Robust Shortest Path Planning and Semicontractive Dynamic Programming

Dimitri P. Bertsekas

Department of Electrical Engineering and Computer Science, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Massachusetts 02139

Received 3 February 2015; revised 16 June 2016; accepted 11 July 2016
DOI 10.1002/nav.21697
Published online 8 August 2016 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: In this article, we consider shortest path problems in a directed graph where the transitions between nodes are subject to uncertainty. We use a minimax formulation, where the objective is to guarantee that a special destination state is reached with a minimum cost path under the worst possible instance of the uncertainty. Problems of this type arise, among others, in planning and pursuit-evasion contexts, and in model predictive control. Our analysis makes use of the recently developed theory of abstract semicontractive dynamic programming models. We investigate questions of existence and uniqueness of solution of the optimality equation, existence of optimal paths, and the validity of various algorithms patterned after the classical methods of value and policy iteration, as well as a Dijkstra-like algorithm for problems with nonnegative arc lengths. © 2016 Wiley Periodicals, Inc. Naval Research Logistics 66: 15–37, 2019

Keywords: shortest path planning; minimax formulation; dynamic programming; semicontractive model

1. INTRODUCTION

In this article, we discuss shortest path problems that embody a worst-case view of uncertainty. These problems relate to several other types of problems arising in stochastic and minimax control, model predictive control, Markovian decision processes, planning, sequential games, robust and combinatorial optimization, and solution of discretized large-scale differential equations. Consequently, our analysis and algorithms relate to a large body of existing theory. However, in this article, we rely on a recently developed abstract dynamic programming (DP) theory of semicontractive problems, and capitalize on general results developed in the context of this theory [31]. We first discuss informally these connections and we survey the related literature.

1.1. Relations with Other Problems and Literature Review

The closest connection to our work is the classical shortest path problem where the objective is to reach a destination node with a minimum length path from every other node in a directed graph. This is a fundamental problem that has an enormous range of applications and has been studied extensively (see e.g., the surveys [43, 48], and many textbooks, including [2, 28, 29, 74]). The assumption is that at any node x, we may determine a successor node y from a given set of possible successors, defined by the arcs (x, y) of the graph that are outgoing from x.

In some problems, however, following the decision at a given node, there is inherent uncertainty about the successor node. In a stochastic formulation, the uncertainty is modeled by a probability distribution over the set of successors, and our decision is then to choose at each node one distribution out of a given set of distributions. The resulting problem, known as stochastic shortest path problem (also known as transient programming problem), is a total cost infinite horizon Markovian decision problem, with a substantial analytical and algorithmic methodology, which finds extensive applications in problems of motion planning, robotics, and other problems where the aim is to reach a goal state with probability 1 under stochastic uncertainty (see Refs. 18, 19, 23, 30, 35, 36, 42, 51, 53, 54, 58, 67, 70, 72, 82, 85). Another important area of application is large-scale computation for discretized versions of differential equations (such as the Hamilton-Jacobi-Bellman equation, and the eikonal equation); see [3, 4, 8, 39, 45, 49, 56, 62, 65, 75, 76 79, 81].

In this article, we introduce a sequential minimax formulation of the shortest path problem, whereby the uncertainty is modeled by set membership: at a given node, we may choose one subset out of a given collection of subsets of nodes, and the successor node on the path is chosen from this subset by an antagonistic opponent. Our principal method

Correspondence to: Dimitri P. Bertsekas (dimitrib@mit.edu)

of analysis is dynamic programming (DP for short). Related problems have been studied for a long time, in the context of control of uncertain discrete-time dynamic systems with a set membership description of the uncertainty (starting with the theses [24, 84], and followed up by many other works; see e.g., the monographs [11, 33, 57], the survey [34], and the references given there). These problems are relevant for example in the context of model predictive control under uncertainty, a subject of great importance in the current practice of control theory (see e.g., the surveys [60, 64], and the books [37, 61, 73]; model predictive control with set membership disturbances is discussed in the thesis [55] and the text [29], Section 6.5.2).

Sequential minimax problems have also been studied in the context of sequential games (see, e.g., the books [11, 46], and the references given there). Sequential games that involve shortest paths are particularly relevant; see the works [12, 50, 66, 88]. An important difference with some of the works on sequential games is that in our minimax formulation, we assume that the antagonistic opponent knows the decision and corresponding subset of successor nodes chosen at each node. Thus in our problem, it would make a difference if the decisions at each node were made with advance knowledge of the opponent's choice ("min-max" is typically not equal to "max-min" in our context). Generally shortest path games admit a simpler analysis when the arc lengths are assumed nonnegative (as is done for example in the recent works [12, 50]), when the problem inherits the structure of negative DP (see [77], or the texts [30, 72]) or abstract monotone increasing abstract DP models (see [16, 25, 31]). However, our formulation and line of analysis is based on the recently introduced abstract semicontractive DP model of [31], and allows negative as well as nonnegative arc lengths. Problems with negative arc lengths arise in applications when we want to find the longest path in a network with nonnegative arc lengths, such as critical path analysis. Problems with both positive and negative arc lengths include searching a network for objects of value with positive search costs (cf. Example 4.3), and financial problems of maximization of total reward when there are transaction and other costs.

An important application of our shortest path problems is in pursuit-evasion (or search and rescue) contexts, whereby a team of "pursuers" are aiming to reach one or more "evaders" that move unpredictably. Problems of this kind have been studied extensively from different points of view (see e.g., Refs. 1, 6, 7, 10, 12, 13, 47, 52, 58, 59, 68, 80). For our shortest path formulation to be applicable to such a problem, the pursuers and the evaders must have perfect information about each others' positions, and the Cartesian product of their positions (the state of the system) must be restricted to the finite set of nodes of a given graph, with known transition costs (i.e., a "terrain map" that is known a priori).

We may deal with pursuit-evasion problems with imperfect state information and set-membership uncertainty by means of a reduction to perfect state information, which is based on set membership estimators and the notion of a sufficiently informative function, introduced in the thesis [24] and in the subsequent paper [15]. In this context, the original imperfect state information problem is reformulated as a problem of perfect state information, where the states correspond to subsets of nodes of the original graph (the set of states that are consistent with the observation history of the system, in the terminology of set membership estimation [14, 24, 57]). Thus, since X has a finite number of nodes, the reformulated problem still involves a finite (but much larger) number of states, and may be dealt with using the methodology of this article. Note that the problem reformulation just described is also applicable to general minimax control problems with imperfect state information, not just to pursuit-evasion problems.

Our work is also related to the subject of robust optimization (see e.g., the book [9] and the recent survey [5]), which includes minimax formulations of general optimization problems with set membership uncertainty. However, our emphasis here is placed on the presence of the destination node and the requirement for termination, which is the salient feature and the essential structure of shortest path problems. Moreover, a difference with other works on robust shortest path (RSP) selection (see e.g., [17, 63, 87]) is that in our work the uncertainty about transitions or arc cost data at a given node is decoupled from the corresponding uncertainty at other nodes. This allows a DP formulation of our problem.

Because our context differs in essential respects from the preceding works, the results of the present paper are new to a great extent. The line of analysis is also new, and is based on the connection with the theory of abstract semicontractive DP mentioned earlier. In addition to simpler proofs, a major benefit of this abstract line of treatment is deeper insight into the structure of our problem, and the nature of our analytical and computational results. Several related problems, involving for example an additional stochastic type of uncertainty, admit a similar treatment. Some of these problems are described in the last section, and their analysis and associated algorithms are subjects for further research.

1.2. Robust Shortest Path Problem Formulation

To formally describe our problem, we consider a graph with a finite set of nodes $X \cup \{t\}$ and a finite set of directed arcs $\mathcal{A} \subset \{(x, y) | x, y \in X \cup \{t\}\}$, where t is a special node called the *destination*. At each node $x \in X$ we may choose a control or action u from a nonempty set $U(x)$, which is a subset of a finite set U. Then a successor node y is selected by an antagonistic opponent from a nonempty set $Y(x, u) \subset X \cup \{t\}$, such that $(x, y) \in \mathcal{A}$ for all $y \in Y(x, u)$, and a cost $g(x, u, y)$ is

incurred. The destination node t is absorbing and cost-free, in the sense that the only outgoing arc from t is (t, t) and we have $g(t, u, t) = 0$ for all $u \in U(t)$.

A policy is defined to be a function μ that assigns to each node $x \in X$ a control $\mu(x) \in U(x)$. We denote the finite set of all policies by $\mathcal{M}$. A *possible path under a policy* μ *starting at node* $x_0 \in X$ is an arc sequence of the form

$$p = \{(x_0, x_1), (x_1, x_2), \dots\},$$

such that $x_{k+1} \in Y(x_k, \mu(x_k))$ for all $k \geq 0$. The set of all possible paths under μ starting at x_0 is denoted by $P(x_0, \mu)$; it is the set of paths that the antagonistic opponent may generate starting from x, once policy μ has been chosen. The length of a path $p \in P(x_0, \mu)$ is defined by

$$L_\mu(p) = \sum_{k=0}^{\infty} g(x_k, \mu(x_k), x_{k+1}),$$

if the series above is convergent, and more generally by

$$L_\mu(p) = \limsup_{m\to\infty} \sum_{k=0}^{m} g(x_k, \mu(x_k), x_{k+1}),$$

if it is not. For completeness, we also define the length of a portion

$$\{(x_i, x_{i+1}), (x_{i+1}, x_{i+2}), \dots, (x_m, x_{m+1})\}$$

of a path $p \in P(x_0, \mu)$, consisting of a finite number of consecutive arcs, by

$$\sum_{k=i}^{m} g(x_k, \mu(x_k), x_{k+1}).$$

When confusion cannot arise we will also refer to such a finite-arc portion as a path. Of special interest are *cycles*, that is, paths of the form $\{(x_i, x_{i+1}), (x_{i+1}, x_{i+2}), \dots, (x_{i+m}, x_i)\}$. Paths that do not contain any cycle other than the self-cycle (t, t) are called *simple*.

For a given policy μ and $x_0 \neq t$, a path $p \in P(x_0, \mu)$ is said to be *terminating* if it has the form

$$p = \{(x_0, x_1), (x_1, x_2), \dots, (x_m, t), (t, t), \dots\}, \qquad (1.1)$$

where m is a positive integer, and $x_0, \dots, x_m$ are distinct non-destination nodes. Since $g(t, u, t) = 0$ for all $u \in U(t)$, the length of a terminating path p of the form (1.1), corresponding to μ, is given by

$$L_\mu(p) = g(x_m, \mu(x_m), t) + \sum_{k=0}^{m-1} g(x_k, \mu(x_k), x_{k+1}),$$

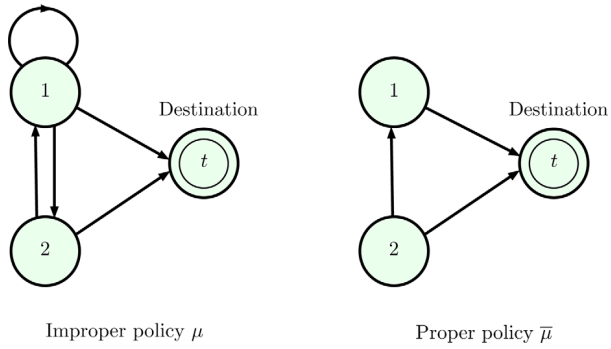


Figure 1. A RSP problem with $X = \{1, 2\}$, two controls at node 1, and one control at node 2. There are two policies, μ and $\overline{\mu}$, corresponding to the two controls at node 1. The figure shows the subgraphs of arcs $\mathcal{A}_\mu$ and $\mathcal{A}_{\overline{\mu}}$. The policy μ is improper because $\mathcal{A}_\mu$ contains the cycle (1, 2, 1) and the (self-)cycle (1, 1). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

and is equal to the finite length of its initial portion that consists of the first $m + 1$ arcs.

An important characterization of a policy μ is provided by the subset of arcs

$$\mathcal{A}_\mu = \cup_{x\in X} \{(x, y) | y \in Y(x, \mu(x))\}.$$

Thus $\mathcal{A}_\mu$, together with the self arc (t, t), consists of the set of paths $\cup_{x\in X} P(x, \mu)$, in the sense that it contains this set of paths and no other paths. We say that $\mathcal{A}_\mu$ is *destination-connected* if for each $x \in X$ there exists a terminating path in $P(x, \mu)$. We say that μ is *proper* if the subgraph of arcs $\mathcal{A}_\mu$ is acyclic (i.e., contains no cycles). Thus μ is proper if and only if all the paths in $\cup_{x\in X} P(x, \mu)$ are simple and hence terminating (equivalently μ is proper if and only if $\mathcal{A}_\mu$ is destination-connected and has no cycles). The term "proper" is consistent with a similar term in stochastic shortest path problems, where it indicates a policy under which the destination is reached with probability 1, see e.g., [18, 19, 67]. If μ is not proper, it is called *improper*, in which case the subgraph of arcs $\mathcal{A}_\mu$ must contain a cycle; see the examples of Fig. 1.

For a proper μ, we associate with every $x \in X$ the worst-case path length over the finite set of possible paths starting from x, which is denoted by

$$J_\mu(x) = \max_{p\in P(x,\mu)} L_\mu(p), \qquad x \in X. \qquad (1.2)$$

Thus $J_\mu(x)$ is the length of the *longest* path from x to t in the acyclic subgraph of arcs $\mathcal{A}_\mu$. Since there are finitely many paths in this acyclic graph, $J_\mu(x)$ may be found either by enumeration and comparison of these paths (in simple cases), or by solving the shortest path problem obtained when the signs of the arc lengths $g(x, \mu(x), y)$, $(x, y) \in \mathcal{A}_\mu$, are reversed.

Our problem is to find an optimal proper policy, i.e., one that minimizes $J_\mu(x)$ over all proper μ, simultaneously for all $x \in X$, under assumptions that parallel those for the classical shortest path problem. We refer to this as the problem of *robust shortest path* (RSP for short) selection. Note that in our problem, *reaching the destination starting from every node is a requirement*, regardless of the choices of the hypothetical antagonistic opponent. In other words the minimization in RSP is over the proper policies only.

Of course for the problem to have a feasible solution and thus be meaningful, there must exist at least one proper policy, and this may be restrictive for a given problem. One may deal with cases where feasibility is not known to hold by introducing for every x an artificial "termination action" $\bar{u}$ into $U(x)$ [i.e., a $\bar{u}$ with $Y(x,\bar{u}) = \{t\}$], associated with very large length [i.e., $g(x,\bar{u},t) = \bar{g} >> 1$]. Then the policy $\overline{\mu}$ that selects the termination action at each x is proper and has cost function $J_{\overline{\mu}}(x) \equiv \bar{g}$. In the problem thus reformulated the optimal cost over proper policies will be unaffected for all nodes x for which there exists a proper policy μ with $J_\mu(x) < \bar{g}$. Since for a proper μ, the cost $J_\mu(x)$ is bounded above by the number of nodes in X times the largest arc length, a suitable value of $\bar{g}$ is readily available.

In Section 2, we will formulate RSP in a way that the semicontractive DP framework can be applied. In Section 3, we will describe briefly this framework and we will quote the results that will be useful to us. In Section 4, we will develop our main analytical results for RSP. In Section 5, we will discuss algorithms of the value and policy iteration (PI) type, by specializing corresponding algorithms of semicontractive DP, and by adapting available algorithms for stochastic shortest path problems. Among others, we will give a Dijkstra-like algorithm for problems with nonnegative arc lengths, which terminates in a number of iterations equal to the number of nodes in the graph, and has low order polynomial complexity. Related Dijkstra-like algorithms were proposed recently, in the context of dynamic games and with an abbreviated convergence analysis, by [12, 50].

2. MINIMAX FORMULATION

In this section, we will reformulate RSP into a minimax problem, whereby given a policy μ, an antagonistic opponent selects a successor node $y \in Y(x,\mu(x))$ for each $x \in X$, with the aim of maximizing the lengths of the resulting paths. The essential difference between RSP and the associated minimax problem is that *in RSP only the proper policies are admissible, while in the minimax problem all policies will be admissible*. Our analysis will be based in part on assumptions under which improper policies cannot be optimal for the minimax problem, implying that optimal policies for the minimax problem will be optimal for the original RSP problem. One such assumption is the following.

ASSUMPTION 2.1:

a. There exists at least one proper policy.
b. For every improper policy μ, all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have positive length.

The preceding assumption parallels and generalizes the typical assumptions in the classical deterministic shortest path problem, i.e., the case where $Y(x,\mu)$ consists of a single node. Then condition (a) is equivalent to assuming that each node is connected to the destination with a path, while condition (b) is equivalent to assuming that all directed cycles in the graph have positive length.[1] Later in Section 4, in addition to Assumption 2.1, we will consider another weaker assumption, whereby "positive length" is replaced with "nonnegative length" in condition (b) above. This assumption will hold in the common case where all arc lengths $g(x,u,y)$ are nonnegative, but there may exist a zero length cycle. As a first step, we extend the definition of the function J_μ to the case of an improper policy. Recall that for a proper policy μ, $J_\mu(x)$ has been defined by Eq. (1.2), as the length of the longest path $p \in P(x,\mu)$,

$$J_\mu(x) = \max_{p \in P(x,\mu)} L_\mu(p), \qquad x \in X. \tag{2.1}$$

We extend this definition to any policy μ, proper or improper, by defining $J_\mu(x)$ as

$$J_\mu(x) = \limsup_{k\to\infty} \sup_{p \in P(x,\mu)} L_p^k(\mu), \tag{2.2}$$

where $L_p^k(\mu)$ is the sum of lengths of the first k arcs in the path p. When μ is proper, this definition coincides with the one given earlier [cf. Eq. (2.1)]. Thus for a proper μ, J_μ is real-valued, and it is the unique solution of the optimality equation (or Bellman equation) for the longest path problem associated with the proper policy μ and the acyclic subgraph of arcs $\mathcal{A}_\mu$:

$$J_\mu(x) = \max_{y \in Y(x,\mu(x))} [g(x,\mu(x),y) + \tilde{J}_\mu(y)] \qquad x \in X, \tag{2.3}$$

[1] To verify the existence of a proper policy [condition (a)] one may apply a reachability algorithm, which constructs the sequence $\{N_k\}$ of sets

$$N_{k+1} = N_k \cup \{x \in X \cup \{t\} \mid \text{there exists} u \in U(x) \text{ with } Y(x,u) \subset N_k\},$$

starting with $N_0 = \{t\}$ (see [24]). A proper policy exists if and only if this algorithm stops with a final set $\cup_k N_k$ equal to $X \cup \{t\}$. If there is no proper policy, this algorithm will stop with $\cup_k N_k$ equal to a strict subset of $X \cup \{t\}$ of nodes starting from which there exists a terminating path under some policy. The problem may then be reformulated over the reduced graph consisting of the node set $\cup_k N_k$, so there will exist a proper policy in this reduced problem.

where we denote by $\tilde{J}_\mu$ the function given by

$$\tilde{J}_\mu(y) = \begin{cases} J_\mu(y) & \text{if } y \in X, \\ 0 & \text{if } y = t. \end{cases} \tag{2.4}$$

Any shortest path algorithm may be used to solve this longest path problem for a proper μ. However, when μ is improper, we may have $J_\mu(x) = \infty$, and the solution of the corresponding longest path problem may be problematic.

We will consider the problem of finding

$$J^*(x) = \min_{\mu \in \mathcal{M}} J_\mu(x), \qquad x \in X, \tag{2.5}$$

and a policy attaining the minimum above, simultaneously for all $x \in X$. Note that *the minimization is over all policies, in contrast with the RSP problem, where the minimization is over just the proper policies*.

2.1. Embedding Within an Abstract DP Model

We will now reformulate the minimax problem of Eq. (2.5) more abstractly, by expressing it in terms of the mapping that appears in Bellman's equation (2.3)-(2.4), thereby bringing to bear the theory of abstract DP. We denote by $E(X)$ the set of functions $J : X \mapsto [-\infty, \infty]$, and by $R(X)$ the set of functions $J : X \mapsto (-\infty, \infty)$. Note that since X is finite, $R(X)$ can be viewed as a finite-dimensional Euclidean space. We introduce the mapping $H : X \times U \times E(X) \mapsto [-\infty, \infty]$ given by

$$H(x, u, J) = \max_{y \in Y(x,u)} [g(x, u, y) + \tilde{J}(y)], \tag{2.6}$$

where for any $J \in E(X)$ we denote by $\tilde{J}$ the function given by

$$\tilde{J}(y) = \begin{cases} J(y) & \text{if } y \in X, \\ 0 & \text{if } y = t. \end{cases} \tag{2.7}$$

We consider for each policy μ, the mapping $T_\mu : E(X) \mapsto E(X)$, defined by

$$(T_\mu J)(x) = H(x, \mu(x), J), \qquad x \in X, \tag{2.8}$$

and we note that the fixed point equation $J_\mu = T_\mu J_\mu$ is identical to the Bellman equation (2.3). We also consider the mapping $T : E(X) \mapsto E(X)$ defined by

$$(TJ)(x) = \min_{u \in U(x)} H(x, u, J), \qquad x \in X, \tag{2.9}$$

also equivalently written as

$$(TJ)(x) = \min_{\mu \in \mathcal{M}} (T_\mu J)(x), \qquad x \in X. \tag{2.10}$$

We denote by T^k and T_μ^k the k-fold compositions of the mappings T and T_μ with themselves, respectively.

Let us consider the zero function, which we denote by $\bar{J}$:

$$\bar{J}(x) \equiv 0, \qquad x \in X.$$

Using Eqs. (2.6)–(2.8), we see that for any $\mu \in \mathcal{M}$ and $x \in X$, $(T_\mu^k \bar{J})(x)$ is the result of the k-stage DP algorithm that computes $\sup_{p \in P(x,\mu)} L_p^k(\mu)$, the length of the longest path under μ that starts at x and consists of k arcs, so that

$$(T_\mu^k \bar{J})(x) = \sup_{p \in P(x,\mu)} L_p^k(\mu), \qquad x \in X.$$

Thus the definition (2.2) of J_μ can be written in the alternative and equivalent form

$$J_\mu(x) = \limsup_{k \to \infty} (T_\mu^k \bar{J})(x), \qquad x \in X. \tag{2.11}$$

We are focusing on optimization over stationary policies because under the assumptions of this article (both Assumption 2.1 and the alternative assumptions of Section 4) the optimal cost function would not be improved by allowing nonstationary policies, as shown in [31], Chapter 3.[2]

The results that we will show under Assumption 2.1 generalize the main analytical results for the classical deterministic shortest path problem, and stated in abstract form, are the following:

a. J^* is the unique fixed point of T within $R(X)$, and we have $T^k J \to J^*$ for all $J \in R(X)$.
b. Only proper policies can be optimal, and there exists an optimal proper policy.[3]
c. A policy μ is optimal if and only if it attains the minimum in Eq. (2.10) for all $x \in X$ when $J = J^*$.

Proofs of these results from first principles are quite complex. However, fairly easy proofs can be obtained by embedding

[2] In the more general framework of [31], nonstationary Markov policies of the form $\pi = \{\mu_0, \mu_1, \dots\}$, with $\mu_k \in \mathcal{M}$, $k = 0, 1, \dots$, are allowed, and their cost function is defined by

$$J_\pi(x) = \limsup_{k \to \infty} (T_{\mu_0} \cdots T_{\mu_{k-1}} \bar{J})(x), \qquad x \in X,$$

where $T_{\mu_0} \cdots T_{\mu_{k-1}}$ is the composition of the mappings $T_{\mu_0}, \dots, T_{\mu_{k-1}}$. Moreover, $J^*(x)$ is defined as the infimum of $J_\pi(x)$ over all such π. However, under the assumptions of the present paper, this infimum is attained by a stationary policy (in fact one that is proper). Hence, attention may be restricted to stationary policies without loss of optimality and without affecting the results from [31] that will be used.

[3] Since the set of policies is finite, there exists a policy minimizing $J_\mu(x)$ over the set of proper policies μ, for each $x \in X$. However, the assertion here is stronger, namely that there exists a proper μ^* minimizing $J_\mu(x)$ over all $\mu \in \mathcal{M}$ and *simultaneously for all* $x \in X$, i.e., a proper μ^* with $J_{\mu^*} = J^*$

the problem of minimizing the function J_μ of Eq. (2.11) over $\mu \in \mathcal{M}$, within the abstract semicontractive DP framework introduced in [31]. In particular, we will use general results for this framework, which we will summarize in the next section.

3. SEMICONTRACTIVE DP ANALYSIS

We will now view the problem of minimizing over $\mu \in \mathcal{M}$ the cost function J_μ, given in the abstract form (2.11), as a special case of a semicontractive DP model. We first provide a brief review of this model, with a notation that corresponds to the one used in the preceding section.

The starting point is a set of states X, a set of controls U, and a control constraint set $U(x) \subset U$ for each $x \in X$. For the general framework of this section, X and U are arbitrary sets; we continue to use some of the notation of the preceding section to indicate the relevant associations. A policy is a mapping $\mu : X \mapsto U$ with $\mu(x) \in U(x)$ for all $x \in X$, and the set of all policies is denoted by $\mathcal{M}$. For each policy μ, we are given a mapping $T_\mu : E(X) \mapsto E(X)$ that is monotone in the sense that for any two $J, J' \in E(X)$,

$$J \leq J' \qquad \Rightarrow \qquad T_\mu J \leq T_\mu J'.$$

We define the mapping $T : E(X) \mapsto E(X)$ by

$$(TJ)(x) = \inf_{\mu \in \mathcal{M}} (T_\mu J)(x), \qquad x \in X.$$

The cost function of μ is defined as

$$J_\mu(x) = \limsup_{m \to \infty} (T_\mu^k \bar{J})(x), \qquad x \in X,$$

where $\bar{J}$ is some given function in $E(X)$. The objective is to find

$$J^*(x) = \inf_{\mu \in \mathcal{M}} J_\mu(x)$$

for each $x \in X$, and a policy μ such that $J_\mu = J^*$, if one exists. Based on the correspondences with Eqs. (2.6)–(2.11), it can be seen that the minimax problem of the preceding section is the special case of the problem of this section, where X and U are finite sets, T_μ is defined by Eq. (2.8), and $\bar{J}$ is the zero function.

In contractive models, the mappings T_μ are assumed to be contractions, with respect to a common weighted sup-norm and with a common contraction modulus, in the subspace of functions in $E(X)$ that are bounded with respect to the weighted sup-norm. These models have a strong analytical and algorithmic theory, which dates to [41]; see also [16], Ch. 3, and recent extensive treatments given in Chapters 1–3 of [30], and Ch. 2 of [31]. In semicontractive models, only some policies have a contraction-like property. This property is captured by the notion of S-regularity of a policy introduced in [31] and defined as follows.

DEFINITION 3.1: Given a set of functions $S \subset E(X)$, we say that a policy μ is *S-regular* if:

(a) $J_\mu \in S$ and $J_\mu = T_\mu J_\mu$.
(b) $\lim_{k \to \infty} T_\mu^k J = J_\mu$ for all $J \in S$.

A policy that is not S-regular is called *S-irregular*.

Roughly, μ is S-regular if J_μ is an asymptotically stable equilibrium point of T_μ within S. An important case of an S-regular μ is when S is a complete subset of a metric space and T_μ maps S to S and, when restricted to S, is a contraction with respect to the metric of that space.

There are several different choices of S, which may be useful depending on the context, such as for example $R(X)$, $E(X)$, $\{J \in R(X) | J \geq \bar{J}\}$, $\{J \in E(X) | J \geq \bar{J}\}$, and others. There are also several sets of assumptions and corresponding results, which are given in [31] and will be used to prove our analytical results for the RSP problem. In this article, we will use $S = R(X)$, but for ease of reference, we will quote results from [31] with S being an arbitrary subset of $R(X)$.

We give below an assumption relating to semicontractive models, which is Assumption 3.2.1 of [31]. A key part of this assumption is part (c), which implies that S-irregular policies have infinite cost for at least one state x, so they cannot be optimal. This part will provide a connection to Assumption 2.1(b).

ASSUMPTION 3.1: In the semicontractive model of this section with a set $S \subset R(X)$ the following hold:

a. S contains $\bar{J}$, and has the property that if J_1, J_2 are two functions in S, then S contains all functions J with $J_1 \leq J \leq J_2$.
b. The function $\hat{J}$ given by

$$\hat{J}(x) = \inf_{\mu:\ S\text{-regular}} J_\mu(x), \qquad x \in X,$$

belongs to S.
c. For each S-irregular policy μ and each $J \in S$, there is at least one state $x \in X$ such that

$$\limsup_{k \to \infty} (T_\mu^k J)(x) = \infty.$$

d. The control set U is a metric space, and the set

$$\{\mu(x) | (T_\mu J)(x) \leq \lambda\}$$

is compact for every $J \in S$, $x \in X$, and $\lambda \in \Re$.

e. For each sequence $\{J_m\} \subset S$ with $J_m \uparrow J$ for some $J \in S$ we have

$$\lim_{m\to\infty}(T_\mu J_m)(x) = (T_\mu J)(x), \qquad \forall\, x \in X,\ \mu \in \mathcal{M}.$$

f. For each function $J \in S$, there exists a function $J' \in S$ such that $J' \le J$ and $J' \le TJ'$.

The following two propositions are given in [31] as Prop. 3.2.1 and Lemma 3.2.4, respectively.[4] Our analysis will be based on these two propositions.

PROPOSITION 3.1: Let Assumption 3.1 hold. Then:

a. The optimal cost function J^* is the unique fixed point of T within the set S.
b. A policy μ^* is optimal if and only if $T_{\mu^*}J^* = TJ^*$. Moreover, there exists an optimal S-regular policy.
c. We have $T^kJ \to J^*$ for all $J \in S$.
d. For any $J \in S$, if $J \le TJ$ we have $J \le J^*$, and if $J \ge TJ$ we have $J \ge J^*$.

PROPOSITION 3.2: Let Assumption 3.1(b),(c),(d) hold. Then:

a. The function $\hat{J}$ of Assumption 3.1(b) is the unique fixed point of T within S.
b. Every policy μ satisfying $T_\mu\hat{J} = T\hat{J}$ is optimal within the set of S-regular policies, i.e., μ is S-regular and $J_\mu = \hat{J}$. Moreover, there exists at least one such policy.

The second proposition is useful for situations where only some of the conditions of Assumption 3.1 are satisfied, and will be useful in the proof of an important part of Prop. 4.3 in the next section.

4. SEMICONTRACTIVE MODELS AND SHORTEST PATH PROBLEMS

We will now apply the preceding two propositions to the minimax formulation of the RSP problem: minimizing over all $\mu \in \mathcal{M}$ the shortest path cost $J_\mu(x)$ as given by Eq. 3.2 for both proper and improper policies. We will first derive some preliminary results. The following proposition clarifies the properties of J_μ when μ is improper.

[4] As noted in the preceding section, a more general problem is defined in [31], whereby nonstationary Markov policies are allowed, and J^* is defined as the infimum over these policies. However, under our assumptions, attention may be restricted to stationary policies without loss of optimality and without affecting the validity of the two propositions.

PROPOSITION 4.1: Let μ be an improper policy and let J_μ be its cost function as given by Eq. (2.2).

a. If all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have nonpositive length, $J_\mu(x) < \infty$ for all $x \in X$.
b. If all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have nonnegative length, $J_\mu(x) > -\infty$ for all $x \in X$.
c. If all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have zero length, J_μ is real-valued.
d. If there is a positive length cycle in the subgraph of arcs $\mathcal{A}_\mu$, we have $J_\mu(x) = \infty$ for at least one node $x \in X$. More generally, for each $J \in R(X)$, we have $\limsup_{k\to\infty}(T_\mu^kJ)(x) = \infty$ for at least one $x \in X$.

PROOF: Any path with a finite number of arcs, can be decomposed into a simple path, and a finite number of cycles (see e.g., the path decomposition theorem of [28], Prop. 1.1, and Exercise 1.4). Since there is only a finite number of simple paths under μ, their length is bounded above and below. Thus in part (a) the length of all paths with a finite number of arcs is bounded above, and in part (b) it is bounded below, implying that $J_\mu(x) < \infty$ for all $x \in X$ or $J_\mu(x) > -\infty$ for all $x \in X$, respectively. Part (c) follows by combining parts (a) and (b).

To show part (d), consider a path p, which consists of an infinite repetition of the positive length cycle that is assumed to exist. Let $C_\mu^k(p)$ be the length of the path that consists of the first k cycles in p. Then $C_\mu^k(p) \to \infty$ and $C_\mu^k(p) \le J_\mu(x)$ for all k [cf. Eq. (2.2)], where x is the first node in the cycle, thus implying that $J_\mu(x) = \infty$. Moreover for every $J \in R(X)$ and all k, $(T_\mu^kJ)(x)$ is the maximum over the lengths of the k-arc paths that start at x, plus a terminal cost that is equal to either $J(y)$ (if the terminal node of the k-arc path is $y \in X$), or 0 (if the terminal node of the k-arc path is the destination). Thus we have,

$$(T_\mu^k\bar{J})(x) + \min\left\{0,\ \min_{x\in X}J(x)\right\} \le (T_\mu^kJ)(x).$$

Since $\limsup_{k\to\infty}(T_\mu^k\bar{J})(x) = J_\mu(x) = \infty$ as shown earlier, it follows that $\limsup_{k\to\infty}(T_\mu^kJ)(x) = \infty$ for all $J \in R(X)$. □

Note that if there is a negative length cycle in the subgraph of arcs $\mathcal{A}_\mu$, it is not necessarily true that for some $x \in X$ we have $J_\mu(x) = -\infty$. Even for x on the negative length cycle, the value of $J_\mu(x)$ is determined by the *longest* path in $P(x,\mu)$, which may be simple in which case $J_\mu(x)$ is a real number, or contain an infinite repetition of a positive length cycle in which case $J_\mu(x) = \infty$.

A key fact in our analysis is the following characterization of the notion of $R(X)$-regularity and its connection to the notion of properness. It shows that proper policies are

$R(X)$-regular, but the set of $R(X)$-regular policies may contain some improper policies, which are characterized in terms of the sign of the lengths of their associated cycles.

PROPOSITION 4.2: Consider the minimax formulation of the RSP problem, viewed as a special case of the abstract semicontractive DP model of Section 3.1 with T_μ given by Eqs. (2.6)–(2.8), and $\bar{J}$ being the zero function. The following are equivalent for a policy μ:

1. μ is $R(X)$-regular.
2. The subgraph of arcs $\mathcal{A}_\mu$ is destination-connected and all its cycles have negative length.
3. μ is either proper or else, if it is improper, all the cycles of the subgraph of arcs $\mathcal{A}_\mu$ have negative length, and $J_\mu \in R(X)$.

PROOF: To show that (1) implies (2), let μ be $R(X)$-regular and to arrive at a contradiction, assume that $\mathcal{A}_\mu$ contains a nonnegative length cycle. Let x be a node on the cycle, consider the path p that starts at x and consists of an infinite repetition of this cycle, and let $L^k_\mu(p)$ be the length of the first k arcs of that path. Let also J be a nonzero constant function, $J(x) \equiv r$, where r is a scalar. Then we have

$$L^k_\mu(p) + r \le (T^k_\mu J)(x),$$

since from the definition of T_μ, we have that $(T^k_\mu J)(x)$ is the maximum over the lengths of all k-arc paths under μ starting at x, plus r, if the last node in the path is not the destination. Since μ is $R(X)$-regular, we have $\limsup_{k\to\infty}(T^k_\mu J)(x) = J_\mu(x) < \infty$, so that for all scalars r,

$$\limsup_{k\to\infty}(L^k_\mu(p) + r) \le J_\mu(x) < \infty.$$

Taking infimum over $r \in \Re$, we have $\limsup_{k\to\infty} L^k_\mu(p) = -\infty$, which contradicts the nonnegativity of the cycle of p. Thus all cycles of $\mathcal{A}_\mu$ have negative length. To show that $\mathcal{A}_\mu$ is destination-connected, assume the contrary. Then there exists some node $x \in X$ such that all paths in $P(x,\mu)$ contain an infinite number of cycles. Since the length of all cycles is negative, as just shown, it follows that $J_\mu(x) = -\infty$, which contradicts the $R(X)$-regularity of μ.

To show that (2) implies (3), we assume that μ is improper and show that $J_\mu \in R(X)$. By (2) $\mathcal{A}_\mu$ is destination-connected, so the set $P(x,\mu)$ contains a simple path for all $x \in X$. Moreover, since by (2) the cycles of $\mathcal{A}_\mu$ have negative length, each path in $P(x,\mu)$ that is not simple has smaller length than some simple path in $P(x,\mu)$. This implies that $J_\mu(x)$ is equal to the largest path length among simple paths in $P(x,\mu)$, so $J_\mu(x)$ is a real number for all $x \in X$.

To show that (3) implies (1), we note that if μ is proper, it is $R(X)$-regular, so we focus on the case where μ is improper. Then by (3), $J_\mu \in R(X)$, so to show $R(X)$-regularity of μ, we must show that $(T^k_\mu J)(x) \to J_\mu(x)$ for all $x \in X$ and $J \in R(X)$, and that $J_\mu = T_\mu J_\mu$. Indeed, from the definition of T_μ, we have

$$(T^k_\mu J)(x) = \sup_{p\in P(x,\mu)} [L^k_\mu(p) + J(x^k_p)], \tag{4.1}$$

where x^k_p is the node reached after k arcs along the path p, and $J(t)$ is defined to be equal to 0. Thus as $k \to \infty$, for every path p that contains an infinite number of cycles (each necessarily having negative length), the sequence $L^k_p(\mu) + J(x^k_p)$ approaches $-\infty$. It follows that for sufficiently large k, the supremum in Eq. (4.1) is attained by one of the simple paths in $P(x,\mu)$, so $x^k_p = t$ and $J(x^k_p) = 0$. Thus the limit of $(T^k_\mu J)(x)$ does not depend on J, and is equal to the limit of $(T^k_\mu \bar{J})(x)$, i.e., $J_\mu(x)$. To show that $J_\mu = T_\mu J_\mu$, we note that by the preceding argument, $J_\mu(x)$ is the length of the longest path among paths that start at x and terminate at t. Moreover, we have

$$(T_\mu J_\mu)(x) = \max_{y\in Y(x,\mu(x))}[g(x,\mu(x),y) + J_\mu(y)],$$

where we denote $J_\mu(t) = 0$. Thus $(T_\mu J_\mu)(x)$ is also the length of the longest path among paths that start at x and terminate at t, and hence it is equal to $J_\mu(x)$. □

We illustrate the preceding proposition with a two-node example involving an improper policy with a cycle that may have positive, zero, or negative length.

EXAMPLE 4.1: Let $X = \{1\}$, and consider the policy μ where at state 1, the antagonistic opponent may force either staying at 1 or terminating, i.e., $Y(1,\mu(1)) = \{1,t\}$. Then μ is improper since its subgraph of arcs $\mathcal{A}_\mu$ contains the self-cycle (1, 1); cf. Fig. 2. Let

$$g(1,\mu(1),1) = a, \qquad g(1,\mu(1),t) = 0.$$

Then,

$$(T_\mu J_\mu)(1) = \max[0,\ a + J_\mu(1)],$$

and

$$J_\mu(1) = \begin{cases} \infty & \text{if } a > 0, \\ 0 & \text{if } a \le 0. \end{cases}$$

Consistently with Prop. 4.2, the following hold:

a. For $a > 0$, the cycle (1, 1) has positive length, and μ is $R(X)$-irregular because $J_\mu(1) = \infty$.

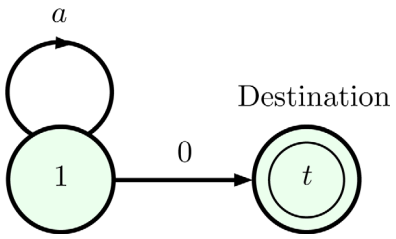


Figure 2. The subgraph of arcs $\mathcal{A}_\mu$ corresponding to an improper policy μ, for the case of a single node 1 and a destination node t. The arcs lengths are shown in the figure. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

b. For $a=0$, the cycle (1, 1) has zero length, and μ is $R(X)$-irregular because for a function $J \in R(X)$ with $J(1) > 0$,

$$\limsup_{k\to\infty}(T_\mu^k J)(x) = J(1) > 0 = J_\mu(1).$$

c. For $a < 0$, the cycle (1, 1) has negative length, and μ is $R(X)$-regular because $J_\mu(1) = 0$, and we have $J_\mu \in R(X)$, $J_\mu(1) = \max[0, a + J_\mu(1)] = (T_\mu J_\mu)(1)$, and for all $J \in R(X)$,

$$\limsup_{k\to\infty}(T_\mu^k J)(1) = 0 = J_\mu(1).$$

We now show one of our main results.

PROPOSITION 4.3: Let Assumption 2.1 hold. Then:

a. The optimal cost function J^* of RSP is the unique fixed point of T within $R(X)$.
b. A policy μ^* is optimal for RSP if and only if $T_{\mu^*}J^* = TJ^*$. Moreover, there exists an optimal proper policy.
c. We have $T^k J \to J^*$ for all $J \in R(X)$.
d. For any $J \in R(X)$, if $J \le TJ$ we have $J \le J^*$, and if $J \ge TJ$ we have $J \ge J^*$.

PROOF: We verify the parts (a)-(f) of Assumption 3.1 with $S = R(X)$. The result then will be proved using Prop. 3.1. To this end we argue as follows:

1. Part (a) is satisfied since $S = R(X)$.
2. Part (b) is satisfied since by Assumption 2.1(a), there exists at least one proper policy, which by Prop. 4.2 is $R(X)$-regular. Moreover, for each $R(X)$-regular policy μ, we have $J_\mu \in R(X)$. Since the number of all policies is finite, it follows that $\hat{J} \in R(X)$.
3. To show that part (c) is satisfied, note that since by Prop. 4.2 every $R(X)$-irregular policy μ must be improper, it follows from Assumption 2.1(b) that the subgraph of arcs $\mathcal{A}_\mu$ contains a cycle of positive length. By Prop. 4.1(d), this implies that for each $J \in R(X)$, we have $\limsup_{k\to\infty}(T_\mu^k J)(x) = \infty$ for at least one $x \in X$.
4. Part (d) is satisfied since $U(x)$ is a finite set.
5. Part (e) is satisfied since X is finite and T_μ is a continuous function mapping the finite-dimensional space $R(X)$ into itself.
6. To show that part (f) is satisfied, we note that by applying Prop. 3.2 with $S = R(X)$, we have that $\hat{J}$ is the unique fixed point of T within $R(X)$. It follows that for each $J \in R(X)$, there exists a sufficiently large scalar $r > 0$ such that the function J' given by

$$J' = \hat{J} - re, \qquad \forall\, x \in X, \tag{4.2}$$

where e is the unit function, $e(x) \equiv 1$, satisfies $J' \le J$ as well as

$$J' = \hat{J} - re = T\hat{J} - re \le T(\hat{J} - re) = TJ', \tag{4.3}$$

where the inequality holds in view of Eqs. (2.6) and (2.9), and the fact $r > 0$.

Thus all parts of Assumption 3.1 with $S = R(X)$ are satisfied, and Prop. 3.1 applies with $S = R(X)$. Since under Assumption 2.1, improper policies are $R(X)$-irregular [cf. Prop. 4.1(d)] and so cannot be optimal, the minimax formulation of Section 2 is equivalent to RSP, and the conclusions of Prop. 3.1 are precisely the results we want to prove. □

The following variant of the two-node Example 4.1 illustrates what may happen in the absence of Assumption 2.1(b), when there may exist improper policies that involve a nonpositive length cycle.

EXAMPLE 4.2: Let $X = \{1\}$, and consider the improper policy μ with $Y(1, \mu(1)) = \{1, t\}$ and the proper policy $\overline{\mu}$ with $Y(1, \overline{\mu}(1)) = \{t\}$ (cf. Fig. 3). Let

$$g(1, \mu(1), 1) = a \le 0, \qquad g(1, \mu(1), t) = 0,$$
$$g(1, \overline{\mu}(1), t) = 1.$$

Then it can be seen that under both policies, the longest path from 1 to t consists of the arc $(1, t)$. Thus,

$$J_\mu(1) = 0, \qquad J_{\overline{\mu}}(1) = 1,$$

so the improper policy μ is optimal for the minimax problem (2.5), and strictly dominates the proper policy $\overline{\mu}$ (which is optimal for the RSP version of the problem). To explain what is happening here, we consider two different cases:

1. $a=0$: In this case, the optimal policy μ is both improper and $R(X)$-irregular, but with $J_\mu(1) < \infty$.

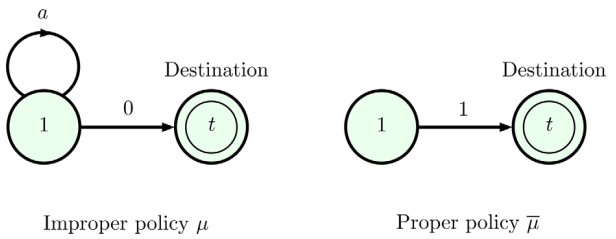


Figure 3. A counterexample involving a single node 1 in addition to the destination t. There are two policies, μ and $\overline{\mu}$, with corresponding subgraphs of arcs $\mathcal{A}_\mu$ and $\mathcal{A}_{\overline{\mu}}$, and arc lengths shown in the figure. The improper policy μ is optimal when $a \leq 0$. It is $R(X)$-irregular if $a=0$, and it is $R(X)$-regular if $a<0$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Thus the conditions of both Props. 3.1 and 4.3 do not hold because Assumptions 3.1(c) and Assumption 2.1(b) are violated.

2. $a<0$: In this case, μ is improper but $R(X)$-regular, so there are no $R(X)$-irregular policies. Then all the conditions of Assumption 3.1 are satisfied, and Prop. 3.1 applies. Consistent with this proposition, there exists an optimal $R(X)$-regular policy (i.e., optimal over both proper and improper policies), which however is improper and hence not an optimal solution for RSP.

We will next discuss modifications of Prop. 4.3, which address the difficulties illustrated in the two cases of the preceding example.

4.1. The Case of Improper Policies with Negative Length Cycles

We note that by Prop. 4.2, the set of $R(X)$-regular policies includes not just proper policies, but also some improper ones (those μ for which $\mathcal{A}_\mu$ is destination-connected and all its cycles have negative length). As a result we can weaken Assumption 2.1 as long as it still implies Assumption 3.1 so we can use Prop. 3.1 to obtain corresponding versions of our main result of Prop. 4.3. Here are two such weaker versions of Assumption 2.1.

ASSUMPTION 4.1: Every policy μ is either proper or else it is improper and its subgraph of arcs $\mathcal{A}_\mu$ is destination-connected with all cycles having negative length.

From Prop. 4.2, it follows that the preceding assumption is equivalent to all policies being $R(X)$-regular. The next assumption is weaker in that it allows policies μ that are $R(X)$-irregular, as long as some cycle of $\mathcal{A}_\mu$ has positive length.

ASSUMPTION 4.2:

a. There exists at least one $R(X)$-regular policy.
b. For every $R(X)$-irregular policy μ, some cycle in the subgraph of arcs $\mathcal{A}_\mu$ has positive length.

Now by essentially repeating the proof of Prop. 4.3, we see that Assumption 4.2 implies Assumption 3.1, so that Prop. 3.1 applies. Then we obtain the following variant of Prop. 4.3.

PROPOSITION 4.4: Let either Assumption 4.1 or (more generally) Assumption 4.2 hold. Then:

a. The function J^* of Eq. (2.11) is the unique fixed point of T within $R(X)$.
b. A policy μ^* satisfies $J_{\mu^*} = J^*$, where J^* is the minimum of J_μ over all $\mu \in \mathcal{M}$ [cf. Eq. (2.5)], if and only if $T_{\mu^*} J^* = T J^*$. Moreover, there exists an optimal $R(X)$-regular policy.
c. We have $T^k J \to J^*$ for all $J \in R(X)$.
d. For any $J \in R(X)$, if $J \leq T J$ we have $J \leq J^*$, and if $J \geq T J$ we have $J \geq J^*$.

It is important to note that the optimal $R(X)$-regular policy μ^* of part (b) above may not be proper, and hence needs to be checked to ensure that it solves the RSP problem (cf. Example 4.2 with $a<0$). Thus one would have to additionally prove that at least one of the optimal $R(X)$-regular policies is proper for the proposition to fully apply to RSP.

4.2. The Case of Improper Policies with Zero Length Cycles

In some problems, it may be easier to guarantee nonnegativity rather than positivity of the lengths of cycles corresponding to improper policies, which is required by Assumption 2.1(b). This is true for example in the important case where all arc lengths are nonnegative, i.e., $g(x,u,y) \geq 0$ for all $x \in X$, $u \in U(x)$, and $y \in Y(x,u)$, as in case (1) of Example 4.2. Let us consider the following relaxation of Assumption 2.1.

ASSUMPTION 4.3:

a. There exists at least one proper policy.
b. For every improper policy μ, all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have nonnegative length.

Note that similar to the case of Assumption 2.1, we may guarantee that part (a) of the preceding assumption is satisfied by introducing a high cost termination action at each node. Then the policy that terminates at each state is proper.

For an analysis under the preceding assumption, we will use a perturbation approach that was introduced in Section

3.2.2 of [31]. The idea is to consider a scalar $\delta > 0$ and a δ-perturbed problem, whereby each arc length $g(x,u,y)$ with $x \in X$ is replaced by $g(x,u,y)+\delta$. As a result, a nonnegative cycle length corresponding to an improper policy as per Assumption 4.3(b) becomes strictly positive, so Assumption 2.1 is satisfied for the δ-perturbed problem, and Prop. 4.3 applies. We thus see that J_δ^*, the optimal cost function of the δ-perturbed problem, is the unique fixed point of the mapping T_δ given by

$$(T_\delta J)(x) = \min_{u\in U(x)} H_\delta(x,u,J), \qquad x \in X,$$

where $H_\delta(x,u,J)$ is given by

$$H_\delta(x,u,J) = H(x,u,J) + \delta.$$

Moreover there exists an optimal proper policy μ_δ for the δ-perturbed problem, which by Prop. 4.3(b), satisfies the optimality equation

$$T_{\mu_\delta,\delta} J_\delta^* = T_\delta J_\delta^*,$$

where $T_{\mu,\delta}$ is the mapping that corresponds to a policy μ in the δ-perturbed problem:

$$(T_{\mu,\delta} J)(x) = H_\delta(x,\mu(x),J), \qquad x \in X.$$

We have the following proposition.

PROPOSITION 4.5: Let Assumption 4.3 hold, and let $\hat J$ be the optimal cost function over the proper policies only,

$$\hat J(x) = \min_{\mu:\ \text{proper}} J_\mu(x), \qquad x \in X.$$

Then:

a. $\hat J = \lim_{\delta\downarrow 0} J_\delta^*$.
b. $\hat J$ is the unique fixed point of T within the set $\left\{J \in R(X) \mid J \ge \hat J\right\}$.
c. We have $T^k J \to \hat J$ for every $J \in R(X)$ with $J \ge \hat J$.
d. Let μ be a proper policy. Then μ is optimal within the class of proper policies (i.e., $J_\mu = \hat J$) if and only if $T_\mu \hat J = T\hat J$.
e. There exists $\overline{\delta} > 0$ such that for all $\delta \in (0,\overline{\delta}]$, an optimal policy for the δ-perturbed problem is an optimal proper policy for the original RSP.

PROOF:

a. For all $\delta > 0$, consider an optimal proper policy μ_δ of the δ-perturbed problem, i.e., one with cost $J_{\mu_\delta,\delta} = J_\delta^*$. We have

$$\hat J \le J_{\mu_\delta} \le J_{\mu_\delta,\delta} = J_\delta^* \le J_{\mu',\delta} \le J_{\mu'} + N\delta, \qquad \forall\, \mu' : \text{proper},$$

where N is the number of nodes of X (since an extra δ cost is incurred in the δ-perturbed problem every time a path goes through a node $x \ne t$, and any path under a proper μ' contains at most N nodes $x \ne t$). By taking the limit as $\delta \downarrow 0$ and then the minimum over all μ' that are proper, it follows that

$$\hat J \le \lim_{\delta\downarrow 0} J_\delta^* \le \hat J,$$

so $\lim_{\delta\downarrow 0} J_\delta^* = \hat J$.

b. For all proper μ, we have $J_\mu = T_\mu J_\mu \ge T_\mu \hat J \ge T\hat J$. Taking minimum over proper μ, we obtain $\hat J \ge T\hat J$. Conversely, for all $\delta > 0$ and $\mu \in \mathcal{M}$, we have

$$J_\delta^* = T J_\delta^* + \delta e \le T_\mu J_\delta^* + \delta e.$$

Taking limit as $\delta \downarrow 0$, and using part (a), we obtain $\hat J \le T_\mu \hat J$ for all $\mu \in \mathcal{M}$. Taking minimum over $\mu \in \mathcal{M}$, it follows that $\hat J \le T\hat J$. Thus $\hat J$ is a fixed point of T. The uniqueness of $\hat J$ will follow once we prove part (c).

c. For all $J \in R(X)$ with $J \ge \hat J$ and proper policies μ, we have using the relation $\hat J = T\hat J$ just shown in part (b),

$$\hat J = \lim_{k\to\infty} T^k \hat J \le \lim_{k\to\infty} T^k J \le \lim_{k\to\infty} T_\mu^k J = J_\mu.$$

Taking the minimum over all proper μ, we obtain

$$\hat J \le \lim_{k\to\infty} T^k J \le \hat J, \qquad \forall\, J \ge \hat J.$$

d. If μ is a proper policy with $J_\mu = \hat J$, we have $\hat J = J_\mu = T_\mu J_\mu = T_\mu \hat J$, so, using also the relation $\hat J = T\hat J$ [cf. part (a)], we obtain $T_\mu \hat J = T\hat J$. Conversely, if μ satisfies $T_\mu \hat J = T\hat J$, then from part (a), we have $T_\mu \hat J = \hat J$ and hence $\lim_{k\to\infty} T_\mu^k \hat J = \hat J$. Since μ is proper, we have $J_\mu = \lim_{k\to\infty} T_\mu^k \hat J$, so $J_\mu = \hat J$.

e. For every proper policy μ we have $\lim_{\delta\downarrow 0} J_{\mu,\delta} = J_\mu$. Hence if a proper μ is not optimal for RSP, it is also nonoptimal for the δ-perturbed problem for all $\delta \in [0,\delta_\mu]$, where δ_μ is some positive scalar. Let $\overline{\delta}$ be the minimum δ_μ over the nonoptimal proper policies μ. Then for $\delta \in (0,\overline{\delta}]$, an optimal policy for the δ-perturbed problem cannot be nonoptimal for RSP. □

Note that we may have $J^*(x) < \hat J(x)$ for some x, but in RSP only proper policies are admissible, so by letting $\delta \downarrow 0$ we approach the optimal solution of interest. This happens

for instance in Example 4.2 when $a = 0$. For the same example $\hat{J}$ (not J^*) can be obtained as the limit of $T^k J$, starting from $J \geq \hat{J}$ [cf. part (c)]. The following example describes an interesting problem, where Prop. 4.5 applies.

EXAMPLE 4.3 (Minimax Search Problems) Consider searching a graph with node set $X \cup \{t\}$, looking for an optimal node $x \in X$ at which to stop. At each $x \in X$ we have two options: (1) stopping at a cost $s(x)$, which will stop the search by moving to t, or (2) continuing the search by choosing a control $u \in U(x)$, in which case we will move to a node y, chosen from within a given set of nodes $Y(x, u)$ by an antagonistic opponent, at a cost $g(x, u, y) \geq 0$. Then Assumption 4.3 holds, since there exists a proper policy (the one that stops at every x).

An interesting special case is when the stopping costs $s(x)$ are all nonnegative, while searching is cost-free [i.e., $g(x, u, y) \equiv 0$], but may lead in the future to nodes where a higher stopping cost will become inevitable. Then a policy that never stops is optimal but improper, but if we introduce a small perturbation $\delta > 0$ to the costs $g(x, u, y)$, we will make the lengths of all cycles positive, and Prop. 4.5 may be used to find an optimal policy within the class of proper policies. Note that this is an example where we are really interested in solving the RSP problem (where only the proper policies are admissible), and not its minimax version (where all policies are admissible).

5. COMPUTATIONAL METHODS

We will now discuss computational methods that are patterned after the classical DP algorithms of value iteration and policy iteration (VI and PI for short, respectively). In particular, the methods of this section are motivated by specialized stochastic shortest path algorithms.

5.1. Value Iteration Algorithms

We have already shown as part of Prop. 4.3 that under Assumption 2.1, the VI algorithm, which sequentially generates $T^k J$ for $k \geq 0$, converges to the optimal cost function J^* for any starting function $J \in R(X)$. We have also shown as part of Prop. 4.5 that under Assumption 4.3, the VI sequence $T^k J$ for $k \geq 0$, converges to $\hat{J}$, the optimal cost function over the proper policies only, for any starting function $J \geq \hat{J}$. We can extend these convergence properties to asynchronous versions of VI based on the monotonicity and fixed point properties of the mapping T. This has been known since the paper [26] (see also [18, 27]), and we refer to the discussions in Sections 2.6.1 and 3.3.1 of [31], which apply in their entirety when specialized to the RSP problem of this article.

It turns out that for our problem, under Assumption 2.1 or Assumption 4.3, the VI algorithm also terminates finitely when initialized with $J(x) = \infty$ for all $x \in X$ [it can be seen that in view of the form (2.9) of the mapping T, the VI algorithm is well-defined with this initialization]. In fact the number of iterations for termination is no more than N, where N is the number of nodes in X, leading to polynomial complexity. This is consistent with a similar result for stochastic shortest path problems ([30], Section 3.4.1), which relies on the assumption of acyclicity of the graph of possible transitions under an optimal policy. Because this assumption is restrictive, finite termination of the VI algorithm is an exceptional property in stochastic shortest path problems. However, in the minimax case of this article, an optimal policy μ^* exists and is proper [cf. Prop. 4.3(b) or Prop. 4.5(e)], so the graph of possible transitions under μ^* is acyclic, and it turns out that finite termination of VI is guaranteed to occur. Note that in deterministic shortest path problems the initialization $J(x) = \infty$ for all $x \in X$, leads to polynomial complexity, and generally works better in practice that other initializations (such as $J < J^*$, for which the complexity is only pseudopolynomial, cf. [18], Section 4.1, Prop. 1.2).

To show the finite termination property just described, let μ^* be an optimal proper policy, consider the sets of nodes $X_0, X_1, \ldots$, defined by

$$X_0 = \{t\},$$
$$X_{k+1} = \left\{x \notin \cup_{m=0}^{k} X_m \,|\, y \in \cup_{m=0}^{k} X_m \text{ for all } y \in Y(x, \mu^*(x))\right\}, \quad k = 0, 1, \ldots, \tag{5.1}$$

and let $X_{\bar{k}}$ be the last of these sets that is nonempty. Then in view of the acyclicity of the subgraph of arcs $\mathcal{A}_{\mu^*}$, we have

$$\cup_{m=0}^{\bar{k}} X_m = X \cup \{t\}.$$

We will now show by induction that starting from $J(x) \equiv \infty$ for all $x \in X$, the iterates $T^k J$ of VI satisfy

$$(T^k J)(x) = J^*(x), \qquad \forall\, x \in \cup_{m=1}^{k} X_m,\ k = 1, \ldots, \bar{k}. \tag{5.2}$$

Indeed, it can be seen that this is so for $k = 1$. Assume that $(T^k J)(x) = J^*(x)$ if $x \in \cup_{m=1}^{k} X_m$. Then, since $TJ \leq J$ and T is monotone, $(T^k J)(x)$ is monotonically nonincreasing, so that

$$J^*(x) \leq (T^{k+1} J)(x), \qquad \forall\, x \in X. \tag{5.3}$$

Moreover, by the induction hypothesis, the definition of the sets X_k, and the optimality of μ^*, we have

$$\begin{aligned}(T^{k+1} J)(x) &\leq H(x, \mu^*(x), T^k J) = H(x, \mu^*(x), J^*) \\ &= J^*(x), \quad \forall\, x \in \cup_{m=1}^{k+1} X_m,\end{aligned} \tag{5.4}$$

where the first equality follows from the form (2.6) of H and the fact that for all $x \in \cup_{m=1}^{k+1} X_m$, we have $y \in \cup_{m=1}^{k} X_m$ for

all $y \in Y(x, \mu^*(x))$ by the definition (5.1) of X_{k+1}. The two relations (5.3) and (5.4) complete the induction proof.

Thus under Assumption 2.1, the VI method when started with $J(x) = \infty$ for all $x \in X$, will find the optimal costs of all the nodes in the set $\cup_{m=1}^{k} X_m$ after k iterations; cf. Eq. (5.2). The same is true under Assumption 4.3, except that the method will find the corresponding optimal costs over the proper policies. In particular, all optimal costs will be found after $\bar{k} \leq N$ iterations, where N is the number of nodes in X. This indicates that the behavior of the VI algorithm, when initialized with $J(x) = \infty$ for all $x \in X$, is similar to the one of the Bellman-Ford algorithm for deterministic shortest path problems. Still each iteration of the VI algorithm requires as many as N applications of the mapping T at every node. Thus it is likely that the performance of the VI algorithm can be improved with a suitable choice of the initial function J, and with an asynchronous implementation that uses a favorable order of selecting nodes for iteration, "one-node-at-a-time" similar to the Gauss-Seidel method. This is consistent with the deterministic shortest path case, where there are VI-type algorithms, within the class of label-correcting methods, which are faster than the Bellman-Ford algorithm and even faster than efficient implementations of the Dijkstra algorithm for some types of problems; see e.g., [28]. For the RSP problem, it can be seen that the best node selection order is based on the sets X_k defined by Eq. (5.1), i.e., iterate on the nodes in the set X_1, then on the nodes in X_2, and so on. In this case, only one iteration per node will be needed. While the sets X_k are not known, an algorithm that tries to approximate the optimal order could be much more efficient that the standard "all-nodes-at-once" VI method that computes the sequence $T^k J$, for $k \geq 0$ (for an example of an algorithm of this type for stochastic shortest path problems, see [65]. The development of such more efficient VI algorithms is an interesting subject for further research, which, however, is beyond the scope of the present paper.

EXAMPLE 5.1: Let us illustrate the VI method for the problem of Fig. 4. The optimal policy is shown in this figure, and it is proper; this is consistent with the fact that Assumption 4.3 is satisfied. The table gives the iteration sequence of two VI methods, starting with $J_0 = (\infty, \infty, \infty, \infty)$. The first method is the all-nodes-at-once method $J_k = T^k J_0$, which finds J^* in four iterations. In this example, we have $X_0 = \{t\}$, $X_1 = \{1\}$, $X_2 = \{4\}$, $X_3 = \{3\}$, $X_4 = \{2\}$, and the assertion of Eq. (5.2) may be verified. The second method is the asynchronous VI method, which iterates one-node-at-a-time in the (most favorable) order 1, 4, 3, 2. The second method also finds J^* in four iterations and with four times less computation.

We finally note that in the absence of Assumption 2.1 or Assumption 4.1, it is possible that the VI sequence $\{T^k J\}$ will not converge to J^* starting from any J with $J \neq J^*$. This can be seen with a simple deterministic shortest path problem involving a zero length cycle, a simpler version of Example 4.2. Here there is a single node 1, aside from the destination t, and two choices at 1: stay at 1 at cost 0, and move to t at cost 1. Then we have $J^* = 0$, while T is given by

$$TJ = \min\{J, 1\}.$$

It can be seen that the set of fixed points of T is $(-\infty, 1]$, and contains J^* in its interior. Starting with $J \geq 1$, the VI sequence converges to 1 in a single step, while starting at $J \leq 1$ it stays at J. This is consistent with Prop. 4.5(c), since in this example Assumption 4.3 holds, and we have $\hat{J} = 1$. In the case of Example 4.2 with $a = 0$, the situation is somewhat different but qualitatively similar. There it can be verified that $J^* = 1$, the set of fixed points is $[0, 1]$, $\{T^k J\}$ will converge to 1 starting from $J \geq 1$, will converge to 0 starting from $J \leq 0$, and will stay at J starting from $J \in [0, 1]$.

5.2. Policy Iteration Algorithms

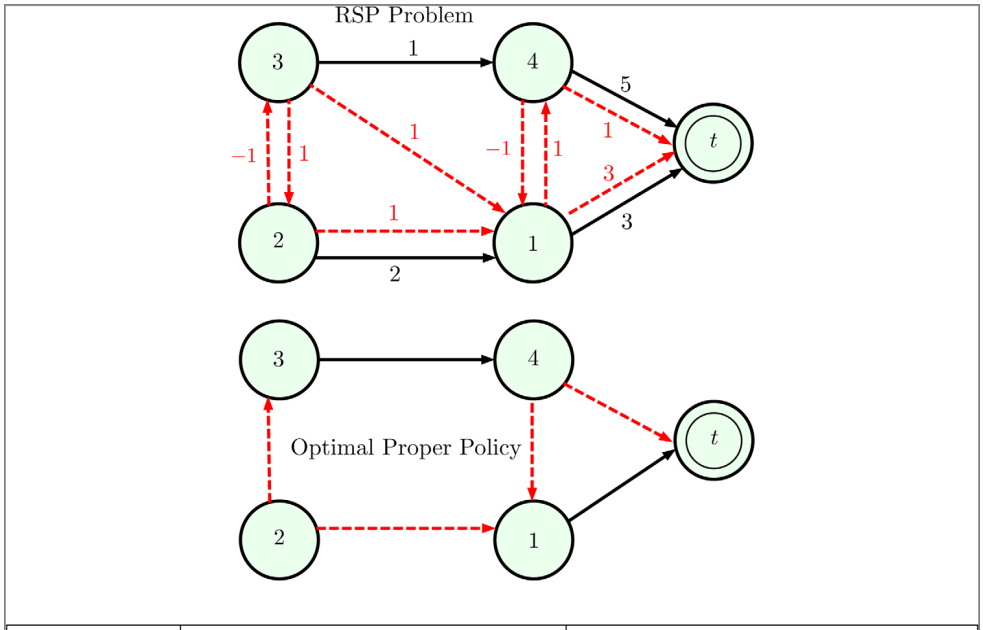
The development of PI algorithms for the RSP problem is straightforward given the connection with semicontractive models. Briefly, under Assumption 2.1, based on the analysis of Section 3.3.2 of [31], there are two types of PI algorithms. The first is a natural form of PI that generates proper policies exclusively. Let μ_0 be an initial proper policy (there exists one by assumption). At the typical iteration k, we have a proper policy μ_k, and first compute J_{μ_k} by solving a longest path problem over the corresponding acyclic subgraph of arcs $\mathcal{A}_{\mu_k}$. We then compute a policy μ_{k+1} such that $T_{\mu_{k+1}} J_{\mu_k} = T J_{\mu_k}$, by minimizing over $u \in U(x)$ the expression $H(x, u, J_{\mu_k})$ of Eq. (2.6), for all $x \in X$. We have

$$\begin{aligned} J_{\mu_k} &= T_{\mu_k} J_{\mu_k} \geq T J_{\mu_k} \\ &= T_{\mu_{k+1}} J_{\mu_k} \geq \lim_{m \to \infty} T_{\mu_{k+1}}^m J_{\mu_k} = J_{\mu_{k+1}}, \end{aligned} \tag{5.5}$$

where the second inequality follows from the monotonicity of $T_{\mu_{k+1}}$. Given that μ_k is proper and hence $J_{\mu_k} \in R(X)$, the next policy μ_{k+1} cannot be improper [in view of Assumption 2.1(b) and Prop. 4.1(d)], so μ_{k+1} must be proper and has improved cost over μ_k. Therefore the sequence of policies $\{\mu_k\}$ is well-defined and proper, and the corresponding sequence $\{J_{\mu_k}\}$ is nonincreasing. It then follows that J_{μ_k} converges to J^* in a finite number of iterations. The reason is that from Eq. (5.5), we have that at the kth iteration, either strict improvement

$$J_{\mu_k}(x) > (T J_{\mu_k})(x) \geq J_{\mu_{k+1}}(x)$$

is obtained for at least one node $x \in X$, or else $J_{\mu_k} = T J_{\mu_k}$, which implies that $J_{\mu_k} = J^*$ [since J^* is the unique fixed



Iteration #	All-Nodes-at-Once VI Method	One-Node-at-a-Time VI Method
0	$J_0 = (\infty, \infty, \infty, \infty)$	$J_0 = (\infty, \infty, \infty, \infty)$
1	$J_1 = TJ_0 = (3, \infty, \infty, 5)$	$J_1 = (3, \infty, \infty, \infty)$
2	$J_2 = T^2J_0 = (3, 5, 6, 2)$	$J_2 = (3, \infty, \infty, 2)$
3	$J_3 = T^3J_0 = (3, 5, 3, 2)$	$J_3 = (3, \infty, 3, 2)$
4	$J_4 = T^4J_0 = (3, 4, 3, 2) = J^*$	$J_4 = (3, 4, 3, 2) = J^*$

Figure 4. An example RSP problem and its optimal policy. At each $X = \{1, 2, 3, 4\}$ there are two controls: one (shown by a solid line) where $Y(x, u)$ consists of a single element, and another (shown by a broken line) where $Y(x, u)$ has two elements. Arc lengths are shown next to the arcs. Both the all-nodes-at-once and the one-node-at-a-time versions of VI terminate in four iterations, but the latter version requires four times less computation per iteration. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

point of T within $R(X)$, by Prop. 4.3(a)] and μ_k is an optimal proper policy.

Unfortunately, when there are improper policies, the preceding PI algorithm is somewhat limited, because an initial proper policy may not be known, and also because when asynchronous versions of the algorithm are implemented, it is difficult to guarantee that all the generated policies are proper. There is another algorithm, combining value and policy iterations, which has been developed in [21, 22, 85, 86] for a variety of DP models, including discounted, stochastic shortest path, and abstract, and is described in Sections 2.6.3 and 3.3.2 of [31]. This algorithm updates a cost function J and a policy μ, but it also maintains an additional function V, which acts as a threshold to keep J bounded and the algorithm convergent. The algorithm not only can tolerate the presence of improper policies, but can also be operated in asynchronous mode, whereby the value iterations, policy evaluation operations, and policy improvement iterations are performed one-node-at-a-time without any regularity. The algorithm is valid even in a distributed asynchronous environment, and in the presence of communication delays between processors. The specialization of this algorithm to RSP under Assumption 2.1 is straightforward, and will be presented briefly in its asynchronous form, but without communication delays between processors.

We consider a distributed computing system with m processors, denoted $1, \ldots, m$, a partition of the node set X into sets $X_1, \ldots, X_m$, and an assignment of each subset X_ℓ to a processor $\ell \in \{1, \ldots, m\}$. The processors collectively maintain two functions $J_k(x)$ and $V_k(x)$, $x \in X$, and a policy $\mu_k \in \mathcal{M}$, $k = 0, 1, \ldots$. We denote by $\min[V_k, J_k]$ the function in $E(X)$ that takes values $\min[V_k(x), J_k(x)]$ for $x \in X$. The initial conditions $J_0(x), V_0(x), \mu_0(x)$, $x \in X$, are arbitrary. For each processor ℓ, there are two infinite disjoint subsets of times $\mathcal{K}_\ell, \overline{\mathcal{K}}_\ell \subset \{0, 1, \ldots\}$, corresponding to local (within the subset X_ℓ) policy improvement and

policy evaluation iterations by that processor, respectively. More specifically, at each time k and for each processor ℓ, we have one of the following three possibilities:

a. *Local policy improvement*: If $k \in \mathcal{K}_\ell$, processor ℓ sets for all $x \in X_\ell$,

$$J_{k+1}(x) = V_{k+1}(x) = \min_{u \in U(x)} H(x, u, \min[V_k, J_k]), \tag{5.6}$$

and sets $\mu_{k+1}(x)$ to a u that attains the above minimum.

b. *Local policy evaluation - Value iteration*: If $k \in \overline{\mathcal{K}}_\ell$, processor ℓ sets for all $x \in X_\ell$,

$$J_{k+1}(x) = H(x, \mu_k(x), \min[V_k, J_k]), \tag{5.7}$$

and leaves V and μ unchanged, i.e., for all $x \in X_\ell$, $V_{k+1}(x) = V_k(x)$, $\mu_{k+1}(x) = \mu_k(x)$.

c. *No local change*: If $k \notin \mathcal{K}_\ell \cup \overline{\mathcal{K}}_\ell$, processor ℓ leaves J, V, and μ unchanged, i.e., for all $x \in X_\ell$,

$$J_{k+1}(x) = J_k(x), \quad V_{k+1}(x) = V_k(x),$$
$$\mu_{k+1}(x) = \mu_k(x).$$

In view of the form (2.6) of the mapping H, the local policy improvement iteration (5.6) involves the solution of a static minimax problem, where the minimizing player chooses $u \in U(x)$ and the maximizing player chooses $y \in Y(x, u)$. The local policy evaluation iteration (5.7) involves a maximization over $y \in Y(x, \mu_k(x))$.

The function V_k in Eqs. (5.6–5.7) is reminiscent of a stopping cost in optimal stopping problems. The use of V_k is essential for the asymptotic convergence of the algorithm to optimality, i.e., $J_k \to J^*$, $V_k \to J^*$, and for finite convergence of $\{\mu_k\}$ to an optimal proper policy. Without V_k the algorithm may potentially oscillate (there is an important counterexample that documents this type of phenomenon, given in [83]; see also the discussion in [21, 22, 30]).

Note that the preceding algorithm includes as a special case a one-node-at-a-time asynchronous PI algorithm, whereby each node is viewed as a processor by itself, and at each iteration a single node is selected and a local policy improvement or local policy evaluation of the form (5.6) or (5.7), respectively, is performed just at that node (see the discussion of Section 2.6 of [30] or Section 2.6 of [31]). This Gauss-Seidel type of algorithm is often considerably faster than all-nodes-at-once versions. The comparative evaluation of PI algorithms that use different initial conditions $J_0(x), V_0(x), \mu_0(x)$, $x \in X$, and different orders of local policy improvement and policy evaluation iterations remains a subject for further research and experimentation.

5.3. A Dijkstra-Like Algorithm for Nonnegative Arc Lengths

One of the most important computational approaches for the classical deterministic shortest path problem with nonnegative arc lengths is based on Dijkstra's algorithm, whereby the shortest distances of the nodes to the destination are determined one-at-a-time in nondecreasing order. When properly implemented, this approach yields shortest path methods with excellent computational complexity and practical performance (see e.g., [2, 28]).

Dijkstra's algorithm has been extended to continuous-space shortest path problems in [79], and finds extensive application in large-scale computational problems involving the eikonal and other equations; see [75, 76]. For recent work in this area, see [4, 36, 39], which give many other references. Dijkstra's algorithm has also been extended to finite-state stochastic shortest path problems, through the notion of a "consistently improving optimal policy" (introduced in the 2001 2nd edition of the author's DP book, and also described in its 4th edition, [30], Section 3.4.1). Roughly, with such a policy, from any node we may only go to a node of no greater optimal cost. While existence of a consistently improving optimal policy is a restrictive condition, the associated Dijkstra-like algorithm has found application in some special contexts, including large-scale continuous-space shortest path problems, where it is naturally satisfied; see [81]. Our Dijkstra-like algorithm is patterned after the Dijkstra-like stochastic shortest path algorithm, but requires less restrictive conditions because an optimal proper policy has the essential character of a consistently improving policy when the arc lengths are nonnegative. As noted earlier, related Dijkstra-like algorithms were proposed by [50] (without the type of convergence analysis that we give here), and by [12] (under the assumption that all arc lengths are strictly positive, and with an abbreviated convergence argument). We will assume the following:

ASSUMPTION 5.1:

a. There exists at least one proper policy.
b. For every improper policy μ, all cycles in the subgraph of arcs $\mathcal{A}_\mu$ have positive length.
c. All arc lengths are nonnegative.

Parts (a) and (b) of the preceding assumption are just Assumption 2.1, under which the favorable results of Prop. 4.3 apply to RSP with both nonnegative and negative arc lengths. The arc length nonnnegativity assumption of part (c) provides additional structure, which provides the basis for the algorithm of this section.

Our Dijkstra-like algorithm maintains and updates a subset of nodes denoted V, and a number $J(x)$ for each $x \in X \cup \{t\}$,

called the *label of* x. Initially,

$$V = \{t\}, \qquad J(x) = \begin{cases} 0 & \text{if } x = t, \\ \infty & \text{if } x \in X. \end{cases}$$

At any given point in the algorithm, let W be the set

$$W = \{x | J(x) < \infty,\ x \notin V\}. \tag{5.8}$$

The algorithm terminates when V is empty. The typical iteration, assuming V is nonempty, is as follows.

Typical Iteration of the Dijkstra-Like Algorithm:

We remove from V a node y^* such that

$$J(y^*) = \min_{y \in V} J(y),$$

and place it in W, i.e., replace W with $W \cup \{y^*\}$. For every $x \notin W$, we let

$$\hat{U}(x) = \left\{u \in U(x) | Y(x,u) \subset W \text{ and } y^* \in Y(x,u)\right\}, \tag{5.9}$$

and we update $J(x)$ and V according to the following two cases:

1. If $\hat{U}(x)$ is nonempty and $J(x) > \min_{u \in \hat{U}(x)} \max_{y \in Y(x,u)} [g(x,u,y) + J(y)]$, we set

$$J(x) = \min_{u \in \hat{U}(x)} \max_{y \in Y(x,u)} [g(x,u,y) + J(y)], \tag{5.10}$$

   and we place x in V if it is not already there.
2. Otherwise, we leave $J(x)$ and V unchanged.

Note that at each iteration of the preceding algorithm, the single node y^* exits V, and enters the set W of Eq. (5.8). Thus W is the set of nodes that have entered V at some previous iteration but are not currently in V. Moreover, from the definition of the algorithm, once a node enters W it stays in W and never returns to V. Also, on entrance of a node into V, its label changes from ∞ to some nonnegative number. In the terminology of Dijkstra-like algorithms, W is the set of nodes that are "permanently labeled," and V is the set of nodes that are "candidates" for permanent labeling. We will show that all the nodes x will enter W in order of nondecreasing $J(x)$, and at the time of entrance, $J(x) = J^*(x)$.

PROPOSITION 5.1: Let Assumption 5.1 hold. Then at the end of an iteration of the Dijkstra-like algorithm, we have $J(x') \geq J(x)$ for all $x' \notin W$ and $x \in W$.

PROOF: We use induction on the iteration count. Clearly the assertion holds at the end of the initial iteration since then $W = \{t\}$, $J(t) = 0$, and according to the formula (5.10) for changing labels and the nonnegativity of the arc lengths, we have $J(x) \geq 0$ for all $x \in X$. Assume that the assertion holds for iteration $k-1$. Let $J(x)$ and $\tilde{J}(x)$ denote the node labels at the start and the end of iteration k, respectively. Then by the minimum label rule for selection of y^*, we have

$$J(x') \geq J(y^*) \geq J(x) = \tilde{J}(x), \qquad \forall\, x' \notin W \cup \left\{y^*\right\},\ x \in W \cup \left\{y^*\right\}, \tag{5.11}$$

where the equality holds because the labels of all $x \in W \cup \{y^*\}$ will not change in iteration k. During iteration k the labels of nodes $x' \notin W \cup \{y^*\}$ will change, if $\hat{U}(x') \neq \emptyset$, according to Eq. (5.10), so that

$$\begin{aligned} \tilde{J}(x') &= \min\left[J(x'),\ \min_{u \in \hat{U}(x')} \max_{y \in Y(x',u)} [g(x',u,y) + J(y)]\right] \\ &\geq \min[J(x'),\ J(y^*)] \\ &\geq J(x) \\ &= \tilde{J}(x), \quad \forall\, x' \notin W \cup \left\{y^*\right\},\ x \in W \cup \left\{y^*\right\}, \end{aligned}$$

where the first inequality holds because $g(x',u,y^*) \geq 0$, and $y^* \in Y(x',u)$ for all $u \in \hat{U}(x')$, and the second inequality and second equality hold because of Eq. (5.11). The induction proof is complete. □

Since no node will enter V twice, while exactly one node exits V at each iteration, the algorithm will terminate after no more than $N+1$ iterations, where N is the number of nodes in X. The next proposition shows that V will become empty after exactly $N+1$ iterations, at which time W must necessarily be equal to $X \cup \{t\}$.

PROPOSITION 5.2: Let Assumption 5.1 hold. The Dijkstra-like algorithm will terminate after exactly $N+1$ iterations with $V = \emptyset$ and $W = X \cup \{t\}$.

PROOF: Assume the contrary, i.e., that the algorithm will terminate after a number of iterations $k < N+1$. Then on termination, W will have k nodes, V will be empty, and the set

$$\bar{V} = \{x \in X | x \notin W\}$$

will have $N+1-k$ nodes and thus be nonempty. Let $\overline{\mu}$ be the proper policy, which is assumed to exist by Assumption 5.1(a). For each $x \in \bar{V}$ we cannot have $Y(x, \overline{\mu}(x)) \subset W$, since then x would have entered V prior to termination, according to the rules of the algorithm. Thus for each $x \in \bar{V}$, there exists a node $y \in Y(x, \overline{\mu}(x))$ with $y \in \bar{V}$. This implies that the subgraph of arcs $\mathcal{A}_{\overline{\mu}}$ contains a cycle of nodes in $\bar{V}$, thus contradicting the properness of $\overline{\mu}$. □

There still remains the question of whether the final node labels $J(x)$, obtained on termination of the algorithm, are equal to the optimal costs $J^*(x)$. This is shown in the following proposition.

PROPOSITION 5.3: Let Assumption 5.1 hold. On termination of the Dijkstra-like algorithm, we have $J(x) = J^*(x)$ for all $x \in X$.

PROOF: For $k = 0, 1, \ldots$, let V_k, W_k, $J_k(x)$ denote the sets V, W, and the labels $J(x)$ at the start of iteration k, respectively, and let y_k^* denote the minimum label node to enter W during iteration k. Thus, we have $W_{k+1} = W_k \cup \{y_k^*\}$,

$$J_{k+1}(x) = J_k(x), \qquad \forall\, x \in W_{k+1},$$

and

$$J_{k+1}(x) = \begin{cases} \min\big[J_k(x),\, \min_{u \in \hat{U}_k(x)} \max_{y \in Y(x,u)} \\ \qquad [g(x,u,y) + J_k(y)]\big] & \text{if } x \in V_{k+1}, \\ \infty & \text{if } x \notin W_{k+1} \cup V_{k+1}, \end{cases} \tag{5.12}$$

where

$$\hat{U}_k(x) = \big\{u \in U(x) \,|\, Y(x,u) \subset W_k \text{ and } y^* \in Y(x,u)\big\}.$$

For each k consider the sets of policies

$$M_k(x) = \{\mu : \text{proper} \,|\, \text{the nodes of all paths } p \in P(x,\mu) \text{ lie in } W_k \cup \{x\}\}.$$

Note that $M_{k+1}(x) \supset M_k(x)$ since $W_{k+1} \supset W_k$, and that from the rule for a node to enter V, we have

$$M_k(x) = \emptyset \quad \text{if and only if} \quad x \notin W_k \cup V_k, \tag{5.13}$$

[the reason why $M_k(x) \neq \emptyset$ for all $x \in W_k \cup V_k$ is that for entrance of x in V at some iteration there must exist $u \in U(x)$ such that $Y(x, u)$ is a subset of $W \cup \{y^*\}$ at that iteration].

We will prove by induction that for all $x \in X \cup \{t\}$, we have

$$J_k(x) = \begin{cases} \min_{\mu \in M_k(x)} \max_{p \in P(x,\mu)} L_p(\mu) & \text{if } x \in W_k \cup V_k, \\ \infty & \text{if } x \notin W_k \cup V_k. \end{cases} \tag{5.14}$$

[In words, we will show that at the start of iteration k, $J_k(x)$ is the shortest "minimax" distance from x to t, using proper policies, which generate paths that start at x and go exclusively through W_k. The idea is that since nodes in W_k have smaller labels than nodes not in W_k and the arc lengths are nonnegative, it would not be optimal to use a path that moves in and out of W_k.] Equation (5.14) will imply that on termination, when $M_{N+1}(x)$ is equal to the set of all proper policies, we have

$$J_{N+1}(x) = \min_{\mu:\text{proper}} \max_{p \in P(x,\mu)} L_p(\mu), \qquad \forall\, x \in X,$$

which will prove the proposition. As can be expected, the proof is based on generalizations of proof ideas relating to the ordinary Dijkstra algorithm for the classical deterministic shortest path problem. We will often use the fact that for all $x \in W_{k+1} \cup V_{k+1}$ and $\mu \in M_{k+1}(x)$ we have

$$\max_{p \in P(x,\mu)} L_p(x) = \max_{y \in Y(x,\mu(x))} \left[g(x, \mu(x), y) + \max_{p' \in P(y,\mu)} L_{p'}(\mu) \right]. \tag{5.15}$$

This is just the optimality equation for the longest path problem associated with μ and the subgraph of arcs A_μ.

Initially, for $k = 0$, we have $W_0 = \emptyset$, $V_0 = \{t\}$, $J_0(t) = 0$, $J_0(x) = \infty$ for $x \neq t$, so Eq. (5.14) holds. Assume that Eq. (5.14) holds for some k. We will show that it holds for $k+1$.

For $x \notin W_{k+1} \cup V_{k+1}$, we have $J_{k+1}(x) = \infty$, since such x have never entered V so far, and therefore their label was never reduced from the initial value ∞. This proves Eq. (5.14) with k replaced by $k+1$ and $x \notin W_{k+1} \cup V_{k+1}$.

For $x = y_k^*$, from the definition (5.13), the set of policies $M_{k+1}(y_k^*)$ is equal to $M_k(y_k^*)$, so using also the induction hypothesis and the fact $J_{k+1}(y_k^*) = J_k(y_k^*)$, it follows that

$$\begin{aligned} J_{k+1}(y_k^*) = J_k(y_k^*) &= \min_{\mu \in M_k(y_k^*)} \max_{p \in P(y_k^*,\mu)} L_p(\mu) \\ &= \min_{\mu \in M_{k+1}(y_k^*)} \max_{p \in P(y_k^*,\mu)} L_p(\mu). \end{aligned}$$

This proves Eq. (5.14) with k replaced by $k+1$ and $x = y_k^*$.

For $x \in W_k \cup V_{k+1}$, we write

$$\min_{\mu \in M_{k+1}(x)} \max_{p \in P(x,\mu)} L_p(\mu) = \min[E_1,\, E_2],$$

where

$$E_1 = \min_{\mu \in M_k(x)} \max_{p \in P(x,\mu)} L_p(\mu),$$

which is equal to $J_k(x)$ by the induction hypothesis, and

$$E_2 = \min_{\mu \in M_{k+1}(x)/M_k(x)} \max_{p \in P(x,\mu)} L_p(\mu).$$

[The set $M_{k+1}(x)/M_k(x)$ may be empty, so here and later we use the convention that the minimum over the empty set is equal to ∞.] Thus for $x \in W_k \cup V_{k+1}$, we have

$$\min_{\mu \in M_{k+1}(x)} \max_{p \in P(x,\mu)} L_p(\mu) = \min\left[J_k(x), \min_{\mu \in M_{k+1}(x)/M_k(x)} \max_{p \in P(x,\mu)} L_p(\mu) \right], \tag{5.16}$$

and we need to show that the right-hand side above is equal to $J_{k+1}(x)$. To estimate the second term of the right-hand side, we consider the two separate cases where $x \in W_k$ and $x \in V_{k+1}$.

Assume first that $x \in W_k$. Then for each $\mu \in M_{k+1}(x)/M_k(x)$, the subgraph of arcs $\mathcal{A}_\mu$ must contain y_k^*, so

$$\max_{p\in P(x,\mu)} L_p(\mu) \geq \max_{p'\in P(y_k^*,\mu)} L_{p'}(y_k^*) \geq J_k(y_k^*) \geq J_k(x),$$

where the first inequality follows in view of the nonnegativity of the arc lengths, the second inequality follows because paths in $P(y_k^*, \mu)$ go exclusively through W_k and thus can also be generated by some policy in $M_k(y_k^*)$, and the last inequality follows since nodes enter W in order of nondecreasing label (cf. Prop. 5.1). Thus the expression (5.16) is equal to $J_k(x)$, and hence is also equal to $J_{k+1}(x)$ (since $x \in W_k$). This proves Eq. (5.14) with k replaced by $k+1$ and $x \in W_k$.

Assume next that $x \in V_{k+1}$. Then to estimate the term $\min_{\mu\in M_{k+1}(x)/M_k(x)} \max_{p\in P(x,\mu)} L_p(\mu)$ in Eq. (5.16), we write

$$M_{k+1}(x)/M_k(x) = \tilde{M}_{k+1}(x) \cup \hat{M}_{k+1}(x),$$

where

$$\tilde{M}_{k+1}(x) = \left\{\mu \in M_{k+1}(x)/M_k(x) | y_k^* \notin Y(x,\mu(x))\right\},$$

is the set of policies for which there exists a path $p \in P(x,\mu)$ that passes through y_k^* after more than one transition, and

$$\hat{M}_{k+1}(x) = \left\{\mu \in M_{k+1}(x)/M_k(x) | y_k^* \in Y(x,\mu(x))\right\},$$

is the set of policies for which there exists a path $p \in P(x,\mu)$ that moves to y_k^* at the first transition.

For all $\mu \in \tilde{M}_{k+1}(x)$, we have using Eq. (5.15),

$$\begin{aligned}
&\max_{p\in P(x,\mu)} L_p(\mu)\\
&\quad= \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + \max_{p'\in P(y,\mu)} L_{p'}(\mu)\right]\\
&\quad\geq \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + \min_{\mu'\in M_{k+1}(y)} \max_{p'\in P(y,\mu')} L_{p'}(\mu')\right]\\
&\quad= \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + J_{k+1}(y)\right]\\
&\quad= \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + J_k(y)\right],
\end{aligned}$$

where the last equality holds because we have shown that $J_{k+1}(y) = J_k(y)$ for all $y \in W_k$. Therefore, since for all $\mu \in \tilde{M}_{k+1}(x)$ there exists a $\mu' \in M_k(x)$ such that $\mu(x) = \mu'(x)$, we have by taking the minimum over $\mu \in \tilde{M}_{k+1}(x)$ in the preceding relation,

$$\begin{aligned}
\min_{\mu\in\tilde{M}_{k+1}(x)} \max_{p\in P(x,\mu)} L_p(\mu) &\geq \min_{\mu'\in M_k(x)} \max_{y\in Y(x,\mu'(x))} [g(x,\mu'(x),y) + J_k(y)]\\
&= \min_{\mu'\in M_k(x)} \max_{p\in P(x,\mu')} L_p(\mu') = J_k(x). \quad (5.17)
\end{aligned}$$

For all $\mu \in \hat{M}_{k+1}(x)$, we have

$$\max_{p\in P(x,\mu)} L_p(x) = \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + \max_{p'\in P(y,\mu)} L_{p'}(\mu)\right];$$

cf. Eq. (5.15). Moreover, for all $\mu \in \hat{M}_{k+1}(x)$, we have $\mu(x) \in \hat{U}_k(x)$ by the definitions of $\hat{M}_{k+1}(x)$ and $\hat{U}_k(x)$. It follows that

$$\begin{aligned}
&\min_{\mu\in\hat{M}_{k+1}(x)} \max_{p\in P(x,\mu)} L_p(\mu)\\
&\quad= \min_{\mu\in\hat{M}_{k+1}(x)} \max_{y\in Y(x,\mu(x))} \left[g(x,\mu(x),y) + \max_{p'\in P(y,\mu)} L_{p'}(\mu)\right]\\
&\quad= \min_{u\in\hat{U}_k(x)} \max_{y\in Y(x,u)} \left[g(x,u,y) + \min_{\mu\in\hat{M}_{k+1}(x)} \max_{p'\in P(y,\mu)} L_{p'}(\mu)\right]\\
&\quad= \min_{u\in\hat{U}_k(x)} \max_{y\in Y(x,u)} \left[g(x,u,y) + \min_{\mu\in M_{k+1}(y)} \max_{p'\in P(y,\mu)} L_{p'}(\mu)\right]\\
&\quad= \min_{u\in\hat{U}_k(x)} \max_{y\in Y(x,u)} [g(x,u,y) + J_{k+1}(y)]\\
&\quad= \min_{u\in\hat{U}_k(x)} \max_{y\in Y(x,u)} [g(x,u,y) + J_k(y)], \quad (5.18)
\end{aligned}$$

where the third equality holds because for $y \in Y(x,\mu(x))$, the collections of paths $P(y,\mu)$ under policies μ in $\hat{M}_{k+1}(x)$ and $M_{k+1}(y)$, are identical, and the last equality holds because we have already shown that $J_{k+1}(y) = J_k(y)$ for all $y \in W_k \cup \{y_k^*\}$. Thus from Eqs. (5.16–5.18) we obtain

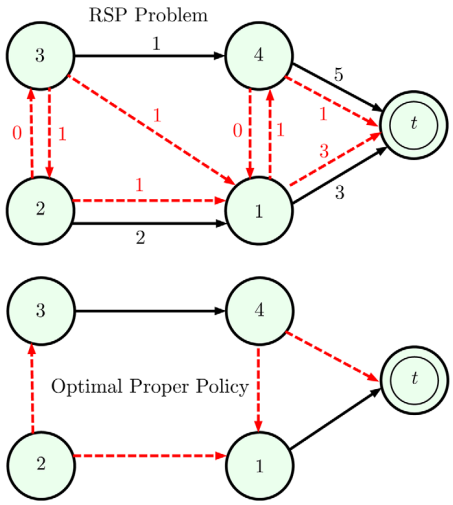
$$\begin{aligned}
&\min_{\mu\in M_{k+1}(x)} \max_{p\in P(x,\mu)} L_p(\mu)\\
&\quad= \min\left[J_k(x), \min_{u\in\hat{U}_k(x)} \max_{y\in Y(x,u)} [g(x,u,y) + J_k(y)]\right].
\end{aligned}$$

Combining this equation with the update formula (5.10) for the node labels, we have

$$\min_{\mu\in M_{k+1}(x)} \max_{p\in P(x,\mu)} L_p(\mu) = J_{k+1}(x),$$

thus proving Eq. (5.14) with k replaced by $k+1$ and $x \in V_{k+1}$. This completes the induction proof of Eq. (5.14) and concludes the proof. □

Since the algorithm terminates in $N+1$ iterations, and each iteration requires at most $O(AL)$ operations, where A is the number of arcs and L is the number of elements of U, the complexity of the algorithm is bounded by $O(NAL)$. This complexity estimate may potentially be improved with the use of efficient data structures of the type used in efficient implementations of Dijkstra's algorithm in deterministic shortest path problems to expedite the selection of the minimum label node [i.e., $y^* \in \operatorname{argmin}_{y\in V} J(y)$]. However, we have not investigated this issue. It is also unclear how



Iteration #	V at Start of Iteration	Labels at Start of Iteration	Min. Label Node out of V
0	$\{t\}$	$(\infty, \infty, \infty, \infty, 0)$	t
1	$\{1, 4\}$	$(3, \infty, \infty, 5, 0)$	1
2	$\{2, 4\}$	$(3, 5, \infty, 3, 0)$	4
3	$\{2, 3\}$	$(3, 5, 4, 3, 0)$	3
4	$\{2\}$	$(3, 4, 4, 3, 0)$	2

Figure 5. The iterations of the Dijkstra-like algorithm for the RSP problem of Example 5.1. The nodes exit V and enter W in the order 1, 4 3, 2. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the Dijkstra-like algorithm compares with the finitely terminating VI algorithm and other asynchronous VI and PI algorithms discussed in Sections 5.1 and 5.2.

EXAMPLE 5.2: We illustrate the Dijkstra-like algorithm with the RSP problem shown in Fig. 5. The table gives the iterations of the algorithm, and the results are consistent with Props. 5.1-5.3, as the reader may verify.

5.4. Approximate Solution by Rollout

Finally let us consider algorithms with approximations. While we have assumed a finite number of nodes, there are many problems of practical interest where the number of nodes is extremely large, and the preceding algorithms are very time-consuming. This is particularly so in minimax control problems with imperfect state information, which are reformulated as problems of perfect state information using the sufficiently informative function approach of [15, 24]. In such cases one may consider minimax analogs of approximate DP or reinforcement learning approaches (see e.g., the books [20, 30, 71, 78]). The development of such methods is an interesting subject for further research. In what follows, we discuss the possibility for approximate solution using a rollout algorithm, which parallels related algorithms for finite horizon and stochastic shortest path problems (see e.g., [29, 30]).

The starting point of the rollout algorithm is a special proper policy μ, called the *base policy*. We define the *rollout policy*, denoted $\overline{\mu}$, as follows: for each $u \in U(x)$ and each $y \in Y(x, u)$, we compute $J_\mu(y)$, and we set

$$\overline{\mu}(x) \in \arg\min_{u \in U(x)} \max_{y \in Y(x,u)} \left\{ g(x, u, y) + \tilde{J}_\mu(y) \right\}, \qquad x \in X, \tag{5.19}$$

where $\tilde{J}_\mu(y)$ is equal to $J_\mu(y)$ for $y \in X$ and $\tilde{J}_\mu(t) = 0$ [cf. Eq. (2.4)]. The computation of $\overline{\mu}(x)$ can be done on-line, only for the nodes x that are encountered in the process of control. Moreover, assuming that $\tilde{J}_\mu(y)$ is precomputed for all y, and that the sets $U(x)$ and $Y(x, u)$ have a relatively small number of elements, the computation in Eq. (5.19) can be performed quite fast. The same is true if for any $u \in U(x)$, $\tilde{J}_\mu(y)$ can be efficiently computed on-line for all $y \in Y(x, u)$.

It can be seen that the rollout policy $\overline{\mu}$ is just the policy obtained from the base policy μ using a *single* PI. In particular, under Assumption 2.1, the rollout policy improves on the base policy in the sense that

$$J_{\overline{\mu}}(x) \leq J_\mu(x), \qquad \forall\, x \in X.$$

This is a well-known property of rollout algorithms for finite horizon and stochastic shortest path problems, and can be verified by writing for all $x \in X$

$$\begin{aligned}(T_{\overline{\mu}} J_\mu)(x) &= (T J_\mu)(x)\\ &= \min_{u \in U(x)} \max_{y \in Y(x,u)} \left\{ g(x,u,y) + \tilde{J}_\mu(y) \right\}\\ &\leq \max_{y \in Y(x,\mu(x))} \left\{ g(x,\mu(x),y) + \tilde{J}_\mu(y) \right\}\\ &= (T_\mu J_\mu)(x)\\ &= J_\mu(x).\end{aligned}$$

Applying $T_{\overline{\mu}}$ repeatedly to both sides of the inequality $T_{\overline{\mu}} J_\mu \leq J_\mu$, we obtain [cf. Eq. (5.5)] that $J_{\overline{\mu}} \leq J_\mu$ and that $\overline{\mu}$ is proper.

As an example of rollout algorithm, consider a pursuit-evasion problem with state $x = (z_1, z_2)$, where z_1 is the location of the minimizer/pursuer and z_2 is the location of the maximizer/evader, in a two- or three-dimensional space. Then a suitable base policy μ is for the pursuer is to follow a shortest path from z_1 to z_2 under the assumption that the evader will stay at his current location z_2 at all future times. To do this for all (z_1, z_2) requires the solution of an all-pairs shortest path problem, which is possible in $O(N^3)$ time using the Floyd-Warshall algorithm [2, 28], where N is the number of possible values of z_1 and z_2. Suppose that we have precomputed $\mu(x)$ for all $x = (z_1, z_2)$ with this shortest path computation. Then the maximization

$$\max_{y \in Y(x,u)} \left\{ g(x,u,y) + \tilde{J}_\mu(y) \right\}$$

that is needed for the on-line computation of the rollout control $\overline{\mu}(x)$ in Eq. (5.19) requires the calculation of $J_\mu(y)$ for each $y \in Y(x,u)$ with $y \neq t$. Knowing μ, each of these calculations is a tractable longest path computation in an acyclic graph of N nodes. Note that the preceding algorithm can be adapted for the imperfect information case where the pursuer knows z_2 imperfectly. This is possible by using a form of assumed certainty equivalence: the pursuer's base policy and the evader's maximization can be computed by using an estimate of the current location z_2 instead of the unknown true location.

In the preceding pursuit-evasion example, the choice of the base policy was facilitated by the special structure of the problem. Generally, however, finding a suitable base policy whose cost function J_μ can be conveniently computed is an important problem-dependent issue. We leave this issue as a subject for further research in the context of more specialized problems. Finally, let us note that a rollout algorithm may be well-suited for on-line suboptimal solution in cases where data may be changing or be revealed during the process of path construction.

6. CONCLUDING REMARKS AND FURTHER RESEARCH

We have considered shortest path problems with set membership uncertainty, and we have shown that they can be fruitfully analyzed in the context of abstract semicontractive models. We have thus proved the existence and uniqueness of the solution of Bellman's equation, and obtained conditions for optimality of a proper policy. Moreover, we have discussed the properties of algorithms of the value and PI type, and we have proposed a finitely terminating Dijkstra-like algorithm for problems with nonnegative arc lengths. The comparative evaluation and the efficient implementation of these algorithms for specific types of applications, such as for example minimax search problems and pursuit-evasion, as well as modifications to take advantage of special problem structures, is an interesting subject for further investigation.

In this article, we have covered the important case of nonnegative arc lengths and improper policies with zero length cycles via the perturbation analysis of Section 4. However, there is an alternative line of analysis, which is based on the fact that when the arc lengths are nonnegative we have $T\bar{J} \geq \bar{J}$, bringing to bear the theory of monotone increasing DP models given in Chapter 4 of [31], which embody the essential structure of negative DP (see [77], or the texts [30, 72]). This theory is somewhat different in character from the analysis of this article.

Let us mention some interesting stochastic extensions of our RSP problem that involve an additional random variable at each stage.[5] In one extension of this type, when at node x, we choose control $u \in U(x)$, then a value of a random variable z is selected from the finite set $\{1, \ldots, n\}$ with probabilities $p_{xz}(u)$, and then the successor node is chosen by an antagonistic opponent from a set $Y_z(x,u) \subset X \cup \{t\}$. To analyze this problem using a semicontractive model, the mapping H of Eq. (2.6) should be replaced by

$$H(x,u,J) = \sum_{z=1}^{n} p_{xz}(u) \max_{y \in Y_z(x,u)} [g(x,u,y) + \tilde{J}(y)], \quad (6.1)$$

[5] This type of stochastic problem arises among others in a context of discretization of the state space of a continuous-space minimax control problem, where randomization in the discretized problem's dynamics is introduced to reduce the error between the optimal cost function of the original continuous-space problem and the optimal cost function of its discretized version (see [10, 79, 65, 75, 76]). There are also other stochastic shortest path-type formulations that involve at least in part a worst case viewpoint, through a risk-sensitive utility or constraints; see [32, 38, 40, 44, 69].

where

$$\tilde{J}(y) = \begin{cases} J(y) & \text{if } y \in X, \\ 0 & \text{if } y = t. \end{cases}$$

A formulation as an abstract DP problem is then possible with an appropriately modified mapping T_μ, similar to Section 2, and the semicontractive theory may be applied similar to Section 4. This analysis and the associated algorithms are, however, beyond our scope. Note that when $n = 1$, from Eq. (6.1) we obtain the mapping (2.6) of the RSP problem of Sections 1 and 2, while when $Y_z(x,u)$ consists of a single node, we obtain the mapping associated with a standard finite-state stochastic shortest path problem (see e.g., [19, 23, 30, 67]). Thus the semicontractive model that is based on the mapping (6.1) generalizes both of these shortest path problems. In the case where $g(x,u,y) \geq 0$ for all $x \in X$, $u \in U(x)$, $z \in \{1,\ldots,n\}$, and $y \in Y_z(x,u)$, we have $T\bar{J} \geq \bar{J}$, and the theory of monotone increasing models of Sections 4.3 and 4.4 of [31] can be used to provide a first layer of analysis without any further assumptions.

In another extension of RSP, which involves randomization, when at node x, we choose control $u \in U(x)$, then a variable w is chosen by an antagonistic opponent from a set $W(x,u)$, and then a successor node $y \in X \cup \{t\}$ is chosen according to probabilities p_{wy}. Here, to apply the line of analysis of the present paper, the mapping H of Eq. (2.6) should be replaced by

$$H(x,u,J) = \max_{w \in W(x,u)} \left[g(x,u,w) + \sum_{y \in X \cup \{t\}} p_{wy} \tilde{J}(y) \right],$$

where

$$\tilde{J}(y) = \begin{cases} J(y) & \text{if } y \in X, \\ 0 & \text{if } y = t. \end{cases} \tag{6.2}$$

A somewhat more complex variation is given by

$$H(x,u,J) = \sum_{z=1}^{n} p'_{xz} \max_{w \in W_z(x,u)} \left[g(z,u,w) + \sum_{y \in X \cup \{t\}} p_{wy} \tilde{J}(y) \right], \tag{6.3}$$

where, for each $x \in X$, z is a random variable taking values in $\{1,\ldots,n\}$ with probabilities p'_{xz} [the two models based on Eqs. (6.2) and (6.3) coincide if z takes values in X and $p'_{xx} = 1$ for all $x \in X$]. The resulting models again combine elements of RSP and a standard finite-state stochastic shortest path problem. They may also be viewed as instances of a generalized aggregation model of the type introduced in Section 7.3.7 of [30].[6] Again the semicontractive theory may be applied, but the corresponding analysis is a subject for further research.

REFERENCES

[1] L. Alfaro, L. T. Henzinger, and O. Kupferman, Concurrent reachability games, Theor Comput Sci 386 (2007), 188–217.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows," in: G. L. Nemhauser, A. H. G. Rinnooy-Kan, and M. J. Todd (Editors), Handbooks in Operations Research and Management Science, Vol. 1, Optimization, North-Holland, Amsterdam, 1989, 211–369.

[3] K. Alton and I. M. Mitchell, An ordered upwind method with precomputed stencil and monotone node acceptance for solving static Hamilton-Jacobi equations, J Sci Comp 51, 2011, 313–348.

[4] J. Andrews and A. Vladimirsky, Deterministic control of randomly-terminated processes, Inter Free Boundaries, 16 (2014), 1–40.

[5] D. Bertsimas, D. B. Brown, and C. Caramanis, Theory and applications of robust optimization, SIAM Rev 53 (2011), 464–501.

[6] M. Bardi, S. Bottacin, and M. Falcone, "Convergence of discrete schemes for discontinuous value functions of pursuit-evasion games, New trends in dynamic games and applications," in: Annals Intern. Soc. Dynamic Games, Birkhauser, Boston, Vol. 3, 1995, pp. 273–304,

[7] S. D. Bopardikar, F. Bullo, and J. P. Hespanha, On discrete-time pursuit-evasion games with sensing limitations, IEEE Trans Robot 24 (2008), 1429–1439.

[8] D. P. Bertsekas, F. Guerriero, and R. Musmanno, "Parallel shortest path methods for globally optimal trajectories," in: J. Dongarra et al., (Editors), High performance computing: Technology, methods, and applications, Elsevier, 1995.

[9] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, Robust optimization, Princeton University Press, Princeton, NJ, 2009.

[10] M. Bardi, P. Soravia, and M. Falcone, "Fully discrete schemes for the value function of pursuit-evasion games," New trends in dynamic games and applications, in: Annals Intern. Soc. Dynamic Games, Birkhauser, Boston, 1994, Vol. 1, pp. 89–105.

[11] T. Basar and P. Bernhard, H-infinity optimal control and related minimax design problems: A dynamic game approach, Birkhauser, Boston, 1991.

[12] M. Bardi and J. P. M. Lopez, A Dijkstra-type algorithm for dynamic games, Dyn Games Appl 6 (2015), 1–14.

[13] M. Bardi and P. Soravia, "Approximation of differential games of pursuit-evasion by discrete-time games, differential games - developments in modeling and computation," in: Lecture Notes in Control and Information Sci., Vol. 156, Springer, Berlin, 1991, pp. 131–143.

[6] In the context of generalized aggregation, for the mapping of Eq. (6.2), we have a high-dimensional RSP problem, whose states are represented by w, and a lower dimensional "aggregate" RSP problem, whose states are represented by the nodes x. Then the p_{wy} play the role of aggregation probabilities in the terminology of [30]. A similar interpretation may be given for the mapping of Eq. (6.3).

[14] D. P., Bertsekas, and I. B., Rhodes, Recursive State Estimation for a Set-Membership Description of the Uncertainty, IEEE Trans. Automatic Control, Vol. AC-16 (1971), 117–128.
[15] D. P. Bertsekas and I. B. Rhodes, Sufficiently informative functions and the minimax feedback control of uncertain dynamic systems, IEEE Trans Auto Control AC-18 (1973), 117–124.
[16] D. P. Bertsekas and S. E. Shreve, Stochastic optimal control: The discrete time case, Academic Press, NY, 1978. Available at: http://web.mit.edu/dimitrib/www/home.html
[17] D. Bertsimas and M. Sim, Robust discrete optimization and network flows, Math Program 98 (2003), 49–71.
[18] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and distributed computation: Numerical methods, Prentice-Hall, Englewood Cliffs, NJ, 1989; republished by Athena Scientific, Belmont, MA, 1997.
[19] D. P. Bertsekas and J. N. Tsitsiklis, An analysis of stochastic shortest path problems, Math OR 16 (1991), 580–595.
[20] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-dynamic programming, Athena Scientific, Belmont, MA, 1996.
[21] D. P. Bertsekas and H. Yu, "Asynchronous distributed policy iteration in dynamic programming," in: Proc. of Allerton Conf. on Communication, Control and Computing, Allerton Park, Ill, 2010, pp. 1368–1374.
[22] D. P. Bertsekas, and H. Yu, Q-learning and enhanced policy iteration in discounted dynamic programming, Math OR, 37 (2012), 66–94.
[23] D. P. Bertsekas and H. Yu, Stochastic shortest path problems under weak conditions, Lab. for Information and Decision Systems Report LIDS-P-2909, MIT, 2016.
[24] D. P. Bertsekas, "Control of uncertain systems with a set-membership description of the uncertainty," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1971.
[25] D. P. Bertsekas, Monotone mappings with application in dynamic programming, SIAM J Control Optim 15 (1977), 438–464.
[26] D. P. Bertsekas, Distributed dynamic programming, IEEE Trans Auto Control AC-27 (1982), 610–616.
[27] D. P. Bertsekas, Asynchronous distributed computation of fixed points, Math Program. 27 (1983), 107–120.
[28] D. P. Bertsekas, Network optimization: Continuous and discrete models, Athena Scientific, Belmont, MA, 1998.
[29] D. P. Bertsekas, Dynamic programming and optimal control, 3rd ed., Vol. I, Athena Scientific, Belmont, MA, 2005.
[30] D. P. Bertsekas, Dynamic programming and optimal control, 4th ed., Vol. II, Athena Scientific, Belmont, MA, 2012.
[31] D. P. Bertsekas, Abstract dynamic programming, Athena Scientific, Belmont, MA, 2013.
[32] D. P. Bertsekas, "Affine monotonic and risk-sensitive models in dynamic programming," in: Lab. for Information and Decision Systems Report LIDS-3204, MIT, 2016.
[33] F. Blanchini and S. Miani, Set-theoretic methods in control, Birkhauser, Boston, MA, 2008.
[34] F. Blanchini, Set invariance in control – A survey, Automatica, 35 (1999), 1747–1768.
[35] B. Bonet, On the speed of convergence of value iteration on stochastic shortest-path problems, Math Oper Res 32 (2007), 365–373.
[36] Z. Clawson, A. Chacon, and A. Vladimirsky, Causal domain restriction for Eikonal equations, SIAM J Sci Comp 36 (2014), A2478–A2505.
[37] E. F. Camacho and C. Bordons, Model predictive control, 2nd ed., Springer-Verlag, New York, NY, 2004.
[38] O. Cavus, and A. Ruszczynski, Computational methods for risk-averse undiscounted transient Markov models, Oper Res 62 (2014), 401–417.
[39] A. Chacon and A. Vladimirsky, Fast two-scale methods for Eikonal equations, SIAM J Sci Comp 34 (2012), A547–A577.
[40] E. V. Denardo and U. G. Rothblum, Optimal stopping, exponential utility, and linear programming, Math Program 16 (1979), 228–244.
[41] E. V. Denardo, Contraction mappings in the theory underlying dynamic programming, SIAM Rev 9 (1967), 165–177.
[42] C. Derman, Finite state Markovian decision processes, Academic Press, NY, 1970.
[43] S. E. Dreyfus, An appraisal of some shortest-path algorithms, 17 (1969), 395–412.
[44] S. Ermon, C. Gomes, B. Selman, and A. Vladimirsky, "Probabilistic planning with non-linear utility functions and worst-case guarantees," in: Proc. of the 11th Intern. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012), Valencia, Spain, 2012.
[45] M. Falcone, A numerical approach to the infinite horizon problem of deterministic control theory, Appl Math Opt 15 (1987), 1–13.
[46] J. Filar and K. Vrieze, Competitive Markov decision processes, Springer, NY, 1996.
[47] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani, A visibility-based pursuit-evasion problem, Intern J Comp Geometry Appl 9 (1999), 471–493.
[48] G. Gallo and S. Pallottino, Shortest path algorithms, Ann Oper Res 7 (1988), 3–79.
[49] R. Gonzalez and E. Rofman, On deterministic control problems: An approximation procedure for the optimal cost, Parts I, II, SIAM J Control Optim 23 (1985), 242–285.
[50] L. Grune and O. Junge, Global optimal control of perturbed systems, J Optim Theory Appl 136 (2008), 411–429.
[51] O. Hernandez-Lerma, O. Carrasco, and Perez-Hernandez. Markov control processes with the expected total cost criterion: Optimality, stability, and transient models, Acta Appl Math 59 (1999), 229–269.
[52] T. C. Hu, A. B. Kahng, and G. Robins, Optimal robust path planning in general environments, IEEE Trans Robotics Auto 9 (1993), 775–784.
[53] K. Hinderer, and K.-H. Waldmann, Algorithms for countable state Markov decision models with an absorbing set, SIAM J Control Optim 43 (2005), 2109–2131.
[54] H. W. James, and E. J. Collins, An analysis of transient Markov decision processes, J Appl Prob 43 (2006), 603–621.
[55] E. C. Kerrigan, Robust constraint satisfaction: Invariant sets and predictive control, Ph.D. Dissertation, Univ. of Cambridge, England, 2000.
[56] H. J. Kushner and P. G. Dupuis, Numerical methods for stochastic control problems in continuous time, Springer-Verlag, NY, 1992.
[57] A. Kurzhanski, and I. Valyi, Ellipsoidal calculus for estimation and control, Birkhauser, Boston, MA, 1997.
[58] S. M. LaValle, Planning algorithms, Cambridge University Press, NY, 2006.
[59] N. Meggido, S. L. Hakimi, M. R. Garey, and D. S. Johnson, The complexity of searching a graph, J ACM 35 (1988), 18–44.
[60] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (2000), 789–814.

[61] J. M. Maciejowski, Predictive control with constraints, Addison-Wesley, Reading, MA, 2002.
[62] J-M. Mirebeau, Efficient fast marching with Finsler metrics, Numerische Mathematik 126 (2014), 515–557.
[63] R. Montemanni and L. M. Gambardella, An exact algorithm for the robust shortest path problem with interval data, Comput Oper Res 31 (2004), 1667–1680.
[64] M. Morari and J. H. Lee, Model predictive control: Past, present, and future, Comput Chem Eng 23 (1999), 667–682.
[65] L. C. Polymenakos, D. P. Bertsekas, and J. N. Tsitsiklis, Efficient algorithms for continuous-space shortest path problems, IEEE Trans Auto Control 43 (1998), 278–283.
[66] S. D. Patek, and D. P. Bertsekas, Stochastic shortest path games, SIAM J Control Optim 36 (1999), 804–824.
[67] R. Pallu de la Barriere, Optimal control theory, Saunders, Phila, 1967; republished by Dover, NY, 1980.
[68] T. D. Parsons, Pursuit-evasion in a graph," in: Y. Alavi and D. Lick (Editors), Theory and applications of graphs, Springer-Verlag, Berlin, 1976, pp. 426–441.
[69] S. D. Patek, On terminating Markov decision processes with a risk averse objective function, Automatica 37 (2001), 1379–1386.
[70] S. R. Pliska, On the transient case for Markov decision chains with general state spaces," in: M. L. Puterman (Editors), Dynamic programming and its applications, Academic Press, NY, 1978.
[71] W. B. Powell, Approximate dynamic programming: Solving the curses of dimensionality, 2nd ed., Wiley, Hoboken, NJ, 2011.
[72] M. L. Puterman, Markov decision processes: Discrete stochastic dynamic programming, Wiley, NY, 1994.
[73] J. B. Rawlings and D. Q. Mayne, Model predictive control theory and design, Nob Hill Publishing, Madison, WI, 2009.
[74] R. T. Rockafellar, Network flows and monotropic programming, Wiley-Interscience, NY, 1984.
[75] J. A. Sethian, Level set methods and fast marching methods evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Cambridge Press, NY, 1999.
[76] J. A. Sethian, Fast marching methods, SIAM Rev 41 (1999), 199–235.
[77] R. Strauch, Negative dynamic programming, Ann Math Statist 37 (1966), 871–890.
[78] R. S. Sutton and A. G. Barto, Reinforcement learning, MIT Press, Cambridge, MA, 1998.
[79] J. N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Trans Auto Control AC-40 (1995), 1528–1538.
[80] R. Vidal, H. J. Kim, D. H. Shim, and S. Sastry, Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation, IEEE Trans Robotics Auto 18 (2002), 662–669.
[81] A. Vladimirsky, Label-setting methods for multimode stochastic shortest path problems on graphs, Math Oper Res 33 (2008), 821–838.
[82] P. Whittle, Optimization over time, Vol. 1, Wiley, NY, 1982.
[83] R. J. Williams and L. C. Baird, Analysis of some incremental variants of policy iteration: First steps toward understanding actor-critic learning systems, in: Report NU-CCS-93-11, College of Computer Science, Northeastern University, Boston, MA, 1993.
[84] H. S. Witsenhausen, Minimax control of uncertain systems, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1966.
[85] H. Yu and D. P. Bertsekas, Q-learning and policy iteration algorithms for stochastic shortest path problems, Ann Oper Res 208 (2013), 95–132.
[86] H. Yu, and D. P. Bertsekas, A mixed value and policy iteration method for stochastic control with universally measurable policies, in: Lab. for Information and Decision Systems Report LIDS-P-2905, MIT, 2013.
[87] G. Yu and J. Yang, On the robust shortest path problem, Comput Oper Res 25 (1998), 457–468.
[88] H. Yu, Stochastic shortest path games and Q-learning, in: Lab. for Information and Decision Systems Report LIDS-P-2875, MIT, 2011.