New Convergence Results for Stochastic Shortest-Path Problems and Real-Time Dynamic Programming

Blai Bonet BONET@CS.UCLA.EDU
Cognitive Systems Laboratory, Department of Computer Science,
University of California, Los Angeles, CA 90024 USA

Abstract

We consider the class of problems known as Stochastic Shortest-Path (SSP) problems. They are an important subclass of Markov Decision Processes that had been used to model a broad range of tasks including robot navigation and localization, game playing and planning under uncertainty and partial information. Two important algorithms for solving SSPs are Value Iteration and Real-Time Dynamic Programming. In this paper, we show new results about the speed of convergence for these algorithms and the suboptimality of the resulting policies when they are stopped before convergence.

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 localization to control of non-deterministic systems, game-playing and planning under uncertainty and partial information (Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998; Bonet & Geffner, 2000). 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 and complex dynamics and cost functions. Second, MDPs provide mathematical foundations for different learning algorithms. And third, general and efficient algorithms for solving MDPs had been developed, the most important being Value Iteration and Policy Iteration.

Plainly, 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. However, quite often we are only interested in how to get to the goal state from a *fixed* initial state, the reason being that usually the complete state space contains many irrelevant states. A *partial* solution is a strategy that leads to the goal with minimum expected cost when it is applied from the initial state. The Real-Time Dynamic Programming algorithm (RTDP) is an algorithm that finds such partial solutions (Barto, Bradtke, & Singh, 1995). However, RTDP only converges *asymptotically* with unknown speed and, if stopped at certain moment, the quality of the resulting strategies is also unknown.

In this paper, we show formal results about the speed of convergence for the Value Iteration and RTDP algorithms over SSPs for a class of initial iterates (or heuristic functions). 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 an stochastic process that is closely related to the SSP problem and algorithms. Thus, properties about the process 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 by us.

The paper is organized as follows. In the next Section we give standard definitions for MDPs and SSPs and review some important results for the Value Iteration and RTDP algorithms. In Section 3, we present the new theoretical results for SSPs and the RTDP algorithm to the case of admissible and monotonic heuristic functions. In Section 4, we show how to obtain such heuristic functions. The paper finishes with a discussion that includes a summary, related and future work.

2. Preliminaries

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 (Bertsekas, 1995); 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 decision functions where μ_k maps states to controls so that the agent applies 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[1]

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

where the x_k's are random variables for the state of the system at time k that are distributed according to the measure induced by the transition probabilities and the policy π. The number $\alpha \in [0, 1]$, called *discount factor*, is used to discount future costs at a geometric rate.

[1] When the limit is not known to exist it is replaced by the inferior or superior limit.

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

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

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 that case, J^* is the unique solution to the *Bellman Optimality* equations:

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

Also, if J^* is a solution for (1) then the *greedy* stationary policy μ^* with respect to J^*:

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

is an optimal policy for the MDP. Therefore, solving the MDP problem is equivalent to solving (1). Such equation can be solved by considering the DP *operators*:

$$\begin{aligned} (T_\mu J)(i) &\overset{\text{def}}{=} g(i,\mu(i)) + \alpha \sum_{j=1}^{n} p(i,\mu(i),j) J(j), \\ (TJ)(i) &\overset{\text{def}}{=} \min_{u \in U(i)} \left\{ g(i,u) + \alpha \sum_{j=1}^{n} p(i,u,j) J(j) \right\} \end{aligned}$$

that map n-dimensional vectors to n-dimensional vectors. When $\alpha < 1$ is not hard to show that such operators are *contraction mappings* with unique fix points J_μ and J^* satisfying:[2]

$$\begin{aligned} J_\mu &= T_\mu J_\mu = \lim_{k \to \infty} T_\mu^k J, \\ J^* &= TJ^* = \lim_{k \to \infty} T^k J \end{aligned}$$

where J is any n-dimensional vector.

The Value Iteration algorithm computes J^* iteratively by using (1) as an update rule. Thus, starting from any vector J, the algorithm computes a succession of vectors $\{J_k\}_{k \ge 0}$ as

$$\begin{aligned} J_0 &\overset{\text{def}}{=} J, \\ J_{k+1} &\overset{\text{def}}{=} TJ_k. \end{aligned}$$

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.

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

2.1 Stochastic Shortest-Path Problems

A Stochastic Shortest-Path problem is an MDP in which the state space $S = \{1, \ldots, n, t\}$ 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 guaranteed 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 incur 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

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

where the x_k's 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

Theorem 1 (Bertsekas, 1995) *Suppose A1 and A2 hold.*[3] *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*

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

That is, the Value Iteration algorithm solves 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 (e.g. 1) to the goal state. The optimal solution in this case is an *partial* optimal 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* 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. Two recent algorithms for finding partial optimal policies are Real-Time Dynamic Programming and LAO* (Barto et al., 1995; Hansen & Zilberstein, 2001). In this paper we focus in RTDP.

[3] The assumptions can be weakened while preserving the validity of the theorem (Bertsekas, 1995).

Algorithm 1: A trial of the RTDP algorithm.

```
begin
    $i \leftarrow 1$   (set initial state);
    while $i$ is not a goal state do
        Evaluate each control $u \in U(i)$ as
            $$Q(i,u) \;=\; g(i,u) + \sum_{j=1}^{n} p(i,u,j)J(j)$$
        where $J(j) = H(j)$ (resp. $= h(j)$) if $j \in H$ (resp. $j \notin H$);
        Choose control $u^*$ that minimizes $Q(i,u)$ breaking ties randomly (or in a systematic way);
        Update hash-table $H(i) \leftarrow Q(i,u^*)$;
        Generate next state $j$ with probability $p(i,u^*,j)$;
        Set $i \leftarrow j$;
    end
end
```

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 (Korf, 1990). RTDP is a randomized learning algorithm that computes a partial optimal policy by performing successive walks, also called trials, over the state space. Each trial starts at the initial state $x_0 = 1$ 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

$$J_0(i) \;\overset{\text{def}}{=}\; h(i), \tag{2}$$

$$J_{k+1}(i) \;\overset{\text{def}}{=}\; \begin{cases} J_k(i) & \text{if } i \neq x_k, \\ g(x_k,u_k) + \sum_{j=1}^{n} p(x_k,u_k,j)J_k(j) & \text{if } i = x_k. \end{cases} \tag{3}$$

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 $H(i)$ is given by the heuristic $h(i)$. Thereafter, every time a control u_k is selected an update of the form of (3) is applied to H such that J_k can be computed from H and h. The Algorithm 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 $i \in S$ (Barto et al., 1995; Bertsekas & Tsitsiklis, 1996). Formally,

Theorem 2 (Barto et al., 1995) *Suppose that assumptions A1 and A2 hold. Assume that an infinite number of* RTDP *trials are 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_∞ vector in the theorem 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 problem 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, yet other important questions are still open like: how fast is the convergence to the final J_∞?, is it an exponential or subexponential convergence?, how long is each RTDP trial?, what is the quality of the resulting policies when RTDP is terminated before convergence?, etc. Below, we answer these questions for an important class of initial heuristic functions for the Value Iteration and the RTDP algorithms.

3. Convergence Results

From now on, we will denote the initial cost-vector of Value Iteration and the heuristic function for RTDP with the same symbol J. We will not make any further difference among both objects since they are just mappings from states to real numbers that ought to be considered as an initial "guess" for J^*. As usual, different initial vectors have dramatic impact on the performance of the algorithms. In this paper, we wil only consider initial vectors from the class $\mathcal{J}$ of vectors satisfying two conditions:

$$0 \;\leq\; J \;\leq\; J^* \qquad \text{(admissibility)}, \tag{4}$$

$$J \;\leq\; TJ \qquad \text{(monotonicity)}. \tag{5}$$

Clearly, $\mathcal{J}$ is non-empty since the zero-constant vector and J^* satisfy the conditions, and $\mathcal{J}$ is *stable* under T, i.e. $T\mathcal{J} = \mathcal{J}$.

In the case of deterministic search, the initial vector J corresponds to the heuristic used to guide the search, and the conditions (4) and (5) refer to the standard *admissible* and *monotonic* properties of heuristic functions (Pearl, 1983). Admissibility is the property that the heuristic never overestimates the minimum cost to the goal while monotonicity refers to a kind of "triangle inequality" that the heuristic estimates must satisfy:

$$h(x) \;\leq\; c(x, x') + h(x')$$

for all states x and its children x', where $c(x, x')$ is the cost to go from x to x'. Using induction it is easy to prove that monotonicity is a stronger condition than admissibility (Pearl, 1983). Both admissibility and monotonicity had proved to be important properties since standard algorithm like A* and IDA* guarantee optimality only if the heuristic function is admissible, and A* is optimal if the heuristic is monotonic (Pearl, 1983; Korf, 1985b, 1985a; Dechter & Pearl, 1985).

In the stochastic case is also true that monotonicity implies admissibility. Indeed, if $0 \leq J \leq J^*$ then

$$0 \;\leq\; J \;\leq\; TJ \;\leq\; T^2 J \;\leq\; \cdots \;\leq\; T^k J \;\rightarrow\; J^*$$

by using Theorem 1 and the *monotonicity* of operator T.[4] Hence, the class $\mathscr{J}$ can be defined as the set of vectors satisfying $0 \leq J \leq TJ$.

Before presenting the results, we will need one more assumption on the problem:

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

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 (Bertsekas, 1995, p.89).

3.1 Stochastic Shortest-Path Problems

We begin by defining the following *stochastic process* that is very useful for deriving facts about the Value Iteration and RTDP algorithms. Denote with $(\mathbf{J}, \mathbf{X}, \mathbf{U}) = (J_k, X_k, U_k)_{k \geq 0}$ the stochastic process defined as

$$\begin{aligned}
J_0 &\overset{\text{def}}{=} J \in \mathscr{J}, \\
X_0 &\overset{\text{def}}{=} 1, \\
U_0 &\overset{\text{def}}{=} \operatorname*{argmin}_{u \in U(1)} \left\{ g(1,u) + \sum_{j=1}^{n} p(1,u,j) J_0(j) \right\}, \\
J_{k+1}(i) &\overset{\text{def}}{=} [\![X_k = i]\!](T^{k+1}J)(i) + [\![X_k \neq i]\!] J_k(i), \\
X_{k+1} &\overset{\text{def}}{=} i \in \{1, \ldots, n, t\} \text{ with probability } p(X_k, U_k, i), \\
U_{k+1} &\overset{\text{def}}{=} \operatorname*{argmin}_{u \in U(X_{k+1})} \left\{ g(X_{k+1}, u) + \sum_{j=1}^{n} p(X_{k+1}, u, j)(T^{k+1}J)(j) \right\}
\end{aligned}$$

where $[\![\varphi]\!]$ is equal to 1 (resp. 0) if φ is true (resp. false), and the ties for U_k are broken randomly (or in a systematic way). In plain words, the process begins at state $X_0 = 1$ and at each time k it chooses a control U_k greedy with respect to $T^k J$ and moves to state X_{k+1} with probability $p(X_k, U_k, X_{k+1})$.

Let us denote with $P_{x,J}(\,\cdot\,)$ and $E_{x,J}(\,\cdot\,)$ the probability distribution and expectation associated with the process when it is started at state x and the initial vector is J. The following results bound the expected time to reach the goal state, show that the DP operator is a pseudo-contraction mapping over $\mathscr{J}$ (see below), and establish bounds on the speed of convergence for Value Iteration.

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

Lemma 3 *For all $J \in \mathcal{J}$ and $k \geq 0$, $T^k J \geq J_k$.*

Proof: The result is trivial for $k = 0$. By definition, $(T^{k+1}J)(X_k) = J_{k+1}(X_k)$. For $i \neq X_k$,

$$(T^{k+1}J)(i) \;=\; (TT^kJ)(i) \;\geq\; (T^kJ)(i) \;\geq\; J_k(i) \;=\; J_{k+1}(i)$$

where the first inequality follows from the monotonicity of T and $J \in \mathcal{J}$, the second by inductive hypothesis, and the last equality by the definition of J_{k+1} and $i \neq X_k$. □

Theorem 4 *Suppose A1–A3 hold. Let τ be the random arrival time to the goal state, i.e. $\tau \overset{\text{def}}{=} \inf\{k > 0 : X_k = t\}$. Then,*

$$(k+1)P_{x,J}\big(X_k \neq t\big) \;\leq\; E_{x,J}(\tau) \;\leq\; \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i) - J(i) + J^*(X_0)\right) \tag{6}$$

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

Proof: Since $P(\tau = 0) = 0$, 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

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

Then,

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

using the fact $J_k(X_k) = J_{k-1}(X_k)$ since $p(X_{k-1}, \cdot, X_k) = 0$. 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 $\mathcal{F}_k$,

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

where the inequality comes from the Lemma, and $[\![X_k = t]\!] p(X_k, \cdot, j) = 0$ since t is absorbing. The reader with background in probability theory will recognize that W_k is a non-negative submartingale. Integrating both sides we obtain

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

The admissibility of J implies $J_k \leq J^*$ 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 5 *Suppose A1–A3 hold. There exist constants $K > 0$ and $0 \leq \alpha < 1$ that depend on x and J so 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 (6) divided by $K + 1$ is less than 1. For the second claim, note that the right-hand in (6) decreases as J increases. □

Theorem 6 *Suppose A1–A3 hold. There exist constants $K > 0$ and $0 \leq \alpha < 1$ so 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 *over $\mathcal{J}$.*[5]

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

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

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

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

Then, $|(T^K J')(x_0) - (T^K J)(x_0)| \leq P_{x_0, J}\big(X_K \neq t\big) \, \|J' - J\| \leq \alpha \|J' - J\|$ where the last inequality follows from Corollary 5 and the choice of α. □

Corollary 7 *Suppose A1–A3 hold. For all $J \in \mathcal{J}$, and integers $m, k \geq 0$*

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

where K and α are given by Theorem 6.

Proof:

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

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

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

□

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 6 does not say how to compute the constants K and α and, more important, K can be quite large.

These are fairly general results for the speed of convergence and quality of approximation for Value Iteration over general Stochastic Shortest-Path problems. The only strong assumption is that the initial cost vector is admissible and monotonic. Other previous results assume stronger conditions on the problem as requiring all stationary policies to be proper; see (Bertsekas, 1995) and references therein.

3.2 Real-Time Dynamic Programming

For studying the RTDP algorithm, we will consider a small modification of the previous stochastic process. The new process differs from the old one in that the updates and selection of controls are done as in the RTDP algorithm. Let $(\tilde{\mathbf{J}}, \tilde{\mathbf{X}}, \tilde{\mathbf{U}}) = (\tilde{J}_k, \tilde{X}_k, \tilde{U}_k)_{k\geq 0}$ be the process identical to $(\mathbf{J}, \mathbf{X}, \mathbf{U})$ except

$$\begin{aligned}\tilde{J}_{k+1}(i) \ &\overset{\text{def}}{=} \ [\![\tilde{X}_k = i]\!](T\tilde{J}_k)(i) + [\![\tilde{X}_k \neq i]\!]\tilde{J}_k(i),\\ \tilde{U}_{k+1} \ &\overset{\text{def}}{=} \ \operatorname*{argmin}_{u\in U(\tilde{X}_{k+1})} \left\{ g(\tilde{X}_{k+1}, u) + \sum_{j=1}^{n} p(\tilde{X}_{k+1}, u, j)\tilde{J}_{k+1}(j) \right\}.\end{aligned}$$

As before, let $\tilde{P}_{x,J}(\,\cdot\,)$ and $\tilde{E}_{x,J}(\,\cdot\,)$ be the distribution and expectation associated to such process when it is started from $x \in S$ and $J \in \mathcal{J}$. A small modification in the proof of Theorem 4 shows

Theorem 8 *Suppose A1–A3 hold and let* $\tau \overset{\text{def}}{=} \inf\{k > 0 : \tilde{X}_k = t\}$. *Then,*

$$(k+1)\tilde{P}_{x,J}\big(\tilde{X}_k \neq t\big) \ \leq \ \tilde{E}_{x,J}(\tau) \ \leq \ \underline{g}^{-1}\left(\sum_{i=1}^{n} J^*(i) - J(i) + J^*(\tilde{X}_0)\right). \tag{7}$$

Proof: The same proof as Theorem 4. 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 $\geq$. □

Therefore, the expected termination time of each RTDP trial is finite and bounded. By Theorem 2, we know that $\tilde{J}_k$ converges with probability 1 to J^* over the relevant states. 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.

$$R \ \overset{\text{def}}{=} \ \big\{i \in S : i \text{ appears infinitely often in } (\tilde{X}_0, \tilde{X}_1, \tilde{X}_2, \ldots)\big\}.$$

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, 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.

$$\tau_{k+1} \overset{\text{def}}{=} \inf\big\{ j > \tau_k : \{\tilde{X}_{\tau_k+1}, \tilde{X}_{\tau_k+2}, \ldots, \tilde{X}_{\tau_k+j}\} \supseteq R \big\}.$$

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

Theorem 9 *Suppose A1–A3 hold. Let K and α be the constants given by Theorem 6. 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 the set R, i.e. $\|J\|_R = \sup_{i \in R} |J(i)|$.

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

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

□

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 9 suggests that interleaving full DP updates (over the discovered state space) with RTDP might speedup the convergence of the algorithm. This since if τ'_k are the times for the full updates, then $\tau_k \leq \tau'_k$ for large k so the times τ_k can be controlled.

3.3 Suboptimality of Resulting Policies

In this section we offer bounds on the suboptimality of the greedy policies that result when the Value Iteration or RTDP algorithms are stopped. To introduce the notion of suboptimality, suppose for a moment that Value Iteration is stopped before convergence with a final vector J. Iif μ is the greedy policy with respect to J, then the suboptimality of μ is defined as the loss incurred when μ is applied to the system, i.e. $\|J^* - J_\mu\|$. Note that such quantity can be quite different from $\|J^* - J\|$. The problem of bounding $\|J^* - J_\mu\|$ is still an important and active area of research in the fields of Operations Research and AI. We will give bounds only for particular policies.

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

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

Definition 2 (Greedy policies) *A stationary policy μ is called 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 proper and greedy policies of order k are proper and greedy policies in the standard sense. The following results establish properties about proper and greedy policies and their suboptimality.

Theorem 10 *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 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,

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

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

Corollary 11 *Let J be any vector in $\mathcal{J}$ and μ a proper and greedy policy of order k with respect to J. Then,*

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

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

Proof: First note,

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

by definition of T_μ and T, the fact $J \leq J^*$ and Theorem 10. Since $J^* \leq J_\mu$, we have

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

□

4. Admissible and Monotonic Heuristics

In this section we describe how to obtain initial vectors satisfying the conditions of admissibility and monotonicity. The standard method for obtaining such estimates is to consider a *relaxation* of the problem, and use as heuristic the optimal solution to the relaxed problem (Pearl, 1983). We will consider the relaxation of assuming a *deterministic and optimistic* transition function. Such relaxation corresponds to the problem associated with the operator that results of replacing the summation with a minimization:

$$(\hat{T}J)(i) \overset{\text{def}}{=} \min_{u \in U(i)} \left\{ g(i,u) + \min_{j : p(i,u,j) > 0} J(j) \right\}.$$

Let $\hat{J}^*$ denote the optimal solution to the relaxed problem, i.e. $\hat{T}\hat{J}^* = \hat{J}^*$. We claim that $\hat{J}^*$ is an admissible and monotonic heuristic function for the SSP. Indeed, $\hat{J}^*$ is clearly non-negative and the monotonicity follows from

$$\begin{aligned}
(T\hat{J}^*)(i) &= \min_{u \in U(i)} g(i,u) + \sum_{j=1}^{n} p(i,u,j)\hat{J}^*(j) \\
&= \min_{u \in U(i)} g(i,u) + \sum_{j=1}^{n} p(i,u,j) \left[\min_{u',j'} g(j,u') + \hat{J}^*(j') \right] \\
&= \min_{u \in U(i)} g(i,u) + \sum_{j=1}^{n} \min_{u',j'} p(i,u,j) \left[g(j,u') + \hat{J}^*(j') \right] \\
&\geq \min_{u \in U(i)} g(i,u) + \min_{j : p(i,u,j) > 0} \ \min_{u' \in U(j)} g(j,u') + \min_{j' : p(j,u',j') > 0} \hat{J}^*(j') \\
&= (\hat{T}^2 \hat{J}^*)(i) \\
&= \hat{J}^*(i)
\end{aligned}$$

since $\hat{J}^*$ is the fix point of $\hat{T}$. Therefore, $\hat{J}^* \in \mathcal{J}$ so it can be used as an initial iterate for the Value Iteration and RTDP algorithms that guarantee above results. A similar computation shows that $\hat{J}^*$ can be replaced by any admissible and monotonic heuristic function for the relaxed problem.

5. Discussion

Stochastic Shortest-Path models had been used in a broad range of problems going from STRIPS planning, robot navigation in stochastic environment and games with complete information to more complex models involving uncertainty and partial information. The latter models correspond to Partially Observable MDPs (POMDPs) that generate SSPs in *belief space* (Bonet & Geffner, 2000; Hauskrecht, 2000; Kaelbling, Littman, & Cassandra, 1999).

As seen in this paper, the SSP model generalizes the heuristic search model of AI to the case of stochastic transitions and general cost functions. It is interesting to note that although heuristic search became one of the main areas within AI, the interest didn't go to the stochastic case. Thus, today's most important theoretical results for SSPs come from the field of Operations Research and Control Theory.

However, the algorithms from Operations Research compute full optimal policies instead of partial policies yet LRTA* and RTDP, from the AI community, were the first algorithms for computing partial optimal policies. The difference between full and partial optimal policies corresponds, for example, to the difference between the solutions of deterministic SSPs given by the Dijkstra and Bellman-Ford algorithms and the ones given by A* and IDA*. The former algorithms compute shortest paths from all states to the goal state while the latter only one shortest path from the initial state to the goal state. This difference is the main reason for the success of heuristic search algorithms in problems with large state spaces, e.g. the 24-puzzle and Rubik's cube (Korf & Taylor, 1996; Korf, 1997). Hence, we think that RTDP might be successful in SSPs with large state spaces that currently cannot be solved by any other method, e.g. Backgammon or Tetris (Tesauro, 1995; Tsitsiklis & Roy, 1996).

A more recent algorithm for computing partial optimal policies is the LAO* algorithm of Hansen and Zilberstein (2001). LAO* is an off-line algorithm that generalize the stochastic AO* algorithm (Nilsson, 1980) to the case when the graph has cycles.

In this paper, we have shown new convergence results for the Value Iteration and RTDP algorithms. The main results of the paper are in Theorems 4 and 8. They were obtained by considering a stochastic process that is closely related to the algorithms. The idea for the process came when reading a proof about the min-max version of LRTA* (Koenig, 2001). Although, the claims and proofs for the stochastic case are more complex than for the deterministic case, the methodology of generalizing definitions and proofs for the deterministic case had been successful.

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

Acknowledgements

This research was supported in part by grants from NSF, ONR, AFOSR, and DoD MURI program. The author is also supported by a USB/CONICIT fellowship from Venezuela.

References

Barto, A., Bradtke, S., & Singh, S. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, *72*, 81–138.

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control, (2 Vols).* Athena Scientific.

Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-Dynamic Programming.* Athena Scientific.

Bonet, B., & Geffner, H. (2000). Planning with incomplete information as heuristic search in belief space. In *Proceedings of AIPS-2000*, pp. 52–61. AAAI Press.

Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, *32*(3), 505–536.

Hansen, E., & Zilberstein, S. (2001). LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, *129*, 35–62.

Hauskrecht, M. (2000). Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, *13*, 33–94.

Kaelbling, L., Littman, M., & Cassandra, A. (1999). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Koenig, S. (2001). Minimax real-time heuristic search. *Artificial Intelligence*, *129*, 165–197.

Korf, R. (1985a). Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence*, *27*(1), 97–109.

Korf, R. (1985b). Iterative-depeening A*: An optimal admissible tree search. In *Proceedings of IJCAI-85*, pp. 1034–1036, Los Angeles, California. Morgan Kaufmann.

Korf, R. (1990). Real-time heuristic search. *Artificial Intelligence*, *42*, 189–211.

Korf, R. (1997). Finding optimal solutions to rubik's cube using patterns databases. In *Proceedings of AAAI-97*, pp. 700–705, Providence, RI. MIT Press.

Korf, R., & Taylor, L. (1996). Finding optimal solutions to the twenty-four puzzle. In *Proceedings of AAAI-96*, pp. 1202–1207, Protland, Oregon. MIT Press.

Nilsson, N. (1980). *Principles of Artificial Intelligence*. Tioga.

Pearl, J. (1983). *Heuristics*. Morgan Kaufmann.

Sutton, R., & Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, *38*, 58–68.

Tsitsiklis, J., & Roy, B. V. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning*, *22*, 59–94.