Loss Bounds for Uncertain Transition Probabilities in Markov Decision Processes

Andrew Mastin and Patrick Jaillet

***Abstract*— We analyze losses resulting from uncertain transition probabilities in Markov decision processes with bounded nonnegative rewards. We assume that policies are precomputed using exact dynamic programming with the estimated transition probabilities, but the system evolves according to different, true transition probabilities. Given a bound on the total variation error of estimated transition probability distributions, we derive upper bounds on the loss of expected total reward. The approach analyzes the growth of errors incurred by stepping backwards in time while precomputing value functions, which requires bounding a multilinear program. Loss bounds are given for the finite horizon undiscounted, finite horizon discounted, and infinite horizon discounted cases, and a tight example is shown.**

I. INTRODUCTION

With the widespread use of Markov decision processes (MDPs), it is not difficult to find situations where only estimates of transition probabilities must be used to determine polices. These scenarios arise often in inventory and resource allocation problems where historical demands must be used to predict future demands. In other cases, estimated distributions must be derived from a limited number of samples of an exact distribution, or simply estimated by an expert.

Many algorithms have been developed to optimize over transition probability uncertainty in a robust fashion. These approaches often use a max-min criteria under various uncertainty descriptions, and optimality is proved in many cases [1], [2], [3], [4], [5], [6]. This has led to many useful frameworks, such as the Markov decision process with imprecise probabilities (MDPIP), where transition probabilities are described by a set of linear inequalities, and the bounded-parameter Markov decision process (BMDP), where intervals are given for transition probabilities and rewards [7], [8]. However, the case where distribution estimates are used directly in conventional dynamic programming, rather than a robust algorithm, has received less attention. This paper addresses such scenarios.

In other related work, there has been some analysis of parameter sensitivity in dynamic programming. Hopp [9] analyzes the sensitivity of optimal polices under perturbations of problem parameters. Müller [10] studies variations in value functions resulting from transition probabilities that satisfy various stochastic order relations. There has also been recent work applying sensitivity analysis for uncertain rewards in dynamic programming [11], [12].

Loss bounds for uncertain value functions in MDPs have been relatively well explored. Singh and Yee [13] prove an upper bound on losses incurred from a bounded error in value functions for the infinite-horizon discounted case. Similar bounds have been found for finite-horizon undiscounted problems [14], [15], [16]. Loss bounds in approximate policy iteration and approximate value iteration scenarios are given in [17], [18], [19].

This paper provides loss bounds for situations where a policy for a MDP is determined using estimated transition probabilities, but the system evolves according to different, true transition probabilities. Specifically, the policy is precomputed using exact dynamic programming with estimated transition probabilities and stored in the form of a lookup table [20]. During the online phase of the algorithm, the MDP evolves according to its true underlying transition probabilities and decisions are made using the precomputed lookup table. We refer to this decision process as the approximate policy. The optimal policy, on the other hand, uses knowledge of a lookup table that is calculated with true transition probabilities. The loss is defined as the difference between the expected total reward obtained by the optimal policy and the one obtained by the approximate policy.

We derive loss bounds for the finite horizon undiscounted, finite horizon discounted, and infinite horizon discounted scenarios, and show a tight example for the finite horizon undiscounted case. We do not assume stationarity, so the transition probabilities, rewards, and states may be different for all stages. The bounds are derived from bounding errors introduced during the backwards induction process, which requires bounding a multilinear programming problem [21].

The organization of the paper is as follows. In Section II we provide background on Markov decision processes and dynamic programming. Section III shows the full derivation of the loss bounds, and Section IV gives a tight example for the undiscounted finite horizon case.

II. MARKOV DECISION PROCESSES AND DYNAMIC PROGRAMMING

We define a T-stage Markov decision process as follows. At each stage t, the system is in a state $S_t \in \mathcal{S}_t$, where $\mathcal{S}_t$ is the set of all states for stage t. In a given state S_t, we must select an action $x_t \in \mathcal{X}_t(S_t)$, where $\mathcal{X}_t(S_t)$ is the set of admissible actions for state S_t. We assume that there is a finite number of states and actions for all time periods. The selected action results in a reward $R_t(S_t, x_t)$. Rewards are

Both authors are with the Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA {mastin,jaillet}@mit.edu

Supported by NSF grant 1029603. The first author is supported in part by a NSF graduate research fellowship.

time discounted with factor $0 \leq \alpha \leq 1$, so that a reward R_t at time t is worth $\alpha^t R_t$. Transitions to the states at the following stage, S_{t+1}, occur randomly according to the distribution $\mathbb{P}(S_{t+1}|S_t, x_t)$. The system starts in a unique state S_0 and receives a terminal reward that is a function of the terminal state, $V_T(S_T)$. A policy $X_t^\pi : \mathcal{S}_t \rightarrow \mathcal{X}_t(S_t)$ is a mapping of states to actions. Let Π be the set of all polices, indexed by π. The goal is to find a policy maximizing total expected reward

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{T-1} \alpha^t R_t(S_t, X_t^\pi(S_t)) + \alpha^T V_T(S_T)\right], \quad (1)$$

which we refer to as the optimal policy. The optimal policy can be found using dynamic programming. Let $V_t(S_t)$ indicate the expected value of a state assuming that optimal decisions are made in the future. The update (backwards induction) equation is

$$\begin{aligned} V_t(S_t) &= \max_{x_t \in \mathcal{X}_t(S_t)} [R_t(S_t, x_t) \\ &\quad + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t, x_t) V_{t+1}(S_{t+1})], \\ &\quad t = 0, \ldots, T-1, \end{aligned} \quad (2)$$

where the notation $\sum_{\mathcal{S}_{t+1}}(\cdot)$ indicates $\sum_{S_{t+1} \in \mathcal{S}_{t+1}}(\cdot)$. We use the shorthand notation $\mathbb{E}\{V_{t+1}(S_{t+1})|S_t, x_t\}$ for $\sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t, x_t) V_{t+1}(S_{t+1})$. Finally, we omit $x_t \in \mathcal{X}_t(S_t)$ and simply use x_t. This gives

$$V_t(S_t) = \max_{x_t} [R_t(S_t, x_t) + \alpha \mathbb{E}\{V_{t+1}(S_{t+1})|S_t, x_t\}]. \quad (3)$$

Given the value function $V_{t+1}(\cdot)$ at time $t+1$, the value function $V_t(\cdot)$ for stage t can be determined with the above equation.

During the evolution of the MDP, the optimal policy makes decisions $x^*(S_t)$ by solving

$$x_t^*(S_t) = \underset{x_t}{\operatorname{argmax}} \left[R(S_t, x_t) + \alpha \mathbb{E}\left\{V_{t+1}(S_{t+1})|S_t, x_t\right\}\right]. \quad (4)$$

In describing the policy that occurs with estimated transition probabilities for the state S_t, which we will refer to as the approximate policy, it is helpful to distinguish between two sources of error that result in finding the decision with maximum value. The first error results from using the estimated transition probability function, denoted by $\hat{\mathbb{P}}(S_{t+1}|S_t, x_t)$, for the current state. The second error is due to the value function for the following stage, which has been solved using estimated transition probabilities from the end of the horizon. We refer to this function as the approximate value function $\hat{V}_{t+1}(S_{t+1})$. The approximate policy thus makes decisions

$$\hat{x}_t(S_t) = \underset{x_t}{\operatorname{argmax}} \left[R(S_t, x_t) + \alpha \hat{\mathbb{E}}\left\{\hat{V}_{t+1}(S_{t+1})|S_t, x_t\right\}\right], \quad (5)$$

where $\hat{\mathbb{E}}\{\hat{V}_{t+1}(S_{t+1})|S_t, x_t\}$ is used to denote $\sum_{\mathcal{S}_{t+1}} \hat{\mathbb{P}}(S_{t+1}|S_t, x_t)\hat{V}_{t+1}(S_{t+1})$.

The value of a state under the approximate policy, which we refer to simply as the policy value[1], is denoted by $V_t^{\hat{\pi}}(S_t)$ and is given by

$$V_t^{\hat{\pi}}(S_t) = R_t(S_t, \hat{x}_t(S_t)) + \alpha \mathbb{E}\{V_{t+1}^{\hat{\pi}}(S_{t+1})|S_t, \hat{x}_t(S_t)\}. \quad (6)$$

To simplify notation, we use x_t in place of $x_t(S_t)$ for various policies; the state of interest should be clear from context:

$$V_t^{\hat{\pi}}(S_t) = R_t(S_t, \hat{x}_t) + \alpha \mathbb{E}\{V_{t+1}^{\hat{\pi}}(S_{t+1})|S_t, \hat{x}_t\}. \quad (7)$$

Similarly, the value of a state under the optimal policy is given by

$$V_t^*(S_t) = R_t(S_t, x_t^*) + \alpha \mathbb{E}\{V_{t+1}^*(S_{t+1})|S_t, x_t^*\}. \quad (8)$$

It is important to note that the approximate value function $\hat{V}_t(S_t)$ is not in general equal to the policy value $V_t^{\hat{\pi}}(S_t)$. On the other hand, $V_t^*(S_t)$ defined in (8) is identical to $V_t(S_t)$ defined in (3).

The loss under the approximate policy for a state is defined by

$$L_t(S_t) = V_t^*(S_t) - V_t^{\hat{\pi}}(S_t). \quad (9)$$

The total loss of the policy $\mathcal{L}$ is given by the loss of the unique starting state

$$\mathcal{L} = L_0(S_0) = V_0^*(S_0) - V_0^{\hat{\pi}}(S_0). \quad (10)$$

III. Uncertainty in transition probabilities

We now focus on bounding the loss incurred by the approximate policy, where the approximate value function results from backwards induction with uncertain transition probabilities. The strategy is to find a recursion describing the growth of losses while stepping backwards in time. We define the estimation error F_t for a given state as

$$F_t(S_t) = V_t(S_t) - \hat{V}_t(S_t), \quad (11)$$

where we have replaced $V_t^*(S_t)$ with $V_t(S_t)$ for notational convenience, as these terms are equal. The policy error G_t for a given state is given by

$$G_t(S_t) = \hat{V}_t(S_t) - V_t^{\hat{\pi}}(S_t). \quad (12)$$

Note that

$$L_t(S_t) = F_t(S_t) + G_t(S_t). \quad (13)$$

For all states at time t, let f_t and g_t be the bounds on estimation error and policy error, respectively.

$$f_t = \max_{S_t} |F_t(S_t)|, \quad g_t = \max_{S_t} |G_t(S_t)|. \quad (14)$$

This gives

$$\mathcal{L} \leq f_0 + g_0. \quad (15)$$

Assuming $f_T = 0$ and $g_T = 0$, and that bounds on f_t and g_t can be derived in terms of f_{t+1} and g_{t+1}, the loss incurred by the algorithm may be bounded via induction. Our remaining analysis focuses on determining these bounds.

We define the difference between the true and estimated distributions as

$$D(S_{t+1}|S_t, x_t) = \hat{\mathbb{P}}(S_{t+1}|S_t, x_t) - \mathbb{P}(S_{t+1}|S_t, x_t). \quad (16)$$

We can view $D(\cdot|S_t, x_t)$ as vector of length $|\mathcal{S}_{t+1}|$ with entries that sum to zero. We assume that there is bounded

[1]This is more appropriately described as the approximate policy value; this term is used to avoid confusion with the approximate value function.

uncertainty in the transition probabilities for all states and time periods, characterized by L_1-norm error bound of $2k$, where $k \leq 1$. Thus, for all time periods, states, and actions, we have

$$\sum_{\mathcal{S}_{t+1}} \left| \hat{\mathbb{P}}(S_{t+1}|S_t, x_t) - \mathbb{P}(S_{t+1}|S_t, x_t) \right| \leq 2k. \tag{17}$$

This is equivalent to stating that the total variation distance is no greater than k. In the backwards induction process with estimated transition probabilities, values of the approximate value function are given by

$$\begin{aligned} \hat{V}_t(S_t) = \max_{x_t} [& R_t(S_t, x_t) \\ & + \alpha \sum_{\mathcal{S}_{t+1}} \hat{\mathbb{P}}(S_{t+1}|S_t, x_t) \hat{V}_{t+1}(S_{t+1})]. \end{aligned} \tag{18}$$

Equivalently, we have

$$\begin{aligned} & \hat{V}_t(S_t) \\ = \; & \max_{x_t} [R_t(S_t, x_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t, x_t) \hat{V}_{t+1}(S_{t+1}) \\ + \; & \alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|S_t, x_t) \hat{V}_{t+1}(S_{t+1})]. \end{aligned} \tag{19}$$

In order to derive loss bounds, the rewards must be bounded at each stage. While our analysis extends to other scenarios, we assume here that for all time periods t, states S_t, and decisions x_t,

$$0 \leq R_t(S_t, x_t) \leq \bar{R}. \tag{20}$$

The maximum possible value of a state at time t, denoted by $V_t^{\max}$, is given by

$$V_t^{\max} = \bar{R} \sum_{u=t}^{T} \alpha^{u-t}. \tag{21}$$

Similarly, the value of a state cannot be less than zero. The policy value for any state must obey the same properties, so we have that for all time periods t and states S_t,

$$0 \leq V_t^{\hat{\pi}}(S_t) \leq V_t^{\max}. \tag{22}$$

The same holds for the approximate value function, as shown in the following lemma.

Lemma 1: If terminal state values $V_T(S_T)$ are known with certainty and approximate state values $\hat{V}_t(S_t)$ for other time periods are determined via backwards induction with estimated transition probabilities, then for all time periods and states S_t,

$$0 \leq \hat{V}_t(S_t) \leq V_t^{\max}. \tag{23}$$

Proof: At each stage, every state value approximation is formed by taking a convex combination of state values for the following stage. The property holds for all stages by induction. ∎

We begin by assuming that we are given f_{t+1} and we wish to find an upper bound on f_t. From now on we fix the state S_t and make this implicit in the notation.

$$\begin{aligned} & \hat{V}_t - V_t \\ = \; & \max_{x_t} [R_t(x_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t) \hat{V}_{t+1}(S_{t+1}) \\ & + \alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t) \hat{V}_{t+1}(S_{t+1})] - V_t \\ = \; & \max_{x_t} [R_t(x_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t) V_{t+1}(S_{t+1}) \\ & - \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t) F_{t+1}(S_{t+1}) \\ & + \alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t) V_{t+1}(S_{t+1}) \\ & - \alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t) F_{t+1}(S_{t+1})] - V_t \\ \leq \; & \alpha \max [- \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t) F_{t+1}(S_{t+1}) \\ & + \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t) V_{t+1}(S_{t+1}) \\ & - \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t) F_{t+1}(S_{t+1})], \end{aligned} \tag{24}$$

where the last maximum is taken over all possible probability distributions, difference vectors, and value functions. Since the probability distributions and difference vectors are functions of x_t, we do not need to explicitly take the maximum over x_t. To further simplify notation, we abbreviate $F_{t+1}(S_{t+1})$, $V_{t+1}(S_{t+1})$, $\mathbb{P}(S_{t+1}|x_t)$, and $D(S_{t+1}|x_t)$ with $F(s)$, $V(s)$, $P(s)$, and $D(s)$, respectively. Also denote the set $\mathcal{S}_{t+1}$ by $\mathcal{S}$, f_{t+1} by f, and $V_{t+1}^{\max}$ by $\bar{V}$. Reformulating and temporarily ignoring the α term gives the following multilinear program.

$$\begin{aligned} \text{maximize} \quad & \sum_{s \in \mathcal{S}} [-P(s)F(s) + D(s)V(s) - D(s)F(s)] \\ \text{subject to} \quad & \sum_{s \in \mathcal{S}} |D(s)| \leq 2k \\ & \sum_{s \in \mathcal{S}} P(s) = 1 \\ & \sum_{s \in \mathcal{S}} D(s) = 0 \\ & 0 \leq V(s) \leq \bar{V} \qquad \forall s \\ & 0 \leq V(s) - F(s) \leq \bar{V} \quad \forall s \\ & |F(s)| \leq f \qquad \forall s \\ & 0 \leq P(s) \leq 1 \qquad \forall s \\ & 0 \leq P(s) + D(s) \leq 1 \quad \forall s, \end{aligned} \tag{25}$$

where the maximization is taken over all vectors $F(s)$, $V(s)$, $P(s)$, $D(s)$ that satisfy the constraints. Let the objective value be denoted by Z_1, and let Z_1^* be the optimal objective value. The constraint $0 \leq V(s) - F(s) \leq \bar{V}$ comes from the fact that $V(s) - F(s)$ refers to the approximate state value and Lemma 1. In an effort to find an upper bound for this problem, we first show that we can impose additional

assumptions on the probability distribution $P(s)$ without affecting the final bound. We then find optimal choices of variables when other variables are fixed to obtain the bound.

Define the states with the maximum and minimum $V(s) - F(s)$ values as

$$s^+ = \operatorname*{argmax}_s [V(s) - F(s)], \quad (26)$$

$$s^- = \operatorname*{argmin}_s [V(s) - F(s)]. \quad (27)$$

Consider an instance where $P(s)$, $V(s)$, and $F(s)$ are given, $P(s^-) \geq k$, $P(s^+) \leq 1-k$, and we must choose $D(s)$. The result has a simple structure.

Lemma 2: For instances of (25) where $P(s)$, $V(s)$, and $F(s)$ are fixed, $P(s^-) \geq k$, $P(s^+) \leq 1-k$, the optimal choice of $D(s)$ is given by

$$D(s) = \begin{cases} k & s = s^+ \\ -k & s = s^- \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

Proof: The non-constant part of the objective function is $\sum_{s\in\mathcal{S}} D(s)[V(s) - F(s)]$. Let $\mathcal{S}^+ = \{s : D(s) > 0\}$ and $\mathcal{S}^- = \{s : D(s) < 0\}$ and define $\Delta^+ = \sum_{s\in\mathcal{S}^+} D(s)$ and $\Delta^- = \sum_{s\in\mathcal{S}^-} D(s)$. For any given Δ^+ and Δ^-, it is optimal to choose $\mathcal{S}^+ = \{s^+\}$ and $\mathcal{S}^- = \{s^-\}$ as these provide the largest and smallest multipliers for Δ^+ and Δ^-, respectively. Now letting Δ^+, Δ^- vary, we must have $\Delta^+ = -\Delta^-$, and the resulting objective function is increasing in Δ^+, assuming $V(s^+) - F(s^+) \neq V(s^-) - F(s^-)$. Making Δ^+ as large as possible gives $\Delta^+ = k$ and $\Delta^- = -k$. ■

The assumption on $P(s^+)$ and $P(s^-)$ values is without loss of generality, as shown by the following lemma. Define an instance of (25) as a set of states with given $P(s)$, $F(s)$, $V(s)$ values for all states and the optimization is over $D(s)$. Let $\mathcal{P}$ be the set of all problem instances, and let $\bar{\mathcal{P}}$ be the set of problem instances that satisfy $P(s^+) \leq (1-k)$ and $P(s^-) \geq k$

Lemma 3: For every problem instance $\mathcal{I} \in \mathcal{P} \setminus \bar{\mathcal{P}}$ with optimal value Z^*, there exists a problem instance $\bar{\mathcal{I}} \in \bar{\mathcal{P}}$ with optimal value $\bar{Z}^*$ such that $Z^* \leq \bar{Z}^*$.

The proof is given in the Appendix. The lemma shows that without loss of generality, we can consider the problem (29) instead of problem (25).

$$\begin{aligned} \text{maximize} \quad & k[V(s^+) - F(s^+)] - k[V(s^-) - F(s^-)] \\ & - \sum_{s\in\mathcal{S}} P(s)F(s) \\ \text{subject to} \quad & \sum_{s\in\mathcal{S}} P(s) = 1,\ P(s^+) \leq 1-k,\ P(s^-) \geq k \\ & 0 \leq V(s) \leq \bar{V} \qquad \forall s \\ & 0 \leq V(s) - F(s) \leq \bar{V} \qquad \forall s \\ & |F(s)| \leq f \qquad \forall s \\ & 0 \leq P(s) \leq 1 \qquad \forall s \\ & V(s) - F(s) \leq V(s^+) - F(s^+) \qquad \forall s \\ & V(s) - F(s) \geq V(s^-) - F(s^-) \qquad \forall s. \end{aligned} \quad (29)$$

Let the objective value of (29) be denoted by Z_2, and its optimal value by Z_2^*. We have $Z_1^* = Z_2^*$.

We now assume that $V(s)$ and $P(s)$ are given, and we would like to calculate the optimal $F(s)$ values. We can rewrite the objective function as

$$Z_2 = kV_+ - kV_- - (P_+ + k)F_+ - (P_- - k)F_- - \sum_{s\in\mathcal{S}\setminus s_-, s_+} P(s)F(s), \quad (30)$$

where $P_+ = P(s^+)$, $P_- = P(s^-)$, $F_+ = F(s^+)$, $F_- = F(s^-)$, and $V_+ = V(s^+)$, $V_- = V(s^-)$. This makes it clear that all $F(s)$ values should be made as small as possible. The resulting objective function is bounded as follows.

Lemma 4: The optimal value of (29) satisfies

$$Z_2^* \leq k\bar{V} + (1-k)f. \quad (31)$$

Proof: Using the bounds on $F(s)$ gives

$$Z_2^* \leq kV_+ - kV_- - (P_+ + k)F_+ - (P_- - k)F_- + f(1 - P_+ - P_-). \quad (32)$$

We evaluate cases based on values for V_+, V_-.
Case 1: $V_+ \geq \bar{V} - f$, $V_- \geq \bar{V} - f$
The smallest that F_+ and F_- can be is $V_+ - \bar{V}$ and $V_- - \bar{V}$, respectively. This gives

$$\begin{aligned} Z_2^* &\leq kV_+ - kV_- - (P_+ + k)(V_+ - \bar{V}) \\ &\quad -(P_- - k)(V_- - \bar{V}) + f(1 - P_+ - P_-) \\ &\leq f, \end{aligned} \quad (33)$$

where we have used that both $\bar{V} - V_+ - f$ and $\bar{V} - V_+ - f$ are nonpositive by definition of the case. The other cases follow similar reasoning.
Case 2: $V_+ \geq \bar{V} - f$, $V_- \leq \bar{V} - f$

$$Z_2^* \leq k\bar{V} + (1-k)f, \quad (34)$$

where we have set $V_- = 0$, $F_+ = V_+ - \bar{V}$, $F_- = -f$ and used that $\bar{V} - V_+ - f \leq 0$.
Case 3: $V_+ \leq \bar{V} - f$, $V_- \leq \bar{V} - f$

$$Z_2^* \leq k\bar{V} + (1-k)f. \quad (35)$$

We have set both F_+ and F_- equal to $-f$, V_+ equal to its maximum possible value of $\bar{V} - f$, and $V_- = 0$.
Case 4: $V_+ \leq \bar{V} - f$, $V_- \geq \bar{V} - f$
It is optimal to set $F_+ = -f$ and $F_- = V_- - V_+ + F_+$, where the latter is the smallest value of F_- permitted from the constraint $V_+ - F_+ \geq V_- - F_-$. This gives

$$Z_2^* \leq f, \quad (36)$$

where we have used that $V_+ - V_-$ is nonpositive by definition of the case. The maximum bounds are achieved by the second and third cases. ■

An example of a tight solution (i.e. satisfying Lemma 4 with equality) using only three states is shown below.

	V	F	P	D
s_0	0	$-f$	k	$-k$
s_1	0	$-f$	$1-k$	0
s_2	$\bar{V}$	0	0	k

The solution provides an intuitive understanding of the bound. Consider an adversary who wishes to construct an approximate value as large as possible for a state with zero value. The adversary has a total probability weight of $2k$ that may be added/subtracted from various state probabilities in the following stage. To make the approximate state value large, the adversary adds weight k to the state s_2 with maximum value, yielding an objective increase of $k\bar{V}$, and subtracts k weight from the minimum value state s_0, which has zero value. Since adding k weight to $\bar{V}$ leaves at most $(1-k)$ remaining weight for the estimated distribution, this weight is associated with state s_1, as it carries maximum (negative) estimation error. This solution is used as a building block for the tight example shown in the next section.

Returning to our original analysis, we have that $\hat{V}_t(S_t) - V_t(S_t) \leq \alpha k V_{t+1}^{\max} + \alpha(1-k)f_{t+1}$. Finding a lower bound for $\hat{V}_t - V_t$ follows a similar approach.

$$\begin{aligned}
& V_t - \hat{V}_t \\
= \; & V_t - \max_{x_t}[R_t(x_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t)\hat{V}_{t+1}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t)\hat{V}_{t+1}(S_{t+1})] \\
= \; & V_t - \max_{x_t}[R_t(x_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t)V_{t+1}(S_{t+1}) \\
& -\alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t)F_{t+1}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t)V_{t+1}(S_{t+1}) \\
& -\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t)F_{t+1}(S_{t+1})] \\
\leq \; & -\alpha \min[-\sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|x_t)F_{t+1}(S_{t+1}) \\
& + \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t)V_{t+1}(S_{t+1}) \\
& - \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|x_t)F_{t+1}(S_{t+1})], \qquad (37)
\end{aligned}$$

where the last minimum is taken over all probability distributions, difference vectors and value functions. Simplifying notation and ignoring α gives a multilinear program with structure similar to that of (25). Using the appropriate substitutions, it is possible to show that $V_t(S_t) - \hat{V}_t(S_t) \leq \alpha k V_{t+1}^{\max} + \alpha(1-k)f_{t+1}$. The following lemma then follows.

Lemma 5: $f_t \leq \alpha k V_{t+1}^{\max} + \alpha(1-k)f_{t+1}$.

We now move to bounding the policy error, $G_t(S_t)$. Omitting the S_t notation, we have

$$\begin{aligned}
& \hat{V}_t - V_t^{\hat{\pi}} \\
= \; & R_t(\hat{x}_t) + \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)\hat{V}_{t+1}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|\hat{x}_t)\hat{V}_{t+1}(S_{t+1}) \\
& -R_t(\hat{x}_t) - \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)V_{t+1}^{\hat{\pi}}(S_{t+1}) \\
= \; & \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)V_{t+1}^{\hat{\pi}}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)G_{t+1}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|\hat{x}_t)V_{t+1}^{\hat{\pi}}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|\hat{x}_t)G_{t+1}(S_{t+1}) \\
& -\alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)V_{t+1}^{\hat{\pi}}(S_{t+1}) \\
= \; & \alpha \sum_{\mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|\hat{x}_t)G_{t+1}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|\hat{x}_t)V_{t+1}^{\hat{\pi}}(S_{t+1}) \\
& +\alpha \sum_{\mathcal{S}_{t+1}} D(S_{t+1}|\hat{x}_t)G_{t+1}(S_{t+1}). \qquad (38)
\end{aligned}$$

Solving for the minimum and maximum values of this term again leads to multilinear programs similar to (25), giving the following.

Lemma 6: $g_t \leq \alpha k V_{t+1}^{\max} + \alpha(1-k)g_{t+1}$.

Lemma 7: $f_0,\ g_0 \leq$

$$\bar{R}\left[\frac{\alpha k - \alpha^{T+1} + \alpha^{T+2}(1-k) + \alpha^{T+1}(1-k)^{T+1}(1-\alpha)}{(1-\alpha)(1-\alpha(1-k))}\right]. \qquad (39)$$

Proof: From Lemma 5

$$f_t \leq \alpha k V_{t+1}^{\max} + \alpha(1-k)f_{t+1}, \qquad (40)$$

and $f_T = 0$. Using the inductive hypothesis

$$f_t \leq k \sum_{u=t+1}^{T} \alpha^{u-t}(1-k)^{u-t-1}V_u^{\max} \qquad (41)$$

with (21) gives the result. The same expression also holds for g_0. ■

We may now state our final results.

Theorem 1: For a T-stage discounted problem ($\alpha < 1$) with transition probability total variation error no greater than k, the loss of the approximate policy satisfies $\mathcal{L} \leq$

$$2\bar{R}\left[\frac{\alpha k - \alpha^{T+1} + \alpha^{T+2}(1-k) + \alpha^{T+1}(1-k)^{T+1}(1-\alpha)}{(1-\alpha)(1-\alpha(1-k))}\right]. \qquad (42)$$

Proof: Using (39) with (15) gives the result. ■

Theorem 2: For an infinite horizon discounted problem ($\alpha < 1$) with transition probability total variation error no greater than k, the loss of the approximate policy satisfies

$$\mathcal{L} \leq \frac{2\bar{R}\alpha k}{(1-\alpha)(1-\alpha(1-k))}. \tag{43}$$

Proof: Since there is a bounded cost per stage, the limit of (42) as $T \to \infty$ is well defined [22]. ∎

Theorem 3: For a T-stage undiscounted problem ($\alpha = 1$) with transition probability total variation error no greater than k, the loss of the approximate policy satisfies

$$\mathcal{L} \leq \frac{2\bar{R}}{k}\left[-1 + k(T+1) + (1-k)^{T+1}\right], \tag{44}$$

for $k \neq 0$.

Proof: This follows using (41) with $\alpha = 1$ and $V_t^{\max} = (T-t+1)\bar{R}$. ∎

Loss sensitivity results are given simply by finding $\lim_{k\to 0} \frac{\partial \mathcal{L}}{\partial k}$. Since the loss functions are concave in k for all cases, these results give valid first order bounds. Of course, the resulting bounds are nearly tight only for very small values of k.

Corollary 1: For a finite or infinite horizon discounted problem ($\alpha < 1$), the loss of the approximate policy satisfies

$$\lim_{k\to 0} \frac{\partial \mathcal{L}}{\partial k} \leq \frac{2\bar{R}\alpha}{(1-\alpha)^2}, \quad \mathcal{L} \leq \frac{2\bar{R}\alpha k}{(1-\alpha)^2}. \tag{45}$$

Corollary 2: For a finite horizon undiscounted problem ($\alpha = 1$), the loss of the approximate policy satisfies

$$\lim_{k\to 0} \frac{\partial \mathcal{L}}{\partial k} \leq \bar{R}T(T+1), \quad \mathcal{L} \leq k\bar{R}T(T+1). \tag{46}$$

IV. Tight Example

We show a tight example for the undiscounted case assuming that $T \leq \frac{(1-k)}{k}$. Tight examples for the discounted cases can be derived using similar structure. Post-decision states are helpful in describing the example [20]. A post-decision state $S_t^x = (S_t, x_t)$ is defined by a state and an admissible decision for the state. We refer to values and approximate values of post-decision states as $V_t^x(\cdot)$ and $\hat{V}_t^x(\cdot)$, respectively.

The example is described with a directed tree structure, where nodes correspond to pre-decision states (denoted by W), post-decision states (denoted by X), and terminal states (denoted by Y), and arcs correspond to transitions between sequential states (that occur by decision or randomly). The example for $T = 3$ is shown in Fig. 1. The only decision takes place at $t = 0$, where there is a unique pre-decision state W_0 two post-decision states $X_0^A,\ X_0^B$ corresponding to path A and path B. Path A, which has a large expected reward, is defined as the set of all node descendants of X_0^A. Path B, which has a negligible expected reward, is the set of all node descendants of X_0^B.

For $t = 1, \ldots, T-1$, there are only two pre-decision and two post-decision states: $W_t^A,\ W_t^B,\ X_t^A,\ X_t^B$, where W_t^A denotes the pre-decision state at time t on path A, for example. For $t = 1, \ldots, T$, there are four terminal states, two for each path, which are denoted by $Y_t^{A+},\ Y_t^{A-},\ Y_t^{B+},\ Y_t^{B-}$. Finally, for $t = T$, there are two additional terminal states $Y_T^{A\circ},\ Y_T^{B\circ}$. The outgoing arcs for nodes are given as follows, where $\delta^+(S)$ denotes the set of nodes connected to node S with an outgoing arc.

$$\delta^+(W_0) = \{X_0^A,\ X_0^B\}. \tag{47}$$

$$\delta^+(X_t^Q) = \{W_{t+1}^Q,\ Y_{t+1}^{Q+},\ Y_{t+1}^{Q-}\}, \quad Q = A,\ B, \quad t = 0, \ldots, T-2. \tag{48}$$

$$\delta^+(W_t^Q) = \{X_t^Q\}, \quad Q = A,\ B, \quad t = 1, \ldots, T-1. \tag{49}$$

$$\delta^+(X_T^Q) = \{Y_T^{Q+},\ Y_T^{Q\circ}\ , Y_T^{Q-}\}, \quad Q = A,\ B. \tag{50}$$

Arc weights exiting pre-decision states correspond to rewards, and arc weights exiting post-decision states correspond to probabilities. Reward values are given as follows

$$R_0(S_0, \cdot) = \begin{cases} 0 & \text{choose } X_0^A \\ \epsilon & \text{choose } X_0^B, \end{cases} \tag{51}$$

where $0 < \epsilon \ll \bar{R}$. Since all other pre-decision states have only one decision (one exiting arc), we can simply specify the corresponding reward for $t = 1, \ldots, T-1$,

$$R_t(W_t^Q) = \begin{cases} \frac{(T-t+1)k\bar{R}}{1-k} & Q = A \\ 0 & Q = B. \end{cases} \tag{52}$$

With the assumption that $T \leq \frac{(1-k)}{k}$, these rewards to not violate the reward bound $\bar{R}$. Terminal values for $t = 1, \ldots, T$ are given by

$$V_t(S_t) = \begin{cases} (T-t+1)\bar{R} & S_t = Y_t^{Q+} \\ 0 & S_t = Y_t^{Q-}, \end{cases} \quad Q = A,\ B,$$

$$V_T(Y_T^{A\circ}) = \frac{k\bar{R}}{(1-k)}, \quad V_T(Y_T^{B\circ}) = 0. \tag{53}$$

Note that the terminal values with nonzero values do not violate the upper bound on rewards because they can be interpreted as states where the maximum reward is obtained at each time step for the remainder of the horizon. Transition and estimated transition probabilities are given by

$$\mathbb{P}(S_{t+1}|X_0^A) = \begin{cases} k & S_{t+1} = Y_{t+1}^{A+} \\ (1-k) & S_{t+1} = W_{t+1}^A(Y_T^{A\circ}) \\ 0 & S_{t+1} = Y_{t+1}^{A-}, \end{cases} \tag{54}$$

$$\mathbb{P}(S_{t+1}|X_0^B) = \begin{cases} 0 & S_{t+1} = Y_{t+1}^{B+} \\ (1-k) & S_{t+1} = W_{t+1}^B(Y_T^{B\circ}) \\ k & S_{t+1} = Y_{t+1}^{B-}, \end{cases} \tag{55}$$

$$\hat{\mathbb{P}}(S_{t+1}|X_0^A) = \begin{cases} 0 & S_{t+1} = Y_{t+1}^{A+} \\ (1-k) & S_{t+1} = W_{t+1}^A(Y_T^{A\circ}) \\ k & S_{t+1} = Y_{t+1}^{A-}, \end{cases} \tag{56}$$

$$\hat{\mathbb{P}}(S_{t+1}|X_0^B) = \begin{cases} k & S_{t+1} = Y_{t+1}^{B+} \\ (1-k) & S_{t+1} = W_{t+1}^B(Y_T^{B\circ}) \\ 0 & S_{t+1} = Y_{t+1}^{B-}, \end{cases} \tag{57}$$

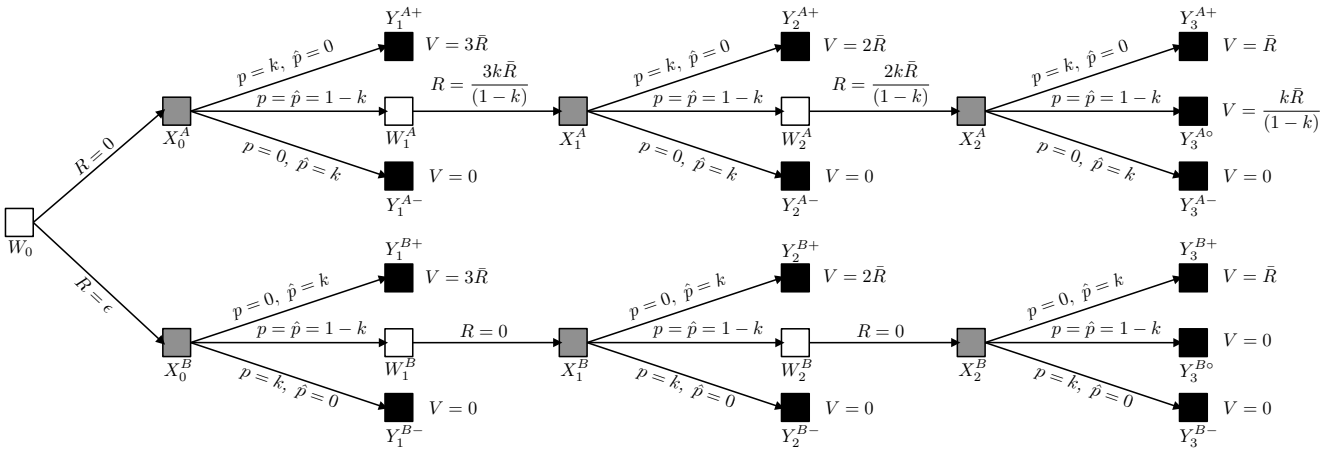

Fig. 1. Tight example for undiscounted case with $T = 3$. White nodes are pre-decision states, gray nodes are post-decision states, and black nodes are terminal states.. Terminal states with nonzero values do not violate the bound on reward $\bar{R}$, because they can be interpreted as paths where reward $\bar{R}$ is received at each following stage.

for $t = 0, \ldots, T-1$. It can be verified by induction that the value of path A is given by

$$V_0^x(X_0^A) = 2k \sum_{u=1}^{T} (1-k)^{u-1} V_u^{\max}, \tag{58}$$

and that $\hat{V}_0^x(X_0^A) = \hat{V}_0^x(X_0^B) = \frac{V_0^x(X_0^A)}{2}$. Inspection of the graph structure shows that $V_0^x(X_0^B) = 0$. The optimal choice for the problem is to choose path A and obtain the expected value given in (58). However, since the immediate reward for choosing path B is larger by ϵ, the approximate policy chooses path B and realizes value ϵ. Letting $\epsilon \to 0$, the loss approaches the term given in (44).

V. CONCLUSIONS

We have presented loss bounds for exact dynamic programming polices that are determined using estimated transition probabilities for the case of both finite and infinite horizon problems. We analyzed the problem from a strictly worst case scenario, so tight instances of our bounds are unlikely to arise in practice. A natural next step would be to bound losses assuming that the transition probabilities are random variables with known distributions. It may also be possible to improve bounds for problems with specific structure.

APPENDIX

Lemma 3: For every problem instance $\mathcal{I} \in \mathcal{P} \setminus \bar{\mathcal{P}}$ with optimal value Z^*, there exists a problem instance $\bar{\mathcal{I}} \in \bar{\mathcal{P}}$ with optimal value $\bar{Z}^*$ such that $Z^* \leq \bar{Z}^*$.

Proof: The presence of $\sum_{s \in \mathcal{S}} -P(s)F(s)$ in the objective function makes the proof non-trivial. Returning to the analysis in Lemma 2, if $P(s^-) < -\Delta^-$, it is optimal to add states to $\mathcal{S}^-$ in increasing order of $V(s) - F(s)$ until $\sum_{s \in \mathcal{S}^-} P(s) \geq -\Delta^-$. If $P(s^+) > 1-k$, then $\Delta^+ = 1 - P(s^+)$. With this in mind, we use a problem transformation algorithm to produce the instance $\bar{\mathcal{I}}$ given $\mathcal{I}$, and show that during each step of the algorithm, the optimal objective value increases. The procedure for generating the new problem instance is shown in Algorithm 1.

Algorithm 1 Problem Transformation

Input: $\mathcal{I} \in \mathcal{P}$
Output: $\bar{\mathcal{I}} \in \bar{\mathcal{P}}$

1: $\mathcal{A} \leftarrow \mathcal{I}$
2: $s^+ \leftarrow \text{argmax}_{s \in \mathcal{A}} V(s) - F(s)$
3: $s^- \leftarrow \text{argmin}_{s \in \mathcal{A}} V(s) - F(s)$
4: **if** $P(s^+) > 1-k$ **then**
5: $\quad c \leftarrow P(s^+) - (1-k)$
6: $\quad P(s^+) \leftarrow P(s^+) - c$
7: $\quad P(s^-) \leftarrow P(s^-) + c$
8: $\quad F(s^-) \leftarrow 0$
9: $\quad V(s^-) \leftarrow 0$
10: **end if**
11: **while** $\sum_{s \in \mathcal{S}^-(\mathcal{A})} P(s) \leq \Delta^-$ **do**
12: $\quad r_1 \leftarrow \text{argmin}_{s \in \mathcal{A}} V(s) - F(s)$
13: $\quad r_2 \leftarrow \text{argmin}_{s \in \mathcal{A} \setminus \{r_1\}} V(s) - F(s)$
14: $\quad P(r') \leftarrow P(r_1) + P(r_2)$
15: $\quad D(r') \leftarrow D(r_1) + D(r_2)$
16: $\quad F(r') \leftarrow \max\{F(r_2) - V(r_2), -f\}$
17: $\quad V(r') \leftarrow 0$
18: $\quad \mathcal{A} \leftarrow (\mathcal{A} \setminus \{r_1, r_2\}) \cup \{r'\}$
19: **end while**
20: $\bar{\mathcal{I}} \leftarrow \mathcal{A}$

Beginning with line 4, if $P(s^+) > 1-k$, the algorithm adjusts the values of states s^+ and s^-. Under the optimal solution, if $P(s^+)$ is decreased by c then $D(s^+)$ increases by c. Let $V'(\cdot)$ $F'(\cdot)$, $D'(\cdot)$ refer to state properties after lines 4-10 of the algorithm have been executed. We use the shorthand notation P_+ for $P(s^+)$, F'_- for $F'(s^-)$, etc. The change in the optimal objective value ΔZ^* for line 6 is

$$\begin{aligned} \Delta Z^* \quad = \quad & -(P_+ - c)F_+ + (D_+ + c)V_+ - (D_+ + c)F_+ \\ & + P_+F_+ - D_+V_+ + D_+F_+ \end{aligned}$$

$$= \quad cV_+, \tag{59}$$

which is always nonnegative. The change in optimal objective value for lines 7-9 is

$$\begin{aligned}\Delta Z^* &= -P'_-F'_- + D'_-V'_- - D'_-F'_- \\ &\quad +P_-F_- - D_-V_- + D_-F_- \\ &= P_-V_-, \end{aligned} \tag{60}$$

where we have used $P_- = -D_-$ under the optimal solution.

Lines 11-19 of the algorithm are used to aggregate states in an iterative fashion until $P(s^-) \geq k$. Define $\mathcal{S}^-(\mathcal{A})$ as the set of states in instance $\mathcal{A}$ that are assigned negative $D(s)$ values under the optimization of (25), as explained in Lemma 2. At each iteration of the process, the algorithm aggregates the two smallest $V(s)-F(s)$ states to produce a new state r' with $V(r') - F(r')$ value smaller than the remaining states. At any point in the process, let r_1 and r_2 be the states with the smallest and second smallest $V(s) - F(s)$ values, respectively. Initially, $r_1 = s^-$. For other iterations, $r_1 = r'$ from the previous iteration. During the aggregation process, we wish only to increase the objective value, so adding the aggregated state and removing the two original states always creates a positive change in the objective function:

$$\begin{aligned}-P'F' + D'(V' - F') \quad &\geq \quad -P_1F_1 - P_2F_2 + D_1(V_1 - F_1) \\ &\quad +D_2(V_2 - F_2), \end{aligned} \tag{61}$$

where $P_1 = P(r_1)$, $P_2 = P(r_2)$, and V', F', D' now refer to state values obtained after one iteration of the aggregation process. The D' and P' values are given by

$$D' = D_1 + D_2, \tag{62}$$

$$P' = P_1 + P_2. \tag{63}$$

The V' value is always equal to zero, and F' is determined according to

$$F' = \max(F_2 - V_2, -f), \tag{64}$$

which results from the constraints that $V' - F'$ must be positive and $V' - F' \leq V_2 - F_2$. The algorithm repeats the process while $P(r') \leq k$, so $D_1 = -P_1$ always holds, as the optimal $D(s)$ places the maximum possible weight on the lowest $V(s) - F(s)$ coefficient before placing weight on other states. The change in objective value ΔZ^* for the aggregation process is always positive. For each iteration, there are two cases; $-f > F_2 - V_2$ and $-f \leq F_2 - V_2$. For the first case, we have

$$\begin{aligned}\Delta Z^* &= -P'F' + D'(V' - F') + P_1F_1 + P_2F_2 - D_1V_1 \\ &\quad +D_1F_1 - D_2V_2 + D_2F_2 \\ &= P_1(F_1 + f) + P_2(F_2 + f) \\ &\quad +D_1(f + F_1 - V_1) + D_2(f + F_2 - V_2). \end{aligned} \tag{65}$$

Since $P_1 = -D_1$,

$$\Delta Z^* = P_1V_1 + P_2(F_2 + f) + D_2(f + F_2 - V_2). \tag{66}$$

The terms D_2 and $f + F_2 - V_2$ are nonpositive (the latter by definition of the case) and the remaining terms are all nonnegative. The second case gives

$$\begin{aligned}\Delta Z^* &= P_1(V_2 - F_2 + F_1) + P_2V_2 \\ &\quad +D_1(V_2 - F_2 - V_1 + F_1). \end{aligned} \tag{67}$$

Again using $P_1 = -D_1$,

$$\Delta Z^* = P_1V_1 + P_2V_2, \tag{68}$$

which is always nonnegative. ■

REFERENCES

[1] E. A. Silver, "Markovian decision processes with uncertain transition probabilities or rewards," Massachusetts Institute of Technology, Tech. Rep. AD0417150, August 1963.

[2] J. K. Satia and R. E. Lave Jr., "Markovian decision processes with uncertain transition probabilities," *Operations Research*, vol. 21, no. 3, pp. 728–740, 1973.

[3] R. S. Filho and F. W. Trevizan, "Multilinear and integer programming for Markov decision processes with imprecise probabilities," in *5th International Symposium on Imprecise Probability: Theories and Applications*, 2007.

[4] M. Kurano, M. Hosaka, Y. Huang, and J. Song, "Controlled Markov set-chains with discounting." *J. Appl. Probab.*, vol. 35, no. 2, pp. 293–302, 1998.

[5] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Oper. Res.*, vol. 53, no. 5, pp. 780–798, September-October 2005.

[6] K. V. Delgado, S. Sanner, and L. N. de Barros, "Efficient solutions to factored MDPs with imprecise transition probabilities," *Artificial Intelligence*, vol. 175, no. 910, pp. 1498 – 1527, 2011.

[7] C. C. White III and H. K. Eldeib, "Markov decision processes with imprecise transition probabilities," *Operations Research*, vol. 42, pp. 739–749, 1994.

[8] R. Givan, S. Leach, and T. Dean, "Bounded-parameter Markov decision processes," *Artificial Intelligence*, vol. 122, pp. 71–109, 2000.

[9] W. J. Hopp, "Sensitivity analysis in discrete dynamic programming," *J. Optim. Theory Appl.*, vol. 56, pp. 257–269, February 1988.

[10] A. Müller, "How does the value function of a Markov decision process depend on the transition probabilities?" *Math. Oper. Res.*, vol. 22, pp. 872–885, November 1997.

[11] C. H. Tan and J. C. Hartman, *Sensitivity Analysis and Dynamic Programming*. John Wiley & Sons, Inc., 2010.

[12] ——, "Sensitivity analysis in Markov decision processes with uncertain reward parameters," *Journal of Applied Probability*, vol. 48, no. 4, pp. 954 – 967, 2011.

[13] S. P. Singh and R. C. Yee, "An upper bound on the loss from approximate optimal-value functions," *Machine Learning*, vol. 16, no. 3, pp. 227–233, 1994.

[14] M. Kearns, Y. Mansour, and A. Y. Ng, "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," in *Machine Learning*, 1999, pp. 1324–1331.

[15] L. Mercier and P. Van Hentenryck, "Performance analysis of online anticipatory algorithms for large multistage stochastic integer programs," in *Proceedings of the 20th international joint conference on Artifical intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 1979–1984.

[16] P. V. Hentenryck and R. Bent, *Online Stochastic Combinatorial Optimization*. The MIT Press, 2009.

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

[18] R. Munos, "Error bounds for approximate policy iteration," in *ICML*, 2003, pp. 560–567.

[19] A. M. Farahmand, R. Munos, and C. Szepesvári, "Error propagation for approximate policy and value iteration," in *NIPS*, 2010, pp. 568–576.

[20] W. B. Powell, *Approximate Dynamic Programming*. John Wiley and Sons, Inc., 2007.

[21] R. F. Drenick, "Multilinear programming: Duality theories," *Journal of Optimization Theory and Applications*, vol. 72, pp. 459–486, 1992.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2007.