Bounded Parameter Markov Decision Processes

Robert Givan and Sonia Leach and Thomas Dean

Department of Computer Science, Brown University
115 Waterman Street, Providence, RI 02912, USA
http://www.cs.brown.edu/people/{rlg,sml,tld}
Phone: (401) 863-7600 Fax: (401) 863-7657
Email: {rlg,sml,tld}@cs.brown.edu

Abstract. In this paper, we introduce the notion of an *bounded parameter Markov decision process* (BMDP) as a generalization of the familiar *exact* MDP. A bounded parameter MDP is a set of exact MDPs specified by giving upper and lower bounds on transition probabilities and rewards (all the MDPs in the set share the same state and action space). BMDPs form an efficiently solvable special case of the already known class of MDPs with *imprecise parameters* (MDPIPs). Bounded parameter MDPs can be used to represent variation or uncertainty concerning the parameters of sequential decision problems in cases where no prior probabilities on the parameter values are available. Bounded parameter MDPs can also be used in aggregation schemes to represent the variation in the transition probabilities for different base states aggregated together in the same aggregate state.

We introduce *interval value functions* as a natural extension of traditional value functions. An interval value function assigns a closed real interval to each state, representing the assertion that the value of that state falls within that interval. An interval value function can be used to bound the performance of a policy over the set of exact MDPs associated with a given bounded parameter MDP. We describe an iterative dynamic programming algorithm called *interval policy evaluation* which computes an interval value function for a given BMDP and specified policy. Interval policy evaluation on a policy π computes the most restrictive interval value function that is sound, *i.e.,* that bounds the value function for π in every exact MDP in the set defined by the bounded parameter MDP. We define *optimistic* and *pessimistic* notions of optimal policy, and provide a variant of value iteration [Bellman, 1957] that we call *interval value iteration* which computes a policies for a BMDP that are optimal in these senses.

1 Introduction

The theory of Markov decision processes (MDPs) provides the semantic foundations for a wide range of problems involving planning under uncertainty [Boutilier *et al.*, 1995a, Littman, 1997]. In this paper, we introduce a generalization of Markov decision processes called *bounded parameter Markov decision processes* (BMDPs) that allows us to model uncertainty in the parameters that comprise an MDP. Instead of encoding a parameter such as the probability of making a transition from one state to another as a single number, we specify a range of possible values for the parameter as a closed interval of the real numbers.

A BMDP can be thought of as a family of traditional (exact) MDPs, *i.e.,* the set of all MDPs whose parameters fall within the specified ranges. From this perspective, we may have no justification for committing to a particular MDP in this family, and wish to analyze the consequences of this lack of commitment. Another interpretation for a BMDP is that the states of the BMDP actually represent sets (aggregates) of more primitive states that we choose to group together. The intervals here represent the ranges of the parameters over the primitive states belonging to the aggregates. While any policy on the original (primitive) states induces a stationary distribution over those states which can be used to give prior probabilities to the different transition probabilities in the intervals, we may be unable to compute these prior probabilities—the original reason for aggregating the states is typically to avoid such expensive computation over the original large state space.

BMDPs are an efficiently solvable specialization of the already known *Markov Decision Processes with Imprecisely Known Transition Probabilities* (MDPIPs). In the related work section we discuss in more detail how BMDPs relate to MDPIPs.

In a related paper, we have shown how BMDPs can be used as part of a strategy for efficiently approximating the solution of MDPs with very large state spaces and dynamics compactly encoded in a factored (or implicit) representation [Dean *et al.*, 1997]. In this paper, we focus exclusively on BMDPs, on the BMDP analog of value functions, called *interval value functions*, and on policy selection for a BMDP. We provide BMDP analogs of the standard (exact) MDP algorithms for computing the value function for a fixed policy (plan) and (more generally) for computing optimal value functions over all policies, called *interval policy evaluation* and *interval value iteration* (IVI) respectively. We define the desired output values for these algorithms and prove that the algorithms converge to these desired values in polynomial-time, for a fixed discount factor. Finally, we consider two different notions of optimal policy for an BMDP, and show how IVI can be applied to extract the optimal policy for each notion. The first notion of optimality states that the desired policy must perform better than any other under the assumption that an adversary selects the model parameters. The second notion requires the best possible performance when a friendly choice of model parameters is assumed.

2 Exact Markov Decision Processes

An (exact) Markov decision process M is a four tuple $M = (\mathcal{Q}, \mathcal{A}, F, R)$ where $\mathcal{Q}$ is a set of states, $\mathcal{A}$ is a set of actions, R is a reward function that maps each state to a real value $R(q)$,[1] and F is a state-transition distribution so that for

[1] The techniques and results in this paper easily generalize to more general reward functions. We adopt a less general formulation to simplify the presentation.

$\alpha \in \mathcal{A}$ and $p, q \in \mathcal{Q}$,

$$F_{pq}(\alpha) = \Pr(X_{t+1} = q | X_t = p, U_t = \alpha)$$

where X_t and U_t are random variables denoting, respectively, the state and action at time t. When needed we will write F^M denote the transition function of the MDP M.

A *policy* is a mapping from states to actions, $\pi : \mathcal{Q} \rightarrow \mathcal{A}$. The set of all policies is denoted Π. An MDP M together with a fixed policy $\pi \in \Pi$ determines a Markov chain such that the probability of making a transition from p to q is defined by $F_{pq}(\pi(p))$. The *expected value function* (or simply the *value function*) associated with such a Markov chain is denoted $V_{M,\pi}$. The value function maps each state to its *expected discounted cumulative reward* defined by

$$V_{M,\pi}(p) = R(p) + \gamma \sum_{q \in \mathcal{Q}} F_{pq}(\pi(p)) V_{M,\pi}(q)$$

where $0 \leq \gamma < 1$ is called the *discount rate.*[2] In most contexts, the relevant MDP is clear and we abbreviate $V_{M,\pi}$ as V_π.

The optimal value function V_M^* (or simply V^* where the relevant MDP is clear) is defined as follows.

$$V^*(p) = \max_{\alpha \in \mathcal{A}} \left(R(p) + \gamma \sum_{q \in \mathcal{Q}} F_{pq}(\alpha) V^*(q) \right)$$

The value function V^* is greater than or equal to any value function V_π in the partial order $\geq_{\mathrm{dom}}$ defined as follows: $V_1 \geq_{\mathrm{dom}} V_2$ if and only if for all states q, $V_1(q) \geq V_2(q)$.

An optimal policy is any policy π^* for which $V^* = V_{\pi^*}$. Every MDP has at least one optimal policy, and the set of optimal policies can be found by replacing the max in the definition of V^* with arg max.

3 Bounded Parameter Markov Decision Processes

An *bounded parameter MDP* is a four tuple $\mathcal{M} = (\mathcal{Q}, \mathcal{A}, \hat{F}, \hat{R})$ where $\mathcal{Q}$ and $\mathcal{A}$ are defined as for MDPs, and $\hat{F}$ and $\hat{R}$ are analogous to the MDP F and R but yield closed real intervals instead of real values. That is, for any action α and states p, q, $\hat{R}(p)$ and $\hat{F}_{p,q}(\alpha)$ are both closed real intervals of the form $[l, u]$ for l and u real numbers with $l \leq u$, where in the case of $\hat{F}$ we require $0 \leq l \leq u \leq 1$.[3] To ensure that $\hat{F}$ admits well-formed transition functions, we require that for

[2] In this paper, we focus on expected discounted cumulative reward as a performance criterion, but other criteria, *e.g.,* total or average reward [Puterman, 1994], are also applicable to bounded parameter MDPs.

[3] To simplify the remainder of the paper, we assume that the reward bounds are always tight, *i.e.,* that for all $q \in \mathcal{Q}$, for some real l, $\hat{R}(q) = [l, l]$, and we refer to l as $R(q)$. The generalization to nontrivial bounds on rewards is straightforward.

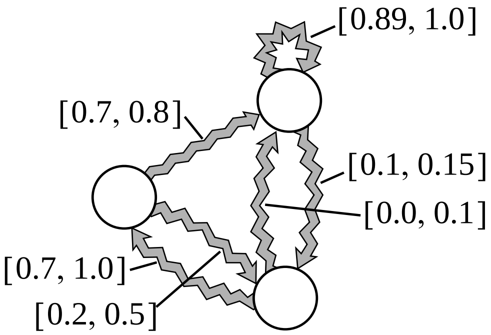

Fig. 1. The state-transition diagram for a simple bounded parameter Markov decision process with three states and a single action. The arcs indicate possible transitions and are labeled by their lower and upper bounds.

any action α and state p, the sum of the lower bounds of $\hat{F}_{pq}(\alpha)$ over all states q must be less than or equal to 1 while the upper bounds must sum to a value greater than or equal to 1. Figure 1 depicts the state-transition diagram for a simple BMDP with three states and one action.

A BMDP $\mathcal{M} = (\mathcal{Q}, \mathcal{A}, \hat{F}, \hat{R})$ defines a set of exact MDPs which, by abuse of notation, we also call $\mathcal{M}$. For exact MDP $M = (\mathcal{Q}', \mathcal{A}', F', R')$, we have $M \in \mathcal{M}$ if $\mathcal{Q} = \mathcal{Q}'$, $\mathcal{A} = \mathcal{A}'$, and for any action α and states p, q, $R'(p)$ is in the interval $\hat{R}(p)$ and $F'_{p,q}(\alpha)$ is in the interval $\hat{F}_{p,q}(\alpha)$. We rely on context to distinguish between the tuple view of $\mathcal{M}$ and the exact MDP set view of $\mathcal{M}$. In the definitions in this section, the BMDP $\mathcal{M}$ is implicit.

An *interval value function* $\hat{V}$ is a mapping from states to closed real intervals. We generally use such functions to indicate that the given state's value falls within the selected interval. Interval value functions can be specified for both exact and BMDPs. As in the case of (exact) value functions, interval value functions are specified with respect to a fixed policy. Note that in the case of BMDPs a state can have a range of values depending on how the transition and reward parameters are instantiated, hence the need for an interval value function.

For each of the interval valued functions $\hat{F}$, $\hat{R}$, $\hat{V}$ we define two real valued functions which take the same arguments and give the upper and lower interval bounds, denoted $\overline{F}$, $\overline{R}$, $\overline{V}$, and $\underline{F}$, $\underline{R}$, $\underline{V}$, respectively. So, for example, at any state q we have $\hat{V}(q) = [\underline{V}(q), \overline{V}(q)]$.

Definition 1. For any policy π and state q, we define the interval value $\hat{V}_\pi(q)$ of π at q to be the interval

$$\left[\min_{M \in \mathcal{M}} V_{M,\pi}(q),\ \max_{M \in \mathcal{M}} V_{M,\pi}(q)\right]$$

In Section 5 we will give an iterative algorithm which we have proven to converge to $\hat{V}_\pi$. In preparation for that discussion we now state that there is at least one specific MDP in $\mathcal{M}$ which simultaneously achieves $\overline{V}_\pi(q)$ for all states q (and likewise a specific MDP achieving $\underline{V}_\pi(q)$ for all q).

Definition 2. For any policy π, an MDP in $\mathcal{M}$ is π-*maximizing* if it is a possible value of $\arg\max_{M \in \mathcal{M}} V_{M,\pi}$ and it is π-*minimizing* if it is in $\arg\min_{M \in \mathcal{M}} V_{M,\pi}$.

Theorem 3. *For any policy π, there exist π-maximizing and π-minimizing MDPs in $\mathcal{M}$.*

This theorem implies that $\underline{V}_\pi$ is equivalent to $\min_{M \in \mathcal{M}} V_{M,\pi}$ where the minimization is done relative to $\geq_{\text{dom}}$, and likewise for $\overline{V}$ using max. We give an algorithm in Section 5 which converges to $\underline{V}_\pi$ by also converging to a π-minimizing MDP in $\mathcal{M}$ (likewise for $\overline{V}_\pi$).

We now consider how to define an optimal value function for a BMDP. Consider the expression $\max_{\pi \in \Pi} \hat{V}_\pi$. This expression is ill-formed because we have not defined how to rank the interval value functions $\hat{V}_\pi$ in order to select a maximum. We focus here on two different ways to order these value functions, yielding two notions of optimal value function and optimal policy. Other orderings may also yield interesting results.

First, we define two different orderings on closed real intervals:

$$[l_1, u_1] \leq_{\text{pes}} [l_2, u_2] \iff \begin{cases} l_1 < l_2, \text{ or} \\ l_1 = l_2 \text{ and } u_1 \leq u_2 \end{cases}$$
$$[l_1, u_1] \leq_{\text{opt}} [l_2, u_2] \iff \begin{cases} u_1 < u_2, \text{ or} \\ u_1 = u_2 \text{ and } l_1 \leq l_2 \end{cases}$$

We extend these orderings to partially order interval value functions by relating two value functions $\hat{V}_1 \leq \hat{V}_2$ only when $\hat{V}_1(q) \leq \hat{V}_2(q)$ for every state q. We can now use either of these orderings to compute $\max_{\pi \in \Pi} \hat{V}_\pi$, yielding two definitions of optimal value function and optimal policy. However, since the orderings are partial (on value functions), we must still prove that the set of policies contains a policy which achieves the desired maximum under each ordering (*i.e.,* a policy whose interval value function is ordered above that of every other policy).

Definition 4. The *optimistic optimal value function* $\hat{V}_{\text{opt}}$ and the *pessimistic optimal value function* $\hat{V}_{\text{pes}}$ are given by:

$$\hat{V}_{\text{opt}} = \max_{\pi \in \Pi} \hat{V}_\pi \text{ using } \leq_{\text{opt}} \text{ to order interval value functions}$$
$$\hat{V}_{\text{pes}} = \max_{\pi \in \Pi} \hat{V}_\pi \text{ using } \leq_{\text{pes}} \text{ to order interval value functions}$$

We say that any policy π whose interval value function $\hat{V}_\pi$ is $\geq_{\text{opt}}$ ($\geq_{\text{pes}}$) the value functions $\hat{V}_{\pi'}$ of all other policies π' is *optimistically (pessimistically) optimal.*

Theorem 5. *There exists at least one optimistically (pessimistically) optimal policy, and therefore the definition of $\hat{V}_{\text{opt}}$ ($\hat{V}_{\text{pes}}$) is well-formed.*

The above two notions of optimal value can be understood in terms of a game in which we choose a policy π and then a second player chooses in which MDP M in $\mathcal{M}$ to evaluate the policy. The goal is to get the highest[4] resulting value function $V_{M,\pi}$. The optimistic optimal value function's upper bounds $\overline{V}_{\text{opt}}$ represent the best value function we can obtain in this game if we assume the second player is cooperating with us. The pessimistic optimal value function's lower bounds $\underline{V}_{\text{pes}}$ represent the best we can do if we assume the second player is our adversary, trying to minimize the resulting value function.

In the next section, we describe well-known iterative algorithms for computing the exact MDP optimal value function V^*, and then in Section 5 we will describe similar iterative algorithms which compute the BMDP variants $\hat{V}_{\text{opt}}$ $(\hat{V}_{\text{pes}})$.

4 Estimating Traditional Value Functions

In this section, we review the basics concerning dynamic programming methods for computing value functions for fixed and optimal policies in traditional MDPs. In the next section, we describe novel algorithms for computing the interval analogs of these value functions for bounded parameter MDPs.

We present results from the theory of exact MDPs which rely on the concept of normed linear spaces. We define operators, VI_π and VI, on the space of value functions. We then use the Banach fixed-point theorem (Theorem 6) to show that iterating these operators converges to unique fixed-points, V_π and V^* respectively (Theorems 8 and 9).

Let $\mathcal{V}$ denote the set of value functions on $\mathcal{Q}$. For each $v \in \mathcal{V}$, define the (sup) *norm* of v by

$$||v|| = \max_{q \in \mathcal{Q}} |v(q)|.$$

We use the term *convergence* to mean convergence in the norm sense. The space $\mathcal{V}$ together with $||\cdot||$ constitute a complete normed linear space, or *Banach Space*. If U is a Banach space, then an operator $T : U \to U$ is a *contraction mapping* if there exists a λ, $0 \leq \lambda < 1$ such that $||Tv - Tu|| \leq \lambda ||v - u||$ for all u and v in U.

Define $VI : \mathcal{V} \to \mathcal{V}$ and for each $\pi \in \mathit{\Pi}$, $VI_\pi : \mathcal{V} \to \mathcal{V}$ on each $p \in \mathcal{Q}$ by

$$VI(v)(p) = \max_{\alpha \in \mathcal{A}} \left(R(p) + \gamma \sum_{q \in \mathcal{Q}} F_{pq}(\alpha) v(q) \right)$$

$$VI_\pi(v)(p) = R(p) + \gamma \sum_{q \in \mathcal{Q}} F_{pq}(\pi(p)) v(q).$$

In cases where we need to make explicit the MDP from which the transition function F originates, we write $VI_{M,\pi}$ and VI_M to denote the operators VI_π and VI as just defined, except that the transition function F is F^M.

Using these operators, we can rewrite the expression for V^* and V_π as

$$V^*(p) = VI(V^*)(p) \quad \text{and} \quad V_\pi(p) = VI_\pi(V_\pi)(p)$$

[4] Value functions are ranked by $\geq_{\text{dom}}$.

for all states $p \in \mathcal{Q}$. This implies that V^* and V_π are fixed points of VI and VI_π, respectively. The following four theorems show that for each operator, iterating the operator on an initial value estimate converges to these fixed points.

Theorem 6. *For any Banach space U and contraction mapping $T : U \rightarrow U$, there exists a unique v^* in U such that $Tv^* = v^*$; and for arbitrary v^0 in U, the sequence $\{v^n\}$ defined by $v^n = Tv^{n-1} = T^n v^0$ converges to v^*.*

Theorem 7. *VI and VI_π are contraction mappings.*

Theorem 6 and Theorem 7 together prove the following fundamental results in the theory of MDPs.

Theorem 8. *There exists a unique $v^* \in \mathcal{V}$ satisfying $v^* = VI(v^*)$; furthermore, $v^* = V^*$. Similarly, V_π is the unique fixed-point of VI_π.*

Theorem 9. *For arbitrary $v^0 \in \mathcal{V}$, the sequence $\{v^n\}$ defined by $v^n = VI(v^{n-1})$ $= VI^n(v^0)$ converges to V^*. Similarly, iterating VI_π converges to V_π.*

An important consequence of Theorem 9 is that it provides an algorithm for finding V^* and V_π. In particular, to find V^*, we can start from an arbitrary initial value function v^0 in $\mathcal{V}$, and repeatedly apply the operator VI to obtain the sequence $\{v^n\}$. This algorithm is referred to as *value iteration*. Theorem 9 guarantees the convergence of value iteration to the optimal value function. Similarly, we can specify an algorithm called *policy evaluation* which finds V_π by repeatedly apply VI_π starting with an initial $v^0 \in \mathcal{V}$.

The following theorem from [Littman *et al.*, 1995] states a convergence rate of value iteration and policy evaluation which can be derived using bounds on the precision needed to represent solutions to a linear program of limited precision (each algorithm can be viewed as solving a linear program).

Theorem 10. *For fixed γ, value iteration and policy evaluation converge to the optimal value function in a number of steps polynomial in the number of states, the number of actions, and the number of bits used to represent the MDP parameters.*

5 Estimating Interval Value Functions

In this section, we describe dynamic programming algorithms which operate on bounded parameter MDPs. We first define the interval equivalent of policy evaluation $I\hat{V}I_\pi$ which computes $\hat{V}_\pi$, and then define the variants $I\hat{V}I_{opt}$ and $I\hat{V}I_{pes}$ which compute the optimistic and pessimistic optimal value functions.

5.1 Interval Policy Evaluation

In direct analogy to the definition of VI_π in Section 4, we define a function $I\hat{V}I_\pi$ (for *interval value iteration*) which maps interval value functions to other interval value functions. We have proven that iterating $I\hat{V}I_\pi$ on any initial interval value function produces a sequence of interval value functions which converges to $\hat{V}_\pi$ in a polynomial number of steps, given a fixed discount factore γ.

$I\hat{V}I_\pi(\hat{V})$ is an interval value function, defined for each state p as follows:

$$I\hat{V}I_\pi(\hat{V})(p) = \left[\min_{M\in\mathcal{M}} VI_{M,\pi(p)}(\underline{V})(p) \;\; \max_{M\in\mathcal{M}} VI_{M,\pi(p)}(\overline{V})(p)\right].$$

We define $\underline{IVI}_\pi$ and $\overline{IVI}_\pi$ to be the corresponding mappings from value functions to value functions (note that for input $\hat{V}$, $\underline{IVI}_\pi$ does not depend on $\overline{V}$ and so can be viewed as a function from $\mathcal{V}$ to $\mathcal{V}$—likewise for $\overline{IVI}_\pi$ and $\underline{V}$).

The algorithm to compute $I\hat{V}I_\pi$ is very similar to the standard MDP computation of VI, except that we must now be able to select an MDP M from the family $\mathcal{M}$ which minimizes (maximizes) the value attained. We select such an MDP by selecting a function F within the bounds specified by $\hat{F}$ to minimize (maximize) the value—each possible way of selecting F corresponds to one MDP in $\mathcal{M}$. We can select the values of $F_{pq}(\alpha)$ independently for each α and p, but the values selected for different states q (for fixed α and p) interact: they must sum up to one. We now show how to determine, for fixed α and p, the value of $F_{pq}(\alpha)$ for each state q so as to minimize (maximize) the expression $\sum_{q\in\mathcal{Q}}(F_{pq}(\alpha)V(q))$. This step constitutes the heart of the IVI algorithm and the only significant way the algorithm differs from standard value iteration.

The idea is to sort the possible destination states q into increasing (decreasing) order according to their $\underline{V}$ ($\overline{V}$) value, and then choose the transition probabilities within the intervals specified by $\hat{F}$ so as to send as much probability mass to the states early in the ordering. Let $q_1, q_2, \ldots, q_k$ be such an ordering of $\mathcal{Q}$—so that, in the minimizing case, for all i and j if $1 \le i \le j \le k$ then $\underline{V}(q_i) \le \underline{V}(q_j)$ (increasing order).

Let r be the index $1 \le r \le k$ which maximizes the following expression without letting it exceed 1:

$$\sum_{i=1}^{r-1} \overline{F}_{p,q_i}(\alpha) + \sum_{i=r}^{k} \underline{F}_{p,q_i}(\alpha)$$

r is the index into the sequence q_i such that below index r we can assign the upper bound, and above index r we can assign the lower bound, with the rest of the probability mass from p under α being assigned to q_r. Formally, we choose $F_{pq}(\alpha)$ for all $q \in \mathcal{Q}$ as follows:

$$F_{pq_j}(\alpha) = \begin{cases} \overline{F}_{p,q_i}(\alpha) \text{ if } j < r \\ \underline{F}_{p,q_i}(\alpha) \text{ if } j > r \end{cases}$$

$$F_{pq_r}(\alpha) = 1 - \sum_{i=1, i\neq r}^{i=k} F_{pq_i}(\alpha)$$

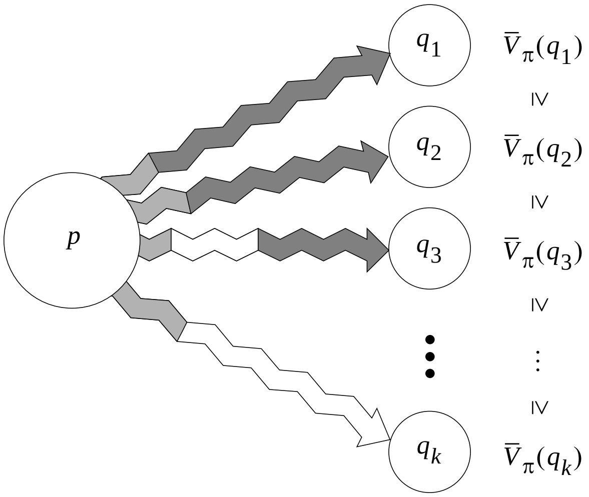

Fig. 2. An illustration of the basic dynamic programming step in computing an approximate value function for a fixed policy and bounded parameter MDP. The lighter shaded portions of each arc represent the required lower bound transition probability and the darker shaded portions represent the fraction of the remaining transition probability to the upper bound assigned to the arc by F.

Figure 2 illustrates the basic iterative step in the above algorithm, for the maximizing case. The states q_i are ordered according to the value estimates in $\overline{V}$. The transitions from a state p to states q_i are defined by the function F such that each transition is equal to its lower bound plus some fraction of the leftover probability mass.

Techniques similar to those in Section 4 can be used to prove that iterating $\underline{IVI}_\pi$ ($\overline{IVI}_\pi$) converges to $\underline{V}_\pi$ ($\overline{V}_\pi$). The key theorems, stated below, assert first that $\underline{IVI}_\pi$ is a contraction mapping, and second that $\underline{V}_\pi$ is a fixed-point of $\underline{IVI}_\pi$, and are easily proven[5].

Theorem 11. *For any policy π, $\underline{IVI}_\pi$ and $\overline{IVI}_\pi$ are contraction mappings.*

Theorem 12. *For any policy π, $\underline{V}_\pi$ is a fixed-point of $\underline{IVI}_\pi$ and $\overline{V}_\pi$ of $\overline{IVI}_\pi$.*

These theorems, together with Theorem 6 (the Banach fixed-point theorem) imply that iterating $I\hat{V}I_\pi$ on any initial interval value function converges to $\hat{V}_\pi$, regardless of the starting point.

Theorem 13. *For fixed γ, interval policy evaluation converges to the desired interval value function in a number of steps polynomial in the number of states, the number of actions, and the number of bits used to represent the MDP parameters.*

[5] The min over members of $\mathcal{M}$ is dealt with using a technique similar to that used to handle the max over actions in the same proof for V^*

5.2 Interval Value Iteration

As in the case of VI_π and VI, it is straightforward to modify $I\hat{V}I_\pi$ so that it computes optimal policy value intervals by adding a maximization step over the different action choices in each state. However, unlike standard value iteration, the quantities being compared in the maximization step are closed real intervals, so the resulting algorithm varies according to how we choose to compare real intervals. We define two variations of interval value iteration—other variations are possible.

$$I\hat{V}I_{opt}(\hat{V})(p) = \max_{\alpha\in\mathcal{A},\ \leq_{\text{opt}}} \left[\min_{M\in\mathcal{M}} VI_{M,\alpha}(\underline{V})(p),\ \max_{M\in\mathcal{M}} VI_{M,\alpha}(\overline{V})(p)\right]$$

$$I\hat{V}I_{pes}(\hat{V})(p) = \max_{\alpha\in\mathcal{A},\ \leq_{\text{pes}}} \left[\min_{M\in\mathcal{M}} VI_{M,\alpha}(\underline{V})(p),\ \max_{M\in\mathcal{M}} VI_{M,\alpha}(\overline{V})(p)\right]$$

The added maximization step introduces no new difficulties in implementing the algorithm. We discuss convergence for $I\hat{V}I_{opt}$—the convergence results for $I\hat{V}I_{pes}$ are similar. We write $\overline{IVI}_{opt}$ for the upper bound returned by $I\hat{V}I_{opt}$, and we consider $\overline{IVI}_{opt}$ a function from $\mathcal{V}$ to $\mathcal{V}$ because $\overline{IVI}_{opt}(\hat{V})$ depends only on $\underline{V}$. $\overline{IVI}_{opt}$ can be easily shown to be a contraction mapping, and it can be shown that $\hat{V}_{\text{opt}}$ is a fixed point of $I\hat{V}I_{opt}$. It then follows that $\overline{IVI}_{opt}$ converges to $\overline{V}_{\text{opt}}$ in polynomially many steps. The analogous results for $\underline{IVI}_{opt}$ are somewhat more problematic. Because the action selection is done according to $\leq_{\text{opt}}$, which focuses primarily on the interval upper bounds, $\underline{IVI}_{opt}$ is not properly a mapping from $\mathcal{V}$ to $\mathcal{V}$, as $\underline{IVI}_{opt}(\hat{V})$ depends on both $\underline{V}$ and $\overline{V}$. However, for any particular value function V and interval value function $\hat{V}$ such that $\overline{V} = V$, we can write $\underline{IVI}_{opt,V}$ for the mapping from $\mathcal{V}$ to $\mathcal{V}$ which carries $\underline{V}$ to $\underline{IVI}_{opt}(\hat{V})$. We can then show that for each V, $\underline{IVI}_{opt,V}$ converges as desired. The algorithm must then iterate $\overline{IVI}_{opt}$ convergence to some upper bound $\overline{V}$, and then iterate $\underline{IVI}_{opt,\overline{V}}$ to converge to the lower bounds $\underline{V}$—each convergence within polynomial time.

Theorem 14. *A. $\overline{IVI}_{opt}$ and $\underline{IVI}_{pes}$ are contraction mappings.*
B. For any value functions V, $\underline{IVI}_{opt,V}$ and $\overline{IVI}_{pes,V}$ are contraction mappings.

Theorem 15. *$\hat{V}_{\text{opt}}$ is a fixed-point of $I\hat{V}I_{opt}$, and $\hat{V}_{\text{pes}}$ of $I\hat{V}I_{pes}$.*

Theorem 16. *For fixed γ, iteration of $I\hat{V}I_{opt}$ converges to $\hat{V}_{\text{opt}}$, and iteration of $I\hat{V}I_{pes}$ converges to $\hat{V}_{\text{pes}}$, in polynomially many iterations in the problem size (including the number of bits used in specifying the parameters).*

6 Policy Selection, Sensitivity Analysis, and Aggregation

In this section, we consider some basic issues concerning the use and interpretation of bounded parameter MDPs. We begin by reemphasizing some ideas introduced earlier regarding the selection of policies.

To begin with, it is important that we are clear on the status of the bounds in a bounded parameter MDP. A bounded parameter MDP specifies upper and lower bounds on individual parameters; the assumption is that we have no additional information regarding individual exact MDPs whose parameters fall with those bounds. In particular, we have no prior over the exact MDPs in the family of MDPs defined by a bounded parameter MDP.

Policy selection Despite the lack of information regarding any particular MDP, we may have to choose a policy. In such a situation, it is natural to consider that the actual MDP, *i.e.,* the one in which we will ultimately have to carry out some policy, is decided by some outside process. That process might choose so as to help or hinder us, or it might be entirely indifferent. To minimize the risk of performing poorly, it is reasonable to think in adversarial terms; we select the policy which will perform as well as possible assuming that the adversary chooses so that we perform as poorly as possible.

These choices correspond to optimistic and pessimistic optimal policies. We have discussed in the last section how to compute interval value functions for such policies—such value functions can then be used in a straightforward manner to extract policies which achieve those values.

There are other possible choices, corresponding in general to other means of totally ordering real closed intervals. We might for instance consider a policy whose average performance over all MDPs in the family is as good as or better than the average performance of any other policy. This notion of average is potentially problematic, however, as it essentially assumes a uniform prior over exact MDPs and, as stated earlier, the bounds do not imply any particular prior.

Sensitivity analysis There are other ways in which bounded parameter MDPs might be useful in planning under uncertainty. For example, we might assume that we begin with a particular exact MDP, say, the MDP with parameters whose values reflect the best guess according to a given domain expert. If we were to compute the optimal policy for this exact MDP, we might wonder about the degree to which this policy is sensitive to the numbers supplied by the expert.

To explore this possible sensitivity to the parameters, we might assess the policy by perturbing the parameters and evaluating the policy with respect to the perturbed MDP. Alternatively, we could use BMDPs to perform this sort of sensitivity analysis on a whole family of MDPs by converting the point estimates for the parameters to confidence intervals and then computing bounds on the value function for the fixed policy via interval policy evaluation.

Aggregation Another use of BMDPs involves a different interpretation altogether. Instead of viewing the states of the bounded parameter MDP as individual primitive states, we view each state of the BMDP as representing a set or *aggregate* of states of some other, larger MDP.

In this interpretation, states are aggregated together because they behave approximately the same with respect to possible state transitions. A little more precisely, suppose that the set of states of the BMDP $\mathcal{M}$ corresponds to the set of *blocks* $\{B_1, \ldots, B_n\}$ such that the $\{B_i\}$ constitutes the partition of another MDP with a much larger state space.

Now we interpret the bounds as follows; for any two blocks B_i and B_j, let $\hat{F}_{B_iB_j}(\alpha)$ represent the interval value for the transition from B_i to B_j on action α defined as follows: $\hat{F}_{B_iB_j}(\alpha) = \left[\min_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha),\ \max_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha)\right]$ Intuitively, this means that all states in a block behave approximately the same (assuming the lower and upper bounds are close to each other) in terms of transitions to other blocks even though they may differ widely with regard to transitions to individual states.

In Dean *et al.* [1997] we discuss methods for using an implicit representation of a exact MDP with a large number of states to construct an explicit BMDP with a possibly much smaller number of states based on an aggregation method. We then show that policies computed for this BMDP can be extended to the original large implicitly described MDP. Note that the original implicit MDP is not even a member of the family of MDPs for the reduced BMDP (it has a different state space, for instance). Nevertheless, it is a theorem that the policies and value bounds of the BMDP can be soundly applied in the original MDP (using the aggregation mapping to connect the state spaces).

7 Related Work and Conclusions

Our definition for bounded parameter MDPs is related to a number of other ideas appearing in the literature on Markov decision processes; in the following, we mention just a few such ideas. First, BMDPs specialize the MDPs with imprecisely known parameters (MDPIPs) described and analyzed in the operations research literature[White and Eldeib, 1994, White and Eldeib, 1986, Satia and Lave, 1973]. The more general MDPIPs described in these papers require more general and expensive algorithms for solution. For example, [White and Eldeib, 1994] allows an arbitrary linear program to define the bounds on the transition probabilities (and allows no imprecision in the reward parameters)—as a result, the solution technique presented appeals to linear programming at each iteration of the solution algorithm rather than exploit the specific structure available in a BMDP. [Satia and Lave, 1973] mention the restriction to BMDPs but give no special algorithms to exploit this restriction. Their general MDPIP algorithm is very different from our algorithm and involves two nested phases of policy iteration—the outer phase selecting a traditional policy and the inner phase selecting a "policy" for "nature", *i.e.,* a choice of the transition parameters to minimize or maximize value (depending on whether optimistic or pessimistic assumptions prevail). Our work, while originally developed independently of the MDPIP literature, follows similar lines to [Satia and Lave, 1973] in defining optimistic and pessimistic optimal policies.

Bertsekas and Castañon [1989] use the notion of aggregated Markov chains and consider grouping together states with approximately the same residuals. Methods for bounding value functions are frequently used in approximate algorithms for solving MDPs; Lovejoy [1991] describes their use in solving partially observable MDPs. Puterman [1994] provides an excellent introduction to Markov decision processes and techniques involving bounding value functions.

Boutilier and Dearden [1994] and Boutilier *et al.* [1995b] describe methods for solving implicitly described MDPs and Dean and Givan [1997] reinterpret this work in terms of computing explicitly described MDPs with aggregate states.

Bounded parameter MDPs allow us to represent uncertainty about or variation in the parameters of a Markov decision process. Interval value functions capture the resulting variation in policy values. In this paper, we have defined both bounded parameter MDP and interval value function, and given algorithms for computing interval value functions, and selecting and evaluating policies.

References

[Bellman, 1957] Bellman, Richard 1957. *Dynamic Programming.* Princeton University Press.

[Bertsekas and Castañon, 1989] Bertsekas, D. P. and Castañon, D. A. 1989. Adaptive aggregation for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control* 34(6):589–598.

[Boutilier and Dearden, 1994] Boutilier, Craig and Dearden, Richard 1994. Using abstractions for decision theoretic planning with time constraints. In *Proceedings AAAI-94.* AAAI. 1016–1022.

[Boutilier *et al.*, 1995a] Boutilier, Craig; Dean, Thomas; and Hanks, Steve 1995a. Planning under uncertainty: Structural assumptions and computational leverage. In *Proceedings of the Third European Workshop on Planning.*

[Boutilier *et al.*, 1995b] Boutilier, Craig; Dearden, Richard; and Goldszmidt, Moises 1995b. Exploiting structure in policy construction. In *Proceedings IJCAI 14.* IJCAII. 1104–1111.

[Dean and Givan, 1997] Dean, Thomas and Givan, Robert 1997. Model minimization in Markov decision processes. In *Proceedings AAAI-97.* AAAI.

[Dean *et al.*, 1997] Dean, Thomas; Givan, Robert; and Leach, Sonia 1997. Model reduction techniques for computing approximately optimal solutions for Markov decision processes. In *Thirteenth Conference on Uncertainty in Artificial Intelligence.*

[Littman *et al.*, 1995] Littman, Michael; Dean, Thomas; and Kaelbling, Leslie 1995. On the complexity of solving Markov decision problems. In *Eleventh Conference on Uncertainty in Artificial Intelligence.* 394–402.

[Littman, 1997] Littman, Michael L. 1997. Probabilistic propositional planning: Representations and complexity. In *Proceedings AAAI-97.* AAAI.

[Lovejoy, 1991] Lovejoy, William S. 1991. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research* 28:47–66.

[Puterman, 1994] Puterman, Martin L. 1994. *Markov Decision Processes.* John Wiley & Sons, New York.

[Satia and Lave, 1973] Satia, J. K. and Lave, R. E. 1973. Markovian decision processes with uncertain transition probabilities. *Operations Research* 21:728–740.

[White and Eldeib, 1986] White, C. C. and Eldeib, H. K. 1986. Parameter imprecision in finite state, finite action dynamic programs. *Operations Research* 34:120–129.

[White and Eldeib, 1994] White, C. C. and Eldeib, H. K. 1994. Markov decision processes with imprecise transition probabilities. *Operations Research* 43:739–749.

This article was processed using the LaTeX macro package with LLNCS style