Stochastic Shortest-Path Problems and Real-Time DP: New Theoretical Results

Blai Bonet

Cognitive Systems Laboratory, Dept. of Computer Science,
University of California, Los Angeles, CA 90024, USA,
bonet@cs.ucla.edu,
WWW home page: http://www.cs.ucla.edu/~bonet

Abstract. We consider the class of problems known as Stochastic Shortest-Path (SSP) problems. They are an important subclass of the Markov Decision Processes that are used to model a broad range of tasks including robot navigation and planning with uncertainty and partial information. An important learning algorithm for solving SSPs is Real-Time Dynamic Programming. In this paper, we show new theoretical results about the speed of convergence and the suboptimality of policies for Real-Time Dynamic Programming and Value Iteration.

1 Introduction

The class of Stochastic Shortest-Path (SSP) problems is a subset of Markov Decision Processes (MDPs) that is of central importance to AI: they are the *natural* generalization of the classic search model to the case of stochastic transitions and general cost functions. SSPs had been recently used to model a broad range of problems going from robot navigation and control of non-deterministic systems to stochastic game-playing and planning under uncertainty and partial information [3, 17, 4]. The theory of MDPs had received great attention from the AI community for three important reasons. First, it provides an easy framework for modeling complex real-life problems that have large state-space (even infinite) and complex dynamics and cost functions. Second, MDPs provide mathematical foundation for independently-developed learning algorithms in Reinforcement Learning. And third, general and efficient algorithms for solving MDPs had been developed, the most important being Value Iteration and Policy Iteration.

As the name suggests, an SSP problem is an MDP problem that has positive costs and an absorbing goal state. A solution for an SSP is a strategy that leads to the goal state with minimum expected cost from any other state. Quite often, however, we are only interested in how to get to the goal from a *fixed* initial state instead of knowing the general solution, the reason being that usually the state space contains many irrelevant states. Real-Time Dynamic Programming (RTDP) [1] is a *probabilistic learning* algorithm that finds such partial solutions. Unfortunately, RTDP converges only asymptotically with *unknown* speed and the quality of the resulting strategies is not known.

In this paper, we present formal results about the speed of convergence for the Value Iteration and RTDP algorithms over SSPs. We also give bounds for the suboptimality of the policies that result when these algorithms are stopped at certain moment. The results are obtained by considering some stochastic processes that are related to the SSP problem and algorithms. Thus, properties about the processes translate into properties about the problem and algorithms. The results are new and, at the time of writing, we are not aware of similar results within an scope as broad as the one given.

The paper is organized as follows. In Sect. 2, we review the formal definitions and some important results for MDPs and SSPs, and the Value Iteration and RTDP algorithms. In Sect. 3, we present the theoretical results about SSPs and the RTDP algorithm. The paper finishes with a discussion that includes a summary, related and future work. The proofs of all the results are in the Appendix.

2 Markov Decision Processes

This section contains a brief review of MDPs, SSPs, and the Value Iteration and RTDP algorithms. We use notation and presentation style close to that in [2]; the reader is referred there for an excellent exposition of the field.

The MDP model assumes the existence of a physical system that evolves in discrete time and that is controlled by an agent. The system dynamics is governed by probabilistic transition functions that maps states and controls to states. At every time, the agent incurs in a cost that depends in the current state of the system and the applied control. Thus, the task is to find a control strategy (also known as a policy) that minimize the expected total cost over the infinite horizon time setting. Formally, an MDP is defined by

(M1) A finite state space $S = \{1, \dots, n\}$,
(M2) a finite set of controls $U(i)$ for each state $i \in S$,
(M3) transition probabilities $p(i, u, j)$ for all $u \in U(i)$ that are equal to the probability of the next state being j after applying control u in state i, and
(M4) a cost $g(i, u)$ associated to $u \in U(i)$ and $i \in S$.

A strategy or policy π is an infinite sequence $(\mu_0, \mu_1, \dots)$ of functions where μ_k maps states to controls so that the agent applies the control $\mu_k(i)$ in state $x_k = i$ at time k, the only restriction being $\mu_k(i) \in U(i)$ for all $i \in S$. If $\pi = (\mu, \mu, \dots)$, the policy is called *stationary* (i.e., the control does not depend on time) and it is simply denoted by μ. The cost associated to policy π when the system starts at state x_0 is defined as

$$J_\pi(x_0) = \lim_{N \to \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)) \right\} \tag{1}$$

where the expectation is taken with respect to the probability distribution induced by the transition probabilities, and where the number $\alpha \in [0, 1]$, called the *discount factor*, is used to discount future costs at a geometric rate.

The MDP *problem* is to find an *optimal policy* π^* satisfying

$$J^*(i) \overset{\text{def}}{=} J_{\pi^*}(i) \leq J_\pi(i), \quad i = 1, \dots, n, \tag{2}$$

for every other policy π. Although there could be none or more than one optimal policy, the optimal cost vector J^* is always unique. The existence of π^* and how to compute it are non-trivial mathematical problems. However, when $\alpha < 1$ the optimal policy always exists and, more important, there exists a stationary policy that is optimal. In such case, J^* is the unique solution to the *Bellman Optimality* equations:

$$J^*(i) = \min_{u \in U(i)} g(i,u) + \alpha \sum_{j=1}^{n} p(i,u,j) J^*(j), \quad i = 1, \ldots, n. \tag{3}$$

Also, if J^* is a solution for (3) then the *greedy* stationary policy μ^* with respect to J^*:

$$\mu^*(i) = \operatorname*{argmin}_{u \in U(i)} \left\{ g(i,u) + \alpha \sum_{j=1}^{n} p(i,u,j) J^*(j) \right\} \tag{4}$$

is an optimal stationary policy for the MDP. Therefore, solving the MDP problem is equivalent to solving (3).

The equation (3) can be solved by considering the DP *operators*

$$(T_\mu J)(i) = g(i,\mu(i)) + \alpha \sum_{j=1}^{n} p(i,\mu(i),j) J(j), \tag{5}$$

$$(TJ)(i) = \min_{u \in U(i)} g(i,u) + \alpha \sum_{j=1}^{n} p(i,u,j) J(j) \tag{6}$$

that map n-dimensional vectors to n-dimensional vectors. Then, it is not hard to show that, when $\alpha < 1$, the operators T_μ and T are contraction mappings with unique fix points J_μ and J^* satisfying[1]

$$J_\mu = T_\mu J_\mu = \lim_{k \to \infty} T_\mu^k J_0, \tag{7}$$

$$J^* = TJ^* = \lim_{k \to \infty} T^k J_0 \tag{8}$$

where J_0 is the zero n-dimensional vector.

The Value Iteration algorithm computes J^* iteratively by using (3) as an update rule. Thus, starting from any vector J, Value Iteration computes a succession of vectors $\{J_k\}$ as

$$J_0 = J, \qquad J_{k+1} = TJ_k. \tag{9}$$

The algorithm stops when $J_{k+1} = J_k$, or when the residual $\max_{i \in S} |J_{k+1}(i) - J_k(i)|$ is sufficiently small. In the latter case, when $\alpha < 1$, the suboptimality of the resulting policy is bounded by a constant multiplied by the residual.

[1] An operator T is a contraction mapping if there is $0 \leq \alpha < 1$ such that $\|TJ - TJ'\| \leq \alpha \|J - J'\|$ for all J, J' in the domain of T.

2.1 Stochastic Shortest-Path Problems

A Stochastic Shortest-Path problem is an MDP problem in which the state space $S = \{1, \ldots, n-1, t\}$ is such that t is a goal (target) state that is absorbing (i.e., $p(t, u, t) = 1$ and $g(t, u) = 0$ for all $u \in U(t)$), and the discount factor $\alpha = 1$. In this case, the existence of optimal policies (and optimal stationary policies) is a major mathematical problem. However, the existence is guarantee under the following reasonable conditions:

(A1) There exists a stationary policy that achieves the goal with probability 1 from any start state.
(A2) All costs are positive; except $g(t, \cdot\,) \equiv 0$.

The first assumption just expresses the fact that the problem admits a well-behaved solution. Such policies are known as *proper* policies. The second assumption, in the other hand, guarantees that all improper policies incurs in infinite cost for at least one state. Thus, both assumptions preclude cases where the optimal solution might "wander" around without ever getting to the goal. For example, a problem having a zero-cost cycle (in state space) violates the second assumption. A useful characterization of proper policies is given in terms of the quantity:

$$\rho_\mu \overset{\text{def}}{=} \max_{i=1,\ldots,n-1} P(x_{n-1} \neq t | x_0 = i, \mu) \tag{10}$$

where $\{x_k\}_{k \geq 0}$ are random variables that stand for the state of the system at time k when it is started at x_0 and operated with policy μ. Hence, a stationary policy μ is proper if and only if $\rho_\mu < 1$. Under above assumptions, we have

Theorem 1 (Bertsekas [2]). *Suppose A1 and A2 hold.*[2] *Then, there exists an optimal policy μ^* that is stationary. Also, if J is a n-dimensional vector J such that $J(t) = 0$, then*

$$\lim_{k\to\infty} T^k_{\mu^*} J = J_{\mu^*} = \lim_{k\to\infty} T^k J = J^*. \tag{11}$$

That is, the Value Iteration algorithm solves the SSPs problems that satisfy assumptions A1 and A2.

As mentioned in the Introduction, often we are only interested in knowing how to go from a fixed initial state, say 1, to the goal state. The optimal solution in this case is an *partial* optimal stationary policy μ such that $\mu(i) = \mu^*(i)$ for all states i that are reachable from 1 when using μ^*, the so-called *relevant states* when starting from 1. Finding a partial optimal policy can be considerably simpler, the extreme case when the set of relevant states is finite and the complete state space is infinite. Thus, the question of how to find partial optimal policies is of great relevance. One algorithm for finding partial optimal policies is Real-Time Dynamic Programming.

[2] Yet the assumptions can be weakened while preserving the validity of the theorem; see the reference.

```
1. **Initialize** current state $x = 1$,
2. **Evaluate** each control $u \in U(x)$ as

   $$Q(x,u) = g(x,u) + \sum_{i=1}^{n} p(x,u,i)J(i)$$

   where $J(i) = H(i)$ (resp. $h(i)$) if $i \in H$ (resp. $i \notin H$).
3. **Choose** control $\mathbf{u}$ that minimizes $Q(x,u)$ breaking ties
   randomly (or in a systematic way),
4. **Update** $H(x)$ to $Q(x,\mathbf{u})$,
5. **Generate** next state $i$ with probability $p(x,\mathbf{u},i)$,
6. **Exit** if $i = t$, else set $x = i$ and **go to** 2.
```

Fig. 1. A trial of the RTDP algorithm.

2.2 Real-Time Dynamic Programming

The RTDP algorithm is the stochastic generalization of Korf's Learning Real-Time A* (LRTA*) for deterministic heuristic search [12]. RTDP is a learning algorithm that computes a partial optimal policy by performing successive walks (also called trials) on the state space. Each trial starts at the initial state 1 and finishes at the goal state t. At all times k, the RTDP algorithm maintains an approximation J_k to J^* that is used to *greedly* select a control u_k to apply in the current state x_k. Initially, J_0 is implicitly stored as an heuristic function $h(\cdot)$. Then, every time a control u_k is selected in state x_k, a new approximation J_{k+1} is computed by

$$J_0(i) = h(i), \tag{12}$$

$$J_{k+1}(i) = \begin{cases} J_k(i) & \text{if } i \neq x_k, \\ g(x_k,u_k) + \sum_{i=1}^{n} p(x_k,u_k,i)J_k(i) & \text{if } i = x_k. \end{cases} \tag{13}$$

Since J_k differs from J_0 at most in k states, J_k can be stored efficiently into a hash-table H. Initially, H is empty and the value $H(x)$ is given by the heuristic $h(x)$. Then, every time a control is selected an update of the form (13) is applied to H such that J_k can be computed from H and h. Figure 1 shows a description of an RTDP trial.

It is known that under assumptions A1 and A2, the RTDP trials eventually transverse minimum-cost paths from the initial state to the goal state if the heuristic function is *admissible*; i.e. if $0 \leq h(i) \leq J^*(i)$ for all $i \in S$ (see [1, 3]). Formally,

Theorem 2 (Barto, Bradtke and Singh [1]). *Suppose that assumptions A1 and A2 hold. Assume that an infinite number of* RTDP *trials is performed each one starting at 1 and that the hash-table is preserved between trials. If h satisfies* $0 \leq h \leq J^*$, *then with probability 1:*

(*i*) *the sequence of vectors* J_k *generated by* RTDP *converges to (some)* J_∞, *(ii)* $J_\infty(i) = J^*(i)$ *for all relevant states* i.

The J_∞ is a random vector that corresponds to the final hash-table denoted by H_∞. Both objects J_∞ and H_∞ are random variables in the formal sense, and the relation between them is that $J_\infty(i)$ is equal to $H_\infty(i)$ (resp. $h(i)$) if $i \in H_\infty$ (resp. if not). Part of the claim in the theorem is that each "sample path" of the algorithm corresponds to an *asynchronous* DP over a random SSP that consists of the states that appear infinitely often in the path. That is each sample path corresponds to an asynchronous DP over a random MDP.

Theorem 2 is the main known result about the RTDP algorithm. Some important open questions are: how fast is the convergence to J_∞? and how long is each trial? In the next section, we answer both questions for an important class of initial heuristic functions.

3 Theoretical Results

From now on, we denote the initial cost-vector for Value Iteration and the heuristic function for RTDP with the same symbol J. We will consider only initial vectors from the set $\mathcal{J}$ of n-dimensional vectors J that satisfy:

$$0 \leq J \leq J^* \quad \text{and} \quad J \leq TJ. \tag{14}$$

Clearly, the zero-constant vector and J^* satisfy (14), and $\mathcal{J}$ is stable under T; i.e., $T\mathcal{J} = \mathcal{J}$. In the case of deterministic search, the initial vector J is the heuristic function used to guide the search, and the conditions in (14) are equivalent to requiring an *admissible* and *monotonic* heuristic function. Admissibility is the property that the heuristic never overestimates the minimum cost to the goal. Monotonicity, on the other hand, refers to a kind of "triangle inequality" that the heuristic estimates must satisfy: for all states x and all of its children x', $h(x) \leq c(x, x') + h(x')$ where $c(x, x')$ is the cost to go from x to x'. It is not hard to check that monotonicity is a stronger condition than admissibility [16]. Admissibility and monotonicity had proved to be important conditions since standard algorithm like A* and IDA* guarantee optimality only if the heuristic function is admissible [16, 11, 10]. Additionally, if the heuristic is monotonic then A* is optimal [5].

In the stochastic case, it is also true that monotonicity implies admissibility. To see that, we use the monotonicity of the operator T as follows:[3] if $0 \leq J \leq J^*$, then

$$0 \leq J \leq TJ \leq T^2J \leq \cdots \leq T^kJ \;\to\; J^* \tag{15}$$

by Theorem 1. Hence, $\mathcal{J}$ can be defined as the set of non-negative vectors satisfying $J \leq TJ$.

To get our results, we need one extra assumption:

(A3) There are no self-loops in the state transition diagram except for t. That is, $p(i, u, i) = 0$ for all $i = 1, \ldots, n-1$ and $u \in U(i)$.

[3] An operator T is monotonic if $J \leq J'$ implies $TJ \leq TJ'$. That T is a monotonic follows easily from its definition.

Fortunately, this assumption is *not restrictive* in the sense that any SSP problem satisfying A1 and A2 can be transformed into an equivalent SSP satisfying also A3; see [2, p. 89].

3.1 Stochastic Shortest-Path Problems

Let us consider the following *stochastic process* that is useful for deriving facts about the Value Iteration and RTDP algorithms. Let $(\mathbf{J}, \mathbf{X}, \mathbf{U}) = (J_k, X_k, U_k)_{k \geq 0}$ be the process defined as

$$J_0 = J \in \mathcal{J}, \tag{16}$$
$$X_0 = 1, \tag{17}$$
$$U_0 = \text{argmin}\,\{Q(J_0, X_0, u) : u \in U(X_0)\}, \tag{18}$$

$$J_{k+1}(i) = [\![X_k = i]\!](T^{k+1}J)(i) + [\![X_k \neq i]\!]J_k(i), \tag{19}$$
$$X_{k+1} = i \in \{1, \ldots, n, t\} \text{ with probability } p(X_k, U_k, i), \tag{20}$$
$$U_{k+1} = \text{argmin}\,\big\{Q(T^{k+1}J, X_{k+1}, u) : u \in U(X_{k+1})\big\} \tag{21}$$

where
$$Q(J, i, u) = g(j, u) + \sum_{j=1}^{n} p(i, u, j)J(j), \tag{22}$$

$[\![\varphi]\!]$ is 1 (resp. 0) if φ is true (resp. false), and the ties for U_k are broken randomly (or in a systematic way). In words, the process begin at $X_0 = 1$, at time k chooses a control U_k (greedy with respect to T^kJ) and move to state X_{k+1} with probability $p(X_k, U_k, X_{k+1})$. Note the similarity of the $(\mathbf{J}, \mathbf{X}, \mathbf{U})$ with the RTDP algorithm (specially the evaluation of controls in (22) and step 2 in Figure 1).

Let $P_{x,J}(\,\cdot\,)$ be the probability distribution of $(\mathbf{J}, \mathbf{X}, \mathbf{U})$ when started at state $x \in S$ and initial vector $J \in \mathcal{J}$, and $E_{x,J}(\,\cdot\,)$ be the expectation taken with respect to such distribution. The following results bound the expected time to reach the goal state, show that the DP operator T is a pseudo-contraction mapping (see below), and establish bounds on the speed of convergence.

Theorem 3. *Suppose A1–A3 hold. Let τ be the (random) arrival time to state t; i.e., $\tau = \inf\{k > 0 : X_k = t\}$. Then,*

$$(k+1)P_{x,J}\big(X_k \neq t\big) \;\leq\; E_{x,J}(\tau) \;\leq\; \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i) - J(i) + J^*(X_0)\right) \tag{23}$$

where $\underline{g}$ is the minimum positive cost.

Corollary 1. *Suppose A1–A3 hold. Then, there exist $K > 0$ and $0 \leq \alpha < 1$, that depend on x and J, such that $P_{x,J}\big(X_K \neq t\big) \leq \alpha$. In addition, if $J' \in \mathcal{J}$ is such that $J' \geq J$, then $P_{x,J'}\big(X_K \neq t\big) \leq \alpha$.*

Theorem 4. *Suppose A1–A3 hold. Then, there exist constants $K > 0$ and $0 \leq \alpha < 1$ such that $\|T^KJ - T^KJ'\| \leq \alpha\|J - J'\|$ for all $J, J' \in \mathcal{J}$ such that $J \leq J'$.*

In particular, $\|J^* - T^K J\| \leq \alpha \|J^* - J\|$ *for all* $J \in \mathcal{J}$*; i.e.,* T^K *is a* pseudo-contraction mapping.[4]

Corollary 2. *Suppose A1–A3 hold. For all* $J \in \mathcal{J}$*, and integers* $m, k \geq 0$

$$\left\|J^* - T^{K(m+k)}J\right\| \ \leq \ \frac{\alpha^k}{1-\alpha}\left\|T^{K(m+1)}J - T^{Km}J\right\| \tag{24}$$

where K *and* α *are given by Theorem 4.*

Therefore, the Value Iteration algorithm converges exponentially fast to J^* when started from any vector $J \in \mathcal{J}$; in particular, the zero vector. Also, the loss incurred in the Km iteration can be bounded in terms of the residual in previous iterations. Unfortunately, Theorem 4 does not say how to compute the constants K and α and, more important, K can be quite large.

3.2 Real-Time Dynamic Programming

For studying the RTDP algorithm, let us consider a modification of the previous process, denoted by $(\tilde{\mathbf{J}}, \tilde{\mathbf{X}}, \tilde{\mathbf{U}}) = (\tilde{J}_k, \tilde{X}_k, \tilde{U}_k)_{k\geq 0}$, that corresponds to the RTDP algorithm. The new process differs from $(\mathbf{J}, \mathbf{X}, \mathbf{U})$ in that the updates for $\tilde{J}_{k+1}$ and the selection of U_k are done as in RTDP; i.e.,

$$\tilde{J}_{k+1}(i) = [\![\tilde{X}_k = i]\!](T\tilde{J}_k)(i) + [\![\tilde{X}_k \neq i]\!]\tilde{J}_k(i), \tag{25}$$

$$\tilde{U}_{k+1} = \operatorname{argmin}\left\{Q(\tilde{J}_{k+1}, \tilde{X}_{k+1}, u) : u \in U(\tilde{X}_{k+1})\right\}. \tag{26}$$

As before, let $\tilde{P}_{x,J}(\,\cdot\,)$ and $\tilde{E}_{x,J}(\,\cdot\,)$ be the distribution and expectation associated to $(\tilde{\mathbf{J}}, \tilde{\mathbf{X}}, \tilde{\mathbf{U}})$ when started from $x \in S$ and $J \in \mathcal{J}$. A small modification in the proof of Theorem 3 shows

Theorem 5. *Suppose A1–A3 hold. Let* $\tau = \inf\{k > 0 : \tilde{X}_k = t\}$*. Then,*

$$(k+1)\tilde{P}_{x,J}\big(\tilde{X}_k \neq t\big) \ \leq \ \tilde{E}_{x,J}(\tau) \ \leq \ \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i) - J(i) + J^*(\tilde{X}_0)\right). \tag{27}$$

Therefore, the expected termination time of each RTDP trial is finite and bounded by the right-hand side of (27). By Theorem 2, we know that $\tilde{J}_k$ converges to J^* over the relevant states with probability 1. To estimate the rate of convergence, define the random variable R as the set of states that are visited infinitely often by RTDP; i.e.,

$$R = \{i \in S : i \text{ is infinitely often in } (\tilde{X}_0, \tilde{X}_1, \tilde{X}_2, \ldots)\}. \tag{28}$$

Since RTDP performs asynchronous DP over the set R we will consider the random times in which a full DP had been performed over all states in R. Thus,

[4] A map $T : \mathcal{S} \to \mathbb{R}$ is a pseudo-contraction mapping over $\mathcal{S}$ if there is a constant $0 \leq \alpha < 1$ and $S^* \in \mathcal{S}$ so that $\|TS^* - TS\| \leq \alpha\|S^* - S\|$ for all $S \in \mathcal{S}$.

let $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$ be the sequence of random times such that all states in R are visited between τ_k and τ_{k+1}; i.e.,

$$\tau_{k+1} = \inf\{j > \tau_k : \{\tilde{X}_{\tau_k+1}, \tilde{X}_{\tau_k+2}, \ldots, \tilde{X}_{\tau_k+j}\} \supseteq R\}. \tag{29}$$

By the definition of R, the times $\tau_k < \infty$ with probability 1. Then,

Theorem 6. *Suppose A1–A3 hold. Let K and α be the constants given by Theorem 4. Then, for all integers $m \geq 0$*

$$\big\|J^* - \tilde{J}_{\tau_{Km}}\big\|_R \;\leq\; \alpha^m \big\|J^* - J\big\|_R, \tag{30}$$

$$\big\|J^* - \tilde{J}_{\tau_{Km}}\big\|_R \;\leq\; \frac{1}{1-\alpha}\big\|\tilde{J}_{\tau_{K(m+1)}} - \tilde{J}_{\tau_{Km}}\big\|_R. \tag{31}$$

where $\|J\|_R$ refers to the sup norm over set R; i.e., $\|J\|_R = \sup_{i\in R}|J(i)|$.

Thus, the RTDP algorithm could converge exponentially fast (in expectation) provided a uniform bound on $\{\tilde{E}_{x,J}(\tau_k) : k \geq 0\}$ exists. We had been unable to obtain such result. Nonetheless, Theorem 6 suggests that interleaving full DP updates (over the discovered state space) with RTDP might speedup the convergence of the algorithm. Since if τ'_k are the times for the full DP updates, then $\tau_k \leq \tau'_k$ (for large k) so the waiting times τ_k can be controlled.

3.3 Suboptimality of Resulting Policies

To bound the loss incurred by the policy that results when stopping the Value Iteration and RTDP algorithms, let us define the more general concept of proper and greedy policies.

Definition 1 (Proper policies of order k). *A stationary policy μ is a* proper policy of order k *if and only if*

$$\rho_\mu^{(k)} \overset{\text{def}}{=} \max_{i=1,\ldots,n} P(x_k \neq t | x_0 = i, \mu) < 1. \tag{32}$$

Definition 2 (Greedy policies of order k). *A stationary policy μ is a* greedy policy of order k *with respect to vector J if and only if $T_\mu T^{j-1} J = T^j J$ for $1 \leq j \leq k$.*

It is not hard to see that a proper policy of order k is a proper policy, and that a greedy policy of order k is also a greedy policy. We have

Theorem 7. *If μ is a proper policy of order k, then $\|T_\mu^k J - T_\mu^k J'\| \leq \rho_\mu^{(k)}\|J - J'\|$ for all n-dimensional vectors J, J'; i.e., T_μ^k is a contraction mapping.*

Then, the suboptimality of proper and greedy policies is given by

Theorem 8. *Let J be any vector in $\mathcal{J}$ and μ a proper and greedy policy of order k with respect to J. Then,*

$$\|J^* - J\| \;\leq\; \frac{1}{1-\rho_\mu^{(k)}}\|T^k J - J\|, \tag{33}$$

$$\|J^* - J_\mu\| \;\leq\; \rho_\mu^{(k)}\|J_\mu - J\| \;\leq\; \|J_\mu - J\| \tag{34}$$

Note that the inequality $\|J^* - J_\mu\| \leq \|J_\mu - J\|$ always holds since $J \leq J^* \leq J_\mu$.

We want to stress that these are new results for the speed of convergence and suboptimality of policies for the Value Iteration and RTDP algorithm. Also, we are not aware of similar results within an scope as large as the one given by assumptions A1–A3: previous results assume stronger conditions. For example, a result about speed of convergence can be obtained under the assumption that all stationary policies are proper, and a result similar to Theorem 8 can be given under stronger conditions and in terms of *unobservable* quantities; see [2].

4 Discussion

Stochastic Shortest-Path models are of central importance in AI since they can be used to model a broad range of problems that go from standard planning problems like robot navigation in stochastic environments and perfect information games to more complex problems involving uncertainty and partial information; the latter are modeled using SSP in belief-space [8, 7, 4].

The SSP model generalizes the important heuristic search model in AI to the case of stochastic transitions and general cost functions. It is interesting to note that although the AI community spent tremendous effort in heuristic search and yielded results as important as the A* and IDA* algorithms, the general interest didn't pass to the stochastic version. Thus, today's most important theoretical results for SSPs come from the field of operations research and control theory (see [2] and references therein). However, the standard algorithms from operations research compute full optimal policies instead of partial policies. The difference between full and partial policies corresponds, for example, to the difference between the solutions to deterministic shortest-path problems given by the Dijkstra or Bellman-Ford algorithms and the one given by A* and IDA*: the former compute shortest paths from all states to the goal while the latter only one shortest path from the initial state to the goal. This difference is the fundamental reason for the success of heuristic search algorithms in problems with very large state spaces as, for example, the 24-puzzle and Rubik's cube [14, 13]. Therefore, we think that RTDP could be used to solve SSPs with very-large state spaces that currently cannot be solved by any other method; e.g., the Backgammon or Tetris [18, 19].

Another interesting algorithm that computes partial policies is the LAO* algorithm in [6]. LAO* is an off-line algorithm that generalizes the standard AO* algorithm for AND/OR graphs [15] to the case in which the AND nodes represent stochastic transitions and the graph has cycles.

In this paper, we have shown new bounds for the speed of convergence and the suboptimality of resulting policies for the Value Iteration and RTDP algorithms. Some of the main results, Theorems 3 and 5, were obtained by considering two stochastic processes associated with the algorithms. The idea for these processes came when reading a deterministic proof related to the min-max version of LRTA* [9]. Although, the claims and proofs in the stochastic case are more complex than the deterministic case, the methodology of generalizing existing definitions and proofs in the deterministic case had proved fruitful. Hence, we believe that new methods and results for SSPs can be obtained by carefully analyzing the extensive work that had been done for deterministic SSPs.

In the future, we want to study the application of RTDP to problems with partial information, to obtain a better understanding of the role of the heuristic function in RTDP, and to compare RTDP with other algorithms like LAO*.

References

1. A. Barto, S. Bradtke, and S. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138, 1995.
2. D. Bertsekas. *Dynamic Programming and Optimal Control, (2 Vols).* Athena Scientific, 1995.
3. D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.
4. B. Bonet and H. Geffner. Planning with incomplete information as heuristic search in belief space. In *Proceedings of AIPS-2000*, pages 52–61. AAAI Press, 2000.
5. R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, 32(3):505–536, 1985.
6. E. Hansen and S. Zilberstein. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129:35–62, 2001.
7. M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.
8. L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1999.
9. S. Koenig. Minimax real-time heuristic search. *Artificial Intelligence*, 129:165–197, 2001.
10. R. Korf. Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109, 1985.
11. R. Korf. Iterative-depeening A*: An optimal admissible tree search. In *Proceedings of IJCAI-85*, pages 1034–1036, Los Angeles, California, 1985. Morgan Kaufmann.
12. R. Korf. Real-time heuristic search. *Artificial Intelligence*, 42:189–211, 1990.
13. R. Korf. Finding optimal solutions to rubik's cube using patterns databases. In *Proceedings of AAAI-97*, pages 700–705, Providence, RI, 1997. MIT Press.
14. R. Korf and L. Taylor. Finding optimal solutions to the twenty-four puzzle. In *Proceedings of AAAI-96*, pages 1202–1207, Protland, Oregon, 1996. MIT Press.
15. N. Nilsson. *Principles of Artificial Intelligence.* Tioga, 1980.
16. J. Pearl. *Heuristics.* Morgan Kaufmann, 1983.
17. R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.
18. G. Tesauron. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38:58–68, 1995.
19. J. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22:59–94, 1996.

Appendix: Proofs

Lemma 1. *For all $J \in \mathcal{J}$ and $k \geq 0$, $T^k J \geq J_k$.*

Proof: It not hard to see that the operator T is monotonic in the sense that if $J \leq J'$ then $TJ \leq TJ'$ (see [2]). So, the result follows using induction, the monotonicity of T and the definition of J_{k+1}. □

Theorem 3. *Suppose A1–A3 hold. Let $\tau = \inf\{k > 0 : X_k = t\}$. Then,*

$$(k+1)P_{x,J}\big(X_k \neq t\big) \;\leq\; E_{x,J}(\tau) \;\leq\; \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i) - J(i) + J^*(X_0)\right)$$

where $\underline{g}$ is the minimum positive cost.

Proof: Since $P(T = 0) = 0$, the first inequality follows from the monotonicity of the events $\{X_k \neq t\} \supseteq \{X_{k+1} \neq t\}$ by

$$(k+1)P_{x,J}\big(X_k \neq t\big) \;\leq\; \sum_{j=0}^{k} P_{x,J}\big(X_j \neq t\big) \;=\; \sum_{j=0}^{k+1} P_{x,J}(\tau \geq j) \;\to\; E_{x,J}(\tau)$$

as k goes to ∞. For the second inequality, define the random sequence $(W_k)_{k\geq 0}$ as

$$W_0 = 0,$$
$$W_{k+1} = \sum_{j=1}^{n} J_{k+1}(j) - J(j) - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) + J(X_0).$$

Then,

$$\begin{aligned}
W_{k+1} &= \sum_{j=1}^{n} J_{k+1}(j) - J(j) - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) + J(X_0) \\
&= \sum_{j\neq X_k} J_{k+1}(j) - J(j) + [\![X_k \neq t]\!]\big(J_{k+1}(X_k) - J(X_k)\big) \\
&\qquad\qquad - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) + J(X_0) \\
&= \sum_{j\neq X_k} J_k(j) - J(j) + [\![X_k \neq t]\!]\big(J_{k+1}(X_k) - J(X_k)\big) \\
&\qquad\qquad - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) + J(X_0) \\
&= \sum_{j=1}^{n} J_k(j) - J(j) + [\![X_k \neq t]\!]\big(J_{k+1}(X_k) - J_k(X_k)\big) \\
&\qquad\qquad - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) + J(X_0) \\
&= \sum_{j=1}^{n} J_k(j) - J(j) - [\![X_k \neq t]\!]J_{k-1}(X_k) + J(X_0) + [\![X_k \neq t]\!]J_{k+1}(X_k) \\
&\qquad\qquad - [\![X_{k+1} \neq t]\!]J_k(X_{k+1}) \\
&= W_k + [\![X_k \neq t]\!]J_{k+1}(X_k) - [\![X_{k+1} \neq t]\!]J_k(X_{k+1})
\end{aligned}$$

using $p(X_{k-1}, \cdot, X_k) = 0$ in the previous to last equality. The reader may want to check the validity of above identity for $k = 0$; i.e., $W_1 = [\![X_0 \neq t]\!]J_1(X_0) - [\![X_1 \neq t]\!]J_0(X_1)$. Define $\mathcal{F}_k = \{X_0, U_0, \ldots, X_k, U_k\}$ as the "random history" of the process until time k. Taking expectations with respect to $\mathcal{F}_k$

$$\begin{aligned}
&E_{x,J}\big(W_{k+1}\big|\mathcal{F}_k\big) \\
&\quad = W_k + [\![X_k \neq t]\!]E_{x,J}\big(J_{k+1}(X_k)\big|\mathcal{F}_k\big) - E_{x,J}\big([\![X_{k+1} \neq t]\!]J_k(X_{k+1})\big|\mathcal{F}_k\big)
\end{aligned}$$

$$
\begin{aligned}
&= W_k + [\![X_k \neq t]\!] E_{x,J}\big(J_{k+1}(X_k)\big|\mathcal{F}_k\big) - \sum_{j=1}^{n} p(X_k, U_k, j) J_k(j) \\
&= W_k + [\![X_k \neq t]\!] g(X_k, U_k) + [\![X_k \neq t]\!] \sum_{j=1}^{n} p(X_k, U_k, j)(T^k J)(j) \\
&\qquad\qquad\qquad\qquad\qquad\qquad - \sum_{j=1}^{n} p(X_k, U_k, j) J_k(j) \\
&\geq W_k + [\![X_k \neq t]\!] g(X_k, U_k) - \big(1 - [\![X_k \neq t]\!]\big) \sum_{j=1}^{n} p(X_k, U_k, j) J_k(j) \\
&= W_k + [\![X_k \neq t]\!] g(X_k, U_k) - [\![X_k = t]\!] \sum_{j=1}^{n} p(X_k, U_k, j) J_k(j) \\
&= W_k + [\![X_k \neq t]\!] g(X_k, U_k)
\end{aligned}
$$

where the inequality follows from Lemma 1. The reader with background in probability theory will recognize that W_k is a non-negative submartingale. Integrating both sides we obtain

$$
E_{x,J}\big(W_{k+1}\big) \;\geq\; \underline{g} \sum_{j=0}^{k} E_{x,J}\big([\![X_j \neq t]\!]\big) \;=\; \underline{g} \sum_{j=0}^{k+1} P_{x,J}(\tau \geq j).
$$

Lemma 1 implies that $J_k \leq J^*$ for all $k \geq 0$. Therefore,

$$
\sum_{j=1}^{n} J^*(j) - J(j) + J^*(X_0) \;\geq\; E_{x,J}\big(W_k\big) \;\geq\; \underline{g} E_{x,J}(\tau)
$$

for all k. The second inequality follows by taking the limit as $k \to \infty$ and using $E_{x,J}(\tau) = \sum_{j \geq 0} P_{x,J}(\tau > j)$ (since $P(\tau = 0) = 0$). □

Corollary 1. *Suppose A1–A3 hold. Then, there exist $K > 0$ and $0 \leq \alpha < 1$, that depend on x and J, such that $P_{x,J}\big(X_K \neq t\big) \leq \alpha$. In addition, if $J' \in \mathcal{J}$ is such that $J' \geq J$, then $P_{x,J'}\big(X_K \neq t\big) \leq \alpha$.*

Proof: For the first claim, choose integer K large enough such that the right-hand side in Theorem 3 over $K + 1$ is less than 1. For the second, note that the right-hand in the first claim decreases as J increases. □

Theorem 4. *Suppose A1–A3 hold. Then, there exist constants $K > 0$ and $0 \leq \alpha < 1$ such that $\|T^K J - T^K J'\| \leq \alpha \|J - J'\|$ for all $J, J' \in \mathcal{J}$ such that $J \leq J'$. In particular, $\|J^* - T^K J\| \leq \alpha \|J^* - J\|$ for all $J \in \mathcal{J}$; i.e., T^K is a* pseudo-contraction mapping *in $\mathcal{J}$.*

Proof: Let $x_0 \in \{1, \ldots, n\}$. Then,

$$
\begin{aligned}
&|(T^k J')(x_0) - (T^k J)(x_0)| \\
&\qquad = \left| \left(\min_{u \in U(x_0)} g(x_0, u) + \sum_{x_1=1}^{n} p(x_0, u, x_1)(T^{k-1} J')(x_1) \right) - \right. \\
&\qquad\qquad\qquad \left. \left(\min_{u \in U(x_0)} g(x_0, u) + \sum_{x_1=1}^{n} p(x_0, u, x_1)(T^{k-1} J)(x_1) \right) \right|
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{x_1=1}^{n} p(x_0,u_0,x_1)\left|(T^{k-1}J')(x_1)-(T^{k-1}J)(x_1)\right| \\
&\leq \sum_{x_1=1}^{n} p(x_0,u_0,x_1)\left|\sum_{x_2=1}^{n} p(x_1,u_1,x_2)\big((T^{k-2}J')(x_2)-(T^{k-2}J)(x_2)\big)\right| \\
&\leq \sum_{x_1=1}^{n}\sum_{x_2=1}^{n} p(x_0,u_0,x_1)p(x_1,u_1,x_2)\left|(T^{k-2}J')(x_2)-(T^{k-2}J)(x_2)\right| \\
&\leq \sum_{x_1=1}^{n}\cdots\sum_{x_k=1}^{n} p(x_0,u_0,x_1)\cdots p(x_{k-1},u_{k-1},x_k)\big|J'(x_k)-J(x_k)\big| \\
&\leq \|J'-J\|\,P_{x_0,J}(X_k\neq t)
\end{aligned}
$$

where $u_k \in U(x_k)$ are the *greedy* controls with respect to T^kJ. Now, choose integer $K>0$ large enough such that

$$
\alpha \overset{\text{def}}{=} \frac{\underline{g}^{-1}}{K+1}\left(\sum_{j=1}^{n} J^*(j)+\max_{j=1,\ldots,n} J^*(j)\right) < 1.
$$

Then,

$$
\big|(T^KJ')(x_0)-(T^KJ)(x_0)\big| \leq \|J'-J\|\,P_{x_0,J}\big(X_K\neq t\big) \leq \alpha\|J'-J\|
$$

where the last inequality follows from Corollary 1 and the choice of α. □

Corollary 2. *Suppose A1–A3 hold. For all $J\in\mathcal{J}$, and integers $m,k\geq 0$*

$$
\big\|J^*-T^{K(m+k)}J\big\| \leq \frac{\alpha^k}{1-\alpha}\big\|T^{K(m+1)}J-T^{Km}J\big\|
$$

where K and α are given by Theorem 4.

Proof:

$$
\begin{aligned}
\big\|J^*-T^{K(m+k)}J\big\| &\leq \big\|J^*-T^{K(m+k+1)}J\big\|+\big\|T^{K(m+k+1)}J-T^{K(m+k)}J\big\| \\
&\leq \alpha\big\|J^*-T^{K(m+k)}J\big\|+\alpha^k\big\|T^{K(m+1)}J-T^{Km}J\big\|.
\end{aligned}
$$

□

Theorem 5. *Suppose A1–A3 hold. Let $\tau=\inf\{k>0:\tilde{X}_k=t\}$. Then,*

$$
(k+1)\tilde{P}_{x,J}\big(\tilde{X}_k\neq t\big) \leq \tilde{E}_{x,J}(\tau) \leq \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i)-J(i)+J^*(\tilde{X}_0)\right).
$$

Proof: The same proof as Theorem 3. The only thing that changes is that $E_{x,J}\big(W_{k+1}\big|\mathcal{F}_k\big)=W_k+[\![\tilde{X}_k\neq t]\!]g(\tilde{X}_k,\tilde{U}_k)$ instead of $\geq$. □

Theorem 6. *Suppose A1–A3 hold. Let K and α be the constants given by Theorem 4. Then, for all integers $m\geq 0$*

$$
\begin{aligned}
\big\|J^*-\tilde{J}_{\tau_{Km}}\big\|_R &\leq \alpha^m\big\|J^*-J\big\|_R, \\
\big\|J^*-\tilde{J}_{\tau_{Km}}\big\|_R &\leq \frac{1}{1-\alpha}\big\|\tilde{J}_{\tau_{K(m+1)}}-\tilde{J}_{\tau_{Km}}\big\|_R.
\end{aligned}
$$

where $\|J\|_R$ refers to the sup norm over set R; i.e., $\|J\|_R=\sup_{i\in R}|J(i)|$.

Proof: Note that $(T^k J)(x) \le \tilde{J}_{\tau_k}(x) \le J^*(x)$ for all $x \in R$. Therefore,

$$\begin{aligned}
\big\|J^* - \tilde{J}_{\tau_{Km}}\big\|_R &\le \big\|J^* - T^{Km}J\big\|_R \le \alpha^m \big\|J^* - J\big\|_R, \\
\big\|J^* - \tilde{J}_{\tau_{Km}}\big\|_R &\le \big\|J^* - \tilde{J}_{\tau_{K(m+1)}}\big\|_R + \big\|\tilde{J}_{\tau_{K(m+1)}} - \tilde{J}_{\tau_{Km}}\big\|_R \\
&\le \big\|J^* - T^K \tilde{J}_{\tau_{Km}}\big\|_R + \big\|\tilde{J}_{\tau_{K(m+1)}} - \tilde{J}_{\tau_{Km}}\big\|_R \\
&\le \alpha\big\|J^* - \tilde{J}_{\tau_{Km}}\big\|_R + \big\|\tilde{J}_{\tau_{K(m+1)}} - \tilde{J}_{\tau_{Km}}\big\|_R.
\end{aligned}$$

□

Theorem 7. *If μ is a proper policy of order k, then $\|T_\mu^k J - T_\mu^k J'\| \le \rho_\mu^{(k)} \|J - J'\|$ for all n-dimensional vectors J, J'; i.e., T_μ^k is a contraction mapping.*

Proof: Let J and J' be two n-dimensional vectors and x_0 any state. Then,

$$\begin{aligned}
&\big|T_\mu^k J(x_0) - T_\mu^k J'(x_0)\big| \\
&\quad\le \sum_{x_1=1}^{n} p(x_0,u_0,x_1) \left|T_\mu^{k-1} J(x_1) - T_\mu^{k-1} J'(x_1)\right| \\
&\quad\le \sum_{x_1=1}^{n}\sum_{x_2=1}^{n} p(x_0,u_0,x_1)\, p(x_1,u_1,x_2) \left|T_\mu^{k-2} J(x_2) - T_\mu^{k-2} J'(x_2)\right| \\
&\quad\le \sum_{x_1=1}^{n}\cdots\sum_{x_k=1}^{n} p(x_0,u_0,x_1)\cdots p(x_{k-1},u_{k-1},x_k) \left|J(x_k) - J'(x_k)\right| \\
&\quad\le \|J - J'\| \sum_{x_1=1}^{n}\cdots\sum_{x_k=1}^{n} p(x_0,u_0,x_1)\cdots p(x_{k-1},u_{k-1},x_k) \\
&\quad\le \rho_\mu^{(k)} \|J - J'\|
\end{aligned}$$

where $u_k = \mu(x_k)$ for $k = 0, \dots, k-1$. □

Theorem 8. *Let J be any vector in $\mathcal{J}$ and μ a proper and greedy policy of order k with respect to J. Then,*

$$\begin{aligned}
\|J^* - J\| &\le \frac{1}{1 - \rho_\mu^{(k)}} \|T^k J - J\|, \\
\|J^* - J_\mu\| &\le \|J_\mu - J\| \le \rho_\mu^{(k)} \|J_\mu - J\|.
\end{aligned}$$

Proof: First note,

$$\|J^* - T^k J\| \le \|T_\mu^k J^* - T^k J\| = \|T_\mu^k J^* - T_\mu^k J\| \le \rho_\mu^{(k)} \|J^* - J\|$$

by definition of T_μ and T, the fact $J \le J^*$ and Theorem 7. Since $J^* \le J_\mu$, we have

$$\begin{aligned}
\|J^* - J\| &\le \|J^* - T^k J\| + \|T^k J - J\| \le \rho_\mu^{(k)} \|J^* - J\| + \|T^k J - J\|, \\
\|J^* - J_\mu\| &\le \|J_\mu - T^k J\| = \|J_\mu - T_\mu^k J\| \le \rho_\mu^{(k)} \|J_\mu - J\|.
\end{aligned}$$

□