\boldsymbol{a}$. Here, a unit vector $\boldsymbol{a}$ is our chosen direction of motion, $f > 0$ is the speed of motion, and (3.3) reduces to the Eikonal PDE:

$$\begin{aligned} |\nabla u| f(\boldsymbol{x}) = K(\boldsymbol{x}) \text{ for } \boldsymbol{x}\in\Omega;\\ u = q, \text{ for } \boldsymbol{x}\in\partial\Omega. \end{aligned} \tag{3.4}$$

Two Dijkstra-like methods for discretizations of this equation were introduced by Tsitsiklis [35, 36] and Sethian [29]. The former was done in the context of semi-Lagrangian discretizations for control-theoretic problems; the latter (the Fast Marching Method) was introduced for upwind finite-difference discretizations of isotropic front propagation problems. A detailed discussion of connections between these two approaches can be found in [32]. The Fast Marching Method was later extended by Sethian and collaborators to higher order upwind discretizations on grids and unstructured meshes in $\boldsymbol{R}^n$ and on manifolds. All Dijkstra-like methods have the same computational complexity of $O(M\log M)$, where M is the number of gridpoints. Another algorithm introduced by Tsitsiklis on an 8-neighbor stencil mirrors the logic of Dial's method and results in $O(M)$ computational complexity (see [36] and the generalizations in [38]).
A more general anisotropic case is obtained when $K = K(\boldsymbol{x},\boldsymbol{a})$, $A = S_{n-1}$, and $\boldsymbol{f} = f(\boldsymbol{x},\boldsymbol{a})\boldsymbol{a}$ (i.e., both the running cost and the speed can depend on the direction of motion). Upwind discretizations of the resulting PDE are generally not causal, making label setting methods inapplicable. Ordered Upwind Methods circumvent this difficulty by dynamically extending the stencil just enough to restore the causality, resulting in Dijkstra-like computational complexity, but with an additional "anisotropy coefficient" [2, 31, 32].

- *Infinite-horizon problems:*
Here, the cost of each control is defined as before, except that $\beta > 0$, T is always taken to be $+\infty$, and there is no terminal cost: $J(\boldsymbol{x},\boldsymbol{a}(\cdot)) = \int_0^{+\infty} e^{-\beta s} K(\boldsymbol{y}(s),\boldsymbol{a}(s))\,ds$. Since we are moving for an infinite time, the value function u is no longer time-dependent. (The fact that u is finite follows from $\beta > 0$ and the boundedness of K. To remain in Ω, the inf is taken over $\mathcal{A}_0(\boldsymbol{x})$.) Similar formal reasoning shows that $u(\boldsymbol{x})$ must satisfy a static Hamilton–Jacobi–Bellman PDE

$$\beta u - \min_{\boldsymbol{a}\in A}\left\{K(\boldsymbol{x},\boldsymbol{a}) + \nabla u\cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\right\} = 0, \text{ for } \boldsymbol{x}\in\Omega. \tag{3.5}$$

We note that there is also a well-known version of this problem, where the process can also be terminated upon hitting $\partial\Omega$; see [4]. In that case, u is defined by taking inf over $\mathcal{A}$, resulting

in Dirichlet boundary conditions on $\partial\Omega$ interpreted *in the viscosity sense*. However, unlike the undiscounted exit-time problems defined above, here the optimal trajectories can be periodic inside Ω even if $K_1 > 0$. This generally precludes any use of causal numerical methods.

- *Optimal stopping problems:*
  Same as the infinite-horizon problem, but with an option of deciding to terminate the process at any point T:

$$J\big(\boldsymbol{x}, \boldsymbol{a}(\cdot), T\big) = \int_0^T e^{-\beta s} K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big)\, ds + e^{-\beta T} q\big(\boldsymbol{y}(T)\big),$$

  where, as in the finite-horizon case, $q : \Omega \mapsto R$. The value function can be defined by taking inf over all controls and all termination times:

$$u(\boldsymbol{x}) = \inf_{\boldsymbol{a}(\cdot)\in\mathfrak{A}_0(\boldsymbol{x}),\, T\geqslant 0} J(\boldsymbol{x}, \boldsymbol{a}(\cdot), T).$$

  The direct consequence of this is the fact that $u(\boldsymbol{x}) \leqslant q(\boldsymbol{x})$ on Ω and, wherever u is less than q, it should be formally a solution of (3.5). In this case, u is the viscosity solution of the following *variational inequality of obstacle type* :

$$\max\Big\{u(\boldsymbol{x}) - q(\boldsymbol{x}),\quad \beta u - \min_{\boldsymbol{a}\in A}\big\{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla u(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\big\}\Big\} = 0. \tag{3.6}$$

  See [4] for a rigorous derivation. Let $M = \{\boldsymbol{x} \in \Omega \mid u(\boldsymbol{x}) = q(\boldsymbol{x})\}$. Then the PDE (3.5) is satisfied on $\Omega\backslash M$ and $B = \partial M$ forms a *free boundary* for this problem. It is not hard to show that the optimal trajectories cannot be periodic, provided $K(\boldsymbol{x}, \boldsymbol{a}) > \beta q(\boldsymbol{x})$ for all $\boldsymbol{x}$ and $\boldsymbol{a}$. In this case, all characteristics of this PDE run into $\Omega\backslash M$ from B and the space-marching numerical approach is feasible. However, since B is a priori unknown, this presents an additional challenge for label-setting methods. (The equivalent properties of uncertain-horizon processes will be considered in detail in the next section.)
  We note that the analysis of optimal stopping problems can be reduced to that of infinite horizon problems if we extend the set of control values by adding a special "motionless" control $\boldsymbol{a}_0$ such that $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}_0) = 0$ and $K(\boldsymbol{x}, \boldsymbol{a}_0) = \beta q(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega$. Similarly, any infinite horizon problem with a motionless control $\boldsymbol{a}_0$ available in every state $\boldsymbol{x}$ can also be viewed as an optimal stopping problem by setting $q(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{a}_0(\boldsymbol{x}))/\beta$.

3.2 *The value function for uncertain-horizon processes*

We now consider a randomly-terminated state-restricted undiscounted problem. The dynamics is yet again restricted to $\overline{\Omega}$, the state evolution is described by (3.1), where $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})$ is Lipschitz-continuous in $\boldsymbol{x}$ and continuous in $\boldsymbol{a}$. We assume that both K and q are lower semi-continuous functions satisfying the following:

$$\begin{gathered} q : \overline{\Omega} \mapsto R \text{ is the terminal cost such that} \\ q_1 \leqslant q(\boldsymbol{x}) \leqslant q_2, \quad \forall \boldsymbol{x} \in \overline{\Omega}, \quad \text{where } q_1 = \min_{\boldsymbol{x}} q(\boldsymbol{x}) \geqslant 0. \end{gathered} \tag{B1}$$

$$\begin{gathered} K : \overline{\Omega} \times A \mapsto R \text{is the running cost such that} \\ 0 \leqslant K_1 \leqslant K(\boldsymbol{x}, \boldsymbol{a}) \leqslant K_2, \quad \forall \boldsymbol{x} \in \overline{\Omega}, \boldsymbol{a} \in A. \end{gathered} \tag{B2}$$

Some of the properties proven below, also rely on a stronger assumption:

$$\begin{aligned}&\text{The velocity } \boldsymbol{f} : \overline{\Omega} \times A \mapsto R^n \text{ is bounded (i.e., } |\boldsymbol{f}| \leqslant F_2) \text{ and the motion in}\\ &\text{every direction is always possible, i.e.,}\\ &\forall \boldsymbol{x} \in \overline{\Omega}, \text{ and all } \boldsymbol{v} \in S_{n-1}, \quad \exists \boldsymbol{a} \in A \text{ s.t. } \boldsymbol{v} \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) = |\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})| \geqslant F_1 > 0.\end{aligned} \tag{B3}$$

We assume that the terminal event is a result of some Poisson process and the time till termination has an exponential probability distribution; i.e., $P(T \geqslant t) = e^{-\lambda t}$. Thus, starting from $\boldsymbol{x}$, the cost of any individual control $\boldsymbol{a}(\cdot) \in \mathfrak{A}_0(\boldsymbol{x})$ is

$$J(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \int_0^{+\infty} \lambda e^{-\lambda t} \left[\int_0^t K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big)\, ds + q\big(\boldsymbol{y}(t)\big) \right] dt \tag{3.7}$$

$$= \int_0^{+\infty} e^{-\lambda s} \left[K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big) + \lambda q\big(\boldsymbol{y}(s)\big) \right] ds \tag{3.8}$$

where the second equality follows from the Fubini's Theorem. This shows that any randomly-terminated problem can be converted to a discounted infinite-horizon problem with $\beta = \lambda$ and $\tilde{K}(\boldsymbol{x}, \boldsymbol{a}) = K(\boldsymbol{x}, \boldsymbol{a}) + \lambda q(\boldsymbol{x})$. This relationship is well-known and has been previously used in [12] to treat even more general random-termination problems (e.g., with $\lambda = \lambda(\boldsymbol{x})$). The value function can then be defined as usual $v(\boldsymbol{x}) = \inf_{\boldsymbol{a}(\cdot) \in \mathfrak{A}_0(\boldsymbol{x})} J(\boldsymbol{x}, \boldsymbol{a}(\cdot))$.

REMARK 3.1 In view of (3.8), adding any constants C_1 and C_2 to functions K and q respectively, will result in adding $C_1/\lambda + C_2$ to the value function v. Thus, the above assumptions about the non-negativity of q_1 and K_1 can be made without any loss of generality.

Assuming $\boldsymbol{f}$, K and q are Lipschitz, the value function v is a $\overline{\Omega}$-*constrained viscosity solution* [4] of a Hamilton–Jacobi PDE:

$$\min_{\boldsymbol{a} \in A} \left\{ \nabla v(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) + K(\boldsymbol{x}, \boldsymbol{a}) \right\} + \lambda\big(q(\boldsymbol{x}) - v(\boldsymbol{x})\big) = 0. \tag{3.9}$$

Before using the fact that $v(\boldsymbol{x})$ is the viscosity solution of the corresponding variational inequality, we first prove a number of properties of the value function directly.

A standard treatment of state-constrained infinite-horizon problems shows that, for sufficiently regular costs, dynamics, and $\partial\Omega$, $v(\boldsymbol{x})$ is bounded uniformly continuous. Using an additional controllability assumption B3 we provide a direct proof of local Lipschitz-continuity of $v(\boldsymbol{x})$.

LEMMA 3.2 Assume B1–B3. Then the value function of the randomly-terminated problem is locally Lipschitz-continuous on Ω with the Lipschitz constant $L_v = (K_2 + \lambda q_2)/F_1$.

Proof. Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \Omega$ are such that the straight line segment connecting these two points lies inside Ω. Suppose $\boldsymbol{a}(\cdot)$ is a control and $\boldsymbol{y}(\cdot)$ is the corresponding trajectory such that we follow that straight line from $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$ and then switch to the optimal control of $\boldsymbol{x}_2$:

$$\boldsymbol{y}(0) = \boldsymbol{x}_1; \qquad \boldsymbol{y}(\tau) = \boldsymbol{x}_2; \qquad J\big(\boldsymbol{x}_2, \boldsymbol{a}(\cdot + \tau)\big) = v(\boldsymbol{x}_2);$$

$$\boldsymbol{y}'(t) \cdot \frac{\boldsymbol{x}_2 - \boldsymbol{x}_1}{|\boldsymbol{x}_2 - \boldsymbol{x}_1|} = |\boldsymbol{y}'(t)| = \big|\boldsymbol{f}\big(\boldsymbol{y}(t), \boldsymbol{a}(t)\big)\big|, \quad \text{for } \forall t \in [0, \tau).$$

The existence of such $\boldsymbol{a}(\cdot)$ and the fact that $\tau \leqslant \frac{|\boldsymbol{x}_2 - \boldsymbol{x}_1|}{F_1}$ follow from the controllability assumption

B3. Then, from the optimality principle,

$$v(\boldsymbol{x}_1) \leqslant J(\boldsymbol{x}_1, \boldsymbol{a}(\cdot)) = \int_0^\tau e^{-\lambda s}\left[K(\boldsymbol{y}(s), \boldsymbol{a}(s)) + \lambda q(\boldsymbol{y}(s))\right] ds + e^{-\lambda\tau} v(\boldsymbol{x}_2)$$
$$\leqslant \tau(K_2 + \lambda q_2) + v(\boldsymbol{x}_2) \;\leqslant\; \frac{|\boldsymbol{x}_2 - \boldsymbol{x}_1|}{F_1}(K_2 + \lambda q_2) + v(\boldsymbol{x}_2) = |\boldsymbol{x}_2 - \boldsymbol{x}_1| L_v + v(\boldsymbol{x}_2).$$

To complete the proof, the above argument can be repeated reversing the roles of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. We note that this proof does not use any additional assumptions on the regularity of K or q. □

We now list three essential assumptions that provide a continuous equivalent of A1–A3, which defined the discrete randomly-terminated problem in Section 2.2. We assume the existence of a special control $\boldsymbol{a}_0 \in A$ such that

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}_0) = 0, \quad \forall \boldsymbol{x} \in \overline{\Omega}. \tag{A1′}$$
$$K(\boldsymbol{x}, \boldsymbol{a}_0) = 0, \quad \forall \boldsymbol{x} \in \overline{\Omega}. \tag{A2′}$$

We will also use $\hat{A} = A \backslash \{\boldsymbol{a}_0\}$ for the set of all other control values.

$$\overline{K}_1 = \inf_{\boldsymbol{x}\in\Omega, \boldsymbol{a}\in\hat{A}} K(\boldsymbol{x}, \boldsymbol{a}) \geqslant 0. \tag{A3′}$$

Assumptions A1′–A3′ allow us to restate (3.9) as an *optimal stopping problem*

$$\max\left\{ v(\boldsymbol{x}) - q(\boldsymbol{x}), \quad \lambda\big(v(\boldsymbol{x}) - q(\boldsymbol{x})\big) - \min_{\boldsymbol{a}\in\hat{A}}\left\{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla v(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\right\} \right\} = 0; \tag{3.10}$$

or, in the isotropic case (when $A = S_{n-1} \cup \{\boldsymbol{a}_0\}$, and $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}) = f(\boldsymbol{x})\boldsymbol{a}$, $K(\boldsymbol{x}, \boldsymbol{a}) = K(\boldsymbol{x})$, $\forall \boldsymbol{a} \neq \boldsymbol{a}_0$),

$$\max\left\{ v(\boldsymbol{x}) - q(\boldsymbol{x}),\ \lambda\left(v(\boldsymbol{x}) - q(\boldsymbol{x})\right) - K(\boldsymbol{x}) + |\nabla v(\boldsymbol{x})| f(\boldsymbol{x}) \right\} = 0. \tag{3.11}$$

Using the notation $[z]^- = \min(z, 0)$, we can rewrite this variational inequality as follows:

$$v(\boldsymbol{x}) = q(\boldsymbol{x}) + \frac{1}{\lambda}\Big[\min_{\boldsymbol{a}\in\hat{A}}\left\{K(\boldsymbol{x}, \boldsymbol{a}) + \nabla v(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a})\right\}\Big]^-, \tag{3.12}$$

or, in the isotropic case,

$$v(\boldsymbol{x}) = q(\boldsymbol{x}) + \frac{1}{\lambda}\big[K(\boldsymbol{x}) - f(\boldsymbol{x})|\nabla v(\boldsymbol{x})|\big]^-. \tag{3.13}$$

The PDE holds (with the $[\cdot]^-$ omitted) on the domain $\overline{\Omega}\backslash M$. In Section 3.3 we discuss the discretized version of the isotropic variational inequality (3.13) and a label-setting algorithm for solving it efficiently. We note that several properties of the value function (e.g., Part 5 of Theorem 3.3) can also be obtained directly from the characteristic ODEs of (3.12) using stronger regularity assumptions on the dynamics and cost. However, our proofs provided below are more general. Examples studied in Sections 4.3 and 4.4 use discontinuous f and K to test our numerical method; additional examples (using discontinuous q) were omitted to save space. A detailed theoretical discussion of viscosity solutions to HJB PDEs with discontinuous Lagrangian can be found in [33].

The rest of this section is based on direct applications of the control-theoretic interpretation rather than on the theory of viscosity solutions. We will refer to a control $\boldsymbol{a}(\cdot)$ as *eventually motionless* (or *e.m.*) if $\exists T$ s.t. $\boldsymbol{a}(t) = \boldsymbol{a}_0,\ \forall t \geqslant T$. We will say that the corresponding e.m. trajectory $\boldsymbol{y}(\cdot)$ becomes motionless at the point $\boldsymbol{y}(T)$. Given a fixed control $\boldsymbol{a}(\cdot)$, we will further use a notation

$$\omega(t) = \int_0^t K\big(\boldsymbol{y}(s), \boldsymbol{a}(s)\big)\, ds + q\big(\boldsymbol{y}(t)\big),$$

for the deterministic cost of it if the termination happens at the time t. We note that, for an e.m. control, formula (3.7) can be rewritten as

$$J(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \int_0^T \lambda e^{-\lambda t}\omega(t)\, dt \ + \ e^{-\lambda T}\omega(T). \tag{3.14}$$

We will also define a "motionless" set $M = \{\boldsymbol{x} \mid v(\boldsymbol{x}) = J(\boldsymbol{x}, \boldsymbol{a}_0) = q(\boldsymbol{x})\}$. We note that, if the value function is continuous, then the set M is closed (since $(v - q)$ is upper semi-continuous).

The following two theorems list a number of properties of the value function and of optimal controls/trajectories.

THEOREM 3.3 Assume B1,B2, and A1′–A3′. Then, for all $\boldsymbol{x} \in \Omega$,

1. $v(\boldsymbol{x}) \in [q_1, q(\boldsymbol{x})]$.
2. $q(\boldsymbol{x}) = q_1$ implies $\boldsymbol{x} \in M$.
3. $\forall \boldsymbol{x} \in \Omega, \varepsilon > 0$, $\exists$ an ε-suboptimal e.m. control $\boldsymbol{a}^\varepsilon(\cdot)$.
4. If $\overline{K}_1 > 0$, then every optimal control is e.m.
5. If $\boldsymbol{y}^*(t)$ is an optimal trajectory, then $v(\boldsymbol{y}^*(t))$ is a monotone non-increasing function of t (and monotone decreasing up to its motionless point if $\overline{K}_1 > 0$).
6. If B3 holds, $\overline{K}_1 > 0$, and $\boldsymbol{y}^*(t)$ is an optimal trajectory for $\boldsymbol{x} = \boldsymbol{y}^*(0)$, then

$$|\boldsymbol{y}^*(t) - \boldsymbol{x}| \geqslant t\frac{\overline{K}_1 F_1}{K_2 + \lambda q_2},$$

   provided the straight line from $\boldsymbol{x}$ to $\boldsymbol{y}^*(t)$ lies within $\overline{\Omega}$.
7. Let $D(\boldsymbol{x}_1, \boldsymbol{x}_2)$ be the minimum distance from $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$ (minimized over all allowable trajectories in $\overline{\Omega}$) and $D = \sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \overline{\Omega}} D(\boldsymbol{x}_1, \boldsymbol{x}_2)$. If B3 also holds, then $v(\boldsymbol{x}) \leqslant q_1 + DL_v$.

Proof. 1. A1′, and A2′ ensure that $v(\boldsymbol{x}) \leqslant \int_0^{+\infty} e^{-\lambda s}\lambda q(\boldsymbol{x})\, ds = q(\boldsymbol{x})$, while A3′ shows that for every control $J(\boldsymbol{x}, \boldsymbol{a}(\cdot)) \geqslant \int_0^{+\infty} e^{-\lambda s}\lambda q(\boldsymbol{y}(s))\, ds \geqslant q_1$.

2. In particular, if $q_1 = q(\boldsymbol{x})$ then $v(\boldsymbol{x}) = q(\boldsymbol{x})$ and $\boldsymbol{a}^*(t) = \boldsymbol{a}_0$ is the optimal control.

3. Suppose $\boldsymbol{a}_1(\cdot)$ is a (possibly non-e.m.) $\frac{\varepsilon}{2}$-suboptimal control and $\boldsymbol{y}_1(\cdot)$ is the corresponding trajectory starting from $\boldsymbol{x}$; i.e.,

$$J\big(\boldsymbol{x}, \boldsymbol{a}_1(\cdot)\big) = \int_0^{+\infty} e^{-\lambda s}\Big[K\big(\boldsymbol{y}_1(s), \boldsymbol{a}_1(s)\big) + \lambda q(\boldsymbol{y}_1(s))\Big]\, ds \leqslant v(\boldsymbol{x}) + \frac{\varepsilon}{2}.$$

Define a control $\boldsymbol{a}_2(t) = \boldsymbol{a}_1(t)$ for $t < \tau$ and $\boldsymbol{a}_2(t) = \boldsymbol{a}_0$ after that. The corresponding trajectory

$\boldsymbol{y}_2$ is a truncated version of $\boldsymbol{y}_1$.

$$\begin{aligned} J\big(\boldsymbol{x},\boldsymbol{a}_2(\cdot)\big) &= \int_0^{\tau} e^{-\lambda s}\Big[K\big(\boldsymbol{y}_1(s),\boldsymbol{a}_1(s)\big)+\lambda q\big(\boldsymbol{y}_1(s)\big)\Big]\,ds + \int_{\tau}^{+\infty} e^{-\lambda s}\lambda q\big(\boldsymbol{y}_1(\tau)\big)\,ds \\ &= \int_0^{+\infty} e^{-\lambda s}\Big[K\big(\boldsymbol{y}_1(s),\boldsymbol{a}_1(s)\big)+\lambda q\big(\boldsymbol{y}_1(s)\big)\Big]\,ds \\ &\qquad + \int_{\tau}^{+\infty} e^{-\lambda s}\Big[\lambda q\big(\boldsymbol{y}_1(\tau)\big) - K\big(\boldsymbol{y}_1(s),\boldsymbol{a}_1(s)\big) - \lambda q\big(\boldsymbol{y}_1(s)\big)\Big]\,ds \\ &\leqslant J\big(\boldsymbol{x},\boldsymbol{a}_1(\cdot)\big) + \Big[q_2 - q_1 - \frac{K_1}{\lambda}\Big]e^{-\lambda\tau}. \end{aligned}$$

Thus, for all sufficiently large τ,

$$J\big(\boldsymbol{x},\boldsymbol{a}_2(\cdot)\big) \leqslant J\big(\boldsymbol{x},\boldsymbol{a}_1(\cdot)\big) + \frac{\varepsilon}{2} \leqslant v(\boldsymbol{x}) + \varepsilon; \qquad \text{i.e., } \boldsymbol{a}_2(\cdot) \text{ is } \varepsilon\text{-suboptimal.}$$

4. Suppose $\boldsymbol{a}^*(t)$ is a non-e.m. optimal control starting from $\boldsymbol{x}$ and $\boldsymbol{y}^*(t)$ is the corresponding trajectory. Let $\tilde{q}_1 = \inf\limits_{t>0} q(\boldsymbol{y}^*(t))$. Then $\exists\tau > 0$ such that $q(\boldsymbol{y}^*(\tau)) - \tilde{q}_1 < \overline{K}_1/\lambda$. Then

$$\int_{\tau}^{+\infty} e^{-\lambda s}\Big[K\big(\boldsymbol{y}^*(s),\boldsymbol{a}^*(s)\big)+\lambda q(\boldsymbol{y}^*(s))\Big]\,ds \geqslant e^{-\lambda\tau}\Big[\overline{K}_1/\lambda + \tilde{q}_1\Big] > e^{-\lambda\tau} q\big(\boldsymbol{y}^*(\tau)\big);$$

i.e., making that trajectory motionless at $\boldsymbol{y}^*(\tau)$ would result in a lower cost, contradicting the optimality of $\boldsymbol{a}^*(t)$.

5. Suppose $\boldsymbol{a}^*(\cdot)$ is an optimal control starting from $\boldsymbol{x}$, $\boldsymbol{y}^*(t)$ is the corresponding optimal trajectory, and define $z(t) = v(\boldsymbol{y}^*(t))$. By the optimality principle,

$$z(0) = \int_0^t e^{-\lambda s}\Big[K\big(\boldsymbol{y}^*(s),\boldsymbol{a}^*(s)\big)+\lambda q\big(\boldsymbol{y}^*(s)\big)\Big]\,ds + e^{-\lambda t} z(t).$$

By Lebesgue's differentiation theorem, $z(t)$ is differentiable for almost all $t > 0$ and

$$0 = e^{-\lambda t}\left[K(\boldsymbol{y}^*(t),\boldsymbol{a}^*(t)) + \lambda q(\boldsymbol{y}^*(t))\right] + e^{-\lambda t} z'(t) - \lambda e^{-\lambda t} z(t);$$

Hence, $z'(t) = \lambda\left[z(t) - q(\boldsymbol{y}^*(t))\right] - K(\boldsymbol{y}^*(t),\boldsymbol{a}^*(t))$ for almost all $t > 0$. Since $z(t) = v(\boldsymbol{y}^*(t)) \leqslant q(\boldsymbol{y}^*(t))$, we have $z'(t) \leqslant 0$; moreover, this inequality becomes strict if $\overline{K}_1 > 0$ and $\boldsymbol{a}^*(t) \neq \boldsymbol{a}_0$.

We note that a similar argument combined with the optimality principle shows that, even for a non-optimal control $\boldsymbol{a}(\cdot)$, the value function along the corresponding trajectory $z(t) = v(\boldsymbol{y}(t))$ will satisfy $z'(t) \geqslant \lambda\left[z(t) - q(\boldsymbol{y}(t))\right] - K(\boldsymbol{y}(t),\boldsymbol{a}(t))$ for almost every $t > 0$.

6. As shown above, $z'(t) \leqslant -\overline{K}_1$; so, $v(\boldsymbol{x}) - v(\boldsymbol{y}^*(t)) \geqslant t\overline{K}_1$. On the other hand, by Lemma 3.2, $v(\boldsymbol{x}) - v\big(\boldsymbol{y}^*(t)\big) = \big|v(\boldsymbol{x}) - v\big(\boldsymbol{y}^*(t)\big)\big| \leqslant L_v\big|\boldsymbol{x} - \boldsymbol{y}^*(t)\big|$. Thus, $|\boldsymbol{y}^*(t) - \boldsymbol{x}| \geqslant t\,\frac{\overline{K}_1 F_1}{K_2 + \lambda q_2}$.

7. Since v is locally Lipschitz by Lemma 3.2, this bound is obtained by following the trajectory from $\boldsymbol{x}$ to $\boldsymbol{x}_0^* = \operatorname{argmin}\limits_{\boldsymbol{x}_0 \in \overline{\Omega}} q(\boldsymbol{x}_0)$.

□

In the rest of this section we will use superscripts to indicate the dependence on λ of the expected control cost $J^\lambda(\boldsymbol{x}, \boldsymbol{a}(\cdot))$, the value function $v^\lambda(\boldsymbol{x})$ and the motionless set $M^\lambda = \{\boldsymbol{x} \mid v^\lambda(\boldsymbol{x}) = q(\boldsymbol{x})\}$. Wherever this superscript is omitted, the properties hold for any fixed $\lambda > 0$. We also introduce v^0 as the solution of a variational inequality:

$$\max\left\{ v^0(\boldsymbol{x}) - q(\boldsymbol{x}), -\min_{\boldsymbol{a}\in A}\left\{K(\boldsymbol{x},\boldsymbol{a}) + \nabla v^0(\boldsymbol{x})\cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\right\}\right\} = 0. \tag{3.15}$$

We see that (3.15) is a variant of (3.6) with $\beta = 0$. Intuitively, v^0 can be interpreted as the value function for the limiting case, in which the termination is so unlikely that we are guaranteed to reach any $\boldsymbol{x} \in \overline{\Omega}$ of our choice before it occurs. Due to A2′, A3′ we can instead minimize over $\hat{A}$ without changing v^0.

Let $D^{\boldsymbol{a}}_+ q(\boldsymbol{x})$ and $\mathcal{H}^{\boldsymbol{a}}_+ q(\boldsymbol{x})$ be respectively the lower Dini and the lower Hadamard derivatives of q along the vector $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})$; i.e.,

$$D^{\boldsymbol{a}}_+ q(\boldsymbol{x}) = \liminf_{t\to 0^+} \frac{q\big(\boldsymbol{x} + t\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\big) - q(\boldsymbol{x})}{t}; \quad H^{\boldsymbol{a}}_+ q(\boldsymbol{x}) = \liminf_{\substack{t\to 0^+ \\ \boldsymbol{b}\to \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})}} \frac{q(\boldsymbol{x}+t\boldsymbol{b}) - q(\boldsymbol{x})}{t};$$

if q is differentiable, then $D^{\boldsymbol{a}}_+ q(\boldsymbol{x}) = H^{\boldsymbol{a}}_+ q(\boldsymbol{x}) = \nabla q(\boldsymbol{x})\cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})$. We also define two sets

$$M^\infty = \left\{\boldsymbol{x} \mid \inf_{\boldsymbol{a}\in\hat{A}}\left\{K(\boldsymbol{x},\boldsymbol{a}) + D^{\boldsymbol{a}}_+ q(\boldsymbol{x})\right\} \geqslant 0\right\} \text{ and } M^\infty_0 = \left\{\boldsymbol{x} \mid \min_{\boldsymbol{a}\in\hat{A}}\left\{K(\boldsymbol{x},\boldsymbol{a}) + H^{\boldsymbol{a}}_+ q(\boldsymbol{x})\right\} > 0\right\}.$$

The minimum in the above definition of M^∞_0 is attained since the lower Hadamard derivatives are lower semi-continuous functions of the direction. Moreover, it is easy to show that for every $\boldsymbol{x} \in M^\infty_0$ there exists $\epsilon > 0$ and $\tau > 0$ such that for all $t \in (0,\tau]$ and all $\boldsymbol{a} \in \hat{A}$

$$K(\boldsymbol{x},\boldsymbol{a}) + \frac{q\big(\boldsymbol{x} + t\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\big) - q(\boldsymbol{x})}{t} \geqslant \epsilon > 0. \tag{3.16}$$

REMARK 3.4 The set $\Omega\backslash M^\infty$ consists of points, starting from which it is never optimal to stay in place regardless of how high λ is. However, the points in $M^\infty\backslash M^\infty_0$ may also remain outside of M^λ for any finite λ. E.g., if $K \equiv 0$ and q is smooth, then every critical point of q will be in M^∞, including all strict local maxima of q, starting from which it is clearly better to move regardless of λ, and all strict local minima of q, which become motionless when λ is sufficiently large. Below we show that all point in M^∞_0 also have the latter property, at least for isotropic cost/dynamics with $\overline{K}_1 > 0$. Interestingly, the isotropy and $\overline{K}_1 > 0$ also imply that all strict local maxima of a smooth q will also lie in M^∞_0. This might seem somewhat counterintuitive, but there is no contradiction: for large λ, the process will likely terminate close to the starting point, and the net decrease in q will be insufficient to compensate for the accumulated running cost.

To simplify the notation, we will refer to a control $\boldsymbol{a}^*(t)$ (and the corresponding trajectory $\boldsymbol{y}^*(t)$) as λ-*optimal* if $J^\lambda(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)) = v^\lambda(\boldsymbol{x})$.

THEOREM 3.5 Bounds and asymptotic behavior of v^λ and M^λ assuming B1, B2, and A1′–A3′. Suppose that an e.m. λ-optimal control exists for every starting location $\boldsymbol{x} \in \Omega$ and $\forall \lambda \geqslant 0$.

1. If $\boldsymbol{y}^*(t)$ is an optimal trajectory starting from $\boldsymbol{x}$ and becoming motionless at some $\boldsymbol{x}_0 = \boldsymbol{y}^*(t_0)$, then $\omega(t_0) \leqslant \omega(t)$ and $q(\boldsymbol{x}_0) \leqslant q(\boldsymbol{y}^*(t))$ for all $t \in [0, t_0)$ (and the second inequality is strict if $\overline{K}_1 > 0$).

2. $v^\lambda(\boldsymbol{x}) \in [v^0(\boldsymbol{x}),\, q(\boldsymbol{x})]$ for all $\lambda > 0$ and all $\boldsymbol{x} \in \overline{\Omega}$.
3. $0 \leqslant \lambda_1 \leqslant \lambda_2 \quad \Longrightarrow \quad v^{\lambda_1}(\boldsymbol{x}) \leqslant v^{\lambda_2}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \overline{\Omega}$.
4. $v^\lambda \to q$ pointwise as $\lambda \to +\infty$.
5. $v^\lambda \to v^0$ pointwise as $\lambda \to 0$.
6. If $\lambda_1 \leqslant \lambda_2$, then $M^{\lambda_1} \subset M^{\lambda_2}$. In particular, $M^0 \subset M^\lambda \subset M^\infty$, for all $\lambda > 0$.
7. Suppose the cost and dynamics are isotropic as in equation (3.11) and $\overline{K}_1 > 0$. Then, for every $\boldsymbol{x} \in M_0^\infty$, there exists a sufficiently large $\lambda > 0$, such that $\boldsymbol{x} \in M^\lambda$.
8. For every $\boldsymbol{x} \notin M^0$, there exists a sufficiently small $\lambda > 0$, such that $\boldsymbol{x} \notin M^\lambda$.
9. We will denote the sets of local minima, strict local minima and global minima of q in $\overline{\Omega}$ as Q_l, Q_l^s and Q_g respectively. Then $Q_g \subset M^0$, and $Q_l \subset M^\infty$. Moreover, if B3 also holds and $K \equiv 0$, then $M_0^\infty \subset Q_l^s$.

Proof. 1. Suppose $t_1 = \operatorname{argmin}_{t\in[0,t_0]} \omega(t)$ and $\boldsymbol{x}_1 = \boldsymbol{y}^*(t_1)$. If $\omega(t_0) > \omega(t_1)$, then formula (3.7) shows that the same trajectory but made motionless earlier (at $\boldsymbol{x}_1$) would have a lower cost, contradicting the optimality of $\boldsymbol{y}^*(\cdot)$. Thus, $\omega(t_0) \leqslant \omega(t)$ for all $t \in [0, t_0)$. As a result,

$$\omega(t) = \int_0^t K\big(\boldsymbol{y}^*(s), \boldsymbol{a}^*(s)\big)\, ds + q\big(\boldsymbol{y}^*(t)\big) \geqslant \omega(t_0) \geqslant \int_0^t K\big(\boldsymbol{y}^*(s), \boldsymbol{a}^*(s)\big)\, ds + q(\boldsymbol{x}_0),$$

where the last inequality uses A3′ and becomes strict if $\overline{K}_1 > 0$. Hence, $q(\boldsymbol{x}_0) \leqslant q(\boldsymbol{y}^*(t))$.
We emphasize that the above argument *does not* imply that $\omega(t)$ is monotone decreasing along optimal trajectories. In fact, for $K = 0$, it is easy to see that any optimal trajectory starting from a local minimum of q (but outside M) would provide a counter-example.

2. Suppose that $\boldsymbol{a}^*(t)$ is a λ-optimal control starting from $\boldsymbol{x}$ and becoming motionless at the time t_0. From the formula (3.14) and using Part 1 of the current theorem, $v^\lambda(\boldsymbol{x}) = J^\lambda(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)) = \int_0^{t_0} \lambda e^{-\lambda t}\omega(t)\, dt \;+\; e^{-\lambda t_0}\omega(t_0) \;\geqslant\; \omega(t_0) \;\geqslant\; v^0(\boldsymbol{x})$, where the last inequality reflects the interpretation of v^0 as the value function of the deterministic optimal stopping problem.

3. Suppose that $\boldsymbol{a}(t)$ is an e.m. λ_2-optimal control and $\boldsymbol{y}(t)$ is the corresponding trajectory starting from $\boldsymbol{x}$. Define $z_2(t) = v^{\lambda_2}(\boldsymbol{y}(t))$ and $z_1(t) = v^{\lambda_1}(\boldsymbol{y}(t))$. Since this trajectory is λ_2-optimal, Part 5 of Theorem 3.3 shows that, for almost all t,

$$z_2'(t) = \lambda_2\big[z_2(t) - q\big(\boldsymbol{y}(t)\big)\big] - K\big(\boldsymbol{y}(t), \boldsymbol{a}(t)\big) \;\leqslant\; \lambda_1\big[z_2(t) - q\big(\boldsymbol{y}(t)\big)\big] - K\big(\boldsymbol{y}(t), \boldsymbol{a}(t)\big),$$

where the inequality uses the fact that $z_2(t) \leqslant q(\boldsymbol{y}(t))$. Since the same trajectory is not necessarily λ_1-optimal, we have

$$z_1'(t) \;\geqslant\; \lambda_1\big[z_1(t) - q\big(\boldsymbol{y}(t)\big)\big] - K\big(\boldsymbol{y}(t), \boldsymbol{a}(t)\big).$$

Subtracting these inequalities, and defining $\theta(t) = z_2(t) - z_1(t)$, we see that $\theta(t)' \leqslant \lambda_1\theta(t)$. On the other hand, when the trajectory becomes motionless at the time $t_0 > 0$, we know that $v^{\lambda_2}(\boldsymbol{y}(t_0)) = q(\boldsymbol{y}(t_0)) \geqslant v^{\lambda_1}(\boldsymbol{y}(t_0))$; i.e., $\theta(t_0) \geqslant 0$. Taken together with the above differential inequality this implies $\theta(0) \geqslant 0$; i.e., $v^{\lambda_2}(\boldsymbol{x}) \geqslant v^{\lambda_1}(\boldsymbol{x})$.

4. Note that $\omega(t)$ is lower semicontinuous for every control $\boldsymbol{a}(\cdot)$. Since q is lower semicontinuous, for every $\boldsymbol{x} \in \Omega$ and every $\epsilon > 0$ there exists δ such that $|\boldsymbol{x} - \tilde{\boldsymbol{x}}| < \delta \Longrightarrow q(\tilde{\boldsymbol{x}}) > q(\boldsymbol{x}) - \epsilon$. Choosing $\tau < \delta/\|\boldsymbol{f}\|_\infty$, we can bound from below the cost of every control

$$J^\lambda\big(\boldsymbol{x}, \boldsymbol{a}(\cdot)\big) \geqslant \int_0^\tau \omega(s)\lambda e^{-\lambda s}\, ds \geqslant \int_0^\tau q\big(\boldsymbol{y}(s)\big)\lambda e^{-\lambda s}\, ds \geqslant \big(q(\boldsymbol{x}) - \epsilon\big)\big(1 - e^{-\lambda\tau}\big).$$

Thus, $q(\boldsymbol{x}) \geqslant v^\lambda(\boldsymbol{x}) \geqslant (q(\boldsymbol{x}) - 2\epsilon)$ for all sufficiently large λ's. Since this argument works for any $\epsilon > 0$, $v^\lambda(\boldsymbol{x}) \to q(\boldsymbol{x})$ as $\lambda \to +\infty$.

5. Suppose a 0-optimal control $\boldsymbol{a}(\cdot)$ becomes motionless at some time T; i.e., $v^0(\boldsymbol{x}) = J^0(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \int_0^T K(\boldsymbol{y}(s), \boldsymbol{a}(s))\, ds + q(\boldsymbol{y}(T))$. Then, from formula (3.14), $\lim_{\lambda \to 0} J^\lambda(\boldsymbol{x}, \boldsymbol{a}(\cdot)) = \omega(T) = v^0(\boldsymbol{x})$.

6. If $\boldsymbol{x} \in M^{\lambda_1}$, then $q(\boldsymbol{x}) = v^{\lambda_1}(\boldsymbol{x}) \leqslant v^{\lambda_2}(\boldsymbol{x}) \leqslant q(\boldsymbol{x})$. Thus, $\boldsymbol{x} \in M^{\lambda_2}$.

7. Suppose $\boldsymbol{x} \in M_0^\infty$ is fixed. To prove that $\boldsymbol{x}$ becomes motionless for some finite λ, it is not enough to show that for every fixed control $\boldsymbol{a}(\cdot)$ we can choose large enough λ to guarantee $J^\lambda(\boldsymbol{x}, \boldsymbol{a}(\cdot)) \geqslant q(\boldsymbol{x})$. We need to show that some finite λ guarantees this inequality for *all* $\boldsymbol{a}(\cdot)$. For isotropic dynamics and cost, the fact that $\boldsymbol{x} \in M_0^\infty$ guarantees that $\exists \delta > 0$ such that $|\boldsymbol{y} - \boldsymbol{x}| \leqslant \delta$ implies

$$K(\boldsymbol{x}) + \frac{q(\boldsymbol{y}) - q(\boldsymbol{x})}{t_{\boldsymbol{y}}} \geqslant \epsilon > 0,$$

where $t_{\boldsymbol{y}}$ denotes the minimum time needed to reach from $\boldsymbol{x}$ to $\boldsymbol{y}$. Without loss of generality, we can assume that $\epsilon < \overline{K}_1$. Let $\boldsymbol{a}(\cdot)$ be an arbitrary control with the corresponding trajectory $\boldsymbol{y}(t)$ starting from $\boldsymbol{x}$. We also choose a small enough τ to ensure that $|\boldsymbol{y}(t) - \boldsymbol{x}| \leqslant \delta$ and $K(\boldsymbol{y}(t)) \geqslant K(\boldsymbol{x}) - \epsilon/2$ for all $t \leqslant \tau$. Then

$$J^\lambda\big(\boldsymbol{x}, \boldsymbol{a}(\cdot)\big) \geqslant \int_0^\tau \omega(s)\lambda e^{-\lambda s}\, ds \geqslant \int_0^\tau \Big[\big(K(\boldsymbol{x}) - \frac{\epsilon}{2}\big)s + q\big(\boldsymbol{y}(s)\big)\Big]\lambda e^{-\lambda s}\, ds.$$

Using the fact that $t_{\boldsymbol{y}(s)} \leqslant s$, we note that

$$q(\boldsymbol{y}(s)) \geqslant q(\boldsymbol{x}) + t_{\boldsymbol{y}(s)}\big(\epsilon - K(\boldsymbol{x})\big) \geqslant q(\boldsymbol{x}) + s\big(\epsilon - K(\boldsymbol{x})\big).$$

Combining this with the above,

$$\begin{aligned} J^\lambda\big(\boldsymbol{x}, \boldsymbol{a}(\cdot)\big) &\geqslant \int_0^\tau \Big[\frac{s\epsilon}{2} + q(\boldsymbol{x})\Big]\lambda e^{-\lambda s}\, ds = (1 - e^{-\lambda\tau})q(\boldsymbol{x}) + \frac{\epsilon}{2\lambda}\Big[1 - e^{-\lambda\tau} - \lambda\tau e^{-\lambda\tau}\Big] \\ &= q(x) + \frac{\epsilon}{2\lambda}\Big[1 - e^{-\lambda\tau} - \lambda\tau e^{-\lambda\tau} - 2\lambda\epsilon^{-1}q(\boldsymbol{x})e^{-\lambda\tau}\Big]. \end{aligned}$$

To complete the proof, we note that, for large enough λ, the expression in the last square brackets is strictly positive, and this inequality holds for all controls $\boldsymbol{a}(\cdot)$.

8. Follows from Part 5.

9. Suppose $\boldsymbol{x} \in Q_g$ and $\boldsymbol{a}(\cdot)$ is its 0-optimal control that becomes motionless at some time T. Then $v^0(\boldsymbol{x}) = \omega(T) \geqslant q(\boldsymbol{y}(T)) \geqslant q(\boldsymbol{x})$; hence, $\boldsymbol{x} \in M^0$. If $\boldsymbol{x} \in Q_l$, then $D_+^{\boldsymbol{a}} q(\boldsymbol{x}) \geqslant 0, \forall \boldsymbol{a} \in A$ and $\boldsymbol{x} \in M^\infty$. Finally, if $K \equiv 0$, then B3 implies that all lower Hadamard derivatives of q are positive at every $\boldsymbol{x} \in M_0^\infty$, making x a strict minimum.

□

We note that the conditions in Theorem 3.5 can be further relaxed by rewriting the proofs in terms of ϵ-suboptimal e.m. trajectories (see Part 3 of Theorem 3.3).

3.3 *Upwind discretization and a modified Fast Marching Method*

We consider a first-order upwind finite differences discretization of the isotropic variational inequality (3.13) and introduce a modified version of Fast Marching Method applicable to it. We

note that Ordered Upwind Methods [2, 31, 32] can be similarly modified to handle anisotropic randomly-terminated problems.

We assume that (3.13) is discretized on a uniform Cartesian grid[2] with M gridpoints and the value function $v(x, y)$ is approximated by a grid-function V:

$$\boldsymbol{x}_{i,j} = (x_i, y_j); \quad x_{i\pm1} = x_i \pm h; \quad y_{j\pm1} = y_j \pm h; \quad v(x_i, y_j) \approx V(x_i, y_j) = V_{i,j}.$$

We also define the set of neighboring gridpoints and the set of neighboring values

$$N(\boldsymbol{x}_{i,j}) = N_{i,j} = \{\boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i,j+1}, \boldsymbol{x}_{i-1,j}, \boldsymbol{x}_{i,j-1}\}; \quad NV_{i,j} = \{V_{i+1,j}, V_{i,j+1}, V_{i-1,j}, V_{i,j-1}\}.$$

Our discretization uses two one-sided, first-order accurate approximations for each partial derivative; i.e.,

$$v_x(x_i, y_j) \approx D_{ij}^{\pm x} V = \frac{V_{i\pm1,j} - V_{i,j}}{\pm h}; \quad v_y(x_i, y_j) \approx D_{ij}^{\pm y} V = \frac{V_{i,j\pm1} - V_{i,j}}{\pm h}.$$

An *upwind discretization* of (3.13) at a gridpoint (x_i, y_j) is obtained as follows:

$$V_{i,j} = q_{i,j} + \frac{1}{\lambda}\left[K_{i,j} - f_{i,j}\sqrt{\left(\max\left(D_{ij}^{-x}V, -D_{ij}^{+x}V, 0\right)\right)^2 + \left(\max\left(D_{ij}^{-y}V, -D_{ij}^{+y}V, 0\right)\right)^2}\right]^{-}. \quad (3.17)$$

If all the $NV_{i,j}$ values are already known, (3.17) has to be solved to obtain $V_{i,j}$. The latter task is significantly simplified once we realize that the equation can be solved on a quadrant-by-quadrant basis. This procedure is described in detail in Appendix B (Section 5) and there we also prove that the finite-difference discretization (3.17) can be obtained from the Kuhn–Tucker optimality conditions for a suitable semi-Lagrangian discretization.

However, the values in $NV_{i,j}$ are not a priori known and equation (3.17) has to hold at every gridpoint, resulting in a system of M coupled non-linear equations. This system can be solved iteratively, but that approach is unnecessarily inefficient. Indeed, (3.17) is related to the upwind scheme used by Rouy and Tourin [28] for the Eikonal equation, which Sethian later showed to possess causal properties, yielding the non-iterative Fast Marching Method [29]. Below we provide an extension of that method to our obstacle problem.

REMARK 3.6 If V satisfies equation (3.17), it is easy to show that
1. $V_{i,j} \leqslant q_{i,j}$; i.e., the discretized version is also an obstacle problem.
2. $V_{i,j}$ is a non-decreasing function of all values in $NV_{i,j}$. (This monotonicity, along with the consistency of the discretization can be used to show the convergence of V to v as $h \to 0$; see [5, 25].)
3. Suppose $\hat{V} \in NV_{i,j}$. Then, either $\hat{V} < V_{i,j}$ or any increase in $\hat{V}$ will not affect $V_{i,j}$. In other words, the value of $V_{i,j}$ depends only on its smaller neighbors, which makes the label-setting method summarized in Algorithm 2 applicable.

[2] For the sake of notational simplicity, we describe the numerical method in R^2; the generalization for $n > 2$ is straightforward.

As with Dijkstra's method, the values V at all gridpoints are at first temporary, equation (3.17) is used to update these temporary values, and the values become permanent when the corresponding gridpoints are *Accepted*. Due to causality of this discretization, by the time the algorithm terminates, (3.17) holds at all gridpoints. As in the original Fast Marching Method for the Eikonal equation, the computational cost of our algorithm is $O(M \log M)$, where the $\log M$ term stems from implementing a sorted list of *Considered* gridpoints using a heap-sort data structure.

Algorithm 2 A modified Fast Marching Method for randomly-terminated isotropic problems.

```
start with all gridpoints marked as Far;
set V(x) := q(x) for all x ∈ X;
mark all local minima of q as Considered;
while (Considered list is not empty)  {
   let x̄ be s.t. V(x̄) is the smallest Considered value;
   mark x̄ Accepted;
   for each not-yet-Accepted x_{i,j} ∈ N(x̄)  {
      update V(x_{i,j});
      if x_{i,j} is Far, mark it Considered;
   }
}
```

A simple implementation of "update $V(\boldsymbol{x}_{i,j})$" is obtained by re-solving (3.17) using all current (possibly temporary) values in $NV_{i,j}$. A more efficient version, using the *Accepted* subset of $NV_{i,j}$ and taking advantage of the fact that only $V(\bar{\boldsymbol{x}})$ has recently changed, is described in Remark B.6 in Appendix B.

4. Numerical examples

4.1 *Convergence study: A trivial free boundary*

We start by studying convergence of our method on a simple randomly-terminated continuous example, where the free boundary is trivial and the analytic formula for the solution is available.

Suppose $\overline{\Omega} = [-2,2] \times [-2,2]$, $q(\boldsymbol{x}) = |\boldsymbol{x}|$, $K = 0$, and $f = 1$. Note that $K = 0$ and the fact that q has only one minimum imply that the free boundary is trivial; i.e., $M = B = \{\mathbf{0}\}$, the global minimum of q.

Since f, K, and q are radially symmetric, then so is the value function: for every $\boldsymbol{x} \neq \mathbf{0}$ it is optimal to move along the straight line toward the origin; i.e., the optimal control is $\boldsymbol{a}^*(t) = -\boldsymbol{x}/|\boldsymbol{x}|$. The expected cost of using this control (with the specified q and f and with any radially symmetric K) is

$$J\big(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)\big) = \int_0^{|\boldsymbol{x}|} e^{-\lambda s} \left[K\Big(\boldsymbol{x}\, \frac{|\boldsymbol{x}| - s}{|\boldsymbol{x}|}\Big) + \lambda\, (|\boldsymbol{x}| - s) \right] ds. \tag{4.1}$$

In particular, when $K = 0$,

$$v(\boldsymbol{x}) = J\big(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)\big) = \int_0^{|\boldsymbol{x}|} \lambda e^{-\lambda s} \big(|\boldsymbol{x}| - s\big)\, ds = |\boldsymbol{x}| - \frac{1}{\lambda}\big(1 - e^{-\lambda |\boldsymbol{x}|}\big).$$

TABLE 1. Errors for the trivial free boundary example with $\lambda = 0.5$

Grid points	1D L_∞ error	2D L_2 error	2D L_∞ error
101×101	0.0073	0.0062	0.0449
201×201	0.0037	0.0035	0.0259
401×401	0.0018	0.0020	0.0147
801×801	0.0009	0.0011	0.0083
1601×1601	0.0005	0.0006	0.0046

Table 1 lists the numerical errors observed in this computational example for $\lambda = 0.5$. The second column reports the maximum error observed on the horizontal gridline passing through the origin. (Since the characteristics are straight lines in this example, this is equivalent to conducting the same experiment in 1D on the domain $[-2, 2]$.) The remaining two columns report L_2 and L_∞ errors computed on the entire $\overline{\Omega}$; the L_2 errors are normalized to account for the non-unit area of $\overline{\Omega}$. The data clearly indicates the first order of convergence.

REMARK 4.1 We note that for an Eikonal equation with a point source, the rate of convergence is often found to be lower due to a local non-smoothness of the viscosity solution at that point source. The techniques to recover the first order of convergence include pre-initializing the true solution in a disk of fixed radius centered at the point source as well as the more recently proposed "singularity removal" method [19]. In the example considered above, we do not face similar issues simply because the leading term in the expansion of v is $|\boldsymbol{x}|^2$ rather than $|\boldsymbol{x}|$.

4.2 *Convergence study: A circular free boundary*

A small modification of the previous example already leads to a non-trivial free boundary. We let $K(\boldsymbol{x}) = |\boldsymbol{x}|$ and take the same f, q and $\overline{\Omega}$ as above.

As before, $v(\boldsymbol{x})$ is radially symmetric, and if $\boldsymbol{x} \notin M$, then the optimal trajectory starting from $\boldsymbol{x}$ is again $\boldsymbol{a}^*(t)$, whose expected cost is evaluated from the integral in (4.1):

$$J\left(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)\right) = \frac{\lambda+1}{\lambda}\left(|\boldsymbol{x}| - \frac{1}{\lambda}\left(1 - e^{-\lambda|\boldsymbol{x}|}\right)\right).$$

The value function is $v(\boldsymbol{x}) = \min\left(q(\boldsymbol{x}), J\left(\boldsymbol{x}, \boldsymbol{a}^*(\cdot)\right)\right)$, and, unlike in the previous example, $q(\boldsymbol{x}) = |\boldsymbol{x}|$ is actually smaller when $|\boldsymbol{x}|$ is sufficiently large. Thus, for this example, $B = \{\boldsymbol{0}\} \cup C(r)$, where $C(r) = \{\boldsymbol{x} \mid |\boldsymbol{x}| = r\}$ and the radius r is such that

$$\frac{\lambda+1}{\lambda}\left(r - \frac{1}{\lambda}\left(1 - e^{-\lambda r}\right)\right) = r. \tag{4.2}$$

The origin is the inflow part and $C(r)$ is the outflow part of B.

This example also illustrates the asymptotic behavior of B. Applying definitions of asymptotic motionless sets from Section 3.2,

$$M^0 = \{(0,0)\} \bigcup \left\{\boldsymbol{x} \;\middle|\; q(\boldsymbol{x}) \leqslant \int_0^{|\boldsymbol{x}|} K\left(\boldsymbol{x}\, \frac{|\boldsymbol{x}| - s}{|\boldsymbol{x}|}\right) ds\right\}$$

and

$$M^{\infty} = \{(0,0)\} \bigcup \{\boldsymbol{x} \mid K(\boldsymbol{x}) - |\nabla q(\boldsymbol{x})| \geqslant 0\}.$$

When $\lambda \to \infty$, Theorem 3.5 shows that $B_{out} \to B_{\infty} = \{\boldsymbol{x} \mid |\boldsymbol{x}| = K(\boldsymbol{x}) = |\nabla q(\boldsymbol{x})| = 1\} = C(1)$. On the other hand, when $\lambda \to 0$, we have $B_{out} \to B_0 = \{\boldsymbol{x} \mid |\boldsymbol{x}| = q(\boldsymbol{x}) = \int_0^{|\boldsymbol{x}|} (|\boldsymbol{x}| - s)\, ds\} = C(2)$. This is also confirmed in Figure 3 obtained by solving (4.2) numerically. Table 1 lists the numerical errors observed in this example for two different values of λ.

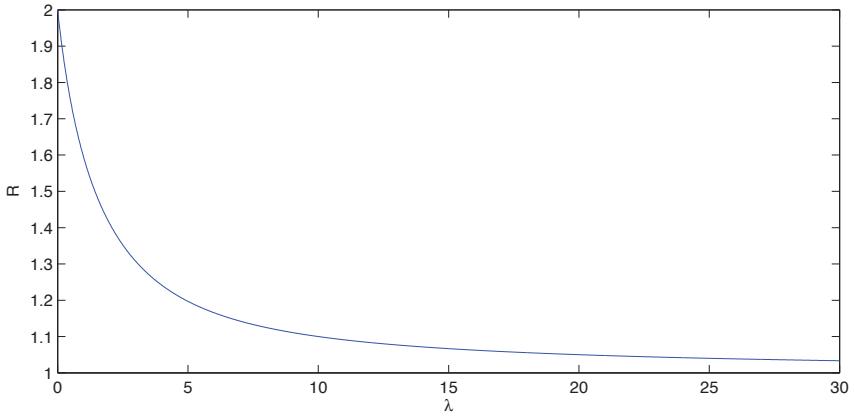


FIG. 3. Radius of the outflow free boundary for different values of λ

TABLE 2. Errors for the circular (outflow) free boundary example

	$\lambda = 0.5$			$\lambda = 25$		
grid points	1D L_∞ error	2D L_2 error	2D L_∞ error	1D L_∞ error	2D L_2 error	2D L_∞ error
101×101	0.0216	0.0048	0.0344	0.0055	0.00015	0.0092
201×201	0.0109	0.0025	0.0173	0.0032	0.00008	0.0053
401×401	0.0055	0.0012	0.0087	0.0017	0.00004	0.0029
801×801	0.0027	0.0006	0.0044	0.0009	0.00002	0.0015
1601×1601	0.0014	0.0003	0.0022	0.0005	0.00001	0.0008

4.3 *Optimal idle-time motion in continuous space*

We now consider a continuous equivalent of the optimal idle-time processing example of Section 2.5. An all-terrain vehicle moving in $\overline{\Omega} \subset R^2$ is tasked with responding to emergency calls. We assume that the arrival of calls is a Poisson process with rate λ, and our goal is to minimize the expected response time to the first caller. We are given a list of possible caller locations $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_r$ inside Ω and the corresponding probabilities $\tilde{P}_1, \ldots, \tilde{P}_r$ of the next call originating at each of these locations. The vehicle's dynamics is assumed to be isotropic; i.e., $\boldsymbol{y}' = f(\boldsymbol{y})\boldsymbol{a}$, where $\boldsymbol{a} \in S_1$ is the current direction of motion. We then use the Fast Marching Method to solve r Eikonal problems:

$$|\nabla u_i(\boldsymbol{x})| f(\boldsymbol{x}) = 1, \quad \boldsymbol{x} \in \Omega \backslash \{\tilde{\boldsymbol{x}}_i\}; \qquad u_i(\tilde{\boldsymbol{x}}_i) = 0, \quad \text{and } u = +\infty \text{ on } \partial\Omega; \qquad i = 1, \ldots, r.$$

The resulting $u_i(\boldsymbol{x})$ is the minimum time to reach $\tilde{\boldsymbol{x}}_i$ for a vehicle starting from $\boldsymbol{x}$ and constrained to move within Ω. If the call is received at $\boldsymbol{x}$, the expected time from there to the caller is

$$q(\boldsymbol{x}) = \sum_{i=1}^{r} \tilde{P}_i u_i(\boldsymbol{x}). \tag{4.3}$$

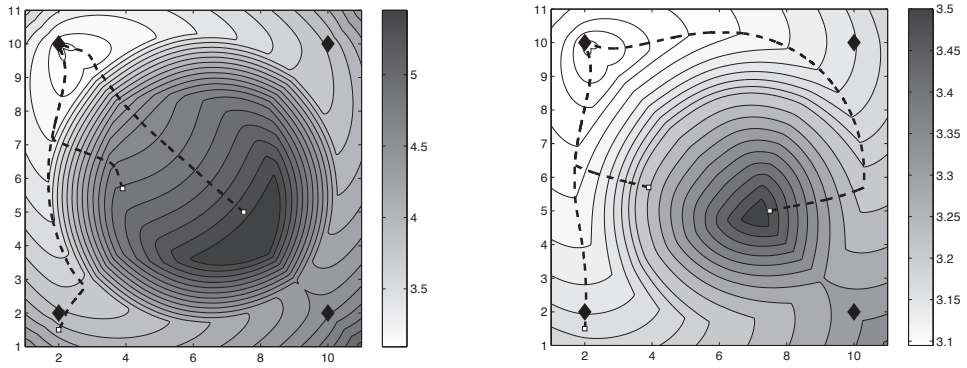
(We note that thus defined $q(\boldsymbol{x})$ is always Lipschitz-continuous.)

A global minimum of q is obviously the optimal place to "park" the vehicle while expecting a call. But what if the current position is not a global minimum of q? The most intuitive approach, based on a gradient descent in q, is far from optimal. (E.g., it would prescribe not moving away from any local minimum of q – a clearly bad strategy when λ is small enough and there is a good chance of reaching the global minimum of q before the next call.) Since we are trying to minimize the expected response time, it is logical to set $K = 0$, since in this continuous-time control, the vehicle starts responding instantaneously. (This is in contrast to the discrete-transitions scenario considered in Section 2.5.)

We consider an example in which f is piecewise-constant: $f = 0.2$ in a large circular "slow" region in the center and $f = 1$ everywhere else. We use four different call locations $(\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_4)$ symmetric relative to the slow region and numbered counter-clockwise starting from the lower left; see Figures 4 and 5.

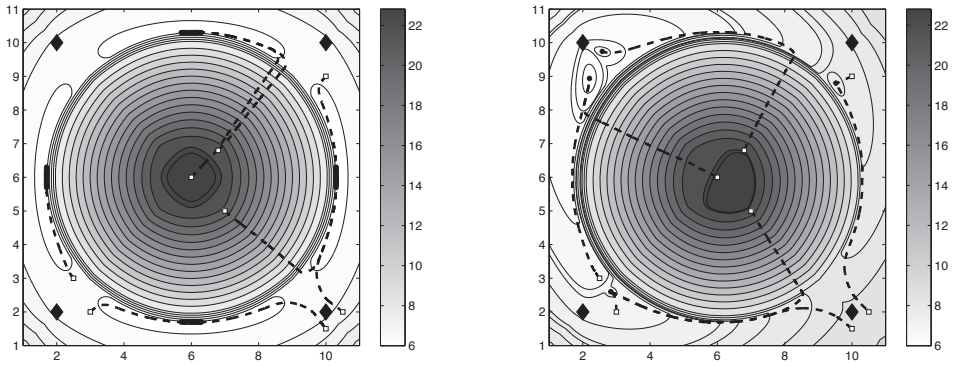
We perform this experiment with several sets of parameter values. First, we set $\lambda = 0.05$, $\tilde{P}_1 = \tilde{P}_2 = \tilde{P}_3 = 0.2$ and $\tilde{P}_4 = 0.4$. The results in Figure 4 highlight the differences between q and v.

We then set $\lambda = 25$ and repeat the experiment two more times: with equal probabilities ($\tilde{P}_1 = \ldots = \tilde{P}_4 = 0.25$) in Figure 5A and with varying probabilities ($\tilde{P}_1 = \tilde{P}_2 = 0.2$; $\tilde{P}_3 = 0.25$; $\tilde{P}_4 = 0.35$) in Figure 5B. In all three cases the computations are performed on a 1001×1001 grid and the level sets are selected to highlight the complex structure of the solution outside of the slow region.



A B

FIG. 4. $\lambda = 0.05$ Four emergency call locations (shown by black diamonds) around a circular "slow region". Three sample starting locations are shown by small white squares. Level curves of q (left) and of v (right) shown by solid lines. "Optimal" trajectories (shown by dotted lines) found by gradient descent in q (on the left) are quite different from the truly optimal trajectories found by gradient descent in v (on the right).



A B

FIG. 5. $\lambda = 25$. Four emergency call locations (shown by black diamonds) around a circular "slow region". Level curves of v shown by solid lines. Optimal trajectories are shown by dotted lines, running from several starting locations (shown by small squares) to the free boundary $\boldsymbol{B}$ (found numerically and shown by thick black dots).

As the figures show, two nearby starting locations can easily produce dramatically different optimal trajectories when the locations are on different sides of the shock line (where ∇v is undefined).

REMARK 4.2 We note that $K \equiv 0$ leads to M consisting of (a subset of) local minima locations of q; these are found numerically (and indicated by thick black dots in the corresponding figures). As a result, the true shape of M may not become apparent even on fairly refined computational grids. For example, when all caller locations are equally likely, it is relatively easy to show analytically that the minima of q are attained at four isolated points only, while Figure 5A shows a larger motionless set. This, however, is due to the fact that the value function varies very slowly in this region (e.g., order of 10^{-6} variation of q on the numerically found M). This also presents an additional challenge in recovering optimal trajectories, which for isotropic problems is done by a gradient descent in v. The characteristic equations show that, when $K \equiv 0$, the directional derivative of v along the optimal trajectory becomes zero at B. To circumvent this difficulty, our current implementation forces the trajectory to take a straight line path to B, when the distance to it decreases below an h-dependent threshold. No such heuristic adjustments are needed when $\overline{K}_1 > 0$.

4.4 *Navigating a maze*

The following example illustrates the effect of changing λ on the free boundary for problems with non-zero running cost K.

We assume two possible locations for emergency calls: $\tilde{\boldsymbol{x}}_1 = (1.0, 0.1)$ and $\tilde{\boldsymbol{x}}_2 = (9.0, 0.1)$ with the corresponding probabilities of calls $\tilde{P}_1 = 0.2$ and $\tilde{P}_2 = 0.8$ inside the domain $\overline{\Omega} = [0, 10] \times [0, 10]$. We further assume that the domain contains a "maze" with K high / f low within its "walls" and K low / f high everywhere else. Figure 6 shows the described f, K, and the resulting terminal cost function q computed from (4.3).

Figure 7 shows the level curves of v and the optimal trajectory starting from the center of the maze computed for different values of λ. The free boundary B is also indicated by a thick line in each case. We note that the running cost matters only until the emergency call arrives. Thus, for very

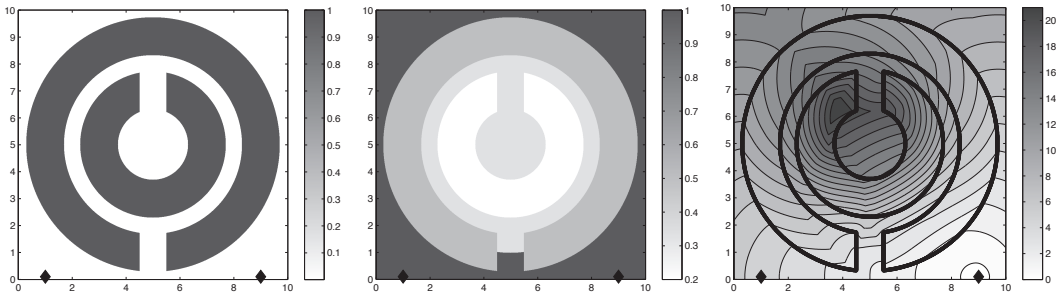


FIG. 6. The "maze example" : running cost K (Left), speed f (Center), and the terminal cost q (Right). The caller locations are shown by black diamonds. The thick solid lines in the last subfigure show ∂M^{∞}.

small λ's, it is likely that the global minimum of q (i.e., the point $\tilde{\boldsymbol{x}}_2$) can be reached avoiding the walls before the first call arrives. When λ increases, it becomes more optimal to head toward (and then through) the closest wall, hoping that the call arrives before we reach it. Finally, for large λ's it is optimal to stop and wait for the call at the wall boundary.

5. Conclusions

We have considered a wide class of uncertain-horizon problems and showed that non-iterative methods can be used to compute their value functions both in discrete and continuous settings. The numerical examples in Section 4 have illustrated both the convergence properties and the asymptotic behavior of the free boundary ∂M. Our modification of the Fast Marching Method in Section 3.3 addressed the isotropic case only, but similarly modified Ordered Upwind Methods [2, 31, 32] can be used to treat the anisotropic cost and dynamics in randomly-terminated problems. A Dial-like version of the Fast Marching Method will be similarly applicable if (3.13) is discretized on an acute triangulated mesh [38]. Another fairly straightforward generalization is to treat inhomogeneous termination rates in the continuous case; i.e., $\lambda = \lambda(\boldsymbol{x})$; this would require only minimal changes to the label-setting algorithms.

In this paper we have not discussed the label-correcting methods [7], whose asymptotic complexity is worse but practical performance is sometimes better than that of label-setting methods. Their applicability to uncertain-horizon problems is clearly also of interest. In the continuous case, we believe that fast sweeping methods (e.g., [11, 34, 40]) and various fast-iterative methods (e.g., [3, 10, 22, 26]) should be easy to extend to randomly-terminated problems. The same is also true for hybrid two-scale methods that aim to combine the advantages of marching and sweeping [13, 14]. Careful testing would be needed to compare the computational efficiency of these alternatives to that of Dijkstra-like methods considered here.

The second author has previously studied the so-called Multimode SSP (MSSP) in [38] and derived sufficient conditions for the applicability of the label-setting methods to them. We emphasize that the uncertain-horizon problems considered here cannot be cast as MSSPs and the results from [38] do not apply. However, the results from both papers can be easily combined to address the randomly-terminated MSSPs. More generally, we believe that label-setting methods will be applicable for a broader class of *hierarchically causal* SSPs. A natural example from this category is the "SSP with recourse" problem, where the transition-costs are known only probabilistically at

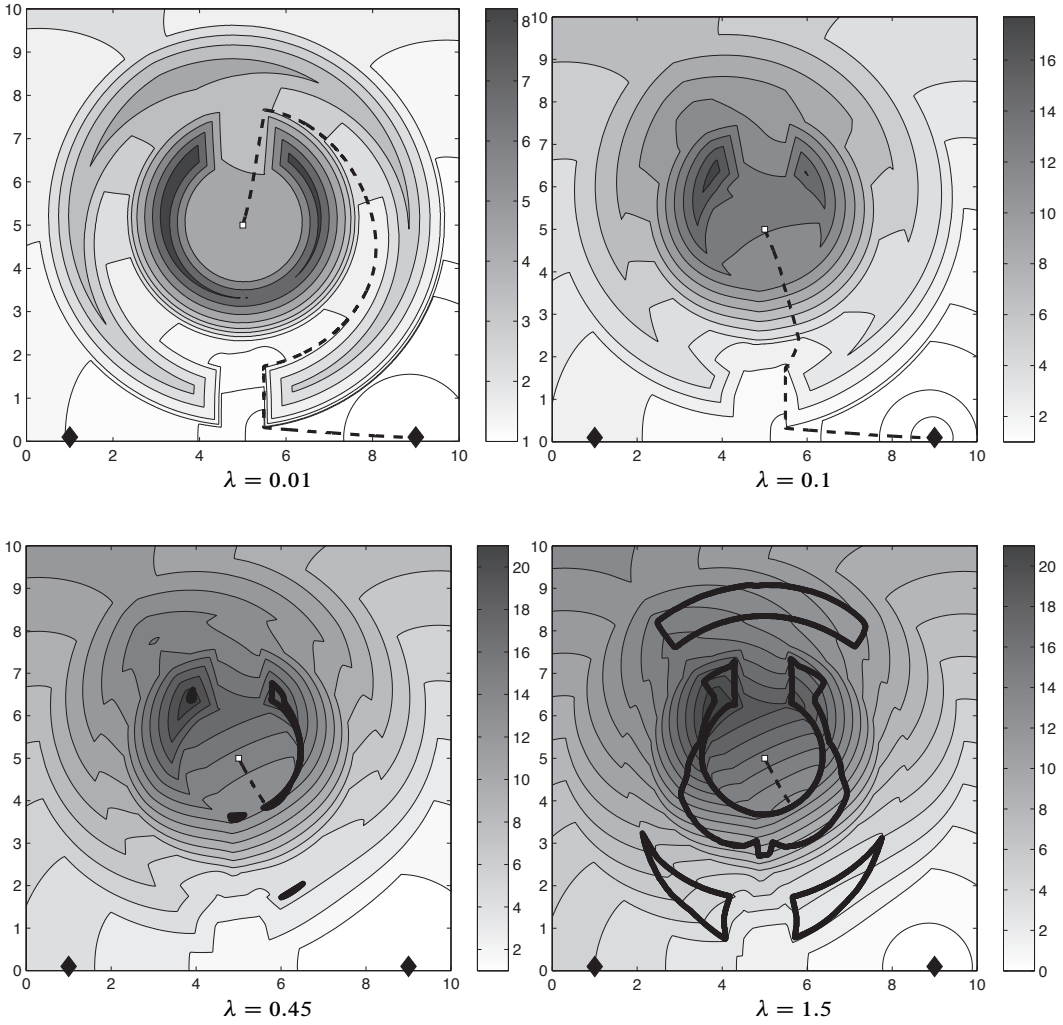


FIG. 7. The "maze example": level-curves of the value function $v(\boldsymbol{x})$ computed for 4 different values of λ. In each case, the optimal trajectory starting from the center is indicated by a dotted line. For larger λ's, when $\boldsymbol{M}$ is not just the global minimum of q, the free boundary is shown by a thick solid line.

first and their true values are learned in the process of traveling through the graph. A Dijkstra-like method for such problems was previously introduced by Polychronopoulos and Tsitsiklis in [37].

On the continuous side, the randomly-terminated problems form a simple subclass of *piecewise-deterministic* problems [16, 20]. The latter arise when a Poisson process governs random switches between several known types of deterministic dynamics/cost, yielding a system of weakly-coupled static non-linear HJB PDEs. We believe that these more general problems can also be treated by non-iterative numerical methods provided all non-deterministic transitions (and the resulting couplings between the PDEs) are hierarchically causal.

Finally, we note that in all randomly-terminated problems considered in the current paper, the only goal was to minimize the expected value of the total cost. An interesting direction for future work is to incorporate simultaneous optimization and/or constraints based on the worst case scenario. Efficient algorithms for this more difficult problem can be built using the recent method for multiobjective optimal control introduced in [23].

Appendix A: Optimality of stationary policies

This section contains the results on existence of optimal stationary policies for the general randomly-terminated processes on graphs. We note that the following proofs do not make any use of assumptions A1–A3; i.e., both the transition penalties K_{ij} and the terminal costs q_j can be positive or negative and the self-transitions need not be allowed.

A function $\mu : X \mapsto X$ is a *control mapping* if $\mu(x) \in N(x)$ for $\forall x \in X$. A *policy* is an infinite sequence of control mappings $\pi = (\mu_0, \mu_1, \ldots)$. Starting from any $x \in X$ a policy will generate a particular path $\boldsymbol{y}^\pi$ as follows: $y_0^\pi = x$, $y_{k+1}^\pi = \mu_k(y_k^\pi)$ for $\forall k \geqslant 0$. Defining $J(x, \pi) = J(\boldsymbol{y}^\pi)$, we can also re-write the value function as

$$V(x) = \min_{\pi} J(x, \pi). \tag{A.1}$$

If $\mu = \mu_0 = \mu_1 = \ldots$, the corresponding policy $\pi = (\mu, \mu, \ldots)$ is called *stationary*. (We will also somewhat abuse the notation and refer to a stationary policy μ.) Since stationary policies generate only simple paths, Theorem A.5 proves the existence of an optimal stationary policy. Formula (2.9) shows that for any stationary policy, $J(x, \mu) = K(x, \mu(x)) + pq(\mu(x)) + (1 - p)J(\mu(x), \mu))$, which implies the dynamic programming equation (2.10).

REMARK A.1 Our problem can be easily recast as a Stochastic Shortest Path (SSP) problem by adding a special absorbing terminal node $x_{\boldsymbol{t}} = x_{M+1}$, and considering $N(x_i)$ to be the set of controls available at x_i. A choice of the control $x_j \in N(x_i)$ then results in a transition to the node x_j with probability $(1 - p)$ and to the node $x_{\boldsymbol{t}}$ with probability p. The cost associated with this control is $(K_{ij} + pq_j)$. The process terminates upon reaching $x_{\boldsymbol{t}}$. In [9] Tsitsiklis and Bertsekas proved the equivalents of Lemma A.2 and Theorem A.5 for a broader class of general SSPs. However, in our setting, direct proofs are much simpler and exploit the special structure of this problem.

We first note the following useful generalization of the recursive formula (2.9). Starting from (2.11),

$$\begin{aligned} J(\boldsymbol{y}) &= E\big[Cost(y_0, \ldots, y_k, \ldots)\big] \\ &= \sum_{i=0}^{k-1} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^i \;+\; \sum_{i=k}^{\infty} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^i \\ &= \sum_{i=0}^{k-1} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^i \;+\; (1-p)^k\, J(y_k, y_{k+1}, \ldots). \end{aligned} \tag{A.2}$$

LEMMA A.2 For every $x \in X$, there exists $\bar{\boldsymbol{y}} \in Y(x)$ such that $J(\bar{\boldsymbol{y}}) = \inf_{\boldsymbol{y} \in Y(x)} J(\boldsymbol{y})$.

Proof. Given any two paths $\boldsymbol{y}, \tilde{\boldsymbol{y}} \in Y$, let k be the first stage of the process where the paths become different; i.e., $k(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \min\{j \mid y_j \neq \tilde{y}_j\}$. A natural metric on Y is defined by

$$dist(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \begin{cases} (1-p)^{k(\boldsymbol{y},\tilde{\boldsymbol{y}})}, & \text{if } \boldsymbol{y} \neq \tilde{\boldsymbol{y}}; \\ 0, & \text{if } \boldsymbol{y} = \tilde{\boldsymbol{y}}. \end{cases}$$

and the induced topology makes Y totally disconnected. The compactness of Y follows from the standard diagonalization argument. $Y(x)$ is thus a closed subset of Y and also compact. On the other hand, based on (A.2), $J : Y \mapsto R$ is Lipschitz-continuous. As a continuous function on a compact set it must attain the minimum at some $\bar{\boldsymbol{y}} \in Y(x)$. □

We will use $Y^*(x)$ to denote the set of all such *minimizing paths* starting from x.

LEMMA A.3 Suppose $\boldsymbol{y} = (y_0, y_1, \ldots) \in Y^*(x)$, while k and m are non-negative integers such that $k < m$ and $y_k = y_m$. Then $J(\boldsymbol{y}) = J(y_0, \ldots, y_k, y_{m+1}, y_{m+2}, \ldots)$.

Proof. We show that any loop can be removed from an optimal (not necessarily simple) path without increasing that path's total cost. By the optimality of $\boldsymbol{y}$, $J(\boldsymbol{y}) \leqslant J(y_0, \ldots, y_k, y_{m+1}, y_{m+2}, \ldots)$. On the other hand, if $J(\boldsymbol{y}) < J(y_0, \ldots y_k, y_{m+1}, y_{m+2}, \ldots)$, then, by formula (A.2), $J(y_k, y_{k+1}, y_{k+2}, \ldots) < J(y_k, y_{m+1}, y_{m+2}, \ldots)$. The latter contradicts the optimality of $\boldsymbol{y}$ since it implies that

$$J(\boldsymbol{y}) = J(y_0, \ldots, y_m, y_{m+1}, y_{m+2}, \ldots) > J(y_0, \ldots, y_m, y_{k+1}, y_{k+2}, \ldots).$$

□

LEMMA A.4 Suppose $\boldsymbol{y} = (y_0, y_1, \ldots) \in Y^*(x)$, while k and m are non-negative integers such that $k < m$ and $y_k = y_m$. Define $\tilde{\boldsymbol{y}}$ by replacing the tail $(y_k, \ldots)$ with an infinitely repeated loop $(y_k, y_{k+1}, \ldots, y_{m-1})$. Then $J(\boldsymbol{y}) = J(\tilde{\boldsymbol{y}})$.

Proof. First, note that proving the equality for $k = 0$ combined with the formula (A.2) yields the proof for the general case. Assuming $k = 0$,

$$\begin{aligned} J(\tilde{\boldsymbol{y}}) &= \sum_{r=0}^{\infty} \Big(\sum_{i=0}^{m-1} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^{i+rm} \Big) \\ &= \Big(\sum_{i=0}^{m-1} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^{i} \Big) \frac{1}{1-(1-p)^m}. \end{aligned} \tag{A.3}$$

By Lemma A.3 and formula (A.2),

$$\begin{aligned} J(\boldsymbol{y}) &= \sum_{i=0}^{m-1} \big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^{i} \; + \; (1-p)^m \, J(\boldsymbol{y}); \\ \sum_{i=0}^{m-1} &\big(K(y_i, y_{i+1}) + pq(y_{i+1})\big)(1-p)^{i} = \big(1-(1-p)^m\big) \, J(\boldsymbol{y}), \end{aligned}$$

which yields $J(\tilde{\boldsymbol{y}}) = J(\boldsymbol{y})$. □

THEOREM A.5 There exists a stationary policy μ such that $J(x,\mu) = V(x)$.

Proof. By Lemma A.2, there exists some optimal path $\hat{\boldsymbol{y}} \in Y^*(x)$. We note that, given any path, it is easy to define a policy generating it. Moreover, if the path is *simple*, then it can be produced by a single control mapping (and the corresponding policy will be stationary). So, we simply need to prove the existence of some $\bar{\boldsymbol{y}} \in Y^s(x)$ such that $J(\bar{\boldsymbol{y}}) = J(\hat{\boldsymbol{y}})$.
For each non-negative integer k and each path $\boldsymbol{y} \in Y(x)$, let $M(\boldsymbol{y},k) = \{m \mid m > k,\ y_m = y_k\}$. We will define a function $D_k : Y(x) \mapsto Y(x)$ as follows:
- if $M(\boldsymbol{y},k)$ is not empty, then $D_k(\boldsymbol{y}) = \tilde{\boldsymbol{y}}$ produced in Lemma A.4 using k and $m = \min M(\boldsymbol{y},k)$,
- and $D_k(\boldsymbol{y}) = \boldsymbol{y}$ otherwise.

Note, that if $\boldsymbol{y} \in Y^*(x)$, then $D_k(\boldsymbol{y}) \in Y^*(x)$ as well. Defining $\tilde{\boldsymbol{y}}^0 = \hat{\boldsymbol{y}}$ and $\tilde{\boldsymbol{y}}^k = D_{k-1}(\tilde{\boldsymbol{y}}^{k-1})$ we obtain a simple path in at most M steps; i.e., $\bar{\boldsymbol{y}} = \tilde{\boldsymbol{y}}^M \in Y^s(x)$. □

Appendix B: A quadrant-by-quadrant update formula

To simplify the discussion, we will focus on one node $\boldsymbol{x} = \boldsymbol{x}_{i,j}$, renaming its neighbors as in Figure 8 and slightly abusing the notation as follows:

$$V = V_{i,j}, \qquad K = K_{i,j}, \qquad q = q_{i,j}, \qquad f = f_{i,j},$$
$$V_1 = V_{i+1,j}, \qquad V_2 = V_{i,j+1}, \qquad V_3 = V_{i-1,j}, \qquad V_4 = V_{i,j-1}.$$

First, suppose that $V < q$, $\max(D_{ij}^{-y}V, -D_{ij}^{+y}V, 0) = 0$, and

$$\max\left(D_{ij}^{-x}V, -D_{ij}^{+x}V, 0\right) = -D_{ij}^{+x}V \quad \Longrightarrow \quad V \geqslant V_1, \tag{B.4}$$

Then (3.17) reduces to $V = q + \left[K - \frac{f}{h}(V - V_1)\right]/\lambda$, with the solution

$$V = (hK + \lambda hq + fV_1)/(\lambda h + f). \tag{B.5}$$

We note that $V < q$ implies $V - V_1 = h(K + \lambda(q - V))/f > 0$, which is consistent with (B.4).

Now suppose that $V < q$ and (B.4) hold, but

$$\max\left(D_{ij}^{-y}V, -D_{ij}^{+y}V, 0\right) = -D_{ij}^{+y}V, \quad \Longrightarrow \quad V \geqslant V_2, \tag{B.6}$$

Then (3.17) reduces to a quadratic equation

$$f^2\left[\left(\frac{V - V_1}{h}\right)^2 + \left(\frac{V - V_2}{h}\right)^2\right] = [K + \lambda q - \lambda V]^2. \tag{B.7}$$

Suppose $V^\#$ is the smallest number satisfying both (B.7) and $V^\# \geqslant \max(V_1, V_2)$ (for consistency with (B.4) and (B.6)). If such $V^\#$ exists, we use it as an "update" from this first quadrant; i.e., $V^{12} = V^\#$. Otherwise, we define the update similarly to (B.5) as $V^{12} = (hK + \lambda hq + f\min(V_1, V_2))/(\lambda h + f)$. The updates from the other quadrants are similarly defined using (V_2, V_3), (V_3, V_4), and (V_4, V_1). If the values in $NV_{i,j}$ are fixed, it is easy to check that

$$V_{i,j} = \min\left(q, V^{12}, V^{23}, V^{34}, V^{41}\right) \tag{B.8}$$

is the unique solution of (3.17).

REMARK B.6 The modified version of the Fast Marching Method in Section 3.3 calls for updating the not-yet-*Accepted* value of $V_{i,j}$ whenever one of its neighbors is *Accepted*. In principle, the above procedure and formula (B.8) may be used for that update, but a more efficient implementation can be built utilizing the quadrant-by-quadrant approach and the fact that $V_{i,j}$ depends on the smaller neighbors only.

First, each quadrant-update should be considered only if at least one of the neighbors defining this quadrant is already *Accepted* (by the causality of (3.17), the non-*Accepted* values can be replaced by $+\infty$ without affecting $V_{i,j}$). Second, if $\bar{\boldsymbol{x}}$ is the last *Accepted* gridpoint, then only quadrants in which it participates will need to be considered (since the updates from other quadrants have already been previously stored in $V_{i,j}$ – this is similar to the idea behind the update formula (2.18) for Dijkstra's method). Third, the monotonicity of (3.17) guarantees that at most one quadrant is relevant.

For definiteness sake, suppose that the recently accepted $\bar{\boldsymbol{x}} = \boldsymbol{x}_{i+1,j} = \boldsymbol{x}_1$ in the notation of Figure 8. The efficient update procedure would set $V(\boldsymbol{x}_{i,j}) := \min\big(V(\boldsymbol{x}_{i,j}), V\big)$, where V is

1. computed by formula (B.5) if neither $\boldsymbol{x}_2$ nor $\boldsymbol{x}_4$ is *Accepted*;
2. equal to V^{12} if $\boldsymbol{x}_2$ is *Accepted* and $\boldsymbol{x}_4$ is not (or if both of them are *Accepted*, but $V_2 \leqslant V_4$);
3. equal to V^{41} if $\boldsymbol{x}_4$ is *Accepted* and $\boldsymbol{x}_2$ is not (or if both of them are *Accepted*, but $V_2 > V_4$).

B.1 *A semi-Lagrangian discretization*

The upwind finite difference discretization (3.17) and its quadrant-by-quadrant version in the previous section do not appear natural from the point of view of optimal control. A more natural semi-Lagrangian scheme is based on a direct discretization of the optimality principle. Here we show that the former is in fact equivalent to the latter.

Suppose $\boldsymbol{x} = (x_i, y_j)$ and $v(\boldsymbol{x}) < q(\boldsymbol{x})$ (i.e., $\boldsymbol{a}_0$ is not the optimal control value at $\boldsymbol{x}$). Suppose f and K are locally constant, v is smooth, and the optimal direction of motion from $\boldsymbol{x}$ is $\boldsymbol{a} \in S_1$ lying in the first quadrant; see Figure 8. We assume that the motion continues in the direction $\boldsymbol{a}$ until crossing the segment $\boldsymbol{x}_1\boldsymbol{x}_2$ at the point $\tilde{\boldsymbol{x}}$ after $\tau = |\boldsymbol{x} - \tilde{\boldsymbol{x}}|/f$ units of time. The corresponding running cost is τK and the probability of termination signal received while moving from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ is $(1 - e^{-\lambda\tau})$. In case of termination, we incur the terminal cost of q; otherwise, the motion along the

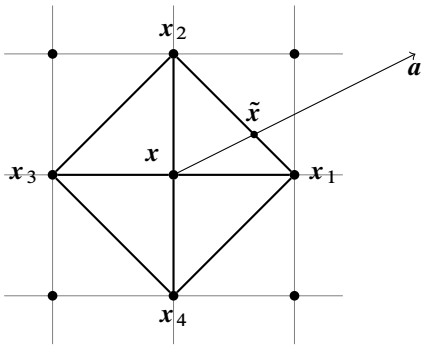


FIG. 8. A semi-Lagrangian scheme using a simple 5-point stencil on a Cartesian grid

approximate optimal trajectory continues (with the expected cost of $v(\tilde{\boldsymbol{x}})$). Thus,

$$v(\boldsymbol{x}) = K\tau + (1-e^{-\lambda\tau})q + e^{-\lambda\tau}v(\tilde{\boldsymbol{x}}) + O(\tau^2) = \frac{K\tau e^{\lambda\tau}}{e^{\lambda\tau}} + \frac{(e^{\lambda\tau}-1)q}{e^{\lambda\tau}} + \frac{v(\tilde{\boldsymbol{x}})}{e^{\lambda\tau}} + O(\tau^2)$$
$$= \frac{K\tau}{1+\lambda\tau} + \frac{q\lambda\tau}{1+\lambda\tau} + \frac{v(\tilde{\boldsymbol{x}})}{1+\lambda\tau} + O(\tau^2).$$

Let $\Xi = \{\xi = (\xi_1, \xi_2) \mid \xi_1 + \xi_2 = 1 \text{ and } \forall \xi_1, \xi_2 \geqslant 0\}$. If $\xi \in \Xi$ is such that $\tilde{\boldsymbol{x}} = \xi_1 \boldsymbol{x}_1 + \xi_2 \boldsymbol{x}_2$, then

$$\tau(\xi) = (h/f)\sqrt{\xi_1^2 + \xi_2^2} \qquad \text{and} \qquad v(\tilde{\boldsymbol{x}}) \approx \xi_1 V_1 + \xi_2 V_2.$$

This suggests a semi-Lagrangian scheme for the grid function V:

$$V^{12} = \min_{\xi\in\Xi} C(\xi) = \min_{\xi\in\Xi} \left\{ \frac{(K+\lambda q)\tau(\xi) \; + \; (\xi_1 V_1 + \xi_2 V_2)}{1+\lambda\tau(\xi)} \right\}. \tag{B.9}$$

Given similarly defined updates from all other quadrants, we can again set

$$V = \min\left(q, V^{12}, V^{23}, V^{34}, V^{41}\right).$$

Kuhn–Tucker optimality conditions can be used to relate this scheme to (3.17). A similar connection was previously demonstrated for Eikonal PDEs on a uniform Cartesian grid by Tsitsiklis [36], and then on triangulated meshes and for more general Hamilton–Jacobi-Bellman PDEs by Sethian and Vladimirsky [32, Appendix]; see also the detailed discussion of connections to MSSP problems in [38]. The following proof applies the same ideas to the variational inequality (3.11), with additional technical details due to the direct dependence of the Hamiltonian on v.

THEOREM B.7 Let ξ^* denote the minimizer in (B.9), and suppose that $V = V^{12} < q$. Then
1. $\xi_i^* > 0 \implies V^{12} > V_i \qquad \text{for } i = 1, 2.$
2. $\xi_1^*, \xi_2^* > 0 \implies V^{12}$ defined by (B.9) is also a solution of (B.7).

Proof. We note three useful properties of the function $\tau(\xi)$:

$$\frac{\partial\tau}{\partial\xi_i}(\xi) = \frac{h^2}{f^2}\frac{\xi_i}{\tau(\xi)}; \tag{B.10}$$

$$\tau(\xi) = \xi_1 \frac{\partial\tau}{\partial\xi_1}(\xi) + \xi_2 \frac{\partial\tau}{\partial\xi_2}(\xi); \tag{B.11}$$

$$\frac{h^2}{f^2} = \left(\frac{\partial\tau}{\partial\xi_1}(\xi)\right)^2 + \left(\frac{\partial\tau}{\partial\xi_2}(\xi)\right)^2. \tag{B.12}$$

To simplify the notation, we will suppress the arguments – in what follows, τ and its partial derivatives are always evaluated at ξ^*.

First, note that if $\xi^* = (1, 0)$, then $\tau = h/f$ and (B.9) reduces to (B.5); a similar formula (with V_2 replacing V_1) holds when $\xi^* = (0, 1)$. Now suppose $\xi_1^*, \xi_2^* > 0$. By the Kuhn–Tucker optimality conditions applied to $C(\xi)$ defined in formula (B.9), there exists a Lagrange multiplier μ such that

$$\mu = \frac{\partial C}{\partial\xi_i}(\xi^*) = \frac{(K+\lambda q)\frac{\partial\tau}{\partial\xi_i} + V_i}{1+\lambda\tau} - \frac{\lambda\frac{\partial\tau}{\partial\xi_i}[\tau(K+\lambda q) + \xi_1^* V_1 + \xi_2^* V_2]}{(1+\lambda\tau)^2}$$
$$= \frac{(K+\lambda q - \lambda V^{12})\frac{\partial\tau}{\partial\xi_i} + V_i}{1+\lambda\tau}, \qquad \text{for } i = 1, 2. \tag{B.13}$$

Multiplying the above by ξ_i^*, adding up (for $i = 1, 2$) and using (B.11), we see that

$$\mu = \mu(\xi_1 + \xi_2) = \frac{(K + \lambda q - \lambda V^{12})\tau + \xi_1^* V_1 + \xi_2^* V_2}{1 + \lambda\tau} = V^{12} - \frac{\lambda\tau V^{12}}{1 + \lambda\tau} = \frac{V^{12}}{1 + \lambda\tau}. \quad \text{(B.14)}$$

Combining (B.13) and (B.14), we obtain

$$V^{12} - V_i = (K + \lambda q - \lambda V^{12})\frac{\partial\tau}{\partial\xi_i} > 0, \quad \text{for } i = 1, 2 \quad \text{(B.15)}$$

where the inequality follows from $K \geqslant 0$, $V^{12} < q$, and the fact that ($\xi_i^* > 0 \implies \frac{\partial\tau}{\partial\xi_i}(\xi^*) > 0$) by formula (B.10). This shows the causality of the semi-Lagrangian discretization.

To prove the second half of the theorem, we take a square of both sides of (B.15), sum over $i = 1, 2$ and use (B.12) to obtain

$$\left(V^{12} - V_1\right)^2 + \left(V^{12} - V_2\right)^2 = \frac{h^2}{f^2}\left(K + \lambda q - \lambda V^{12}\right)^2,$$

which is equivalent to (B.7). □

Acknowledgments. The authors are grateful to anonymous reviewers for their suggestions on improving this paper. The second author would also like to thank A. Dukhovny and V. Protsak. This research was supported by the National Science Foundation grants DMS-0514487 and DMS-1016150. The first author's work was also supported by the NSF Graduate Fellowship.

REFERENCES

1. AHUJA, R.K., MAGNANTI, T.L., & ORLIN, J.B., *Network flows*, Prentice Hall, 1993. Zbl1201.90001 MR1205775
2. ALTON, K. & MITCHELL, I. M., An ordered upwind method with precomputed stencil and monotone node acceptance for solving static Hamilton-Jacobi equations, *Journal of Scientific Computing* **51** (2012) 313–348. Zbl1256.65099 MR2902209
3. BAK, S., MCLAUGHLIN, J. & RENZI, D., Some improvements for the fast sweeping method, *SIAM J. Sci. Comp.* **32** (2010) 2853–2874. Zbl1219.65124 MR2729443
4. BARDI, M. & CAPUZZO DOLCETTA, I., *Optimal control and viscosity solutions of Hamilton–Jacobi-Bellman equations*, Birkhäuser (1997). Zbl0890.49011 MR1484411
5. BARLES, G. & SOUGANIDIS, P. E., Convergence of approximation schemes for fully nonlinear second order equations, *Asymptot. Anal.* **4** (1991), 271–283. Zbl0729.65077 MR1115933
6. BELLMAN, R., On the theory of dynamic programming, *Proc. Nat. Acad. Sci.* **38** (1952), 716–719. Zbl0047.13802 MR0050856
7. BERTSEKAS, D.P., *Network optimization: continuous & discrete models*, Athena Scientific (1998). Zbl0997.90505
8. BERTSEKAS, D.P., *Dynamic programming and optimal control*, 2nd edition, Volumes I and II, Athena Scientific (2001). Zbl1083.90044 MR2183196 (Vol. I), MR2182753 (Vol. II)
9. BERTSEKAS, D.P. & TSITSIKLIS, J.N., An analysis of stochastic shortest path problems, *Mathematics of Operations Research* **16** (1991), 580–595. Zbl0751.90077 MR1120471
10. BORNEMANN, F. & RASCH, C., Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle, *Comp. and Visualization in Science* **9** (2006), 57–69. MR2247685

11. BOUÉ, M. & DUPUIS, P., Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, *SIAM J. Numer. Anal.* **36** (1999), 667–695. Zbl0933.65073 MR1681057
12. BOUKAS, E.K., HAURIE, A., MICHAEL, P., An optimal control problem with a random stopping time, *J. of Optimization Theory and Applications* **64** (1990), 471–480. Zbl0681.93070 MR1043735
13. CHACON, A. & VLADIMIRSKY, A., Fast two-scale methods for Eikonal equations, *SIAM J. Sci. Comp.* **33** (2012), A547–A578. Zbl1244.49047 MR2914295
14. CHACON, A. & VLADIMIRSKY, A., A parallel two-scale method for Eikonal equations, submitted to *SIAM J. Sci. Comp.* in 2012.
15. CRANDALL, M.G. & LIONS, P-L., Viscosity solutions of Hamilton-Jacobi equations, *Tran. AMS* **277** (1983), 1–43. Zbl0599.35024 MR0690039
16. DAVIS, M.H.A., *Markov models and optimization*, Chapman and Hall, London, 1993. Zbl0780.60002 MR1283589
17. DIAL, R., Algorithm 360: shortest path forest with topological ordering, *Comm. ACM* (1969), 632–633.
18. DIJKSTRA, E.W., A note on two problems in connection with graphs, *Numerische Mathematik* **1** (1959), 269–271. Zbl0092.16002 MR0107609
19. FOMEL, S., LUO, S. & ZHAO, H., Fast sweeping method for the factored Eikonal equation, *J. Comp. Phys.* **228** (2009), 6440–6455. Zbl1175.65125 MR2549850
20. HAURIE, A., A multigenerational game model to analyze sustainable development, *Annals of Operations Research* **137** (2005), 369–386. Zbl1138.91348 MR2166448
21. HAURIE, A. & MORESINO, F., A stochastic control model of economic growth with environmental disaster prevention, *Automatica (Journal of IFAC)* **42** (2006), 1417–1428. Zbl1108.93078 MR2242926
22. JEONG, W.-K. & WHITAKER, R. T., A fast iterative method for Eikonal equations, *SIAM J. Sci. Comput.* **30** (2008), 2512–2534. Zbl1246.70003 MR2429477
23. KUMAR, A. & VLADIMIRSKY, A., An efficient method for multiobjective optimal control and optimal control subject to integral constraints, *Journal of Computational Mathematics* **28** (2010), 517–551. Zbl1240.90345 MR2666839
24. MAXWELL, M.S., RESTREPO, M., HENDERSON, S.G. & TOPALOGLU, H., Approximate dynamic programming for ambulance redeployment, *INFORMS Journal on Computing* **22** (2010), 266–281. Zbl1243.90109 MR2677214
25. OBERMAN, A. M., Convergent difference schemes for nonlinear elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems, *SIAM J. Numer. Anal.* **44** (2006), 879–895. Zbl1124.65103 MR2218974
26. POLYMENAKOS, L. C., BERTSEKAS, D. P. & TSITSIKLIS, J. N., Implementation of efficient algorithms for globally optimal trajectories, *IEEE Trans. on Automatic Control* **43** (1998), 278–283. Zbl1032.49037 MR1605994
27. POWELL, W. B., *Approximate dynamic programming: solving the curses of dimensionality*, J. Wiley & Sons (2007). Zbl1156.90021 MR2347698
28. ROUY, E. & TOURIN, A., A viscosity solutions approach to shape-from-shading, *SIAM J. Num. Anal.* **29** (1992), 867–884. Zbl0754.65069 MR1163361
29. SETHIAN, J.A., A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.* **93** (1996), 1591–1595. Zbl0852.65055 MR1374010
30. SETHIAN, J.A., Fast marching methods, *SIAM Review* **41** (1999), 199–235. Zbl0926.65106 MR1684542
31. SETHIAN, J.A. & VLADIMIRSKY, A., Ordered upwind methods for static Hamilton–Jacobi equations, *Proc. Nat. Acad. Sci.* **98** (2001), 11069–11074. Zbl1002.65112 MR1854545
32. SETHIAN, J.A. & VLADIMIRSKY, A., Ordered upwind methods for static Hamilton–Jacobi equations: Theory & algorithms, *SIAM J. on Numerical Analysis* **41** (2003), 325–363. Zbl1040.65088 MR1974505

33. SORAVIA, P., Boundary value problems for Hamilton–Jacobi equations with discontinuous Lagrangian, *Indiana Univ. Math. J.* **51** (2002), 451–477. Zbl1032.35055 MR1909297
34. TSAI, Y.-H.R., CHENG, L.-T., OSHER, S., & ZHAO, H.-K., Fast sweeping algorithms for a class of Hamilton–Jacobi equations, *SIAM J. Numer. Anal.* **41** (2003) 673–694. Zbl1049.35020 MR2004194
35. TSITSIKLIS, J.N., Efficient algorithms for globally optimal trajectories, Proceedings, *IEEE 33rd Conference on Decision and Control*, 1368–1373, Lake Buena Vista, Florida (1994).
36. TSITSIKLIS, J. N., Efficient algorithms for globally optimal trajectories, IEEE TRAN. AUTOMATIC CONTROL **40** (1995), 1528–1538. Zbl0831.93028 MR1347833
37. POLYCHRONOPOULOS, G. H. & TSITSIKLIS, J. N., Stochastic shortest path problems with recourse, *Networks* **27** (1996), 133–143. Zbl0851.90129 MR1375843
38. VLADIMIRSKY, A., Label-setting methods for multimode stochastic shortest path problems on graphs, *Mathematics of Operations Research* **33** (2008), 821–838. Zbl1218.90205 MR2464644
39. ZHANG, L., MASON, A. & PHILPOTT, A., Optimization of a single ambulance move up, Technical report, University of Auckland Faculty of Engineering (2010).
40. ZHAO, H. K., Fast sweeping method for Eikonal equations, *Math. Comp.* **74** (2005), 603–627. Zbl1070.65113 MR2114640