Partially Observed Stochastic Shortest Path Problems with Approximate Solution by Neuro-Dynamic Programming[*]

Systems and Information Engineering Technical Report, SIE-030002

Stephen D. Patek
Department of Systems and Information Engineering
University of Virginia

Abstract

We analyze a class of Markov decision processes with imperfect state information that evolve on an infinite time horizon and have a total cost criterion. Particularly, we are interested in problems with stochastic shortest path sturcture, assuming (1) the existence of a policy that guarantees termination with probability one and (2) the property that any policy that fails to guarantee termination has infinite expected cost from some initial state. We also assume that termination is perfectly recognized. In this paper, we expand upon arguments (given in [11]) for establishing the existence, uniqueness, and characterization of stationary optimal policies and the convergence of value and policy iteration. We also present an illustrative example, involving the search for partially observed target which moves randomly on a grid, and we develop a simulation-based algorithm (based on neuro-dynamic programming techniques) for computing policies that approximately minimize the expected number of stages to complete the search.

1 Introduction

This paper considers a class of imperfectly observed Markov decision processes known as *partially observed stochastic shortest path problems*. The model we analyze extends the *stochastic shortest path* problems of Bertsekas and Tsitsiklis [2] to the case where, instead of knowing precisely the state of an underlying finite-state Markov chain, we receive noisy observations that are correlated with the actual state of the system. The defining characteristics of a partially observed stochastic shortest path problem are (1) that there exists a stationary policy that leads to termination with probability one and (2) that any policy which fails to guarantee termination has infinite expected cost from some initial state. Moreover, in this paper, we assume that termination of the process is perfectly recognized. Building on arguments in [11], we establish the existence of a stationary policy that is optimal within the class of Markov policies, along with the convergence of two standard dynamic programming recursions: value iteration and policy iteration. As another contribution, we present a pursuit/evasion-type example of the analytical framework, with numercial solution by neuro-dynamic programming [3].

1.1 Related Literature

The subject of this paper is closely related to the theory of optimal search and pursuit. Our formulation may be traced back to Eaton and Zadeh in [5], who considered problems with perfect state information. This early research underwent a sequence of extensions and refinements, leading to the stochastic shortest path theory of Bertsekas and Tsitsiklis [2] and to the search theory of Feinberg [6], both of which are set in the context of perfect state information. Other researchers have incorporated *imperfect* state information into their optimal search formulations (e.g. Stone's monograph [15]), although generally these models do not take the form of Markov decision processes. Subsequently to [2], the stochastic shortest path model has been generalized in a number of ways. In [9] and [12], Patek and Bertsekas analyzed the case of two players, where one player seeks to drive the system to termination along a least-cost path and the other seeks to prevent termination altogether. In [10], Patek reexamined the stochastic shortest path formulation in the context of Markov decision processes with an exponential utility function.

[*] This work is supported in part by the National Science Foundation through grant ECS-9875688 (CAREER).

The results in this paper contribute to the literature on *infinite-horizon* POMDPs (partially observed Markov decision processes). For background on this class of problems, we refer generally to the monographs by Bertsekas and Shreve [1] and Dynkin and Yushkevich [4]. Our results are most closely related to those in Sondijk [14] and Platzman [13]. In [14], the discounted case is considered, leading to an analysis similar (but not identical) to ours for the case where termination is inevitable. In [13], relationships between the average cost and discounted cost formulations are explored, with the main results dependent on conditions of reachability and detectability. Similar conditions are imposed in this paper that make our problem well-posed. Many other researchers have considered undiscounted Markov decision processes, focusing particularly on the relationship between the average reward and total reward criteria in their various forms. We cite the monographs by Hernández-Lerma and Lasserre [7, 8] for a recent and very general treatment of this topic. While much of the theory for problems with general state and action spaces applies within the context of partially observed stochastic shortest path problems, we are able to derive sharper results by exploiting the special structure of the model. In particular, our structural assumptions (i.e. the existence of a stationary policy that leads to termination with probability one and the property that any policy which fails to guarantee termination has infinite expected cost from some initial state) imply the existence of a unique bounded functional solution to the appropriate optimality equation and the convergence of value and policy iteration to this solution. No auxiliary tests are required to establish the existence of a solution or the convergence of the standard recursions.

1.2 Outline

In Section 2, we formally define the class of models known as partially observed stochastic shortest path problems. Part of this formulation is the introduction of an "information state" that allows us to embed the problem in one of perfect state information over a generalized state space. We also state a number of standard results for the case where termination is inevitable regardless of the sequence of actions applied. For this "easy" case, the dynamic programming operator exhibits an m-stage contraction property, and the usual results (e.g. the existence and uniqueness of a solution to Bellman's equation) can be established by the usual means. In Section 3, we state our results for the "hard" case where there may exist "improper" policies that do not guarantee termination. Here, the contraction property for the dynamic programming operator disappears, and the analysis of the model is substantially more complex. In Section 4 we present an illustrative example, involving the search for a randomly moving target with imperfect observations, and we develop a simulation-based algorithm (inspired by neuro-dynamic programming methods [3]) for minimizing the expected cost of the search. In Section 5, we summarize and discuss our results. As many of the results of this paper appear in [11], the main contributions here are that (i) we give expanded arguments for the main theoretical results and (ii) we present an illustrative example of the framework, with approximate solution by neuro-dynamic programming.

2 Formulation

We consider a controlled Markov chain with state space S and action space (control constraint set) U. The probability of transitioning from $i \in S$ to $j \in S$ under $u \in U$ is $p_{ij}(u)$. The expected cost of transitioning from $i \in S$ under $u \in U$ is $g(i,u)$. After each transition, the system outputs a symbol z from a set of observation symbols Z. The probability of observing z given that we have transitioned to $j \in S$ under the control $u \in U$ is $p_z(j,u)$. We assume the following throughout the entire paper.

Assumption A *The sets S, U, and Z are finite, and we may list the elements of S as $\{1,\dots,n\}$. There is an extra state $\Omega \notin S$ which is zero-cost and absorbing. That is,*

$$p_{\Omega\Omega}(u) = 1 \quad \textit{and} \quad g(\Omega,u) = 0, \qquad \forall\, u \in U. \tag{1}$$

Finally, there is an extra observation symbol $z_\Omega \notin Z$ which is unique to transitions to Ω. That is,

$$p_{z_\Omega}(\Omega,u) = 1 \quad \textit{and} \quad p_{z_\Omega}(j,u) = 0, \qquad \forall\, u \in U,\ j \in S. \tag{2}$$

Assumption A is analogous to Assumption 2 in [2]. The last part of Assumption A is a formal statement of the property that Ω is perfectly recognized.

In controlling the Markov chain, we restrict attention to deterministic Markov policies that select actions at each stage based on the posterior probability distribution for the current state given the priors, controls, and observations available up to the current time. We let x_i^0 denote the prior probability that the system starts out in state i. Then, $x^0 = (x_1^0, \ldots, x_n^0) \in X$ is the initial *information state*, where

$$X = \{x \in \Re^n \mid \sum_{i \in S} x_i = 1 \text{ and } x_i \geq 0,\ i = 1, \ldots, n\}. \tag{3}$$

We let x_i^t denote the conditional probability that the system is in state i at stage t given x^0, $u_0, \ldots, u_{t-1}$, and $z_1, \ldots, z_t$, where u_s is the control applied at stage s and z_s is the observation made just before the s-th control is applied. If the process has not terminated by stage t, then $x^t = (x_1^t, \ldots, x_n^t) \in X$ is the information state at stage t, otherwise $x^t = (0, \ldots, 0) \notin X$. Bayes' rule describes the information state dynamics: given that we observe z_{t+1} after applying the control u_t from x^t,

$$\begin{aligned} x_i^{t+1} &= \frac{\sum_{k \in S} x_k^t p_{ki}(u_t) p_{z_{t+1}}(i, u_t)}{\sum_{j \in S \cup \{\Omega\}} \sum_{k \in S} x_k^t p_{kj}(u_t) p_{z_{t+1}}(j, u_t)} && (4) \\ &\stackrel{\triangle}{=} f_i(x^t, u_t, z_{t+1}). && (5) \end{aligned}$$

With $f : X \times U \times Z \mapsto X \cup \{(0, \ldots, 0)\}$ denoting the vector function whose components are the f_i above, we have $x^{t+1} = f(x^t, u_t, z_{t+1})$. Note that f is continuous as a function of $x \in X$, even if the next observation symbol is z_Ω. We pause to note that by introducing the information state $x \in X$ we are essentially embedding the original problem in a new problem with perfect state information, though one with a polyhedral state space.

We use M to denote the set of decision maps, i.e. functions $\mu : X \mapsto U$. The set of nonstationary policies is $\overline{M} = \{(\mu_0, \mu_1, \ldots) | \mu_k \in M\}$, where, for a given policy $\pi = (\mu_0, \mu_1, \ldots)$, each μ_k describes how the information state at stage k will be mapped to an action in U. Since the dependence is always on the current information state, we may think of $\overline{M}$ as the set of deterministic Markov policies. There is no explicit dependence on past information states and controls. We will sometimes abuse notation slightly by using $\mu \in M$ to denote the stationary policy $(\mu, \mu, \ldots) \in \overline{M}$.

Let J be the space of functions $J : X \mapsto \Re$, equipped with metric topology induced by the sup-norm $\|J\|_\infty = \sup_{x \in X} |J(x)|$. Let J_B and J_C be the subspaces of bounded and continuous functions $J \in J$, both with topologies induced by $\|\cdot\|_\infty$. (The main results of this paper will be stated in terms of J_B and J_C.) Note that J_B is not the same as the space $L^\infty(X)$ because we do not assert the equivalence of functions that are equal "almost everywhere" and $\|\cdot\|_\infty$ is not defined by the "essential supremum." Even so, both J_B and J_C turn out to be Banach spaces. For convenience, let $\mathbf{0}$ and $\mathbf{1}$ be the constant functions in J_C evaluating to zero and one, respectively. Also, given $J, J' \in J$, if $J(x) \leq J'(x)$ for all $x \in X$, then we write $J \leq J'$. A final piece of notation before moving on: if $\{J_k\}_{k=0}^\infty \subset J$ converges pointwise to $J \in J$, we write $J_k \xrightarrow{pw.} J$.

2.1 Dynamic Programming Formuation

Given $\mu \in M$, we define $c_\mu \in J_B$ as

$$c_\mu(x) = \sum_{i \in S} x_i g(i, \mu(x)). \tag{6}$$

Note that c_μ may not be continuous as a function of $x \in X$, depending on $\mu \in M$. We define the operator $P_\mu : J \mapsto J$, such that

$$[P_\mu J](x) = \sum_{z \in Z} \sum_{k \in S} \sum_{i \in S} x_i p_{ik}(\mu(x)) p_z(k, \mu(x)) J(f(x, \mu(x), z)). \tag{7}$$

For convenience, we define

$$\bar{p}_z(x, u) = \sum_{k \in S} \sum_{i \in S} x_i p_{ik}(u) p_z(k, u), \tag{8}$$

so that

$$[P_\mu J](x) = \sum_{z \in Z} \bar{p}_z(x, \mu(x)) J(f(x, \mu(x), z)).$$

Note that for any $J \in J$ we have $\|P_\mu J\|_\infty \leq \|J\|_\infty$, so that $P_\mu : J_B \mapsto J_B$. The expression $c_\mu(x) + [P_\mu J](x)$ may be interpreted as the expected cost associated with one state transition under the decision map μ from the prior $x^0 = x$ plus the expected cost $J(x^1)$ of winding up with the information state x^1. (Implicit in this interpretation is the fact that the cost remaining to be experienced from the terminal state Ω is zero. Since termination is perfectly recognized, there is no need to include z_Ω in the summation over observation symbols.) The operator $T_\mu : \mathcal{J} \mapsto \mathcal{J}$ associated with $\mu \in M$ is defined by the mapping,

$$[T_\mu J](x) = c_\mu(x) + [P_\mu J](x). \tag{9}$$

The dynamic programming operator $T : \mathcal{J} \mapsto \mathcal{J}$ is defined by the mapping,

$$TJ(x) = \min_{u \in U} \left[\sum_{i \in S} x_i g(i,u) + \sum_{z \in Z} \bar{p}_z(x,u) J(f(x,u,z)) \right], \tag{10}$$

which in shorthand notation may be written as $TJ = \min_{\mu \in M} T_\mu J$. Note that if $J \in \mathcal{J}$ is continuous, then so is TJ. Thus, T "preserves" continuity (and therefore also boundedness). In general, T_μ only preserves boundedness. The following standard properties also hold.

(i) **(Monotonicity)** If $J, J' \in \mathcal{J}$ are such that $J \geq J'$, then $TJ \geq TJ'$ and $T_\mu J \geq T_\mu J'$ for any $\mu \in M$.

(ii) **(Cost Shifting)** For all $\varepsilon > 0$ we have, $T(J + \varepsilon \mathbf{1}) \leq TJ + \varepsilon \mathbf{1}$. The inequality reverses with $\varepsilon < 0$.

(iii) **(Continuity with respect to $\|\cdot\|_\infty$)** For all $J, J' \in \mathcal{J}$, we have $\|TJ - TJ'\|_\infty \leq \|J - J'\|_\infty$.

In Section 3, we introduce an alternative form of continuity which allows us to derive our main results.

Given an allowable policy, $\pi = \{\mu_0, \mu_1, \ldots\} \in \bar{M}$, we define the associated cost (cost-to-go) function $J_\pi : X \mapsto \Re$ as

$$J_\pi(x) = \liminf_{t \to \infty} [(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_t})(\mathbf{0})](x). \tag{11}$$

We interpret $J_\pi(x)$ to be the expected long term cost of π from the prior state distribution x. If $\pi = \{\mu, \mu, \ldots\}$, then J_μ denotes the cost function for π as defined above. (Note that, just as c_μ may not be continuous as a function of $x \in X$, the function J_μ may not be continuous.) The optimal expected long term cost function $J^* : X \mapsto \Re$ is defined as

$$J^*(x) = \inf_{\pi \in \bar{M}} J_\pi(x). \tag{12}$$

2.2 Inevitable Termination

In this subsection we consider the case where termination is inevitable under all stationary policies. To set this context more precisely, let $P_\Omega^m(i, \{u_t\}_{t=0}^{m-1})$ be the probability of terminating within m stages under the sequence of actions $\{u_t\}_{t=0}^{m-1}$ from the known initial state i. Let $P_\Omega^m(x, \{u_t\}_{t=0}^{m-1})$ be the probability of terminating within m stages under the sequence of actions $\{u_t\}_{t=0}^{m-1}$ from the prior state distribution $x \in X$. Let $P_\Omega^m(x, \pi)$ be the probability of terminating within m stages under $\pi = \{\mu_0, \mu_1, \ldots\} \in \bar{M}$ from the prior state distribution $x \in X$. The case of inevitable termination is defined by the following assumption.

Assumption B *There exists a positive integer m such that $P_\Omega^m(i, \{u_t\}_{t=0}^{m-1}) > 0$ for each initial state $i \in S$ and sequence of actions $\{u_t\}_{t=0}^{m-1} \subset U$ applied from i. (Note that this implies $P_\Omega^m(x, \{u_t\}_{t=0}^{m-1}) > 0$ for each prior state distribution $x \in X$.)*

Assumption B implies that regardless of the controls applied the probability of terminating within m stages is positive. We impose Assumption B only for the remainder of this subsection.

Lemma 1 *There exists $\rho > 0$ such that $P_\Omega^m(x, \pi) \geq \rho$ for all $\pi \in \bar{M}$ and $x \in X$.*

Proof: We first establish a lower bound for $P_\Omega^m(i, \pi)$. We do this by defining an auxiliary problem whose solution is the minimal probability of terminating in m stages under perfect state information. Specifically, consider an m-stage stochastic control problem defined on the finite state space $\check{S} = S \cup \{\Omega\}$, where $p_{ij}(u)$ is the probability of transitioning from $i \in S \cup \{\Omega\}$ to $j \in S \cup \{\Omega\}$ under the control u. The control objective is to minimize, under perfect state information, the expected terminal cost $I_\Omega(i^m)$, where $I_\Omega(i) = 0$ if $i \in S$ and $I_\Omega(\Omega) = 1$. Note that this problem can be solved by a backwards dynamic programming recursion, and its optimal value, denoted $\check{J}(i)$, is precisely the minimal probability of terminating within m stages from i in the original decision process. By Assumption B this value is positive for each state $i \in S$. Thus,

$$\rho \stackrel{\triangle}{=} \min_{i \in S} \check{J}(i) > 0. \tag{13}$$

Moreover, because of the nonnegative value of perfect information $P_\Omega^m(i,\pi) \geq \rho$ for any $\pi \in \bar{M}$. (It's harder to evade termination when the state of the system is uncertain after the first transition.) Since $P_\Omega^m(x,\pi) = x_1 P_\Omega^m(1,\pi) + \cdots + x_n P_\Omega^m(n,\pi)$, there exists $\bar{i} \in S$ such that

$$\min_{x \in X} P_\Omega^m(x,\pi) = P_\Omega^m(\bar{i},\pi) \geq \rho.$$

Q.E.D.

Using Lemma 1 we may show that the operator T is an m-stage contraction mapping, and this in turn implies that J^* is the unique bounded solution to the functional equation $T = TJ$, known as Bellman's equation. Moreover, for any initial bounded function $J_0 \in J_B$, the iterates of value iteration $J_{k+1} = TJ_k$ converge to J^*. These results can be established using more or less standard techniques and are stated below without proof.

Lemma 2 *The following property holds:*

$$\|(T^m J) - (T^m J')\|_\infty \leq (1-\rho)\|J - J'\|_\infty, \qquad \forall\, J, J' \in J,$$

where $\rho \in (0,1]$ *is defined in Equation (13). The same property holds for* T_μ *for any* $\mu \in M$.

Lemma 3 *Given* $\mu \in M$,

1. J_μ *is the unique fixed point of the operator* T_μ *on* J_B.
2. *For every* $J \in J_B$ *we have* $\|T_\mu^k J - J_\mu\|_\infty \to 0$.

Proposition 1 *The following are true.*

1. J^* *is continuous and is the unique fixed point of the operator* T *in* J_B.
2. *(Value Iteration) For every* $J \in J_B$ *we have* $\|T^k J - J^*\|_\infty \to 0$.

The fact that J^* is continuous follows from the Banach fixed point theorem applied to T in J_C.

3 Analysis of the Case where Termination is not Inevitable

In this section we relax the assumption that termination is inevitable and allow for the existence of policies that are non-terminating as long as they result in infinite expected cost from some initial state. In effect, we replace Assumption B with the following.

Assumption C *There exists a stationary policy* $\pi = \{\mu, \mu, \ldots\} \in \bar{M}$ *such that for all* $x \in X$

$$\lim_{m \to \infty} P_\Omega^m(x,\pi) = 1. \tag{14}$$

Moreover, any policy $\pi = \{\mu_0, \mu_1, \ldots\} \in \bar{M}$ *that fails to satisfy this condition is such that a subsequence of*

$$\left\{ [(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_t})(\mathbf{0})](x) \right\}_{t=0}^{\infty}$$

tends to infinity for some initial information state $x \in X$.

Our objectives are to (i) characterize J^* as the unique fixed point of T in some subspace of J, (ii) establish the existence of stationary policies $\mu^* \in M$ which achieve J^*, and (iii) establish the convergence properties of the dynamic programming recursions, value and policy iteration. We will refer to stationary policies that satisfy the limit of Equation (14) as *proper*; we'll call stationary policies that don't satisfy the limit *improper*.

The analysis here is complicated by the fact that T fails to be a contraction mapping with respect to any norm, as shown in [2]. Moreover, the dynamic programming operator T is not known to be an m-stage contraction. (The proof of Lemma 1 breaks down.) Because we cannot rely upon standard fixed point theory, we are forced to settle for dynamic programming arguments with pointwise convergence to a solution of the functional equation $J = TJ$. We are led to introduce an alternative form of continuity for the operator T, as in the following lemma.

Lemma 4 (Pointwise Continuity) *Given a sequence of functions $\{J_k\}_{k=0}^{\infty} \subset J$, if $J_k \xrightarrow{pw.} J \in J$, then $TJ_k \xrightarrow{pw.} TJ$.*

Proof. Suppose $J_k \xrightarrow{pw.} J \in J$. We must show that $\lim_{k\to\infty} TJ_k(x) = TJ(x)$ for all $x \in X$. To this end, arbitrarily fix $x \in X$. Let $\bar{X}(x,u)$ be the set of nonterminal information states that can be reached (via Bayes' rule) from x under the control u. Note that $\bar{X}(x,u)$ is finite since Z is finite. Now define $c_k = \max_{u\in U} \max_{\bar{x}\in\bar{X}(x,u)} |J_k(\bar{x}) - J(\bar{x})|$. From the monotonicity and cost-shifting properties of T, we have that $|TJ_k(x) - TJ(x)| \leq c_k$. Since $J_k \xrightarrow{pw.} J$, we can choose k to make c_k arbitrarily small. [Pointwise convergence is sufficient because we care about the limiting behavior of $\{J_k(\bar{x})\}_{k=0}^{\infty}$ only at a finite number of points $\bar{x} \in \cup_{u\in U}\bar{X}(x,u)$.] **Q.E.D.**

The following lemma establishes a convenient test for determining whether a stationary policy is proper. The proof follows similarly to arguments given in [2].

Lemma 5 *Given $\mu \in M$, if there exists a function $\bar{J} \in J_B$ such that $\bar{J} \geq T_\mu \bar{J}$, then μ is proper.*

Proof. To reach a contradiction, suppose μ is improper. By expanding $(T_\mu^t)(\mathbf{0})$, Assumption C implies the existence of $\bar{x} \in X$ such that a subsequence of

$$\left\{ \sum_{k=0}^{t} \left[P_\mu^k c_\mu \right] (\bar{x}) \right\}_{t=0}^{\infty} \tag{15}$$

tends to infinity, where $[P_\mu^k c_\mu](x) = [(P_\mu \circ \cdots \circ P_\mu)c_\mu](x)$. On the other hand, there exists by hypothesis $\bar{J} \in J$ such that $\bar{J} \geq T_\mu \bar{J}$. Applying T_μ to $\bar{J}$, we have that $\bar{J} \geq T_\mu \bar{J} \geq c_\mu + P_\mu \bar{J}$, where the second inequality follows from the definition of T_μ. From the monotonicity of T_μ, we obtain

$$\begin{aligned} \bar{J} \geq T_\mu \bar{J} \geq T_\mu^2 \bar{J} &\geq T_\mu(c_\mu + P_\mu \bar{J}) \\ &\geq P_\mu P_\mu \bar{J} + [c_\mu + P_\mu c_\mu], \end{aligned}$$

where the last inequality follows again from the definition of T_μ. Proceeding inductively we obtain

$$\bar{J} \geq T_\mu^t \bar{J} \geq P_\mu^{t+1} \bar{J} + \sum_{k=0}^{t} P_\mu^k c_\mu. \tag{16}$$

Since $\bar{J}$ is bounded, there exists $M \in \Re_+$ such that $|\bar{J}(x)| \leq M$. Since $\|P_\mu J\|_\infty \leq \|J\|_\infty$ for all $J \in J$, the term in the far right-hand-side of Equation (16) involving $\bar{J}$ is bounded (by M) as $t \to \infty$. Thus, Equation (16) implies that no subsequence of Equation (15) may converge to infinity, and μ must be proper by contradiction. **Q.E.D.**

We now show that there exists a unique solution to Bellman's equation $J = TJ$ in the space of bounded functions J_B. We do this by establishing the pointwise convergence of the "policy iteration" and "value iteration" algorithms. The argument hinges upon the following observations.

1. The results of Lemma 3 apply to any proper policy μ, as though it were the only available policy (resulting in a new problem in which Assumption B holds).
2. The cost function J_μ of any proper policy μ is bounded below by the optimal cost of a revised stochastic shortest path problem where perfect information prevails. (Cf. [2]. An optimal solution to the revised problem exists since the stochastic shortest path aspect of the cost structure remains intact.)

Proposition 2 *The following are true.*

1. *The operator T has a unique fixed point in J_B; this fixed point is exactly J^*. There exists a proper optimal policy $\mu^* \in M$. Finally, a stationary policy $\mu^* \in M$ is optimal if and only if $T_{\mu^*} J^* = TJ^*$.*
2. *(Value Iteration) For every $J \in J_B$, we have that $T^k J \xrightarrow{pw.} J^*$.*
3. *(Policy Iteration) Given a proper policy $\mu_0 \in M$, we have that $J_{\mu_k} \xrightarrow{pw.} J^*$, where $\{\mu_k\}_{k=0}^{\infty}$ is a sequence of policies such that $T_{\mu_k} J_{\mu_{k-1}} = TJ_{\mu_{k-1}}$.*

Proof. The uniqueness of a fixed point in J_B follows from the monotonicity of T and Lemmas 3 and 5, as in the proof of Proposition 2 in [2]. The existence of a fixed point and the pointwise convergence of policy iteration, established below, follow from arguments similar to those in [2]. Let μ_0 proper and $\mu_1 \in M$ such that $T_{\mu_1} J_{\mu_0} = TJ_{\mu_0}$. (Assumption C implies that an initial proper policy μ_0 exists.) We have $T_{\mu_1} J_{\mu_0} = TJ_{\mu_0} \leq T_{\mu_0} J_{\mu_0} = J_{\mu_0}$. By Lemma 5, μ_1 is proper. By the monotonicity of T_{μ_1} and Lemma 3, we have that for all t

$$J_{\mu_0} \geq TJ_{\mu_0} \geq T_{\mu_1}^{t-1} J_{\mu_0} \geq T_{\mu_1}^{t} J_{\mu_0}.$$

Thus,

$$J_{\mu_0} \geq TJ_{\mu_0} \geq \lim_{t \to \infty} T_{\mu_1}^{t} J_{\mu_0} = J_{\mu_1}.$$

Applying this argument iteratively, we construct a sequence $\{\mu_k\}$ of proper policies such that,

$$J_{\mu_k} \geq TJ_{\mu_k} \geq J_{\mu_{k+1}}, \qquad \forall\, k = 0, 1, \ldots\ . \tag{17}$$

Thus, $\{J_{\mu_k}(x)\}_{k=0}^{\infty}$ is monotonically decreasing for each $x \in X$. Moreover, by considering a revised version of the problem where perfect state information prevails, we may establish a lower bound for the sequence $\{J_{\mu_k}(x)\}_{k=0}^{\infty}$. Let $\check{J}^*(i) \in \Re$ be the minimal expected cost-to-go from $i \in S$ for the revised problem (cf. [2].) Because of the value of perfect information, each element of $\{J_{\mu_k}(x)\}_{k=0}^{\infty}$ is bounded below by $\min_{i \in S} \check{J}^*(i)$. Thus, $\{J_{\mu_k}\}_{k=0}^{\infty}$ converges pointwise to some bounded function J^∞. From Equation (17) and the pointwise continuity of T (cf. Lemma 4), we have that $J^\infty = TJ^\infty$, the unique fixed point of T in J_B.

The following arguments for (i) the pointwise convergence of value iteration for all initial $J \in J_B$, (ii) the fact that $J^\infty = J^*$, and (iii) existence of an optimal proper policy and the fact that μ^* is optimal if and only if $T_{\mu^*} J^* = TJ^*$, are similar to arguments given in [2] (cf. Proposition 2.)

First, we show that show that $T^k J \xrightarrow{pw.} J^\infty$ for all $J \in J_B$. This argument is given in the following three paragraphs. Before we proceed, let $\mu^* \in M$ be such that $T_{\mu^*} J^\infty = TJ^\infty = J^\infty$. Note that the final equality implies that μ^* is proper, and from the uniqueness of the fixed point $J_{\mu^*} = J^\infty$. Let δ be some positive scalar.

We claim that there exists a unique bounded function J^δ which satisfies $T_{\mu^*} J^\delta = J^\delta - \delta\mathbf{1}$. To see this, consider a revised problem where are all of the transition costs from states $i \in S$ are increased by δ. Since μ^* is proper, the cost function $\tilde{J}$ associated with μ^* in the revised problem is characterized by the unique solution to Bellman's equation (cf. Lemma 3). Since

$$\begin{aligned} \tilde{J} &= c_{\mu^*} + \delta\mathbf{1} + P_{\mu^*}\tilde{J} \\ &= \delta\mathbf{1} + T_{\mu^*}\tilde{J}, \end{aligned}$$

we have that $J^\delta = \tilde{J}$, establishing our claim. Since $\delta > 0$ we have also shown that $J^\delta \geq T_{\mu^*} J^\delta$. Thus, from the monotonicity of T_{μ^*} we have that for all $k > 0$

$$T_{\mu^*}^{k} J^\delta \leq J^\delta.$$

By taking the limit as $k \to \infty$, we see that $J_{\mu^*} \leq J^\delta$. Using the monotonicity of T and the fact that $J^\infty = J_{\mu^*}$, we get

$$J^\infty = TJ^\infty \leq TJ^\delta \leq T_{\mu^*} J^\delta = J^\delta - \delta\mathbf{1} \leq J^\delta.$$

Proceeding inductively, we get

$$J^\infty \leq T^k J^\delta \leq T^{k-1} J^\delta \leq J^\delta.$$

Hence, $\{T^k J^\delta(x)\}_{k=0}^{\infty}$ is a monotonically decreasing sequence which is bounded below for each $x \in X$ and therefore $\{T^k J^\delta\}_{k=0}^{\infty}$ converges pointwise to some $\bar{J}^\infty \in J_B$. By the pointwise continuity property of the operator T, we must have that $\bar{J}^\infty = T\bar{J}^\infty$. By the uniqueness of the fixed point of T in J_B, we have that $\bar{J}^\infty = J^\infty$. Thus, we have shown that

$$T^k J^\delta \xrightarrow{pw.} J^\infty. \tag{18}$$

We now examine the convergence of the operator T^k applied to $J^\infty - \delta\mathbf{1}$. Note that

$$J^\infty - \delta\mathbf{1} = TJ^\infty - \delta\mathbf{1} \leq T(J^\infty - \delta\mathbf{1}) \leq TJ^\infty = J^\infty,$$

where the first inequality follows from the cost shifting property of T. Once again, the monotonicity of T prevails, implying that $T^k(J^\infty - \delta\mathbf{1})$ is pointwise monotonically increasing and bounded above. From the pointwise continuity of T and the uniqueness of the fixed point we have that

$$T^k(J^\infty - \delta\mathbf{1}) \xrightarrow{pw.} J^\infty. \tag{19}$$

We saw earlier that $J^\delta = T_{\mu^*}J^\delta + \delta\mathbf{1}$ and that $J^\delta \geq J^\infty$. Then,

$$J^\delta = T_{\mu^*}J^\delta + \delta\mathbf{1} \geq T_{\mu^*}J^\infty + \delta\mathbf{1} = J^\infty + \delta\mathbf{1}.$$

Thus, for any $J \in J_B$ we can find $\delta > 0$ such that $J^\infty - \delta\mathbf{1} \leq J \leq J^\delta$. By the monotonicity of T, we then have

$$T^k(J^\infty - \delta\mathbf{1}) \leq T^k J \leq T^k J^\delta, \quad \forall\, k \geq 1.$$

Taking pointwise limits and using Equations (18) and (19) we see that $\lim_{k\to\infty} T^k J(x) = J^\infty(x)$, for all $x \in X$.

All that remains to be shown is that $J^\infty = J^*$. Take any $\pi = \{\mu_0, \mu_1, \ldots\} \in \bar{M}$. From the monotonicity of T and T_μ (for every $\mu \in M$), we have

$$T^k\mathbf{0} = T^{k-1}T\mathbf{0} \leq T^{k-1}T_{\mu_k}\mathbf{0} \leq \cdots \leq T_{\mu_0}T_{\mu_1}\cdots T_{\mu_k}\mathbf{0}, \quad \forall\, k > 0.$$

Taking the pointwise limit inferior of both sides we obtain $J^\infty \leq J_\pi$. Since $J^\infty = J_{\mu^*}$ we have that μ^* is optimal and $J^\infty = J^*$. **Q.E.D.**

3.1 Discussion

From [2], we know that in problems with perfect state information where termination is inevitable under all stationary policies the dynamic programming operator is a contraction mapping with respect to a weighted sup-norm. Apparently, this result hinges on the finiteness of the state space. (Finiteness of the state space is critical in all known proofs for the result.) Since the state space in this paper is X (which is uncountably infinite), it is an open question whether a one-stage contraction property holds under our Assumption B.

Overall, the results of this section generalize [2] to the case of imperfect state information. However, there is one important feature in [2] that we were unable to capture here, namely compact constraint sets U. Our assumption of finite U was necessary to prove Lemma 4, which, in turn, is essential in establishing our main results. Otherwise, the arguments in this paper are structured so that the main results hold even with compact constraint sets (along with lower semicontinuous functions $p_{i,j}(u)$ and $g(i,u)$). The generalization to compact U would be easy as long as an alternative proof to Lemma 4 can be found.

4 Example with Approximate Solution by Neuro-Dynamic Programming

In this section we illustrate the basic model of Sections 2 and 3 by considering a pursuit/evasion problem in which a single player (the "pursuer") seeks to minimize the expected number of stages to identifying the true location of a randomly moving and imperfectly observed target.

4.1 Search Model

We consider an idealized search model that involves a target (think: "submarine") which moves randomly on an $N \times N$ grid. The motion of the target evolves according to a controlled, hidden Markov chain, where, given that the target is presently located grid cell (i_x, i_y), the probability of transitioning to a neighboring cell is given as a function of the search activity of the pursuer. In this model, we take the perspective of the pursuer who seeks to isolate the location of the target according to the following process.

1. Initially:

   (a) The pursuer starts with a probability distribution $x^0 = (x_{(i_x,i_y)})_{i_x,i_y\in\{1,\ldots N\}}$, which describes the prior probability of the target existing in each grid cell (i_x, i_y).

2. At the t-th stage of the process:

   (a) Having the probability distribution x^t, which describes the current location of the target, the pursuer decides upon a grid cell $u_t = (u_x^t, u_y^t)$ to search. The pursuer perceives a cost of one for engaging in this stage of the search process. (Thus, the total cost of the search is equal to the total number of stages to termination.)

   (b) The target randomly moves randomly according to the hidden, controlled Markov chain.

      i. If the target moves into the cell searched by the pursuer, then the search process terminates, and the pursuer accrues no more cost.

      ii. Otherwise, if the target moves into some other cell, then the search results in a noisy observation $z_{t+1} = (z_x^{t+1}, z_y^{t+1})$ of the target's new location, and the pursuer computes (via Bayes' rule) the posterior distribution for the target's location x^{t+1}.

   This process continues until the pursuer arranges to search in the same cell as the new location of the target, as in step 2(b)i.

In seeking to minimize the expected cost to termination, the pursuer faces a partially observed stochastic shortest path problem of the type described earlier in this paper. Particularly, if the pursuer limits attention to grid cells for which the probability of the target arriving is greater than a given minimal threshold $p_\Omega > 0$, then termination is inevitable and the results of Proposition 1 apply. Otherwise, if the pursuer is free to search any cell (regardless of how unlikely it is that the target will arrive), then termination is not inevitable, but the results of Proposition 2 apply. In either case, minimizing the expected cost to termination is an extremely difficult task since the optimal solution is characterized by the unique solution $J^* : X \mapsto \Re$ to Bellman's equation $J = TJ$. We describe a generic, approximate solution methodology for this search problem in Section 4.2, and we test this methodology in Section 4.3 for a particular instance of the search problem.

4.1.1 Target Motion

To give the details of a particular instance of the search problem, we begin by describing the hidden, controlled Markov chain that characterizes the target's motion. Given that the target is presently located grid cell (i_x, i_y), the probability of transitioning to cell $(j_x, j_y) = (i_x + d_x,\, i_y + d_y)$, with d_x and d_y elements of $\{-1, 0, 1\}$, is given by

$$p_{(i_x,i_y),(j_x,j_y)}(u_x,u_y) = \frac{w_{(i_x+d_x,\, i_y+d_y)}(u_x,u_y)}{\sum_{\delta_x=-1}^{1}\sum_{\delta_y=-1}^{1} w_{(i_x+\delta_x,\, i_y+\delta_y)}(u_x,u_y)}, \tag{20}$$

where

1. $w_{(i_x+\delta_x,\, i_y+\delta_y)}(u_x,u_y)$ is a weight that determines the likelihood of transitioning into grid cell $(i_x + \delta_x,\, i_y + \delta_y)$ from neighboring cells and

2. (u_x, u_y) refers to the decision of the pursuer to search in grid cell (u_x, u_y).

The weights $w_{(j_x,j_y)}(u_x,u_y)$ are computed with respect to an underlying set of weights $\tilde{w}_{(j_x,j_y)}(u_x,u_y)$ that are associated with the particular features of the search grid. In particular,

$$w_{(j_x,j_y)}(u_x,u_y) = \begin{cases} \frac{\tilde{w}_{(j_x,j_y)}(u_x,u_y)}{2} & \text{if } j_x = u_x \text{ and } j_y = u_y, \\ \tilde{w}_{(j_x,j_y)}(u_x,u_y) & \text{otherwise.} \end{cases} \tag{21}$$

The "divide-by-two rule" of Equation (21) for the case where $j_x = u_x$ and $j_y = u_y$ serves to model an inherent ability of the target to anticipate the pursuer's search activity. The result is that the target is less likely to visit the searched-cell (u_x, u_y) than would be indicated by the underlying weights.

4.1.2 Observations of the Target

We now lay out the process by which observed locations of the target are generated. Recall that whenever the pursuer searches a cell (u_x, u_y) and fails to find the target there, the pursuer receives a noisy observation that suggests the new location of the target. The observed location of the target (z_x^{t+1}, z_y^{t+1}) can only occur within a set of allowable grid cells, namely those that appear in the same quadrant or half plane as the actual (new) location of the target (j_x, j_y) relative to the search-cell (u_x, u_y). For example, if the pursuer searches $(4,4)$ and the target actually transitions to $(2,3)$, then the set of allowable cells includes all those contained within the region defined by four corners: $(1,1)$, $(1,4)$, $(4,4)$, and $(4,1)$, including the boundaries. If the target had actually transitioned to $(5,5)$ instead, then the set of allowable cells would be the cells contained within $(4,4)$, $(4,N)$, (N,N), and $(N,4)$. If the searched cell and the new location of the target agree in one dimension, then the set of allowable cells is a “halfplane.” For example, if the pursuer searches $(4,4)$ and the target actually transitions to $(2,4)$, then the set of allowable cells would be the cells contained within $(1,1)$, $(N,1)$, $(N,4)$, and $(1,4)$. The observed location of the target is drawn from the set of allowable cells according to a probability distribution that depends on the distance between (j_x, j_y) and (u_x, u_y):

$$d((j_x, j_y), (u_x, u_y)) = \max\{|j_x - u_x|, |j_y - u_y|\}, \tag{22}$$

and also on the distance between (j_x, j_y) and the allowable cells that serve as potential observed locations of the target. To describe the selection process, let D denote the largest possible distance between (j_x, j_y) and any allowable cell:

$$D = \max_{\text{allowable } (z_x, z_y)} d((j_x, j_y), (z_x, z_y)), \tag{23}$$

and let n_d denote the number of allowable cells whose distance from (j_x, j_y) is exactly d. (There is exactly one allowable cell whose distance is zero from (j_x, j_y), namely (j_x, j_y) itself, so $n_0 = 1$.) Define p_d so that $n_d p_d$ is the probability of observing the target d units away from the true location of the target. (Thus, p_d is the probability that a particular allowable cell that is distance d away from (j_x, j_y) will be selected as the observed location of the target.) In the numerical evaluation of Section 4.3, we assign values to $p_0, p_1, \ldots, p_D$ so that

$$\begin{aligned} \alpha \cdot n_D p_D &= n_{D-1} p_{D-1}, \\ \alpha \cdot n_{D-1} p_{D-1} &= n_{D-2} p_{D-2}, \\ &\vdots \\ \alpha \cdot n_1 p_1 &= p_0, \end{aligned} \tag{24}$$

where α is a parameter that describes the accuracy of the sensor computed as a function of the distance between the actual location of the target and the search location:

$$\alpha = 1 + K_0 e^{-K_1 d((j_x, j_y), (u_x, u_y))}, \tag{25}$$

where K_0 and K_1 are parameters. To summarize, the allowable cell (z_x, z_y), which is d distance from the actual location of the target, is selected as the observed location of the target (z_x^{t+1}, z_y^{t+1}) with probability p_d.

4.1.3 Information State Dynamics

Upon searching a grid cell (u_x, u_y), not finding the target there, and receiving (z_x^{t+1}, z_y^{t+1}) as the observed location of the target, the pursuer assesses the situation by computing x^{t+1} via Bayes’ rule:

$$x_{(i_x,i_y)}^{t+1} = \frac{\sum_{k_x=1}^{N} \sum_{k_y=1}^{N} x_{(k_x,k_y)}^{t} p_{(k_x,k_y),(i_x,i_y)}(u_x, u_y) p_{(i_x,i_y)}(z_x^{t+1}, z_y^{t+1})}{\sum_{j_x=1}^{N} \sum_{j_y=1}^{N} \sum_{k_x=1}^{N} \sum_{k_y=1}^{N} x_{(k_x,k_y)}^{t} p_{(k_x,k_y),(j_x,j_y)}(u_x, u_y) p_{(j_x,j_y)}(z_x^{t+1}, z_y^{t+1})}, \tag{26}$$

where

1. $p_{(k_x,k_y),(i_x,i_y)}(u_x, u_y)$ is the probability that the target transitions into (i_x, i_y) given that the pursuer searches in (u_x, u_y) and that the target was in cell (k_x, k_y), as discussed in Section 4.1.1,
2. $p_{(i_x,i_y)}(z_x^{t+1}, z_y^{t+1})$ is the probability that (z_x^{t+1}, z_y^{t+1}) is the observed location of the target given that the actual (new) location of the target is (i_x, i_y), as discussed in Section 4.1.2.

4.1.4 Baseline Policy, μ_0

Computing an optimal policy for this problem is extremely difficult since (as a partially observed stochastic shortest path problem) the optimal cost-to-go function is a function (not a finite-dimensional vector). The value and policy iteration algorithms described in Propositions 1 and 2 are difficult to implement in software, at least precisely, since the limiting structure of the iterates (J_k and J_{μ_k}, respectively) is unknown. A reasonable baseline strategy for the pursuer, which we may take as a heuristic solution to the problem, comes from ignoring the "divide-by-two rule," which reduces the likelihood of the target to move in to the cell. Let $\tilde{x}_{(i_x,i_y)}$ denote the probability that target will transition into (i_x, i_y) in the next stage, given that the "divide-by-two rule" is not in effect:

$$\tilde{x}_{(i_x,i_y)} = \sum_{k_x=1}^{N} \sum_{k_y=1}^{N} x^t_{(k_x,k_y)} \tilde{p}_{(k_x,k_y),(i_x,i_y)}, \tag{27}$$

where $\tilde{p}_{(k_x,k_y),(i_x,i_y)}$ is the probability that the target transitions into (i_x, i_y) given that the target was in (k_x, k_y), computed as in Section 4.1.1 using the underlying weights $\tilde{w}_{(k_x,k_y)}$ without accounting for the search region selected by the pursuer. We define the baseline policy μ_0 as follows:

$$\mu_0(x^t) = \arg \max_{(i_x,i_y)\in\{1,\ldots,N\}^2} \tilde{x}_{(i_x,i_y)}. \tag{28}$$

In other words, μ_0 directs the pursuer to search the cell which is most likely to receive the target ignoring the divide-by-two effect of the search. This policy has the virtue being simple. Moreover, it also ensures the eventual termination of the search process (with probability one) since the per-stage probability of terminating is lower-bounded away from zero. On the other hand, μ_0 is clearly suboptimal since it ignores the "divide-by-two rule" that characterizes the target's motion. More importantly, μ_0 is myopic in the since that it always selects the next search cell based on the likelihood of termination in the *next* stage. An optimal search strategy will take a longer view in selecting cells to search, sometimes selecting search-cells in an attempt to herd the target into regions of the search grid where the likelihood of detecting the target is higher.

4.2 Approximate Solution by Neuro-Dynamic Programming

We describe here a simulation-based, approximate solution methodology for the search problem of Section 4.1. The methodology is based on an algorithm known as approximate policy iteration, which is one of several neuro-dynamic programming algorithms [3] originally designed to solve Markov decision processes with perfect state information. Approximate policy iteration proceeds as a sequence of policy evaluations and policy improvements, as in the (exact) policy iteration algorithm of Proposition 2. The main difference here is that both the evaluation and improvement steps are done approximately, computed with respect to simulation data and neural-network approximations of the cost-to-go function. We the refer the interested reader to [3, 16] for a more comprehensive review of related literature on neuro-dynamic programming and reinforcement learning.

A key component in what follows is the notion of an "approximation architecture," which is the mathematical form of the approximation of the cost-to-go function associated with the policies produced in approximate policy iteration. In this paper, we use a so-called linear approximation architecture, where the approximation of expected cost-to-go from the information state x is computed as the inner product of a *feature vector* $f(x) \in \Re^r$ and a *weights vector* $\omega \in \Re^r$:

$$\text{expected cost-to-go from } x \;\approx\; f(x)'\omega.$$

The elements of $f(x)$ are quantitative *features* that are believed to correlate to the cost that remains to be accrued from the information state x before termination. Generally, the features selected for a particular application are based on the user's insight into the problem at hand. We describe in detail the features we use for the search model in Section 4.2.1.

Another key component of our methodological approach is the notion of a greedy policy with respect to the cost-to-go function approximated by a set of weights ω^{k-1}. We say that the policy μ_k is greedy with respect to ω^{k-1} if

$$\mu(x) = \arg \min_{(u_x,u_y)\in\{1,\ldots,N\}^2} \mathbf{E}\left[\, 1 + f(x^1)'\omega^{k-1} \mid x^0 = x,\; u_0 = (u_x, u_y) \,\right]. \tag{29}$$

The actions specified by μ_k are "greedy" in the sense that they minimize the expected cost of the next transition (each search costs one unit) *plus* the approximate expected cost-to-go from the new information state x^1 reached from

x^0 under the given action. Note that if ω^{k-1} results in an error-free approximation of expected cost-to-go for some baseline policy μ_{k-1}, then the policy μ_k that is greedy with respect to ω^{k-1} is the same as the result of one stage of policy iteration starting with μ_{k-1} (cf. Proposition 2, Part 3). To the extent that ω^{k-1} results in an approximation of the expected cost-to-go associated with μ_{k-1}, the greedy policy with respect to ω^{k-1} is an approximation of the result of one stage of policy iteration.

For some problems it may not be possible to evaluate in closed form the expectation in Equation (29). When this is the case, it is still possible to *approximately* compute greedy actions as needed in the course of the simulation. The idea is to replace the expectation in Equation (29) with a sample average over N_{usim} independent trials as follows.

$$\hat{\mu}_k(x^t) = \arg \min_{(u_x,u_y)\in\{1,\ldots,N\}^2} \frac{1}{N_{usim}} \sum_{s=1}^{N_{usim}} 1 + f(X_s)'\omega^{k-1}, \tag{30}$$

where $\{X_1, X_2, \ldots, X_{N_{usim}}\}$ are sample values of information-state reached in one-stage from x^t under the action (u_x, u_y) being evaluated in the minimization.

Algorithm 1 *Approximate Policy Iteration*

Initialization *Given a baseline policy μ_0:*

1. *Simulate μ_0 from a given initial information state x^0 all the way to termination, storing the information state trajectory and associated sample cost to go for N_{sim} independent simulation runs:*

$$\left\{ \left[\, x^{t,\tau},\ \ (T_\tau - t)\,\right] \right\}_{t=0,\ldots,T_\tau,\ \tau=1,\ldots,N_{sim}},$$

where τ is an index that counts complete simulations from 1 to N_{sim}, T_τ is the number of stages to termination in the τ-th simulated trajectory, and $T_\tau - t$ is the sample cost-to-go from the information state $x^{t,\tau}$ in the t-th stage of the τ-th simulated trajectory.

2. *Compute ω^0 to be the weight vector that minimizes the sum of squared between the sample cost-to-go data collected in the simulation and the approximated value:*

$$\omega^0 = \arg \min_{\omega\in\Re^r} \sum_{\tau=1}^{N_{sim}} \sum_{t=0}^{T_\tau} \left[f(x^{t,\tau})'\omega - (T_\tau - t) \right]^2.$$

***k*-th Iteration** *Given the weight vector ω^{k-1} for the approximation of expected cost-to-go for the preceding policy:*

1. *Use Equation (30) to simulate the greedy policy $\hat{\mu}_k$ from x^0 all the way to termination, storing the information state trajectory and associated sample cost to go for N_{sim} independent simulation runs:*

$$\left\{ \left[\, x^{t,\tau},\ \ (T_\tau - t)\,\right] \right\}_{t=0,\ldots,T_\tau,\ \tau=1,\ldots,N_{sim}}.$$

2. *Compute the weight vector ω^k to minimize the sum of squared error in the sample data collected for $\hat{\mu}^{k-1}$:*

$$\omega^k = \arg \min_{\omega\in\Re^r} \sum_{\tau=1}^{N_{sim}} \sum_{t=0}^{T_\tau} \left[f(x^{t,\tau})'\omega - (T_\tau - t) \right]^2.$$

4.2.1 Approximation Architecture for the Search Model

For the numerical results of Section 4.3, we use a very simple set of features for the linear approximation architecture, as described below.

1. The first feature $f_1(x)$ is simply the constant value $f_1(x) = 1$. This provides the opportunity for the approximation function to have an offset away from zero.

2. The next set of features $f_2(x)$ to $f_{N^2+1}(x)$ are the individual elements of the information state x. That is, $f_2(x) = x_{(1,1)}$, $f_3(x) = x_{(1,2)}$, and so on up to $f_{N^2+1}(x) = x_{(N,N)}$.

3. To allow for some nonlinear dependence in the approximation function, the final set of features we use are the products of all neighboring pairs of elements of the information state. Namely, $f_{N^2+2}(x) = x_{(1,1)}x_{(1,2)}$, $f_{N^2+3}(x) = x_{(1,1)}x_{(2,2)}$, $f_{N^2+4}(x) = x_{(1,1)}x_{(2,1)}$, and so on.

Parameter	Description	Value
N	size of the search grid	6
N_{sim}	number of sample trajectories per evaulation	1000
N_{usim}	number of samples for evluateing greedy actions	10
K_0	observation model parameter	10
K_1	observation model parameter	.5

Table 1: Parameters of the search model for the numerical investigation.

4.3 Numerical Results

To illustrate the application of approximate policy iteration, we implemented Algorithm 1 in MATLAB on a 800 MHz PC, running the Linux operating system. As detailed in Tables 1 and 2, we implemented the search model and optimization algorithm for a 6×6 search grid. The target-motion weight parameters $\tilde{w}_{(i_x,i_y)}$ of Table 2 indicate that the target will never transition to the edges of the grid (because of the zeros on the boundary of the search region). Thus, the target is actually constrained to move in the 4×4 region in the middle. On the other hand, depending on the actual location of the target, relative to the grid cell being searched, any of the 6×6 cells may be drawn as the the observed location of the target, according to the observation process described in Section 4.1.2. The observation model parameters K_0 and K_1 are set so that the closer pursuer is to the actual location of the target the more accurate the observed location of the target is. For example, if the pursuer searches grid cell $(4,4)$ when the target actually transitions into $(2,2)$, then the probability that the observed the location of the target will be correct is .7940. On the other hand, if the pursuer searches grid cell $(3,3)$ when the target actually transitions into $(2,2)$, then the probability that the observed location of the target will be correct is .8760.

In implementing Algorithm 1, particularly in simulating each policy produced by the algorithm, we set the initial information state x^0 so that $x^0_{(i_x,i_y)} = 0$ for all grid cells (i_x, i_y) on the boundary of the search region, whereas we set $x^0_{(i_x,i_y)} = 1/16$ for the 4×4 grid cells in the middle of the search region. Also, as indicated in Table 1, we set the parameters N_{sim} and N_{usim}, respectively, to 1000 independent simulation runs per policy evaluation and 10 independent one-stage samples for calculation of the greedy action per stage (cf. Equation (29)). We chose these values heuristically, based on what seems to work for this application. Finally, to keep the run-time of the algorithm reasonable, in identifying the greedy action at each t of a simulation (for policy evaluation), we limited attention to grid cells (u_x, u_y) such that $x^t_{(u_x,u_y)} \geq .0001$.

The results from two independent trials of approximate policy iteration are given in Figure 1. The results for "Run 1" appear on the left, and the results for "Run 2" appear on the right. Both plots show the evolution of Algorithm 1 in terms of the average cost-to-go from x^0 for N_{sim} independent trials, starting with μ_0. The results from both trials are qualitatively similar: the average number of stages to termination for the baseline policy μ_0 is around 16, whereas the average number of stages to termination for the successive policies generated by Algorithm 1 hover somewhere between 9 and 10. These results are rather typical of those generated by approximate policy iteration [3]. Particularly, we observe that, compared to exact policy iteration, the improvement in expected cost from Algorithm 1 is not monotonically decreasing. Rather, the average number of stages to termination after the second iteration tend to rise or fall erratically. These oscillations could be explained by the relatively small sample size ($N_{sim} = 1000$), but the true explaination probably lies in the power of our linear approximation architecture in approximating expected cost-to-go for this problem. Finally, we observe that, even though the target is constrained to move within the 4×4 middle cells of the search grid, it turns out to be rather difficult to pin it down, as evidenced by the performance of the baseline policy μ_0.

Before concluding this section, we remark that the computational requirements of approximate policy iteration can be extremely heavy. For our MATLAB implementation of Algorithm 1, for the case $N = 6$, each policy evaluation stage required approximately three hours of compute time, with the majority of the time spend in computing the greedy action at each stage. Testing the algorithm for $N > 6$, at least in MATLAB, would prohibitively slow.

$i_y \ddots i_x$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	1	2	4	4	0
3	0	1	2	5	5	0
4	0	2	4	6	6	0
5	0	4	5	7	8	0
6	0	0	0	0	0	0

Table 2: Underlying weight parameters $\tilde{w}_{(i_x,i_y)}$ for target motion.

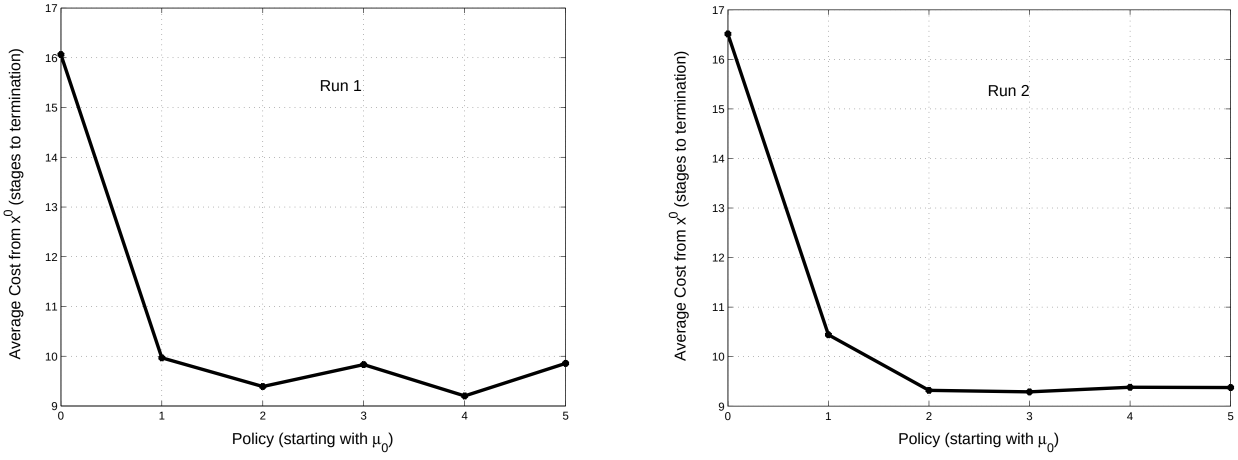

Figure 1: Evolution of approximate policy iteration in independent runs

5 Conclusions

We have analyzed a class of partially observed stochastic shortest path problems. Under standard assumptions of termination inevitable (i.e. Assumptions A and B), the optimal cost-to-go function J^* is characterized as the unique bounded fixed point of the dynamic programming operator T and is continuous as a function of the information-state $x \in X$, and the value iteration algorithm converges geometrically to J^* for any initial continuous approximation of J^*.

For the case where termination is not inevitable, at least under some policies, i.e. under Assumptions A and C, the dynamic programming operator T is not a contraction with respect to any norm [2]. Our main results for this more general class of models include the following.

(i) J^* is again characterized as the unique bounded fixed point of T. (We have not established the continuity of J^* under Assumptions A and C.)

(ii) Value iteration converges pointwise to J^* for any initial bounded approximation of the optimal cost to go function.

(iii) The cost-to-go functions generated by policy iteration also converge pointwise to J^* for any initial proper policy.

Under both sets of assumptions, there exists an optimal stationary policy $\mu^* : X \mapsto U$, and the cost function J_μ for an arbitrary proper policy $\mu \in M$ may be discontinuous.

Our numerical results for the pursuit/evasion search model of Section 4 illustrate the application of our modeling framework, and also indicate the heavy computational requirements to obtain even approximate solutions to partially observed stochastic shortest path problems.

References

[1] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*. Academic Press, New York, 1978.

[2] D. P. Bertsekas and J. N. Tsitsiklis. Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena-Scientific, Belmont, MA, 1996.

[4] E. B. Dynkin and A. A. Yushkevich. *Controlled Markov Processes*. Springer-Verlag, New York, 1979.

[5] J. H. Eaton and L. A. Zadeh. Optimal Pursuit Strategies in Discrete State Probabilistic Systems. *Journal of Basic Engineering (formerly Transactions of the ASME, Series D)*, 84:23–29, 1962.

[6] E. A. Feinberg. A Markov Decision Model of a Search Process. *Contemporary Mathematics*, 125:87–96, 1992.

[7] O. Hernández-Lerma and J. B. Lasserre. Discrete-Time Markov Control Processes: Basic Optimality Criteria. Springer-Verlag, New York, 1996.

[8] O. Hernández-Lerma and J. B. Lasserre. Further Topics on Discrete-Time Markov Control Processes. Springer-Verlag, New York, 1999.

[9] S. D. Patek. *Stochastic Shortest Path Games: Theory and Algorithms*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 1997.

[10] S. D. Patek. On Terminating Markov Decision Processes with a Risk Averse Objective Function. *Automatica*, 37(9):1379–1386, 2001.

[11] S. D. Patek. On Partially Observed Stochastic Shortest Path Problems. *Proceedings of the* 40*-th IEEE Conference on Decision and Control (CDC 2001)*, pp. 5050–5055, 2001.

[12] S. D. Patek and D. P. Bertsekas. Stochastic Shortest Path Games. *SIAM Journal on Control and Optimization*, 37(3):804–824, 1999.

[13] L. K. Platzman. Optimal Infinite-Horizon Undiscounted Control of Finite Probabilistic Systems. *SIAM Journal on Control and Optimization*, 18(4):362–380, 1980.

[14] E. J. Sondijk. The Optimal Control of Partially Observable Process over the Infinite Horizon: Discounted Costs. *Operations Research*, 26(2):282–304, 1978.

[15] L. D. Stone. *Theory of Optimal Search*. Academic Press, New York, 1975.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.