REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30-06-01	Final Technical	01/07-97 -- 30-06-01

4. TITLE AND SUBTITLE
(FY97 AASERT) REPRESENTING AND SOLVING AIR CAMPAIGN PROBLEMS AS PARTIALLY OBSERVABLE MARKOV DECISION PROBLEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-97-1-0477

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Thomas L. Dean

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Brown University
Providence, RI 02912

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
801 North Randlph St, Rm. 732
Arlington, VA 22203-1977

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Publicly available.

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20011016 063

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The original purpose of this project was to design algorithms and architectures for maintenance and deployment scheduling solutions to support large-scale strategic military airlift activities related to the needs of the Air Mobility Command (AMC), and secondarily to adapt these solutions to other military and civilian planning and scheduling problems. The research adopted a stochastic modeling framework and used novel techniques for planning in unpredictable dynamic environments with complex state and action spaces. Many of the original goals of the project were achieved in conjunction with the primary contract; however, several extensions were carried out by students receiving funding from the AASERT supplementary grant.

15. SUBJECT TERMS
Statistical models, planning, scheduling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thomas L. Dean
UNCLASSIFIED	UNCLASSIFIED				19b. TELEPHONE NUMBER *(Include area code)* 401 863 7600

Final Report

The original purpose of this project was to design algorithms and architectures for maintenance and deployment scheduling solutions to support large-scale strategic military airlift activities related to the needs of the Air Mobility Command (AMC). And secondarily to adapt these solutions to other military and civilian planning and scheduling problems. The research adopted a stochastic-modeling framework and made use of novel techniques for planning in unpredictable, dynamic environments with complex state and action spaces.

Many of the goals associated with the project were achieved in conjunction with the primary contract; however, several extensions were carried out by students receiving funding from the AASERT supplementary grant. Originally, Sonia Leach, a graduate student at Brown, was to be the primary beneficiary of the grant and, indeed, Sonia was funded by this grant for one year. Later Sonia's interest turned to the use of mathematically related models and algorithms that were targeted at problems in genomics and computational biology.

Sonia Leach worked with researchers at the National Cancer Institute in Bethesda, Maryland, applying machine learning techniques to analyze biological data. She began collaborations with researchers at University of Colorado Health Sciences Center in Denver, Colorado to analyze gene expression data. She received NIH funding for this work and so, after consultation with AFOSR, the remainder of the AASERT grant was applied to other students working on projects concerned with stochastic modeling methods.

The stochastic models that were at the heart of the scheduling and planning problems have broad application. Sonia's work in genomics is a good example but there are also related models in statistical natural language processing that are receiving a lot of attention recently. This grant paid supplemental stipends for Niyu Gee, Keith Hall, and Don Blaheta for their work on solutions to language learning problems. Niyu Ge completed her PhD dissertation on pronoun anaphora (finding the referents (or "antecedents") of pronouns) and has now taken a position at IBM research. Keith and Don are working on their dissertations and should finish in the coming year with the rest of their funding coming from NSF.

Luis Ortiz also received funding from this grant and his research is directly related to the combinatorial problems that were the primary focus of this AASERT proposal. Luis developed new sampling methods for solving influence (decision) diagrams - an alternative representation for stochastic planning and scheduling problems. He provided bounds on the number of samples required to select "good" actions with high probability for what was considered the "traditional sampling-method" used. He proposed a new method that requires fewer samples (both on expectations and with high probability) to obtain the same results. Luiz will complete his dissertation and defend in September and has accepted a postdoctoral position at AT&T labs.

I've enclosed copies of the following papers co-authored with Sonia Leach and Luis Ortiz as they are the most relevant to the original goals of the AASERT proposal.

- Thomas Dean, Robert Givan, and Sonia Leach, "Model Reduction Techniques for Computing Approximately Optimal Solutions for Markov Decision Processes," Proceedings of the Conference on Uncertainty in AI, 1997.
- Robert Givan, Sonia Leach, and Thomas Dean, "Bounded Parameter Markov Decision Processes", Proceedings of the European Conference on Planning, 1997.
- Robert Givan, Sonia Leach and Tom Dean, Bounded-parameter Markov Decision Processes, *Artificial Intelligence*, Volume 122, Number 1-2, Pages 71-109, 2000.
- Luis E. Ortiz and Leslie Pack Kaelbling, "Adaptive Importance Sampling for Estimation in Structured Domains," , Proceeding of the Sixteenth Conference on Uncertainty in Artificial Intelligence 2000.
- Luis E. Ortiz and Leslie Pack Kaelbling, "Sampling Methods for Action Selection in Influence Diagrams," Proceedings of the Seventeenth National Conference on Artificial Intelligence, 2000.
- Milos Hauskrecht, Luis Ortiz, Ioannis Tsochantaridis, and Eli Upfal, "Computing Global Strategies for Multi-Market Commodity Trading," Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, 2000.

Model Reduction Techniques for Computing Approximately Optimal Solutions for Markov Decision Processes

Thomas Dean and **Robert Givan** and **Sonia Leach**
Department of Computer Science, Brown University
[tld, rlg, sml]@cs.brown.edu
http://www.cs.brown.edu/people/

Abstract

We present a method for solving implicit (factored) Markov decision processes (MDPs) with very large state spaces. We introduce a property of state space partitions which we call ϵ-homogeneity. Intuitively, an ϵ-homogeneous partition groups together states that behave approximately the same under all or some subset of policies. Borrowing from recent work on model minimization in computer-aided software verification, we present an algorithm that takes a factored representation of an MDP and an $0 \leq \epsilon \leq 1$ and computes a factored ϵ-homogeneous partition of the state space.

This partition defines a family of related MDPs—those MDP's with state space equal to the blocks of the partition, and transition probabilities "approximately" like those of any (original MDP) state in the source block. To formally study such families of MDPs, we introduce the new notion of a "bounded parameter MDP" (BMDP), which is a family of (traditional) MDPs defined by specifying upper and lower bounds on the transition probabilities and rewards. We describe algorithms that operate on BMDPs to find policies that are approximately optimal with respect to the original MDP.

In combination, our method for reducing a large implicit MDP to a possibly much smaller BMDP using an ϵ-homogeneous partition, and our methods for selecting actions in BMDP's constitute a new approach for analyzing large implicit MDP's. Among its advantages, this new approach provides insight into existing algorithms to solving implicit MDPs, provides useful connections to work in automata theory and model minimization, and suggests methods, which involve varying ϵ, to trade time and space (specifically in terms of the size of the corresponding state space) for solution quality.

1 Introduction

Markov decision processes (MDP) provide a formal basis for representing planning problems involving uncertainty [Boutilier *et al.*, 1995a]. There exist algorithms for solving MDPs that are polynomial in the size of the state space [Puterman, 1994]. In this paper, we are interested in MDPs in which the states are specified implicitly using a set of state variables. These MDPs have explicit state spaces which are exponential in the number of state variables, and are typically not amenable to direct solution using traditional methods due to the size of the explicit state space.

It is possible to represent some MDPs using space polylog in the size of the state space by factoring the state-transition distribution and the reward function into sets of smaller functions. Unfortunately, this efficiency in representation need not translate into an efficient means of computing solutions. In some cases, however, dependency information implicit in the factored representation can be used to speed computation of an optimal policy [Boutilier and Dearden, 1994, Boutilier *et al.*, 1995b, Lin and Dean, 1995].

The resulting computational savings can be explained in terms of finding a *homogeneous* partition of the state space—a partition such that states in the same block transition with the same probability to each of the other blocks. Such a partition induces a smaller, explicit MDP whose states are the blocks of the partition; the smaller MDP, or *reduced model* is equivalent to the original MDP in a well defined sense. It is possible to take an MDP in factored form and find its smallest reduced model using a number of "partition splitting" operations polynomial in the size of the resulting model; however, these splitting operations are in general propositional logic operations which are $\mathcal{NP}$-hard and are thus only heuristically effective. The states of the reduced process correspond to groups of states (in the original process) that behave the same under all policies. The original and reduced processes are equivalent in the sense that they yield the same solutions, *i.e.*, the same optimal policies and state values.

The basic idea of computing equivalent reduced pro-

cesses has its origins in automata theory [Hartmanis and Stearns, 1966] and stochastic processes [Kemeny and Snell, 1960] and has surfaced more recently in the work on model checking in computer-aided verification [Burch *et al.*, 1994][Lee and Yannakakis, 1992]. Building on the work of Lee and Yannakakis [1992], we have shown [Dean and Givan, 1997] that several existing algorithms are asymptotically equivalent to first constructing the minimal reduced MDP and then solving this MDP using traditional methods that operate on the flat (unfactored) representations.

The minimal model may be exponentially larger than the original compact MDP. In response to this problem, this paper introduces the concept of an ϵ-homogeneous partition of the state space. This relaxation of the concept of homogeneous partition allows states within the same block to transition with different probabilities to other blocks so long as the different probabilities are within ϵ. For $\epsilon > 0$, there are generally ϵ-homogeneous partitions which are smaller and often much smaller than the smallest homogeneous partition. In this paper we discuss *approximate model reduction*—an algorithm for finding an ϵ-homogeneous partition of a factored MDP which is generally smaller and always no larger than the smallest homogeneous partition.

Any ϵ-homogeneous partition induces a family of explicit MDPs, each with state space equal to the blocks of the partition, and transition probabilities from each block nearly identical to those of the underlying states. To formalize and analyze such families we introduce the new concept of a *bounded parameter MDP* (BMDP)—an MDP in which the transition probabilites and rewards are given not as point values but as closed intervals. In Givan *et al.* [1997], we describe algorithms that operate on BMDPs to produce bounds on value functions and thereby compute approximately optimal policies—we summarize these methods here. The resulting bounds and policies apply to the original implicit MDP. Bounded parameter MDPs generalize traditional (exact) MDPs and are related to constructs found in work on aggregation methods for solving MDPs [Schweitzer, 1984, Schweitzer *et al.*, 1985, Bertsekas and Castañon, 1989]. Although BMDPs are introduced here to represent approximate aggregations, they are interesting in their own right and are discussed in more detail in [Givan *et al.*, 1997], The model reduction algorithms and bounded parameter MDP solution methods can be combined to find approximately optimal solutions to large factored MDPs, varying ϵ to trade time and space for solution quality.

The remainder of this paper is organized as follows. In Section 2, we give an overview of the algorithms and representations in this paper and discuss how they fit together. Section 3 reviews traditional and factored MDPs and describes the generalization to bounded parameter MDPs. Section 4 describes an algorithm for ϵ-reducing an MDP to a (possibly) smaller explicit BMDP (an MDP if $\epsilon = 0$). Section 5 summarizes our methods for policy selection in BMDPs, and addresses the applicability of the selected policies to any MDP which ϵ-reduces to the analyzed BMDP. The remaining sections summarize preliminary experimental results and discuss related work.

2 Overview

Here we survey and relate the basic mathematical objects and operations defined later in this paper. We start with a Markov decision process (MDP) M for which we would like to compute an optimal or near optimal policy. Figure 1.a depicts the MDP M as a directed graph corresponding to the state-transition diagram, and its optimal policy π^*_M as found by traditional value iteration.

We assume that the state space for M (and hence the state-transition graph) is quite large. We therefore assume that the states of M are encoded in terms of state variables which represent aspects of the state; an assignment of values to all of the state variables constitutes a complete description of a state. In this paper, we assume that the factored representation is in the form of a Bayesian network, such as that depicted in Figure 1.b with four state variables $\{A, B, C, D\}$.

We speak about operations involving M, but in practice all operations will be performed symbolically using the factored representation: we manipulate sets of states represented as formulas involving the state variables.

Figure 1.c and Figure 1.d depict the unique smallest homogeneous partition of the state space of M, where the blocks are represented (respectively) implicitly and explicitly. The process of finding this partition is called (exact) model minimization. Factored model minimization involves manipulating boolean formulas and is $\mathcal{NP}$-hard, but heuristic manipulation may rarely achieve this worst case.

The smallest homogeneous partition may be exponentially large, so we seek further reduction (at a cost of only approximately optimal solutions) by finding a smaller ϵ-homogeneous partition, depicted in Figure 1.e and Figure 1.f where the blocks are again represented (respectively) implicitly and explicitly.

Any ϵ-homogeneous partition can be used to create a bounded parameter MDP, shown in Figure 1.g and notated as $\mathcal{M}$ —to do this, we treat the partition blocks as (aggregate) states and summarize everything that we know about transitions between blocks in terms of closed real intervals that describe the variation within a block of the transition probabilities to other blocks, *i.e.*, for any action and pair of blocks, we record the upper and lower bounds on the probability of starting in a state in one block and ending up in the other block.[1]

[1] The BMDP $\mathcal{M}$ naturally represents a family of MDPs,

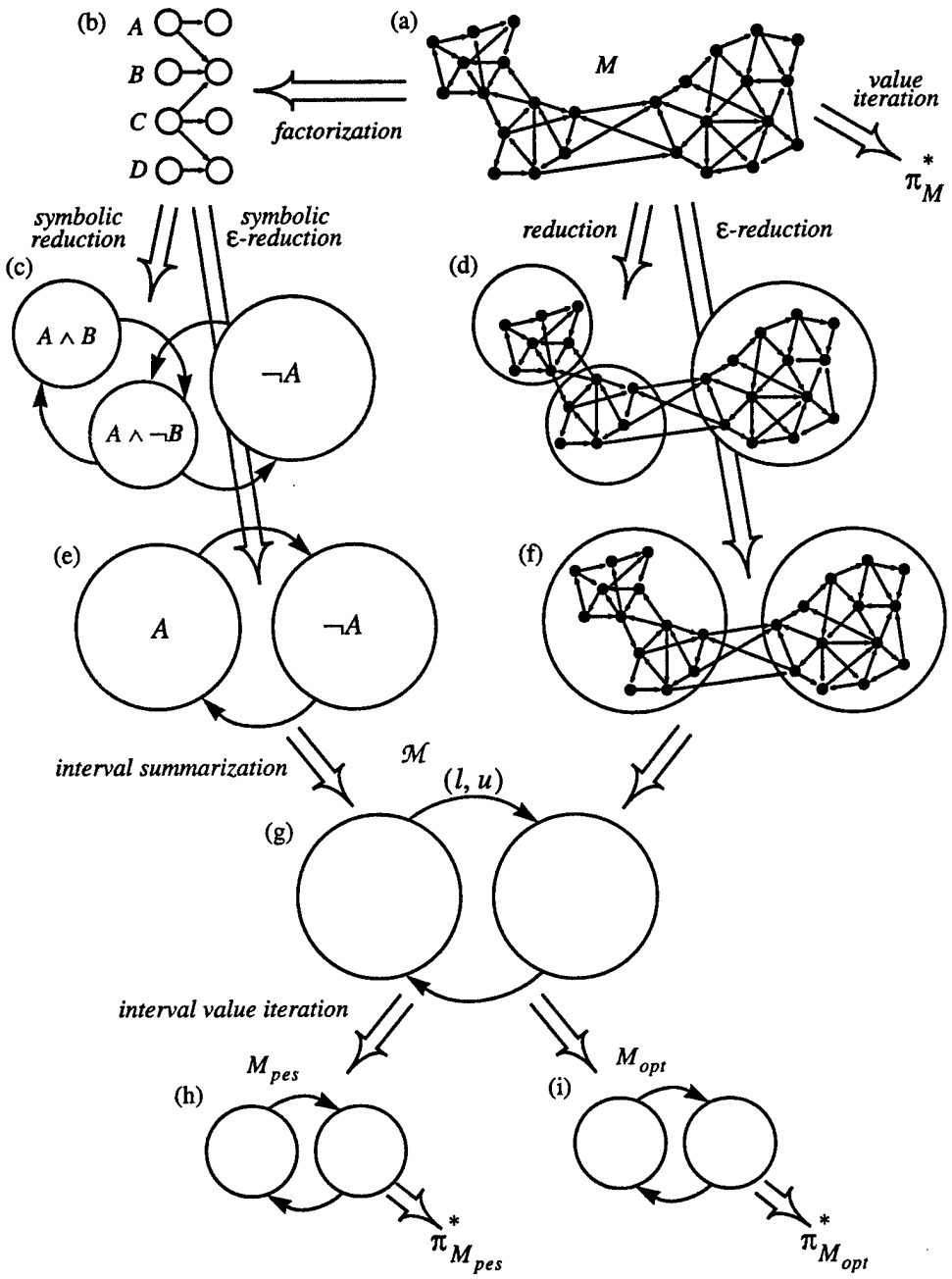


Figure 1: The basic objects and operations described in this paper: (a) depicts the state-transition diagram for an MDP M (only a single action is shown), (b) depicts a Bayesian network as an example of a symbolic representation compactly encoding M, (c) and (d) depict the smallest homogeneous partition in (respectively) its implicit (symbolic) and explicit forms, similarly, (e) and (f) depict an ϵ-homogeneous partition in its implicit and explicit forms, (g) represents the bounded-parameter MDP $\mathcal{M}$ summarizing the variations in the ϵ-homogeneous partition, and, finally, (h), (i), and (j) depict particular (exact) MDPs from the family of MDPs defined by $\mathcal{M}$.

Our BMDP analysis algorithms extract particular MDPs from $\mathcal{M}$ that have intuitive characterizations. The *pessimistic model* M_{pes} is the MDP within $\mathcal{M}$ which yields the lowest optimal value $V^*_{M_{pes}}$ at every state. It is a theorem that M_{pes} is well-defined, and that $V^*_{M_{pes}}$ at each state in $\mathcal{M}$ is a lower bound for following the optimal policy $\pi^*_{M_{pes}}$ in any MDP in $\mathcal{M}$ (as well as in the original M from any state in the corresponding block). Similarly, the *optimistic model* M_{opt} has the best value function $V_{M_{opt}}$. $V_{M_{opt}}$ gives upper-bounds for following any policy in M. In summary, $V^*_{M_{pes}}$ and $V^*_{M_{opt}}$ give us lower and upper bounds on the optimal value function we are really interested in, V^*_M, and following $\pi^*_{M_{pes}}$ in M is guaranteed to achieve at least the lower bound.

Now, armed with this high-level overview to serve as a road map, we descend into the details.

3 Markov Decision Processes

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 $R(q)$,[2] F assigns a probability to each state transition for each action, so that for $\alpha \in \mathcal{A}$ 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.

A *policy* is a mapping from states to actions, $\pi : \mathcal{Q} \rightarrow \mathcal{A}$. The *value function* $V_{\pi,M}$ for a given policy maps states to their expected discounted cumulative reward given that you start in that state and act according the given policy:

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

where γ is the *discount rate*, $0 \leq \gamma < 1$. [Puterman, 1994].

Bounded Parameter MDPs A *bounded parameter MDP* (BMDP) is a four tuple $\mathcal{M} = (\mathcal{Q}, \mathcal{A}, \hat{F}, \hat{R})$ where $\mathcal{Q}$ and $\mathcal{A}$ are 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 $0 \leq l \leq u \leq 1$. For convenience, we define $\underline{F}$ and $\overline{F}$ to be real valued functions which give the lower and upper bounds of the intervals; likewise for $\underline{R}$ and $\overline{R}$. [3] To ensure that $\hat{F}$ admits well-formed transition functions, we require that, for any action α and state p, $\sum_{q \in \mathcal{Q}} \underline{F}_{p,q}(\alpha) \leq 1 \leq \sum_{q \in \mathcal{Q}} \overline{F}_{p,q}(\alpha)$.

A BMDP $\mathcal{M} = (\mathcal{Q}, \mathcal{A}, \hat{F}, \hat{R})$ defines a set of exact MDPs $\mathcal{F}_{\mathcal{M}} = \{M | \mathcal{M} \models M\}$ where $\mathcal{M} \models M$ iff $M = (\mathcal{Q}, \mathcal{A}, F, R)$ and F and R satisfy the bounds provided by $\hat{F}$ and $\hat{R}$ respectively. We will write of *bounding the (optimal or policy specific) value* of a state in a BMDP—by this we mean providing an upper or lower bound on the corresponding state value over the entire family of MDPs $\mathcal{F}_{\mathcal{M}}$. For a more thorough treatment of BMDPs, please see [Givan *et al.*, 1997].

Factored Representations In the remainder of this paper, we make use of *Bayesian networks* [Pearl, 1988] to encode implicit (or *factored*) representations; however, our methods apply to other factored representations such as probabilistic STRIPS operators [Kushmerick *et al.*, 1995]. Let $\mathcal{X} = \{X_1, \ldots, X_m\}$ be a set of state variables. We assume the variables are boolean, and refer to them also as *fluents*. We represent the state at time t as a vector $X_t = \langle X_{1,t}, \ldots, X_{m,t} \rangle$ where $X_{i,t}$ denotes the value of the ith state variable at time t.

The state transition probabilities can be represented using Bayes networks. A *two-stage temporal Bayesian network* (2TBN) is a directed acyclic graph consisting of two sets of variables $\{X_{i,t}\}$ and $\{X_{i,t+1}\}$ in which directed arcs indicating dependence are allowed from the variables in the first set to variables in the second set and between variables in the second set.[Dean and Kanazawa, 1989] The state-transition probabilities are now factored as

$$\Pr(X_{t+1} | X_t, U_t) = \prod_{i=1}^{m} \Pr(X_{i,t+1} | \text{Parents}(X_{i,t+1}), U_t)$$

where $\text{Parents}(X)$ denotes the parents of X in the 2TBN and each of the conditional probability distributions $\Pr(X_{i,t+1} | \text{Parents}(X_{i,t+1}), U_t)$ can be represented as a conditional probability table or as a decision tree—we choose the latter in this paper following [Boutilier *et al.*, 1995b]. We enhance the 2TBN representation to include actions and reward functions; the resulting graph is called an *influence diagram* [Howard and Matheson, 1984].

Figure 2 illustrates a factored representation with three state variables, $\mathcal{X} = \{P, Q, S\}$, and describes the transition probabilities and rewards for a particular action. The factored form of the transition probabilities

but note that the original M is not generally in this family. Nevertheless, our BMDP algorithms compute policies and value bounds which can be soundly applied to the original M.

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

[3] To simplify the remainder of the paper, we assume that the reward bounds are always tight, *i.e.*, that $\underline{R} = \overline{R}$. The generalization to nontrivial bounds on rewards is straightforward.

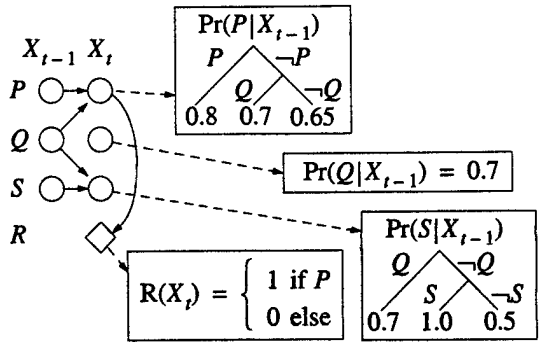


Figure 2: A factored representation with three state variables, P, Q and S, and reward function R.

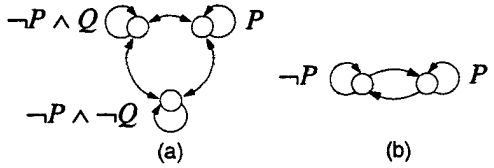


Figure 3: Two ϵ-homogeneous partitions for the MDP described in Figure 2: (a) the smallest exact homogeneous partition ($\epsilon = 0$) and (b) a smaller partition for $\epsilon = 0.05$.

is

$$\Pr(X_{t+1}|X_t, U_t) = \Pr(P_{t+1}|P_t, Q_t) \cdot \Pr(Q_{t+1}) \cdot \Pr(S_{t+1}|S_t, Q_t)$$

where in this case $X_t = \langle P_t, Q_t, S_t \rangle$.

4 Model Reduction Methods

In this section, we describe a family of algorithms that take as input an MDP and a real value ϵ between 0 and 1 and compute a bounded parameter MDP where each closed real interval has extent less than or equal to ϵ. The states in this BMDP correspond to the blocks of a partition of the state space in which states in the same block behave *approximately* the same with respect to the other blocks. The upper and lower bounds in the BMDP correspond to bounds on the transition probabilities (to other blocks) for states that are grouped together.

We first define the property sought in the desired state space partition. Let $\mathcal{P} = \{B_1, \ldots, B_n\}$ be a partition of $\mathcal{Q}$.

Definition 1 *A partition $\mathcal{P} = \{B_1, \ldots, B_n\}$ of the state space of an MDP M has the property of* ϵ-approximate stochastic bisimulation homogeneity *with respect to M for ϵ such that $0 \leq \epsilon \leq 1$ if and only if for each $B_i, B_j \in \mathcal{P}$, for each $\alpha \in \mathcal{A}$, for each $p, q \in B_i$,*

$$|R(p) - R(q)| \leq \epsilon, \quad \text{and}$$

$$\left| \sum_{r \in B_j} F_{pr}(\alpha) - \sum_{r \in B_j} F_{qr}(\alpha) \right| \leq \epsilon$$

For conciseness, we say $\mathcal{P}$ is ϵ-homogeneous.[4]

Figure 3 shows two ϵ-homogeneous partitions for the MDP described in Figure 2.

We now explain how we construct an ϵ-homogeneous partition. We first describe the relationship between every ϵ-homogeneous partition and a particular simple partition based on immediate reward.

Definition 2 *A partition $\mathcal{P}'$ is a* refinement *of a partition $\mathcal{P}$ if and only if each block of $\mathcal{P}'$ is a subset of some block of $\mathcal{P}$; in this case, we say that $\mathcal{P}$ is* coarser *than $\mathcal{P}'$, and is a* clustering *of $\mathcal{P}'$*

Definition 3 *The* immediate reward partition *is the partition in which two states, p and q, are in the same block if and only if they have the same reward.*

Definition 4 *A partition $\mathcal{P}$ is* ϵ-uniform *with respect to a function $f : \mathcal{Q} \to \mathcal{R}$ if for every two states p and q in the same block of $\mathcal{P}$, $|f(p) - f(q)| \leq \epsilon$.*

Every ϵ-homogeneous partition is a refinement of some ϵ-uniform clustering (with respect to reward) of the immediate reward partition. Our algorithm starts by constructing an ϵ-uniform reward clustering $\mathcal{P}_0$ of the immediate reward partition.[5] We then refine this initial partition by splitting[6] blocks repeatedly to achieve ϵ-homogeneity. We can decide which blocks are candidates for splitting using the following local property of the blocks of an ϵ-homogenous partition:

Definition 5 *We say that a block C of a partition $\mathcal{P}$ is* ϵ-stable with respect to a block B *iff for all actions α and all states $p \in C$ and $q \in C$ we have*

$$\left| \sum_{r \in B} F_{pr}(\alpha) - \sum_{r \in B} F_{qr}(\alpha) \right| \leq \epsilon$$

We say that C is ϵ-stable *if C is ϵ-stable with respect to every block of $\mathcal{P}$ and action in $\mathcal{A}$.*

The definitions immediately imply that a partition is ϵ-homogenous iff every block in the partition is ϵ-stable.

The *model ϵ-reduction algorithm* simply checks each block for ϵ-stability, splitting unstable blocks until quiescence, *i.e.*, until there are no unstable blocks left to split. Specifically, when a block C is found to be unstable with respect to a block B, we replace C in the partition by a set[7] of sub-blocks $C_1, \ldots, C_k$ such that each

[4] For the case of $\epsilon = 0$, ϵ-approximate stochastic bisimulation homogeneity is closely related to the *substitution property* for finite automata developed by Hartmanis and Stearns [1966] and the notion of *lumpability* for Markov chains [Kemeny and Snell, 1960].

[5] There may be many such clusterings, we currently choose a coarsest one arbitrarily.

[6] The term *splitting* refers to the process whereby a block of a partition is divided into two or more sub-blocks to obtain a refinement of the original partition.

[7] There may be more than one choice, as discussed below.

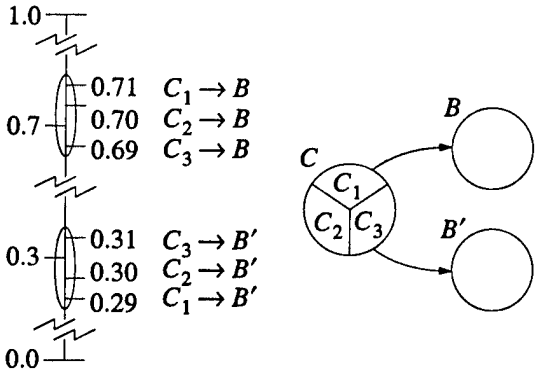


Figure 4: Clustering sub blocks that behave approximately the same. With $\epsilon = 0.01$ there are two smallest clusterings.

C_i is a maximal sub-block of C that is ϵ-stable with respect to B. *Note that at all times the blocks of the partition are represented in factored form,* e.g., *as DNF formulas over the state variables. The block splitting operation manipulates these factored representations, not explicit states.* This method is an extension to Markov decision processes of the deterministic model reduction algorithm of Lee and Yannakakis [1992].

If $\epsilon = 0$, the above description fully defines the block splitting operation, as there exists a unique set of maximal, stable sub-blocks. Furthermore, in this case, the algorithm finds the unique smallest homogeneous partition, independent of the order in which unstable blocks are split. We call this partition the *minimal model* (we also use this term to refer to the MDP derived from this partition by treating its blocks as states).

However, if $\epsilon > 0$, then we may have to choose among several possible ways of splitting C as shown in the following example. Figure 4 depicts a block, C, and two other blocks, B and B', such that states in C transition to states in B and B' under some action α. We partition C into three sub blocks $\{C_1, C_2, C_3\}$ such that states in each sub block have the same transition probabilities with respect to α, B, and B'. In building an 0.01-approximate model, we might replace C by the two blocks C_1 and $C_2 \cup C_3$, or by the two blocks C_3 and $C_1 \cup C_2$; it is possible to construct examples in which each of these is the most appropriate choice because the splits of other blocks induced later[8]. We require only that the clustering selected is not the refinement of another ϵ-uniform clustering, *i.e.*, that it is as coarse as possible.

Because we make the clustering decisions arbitrarily, our algorithm does not guarantee finding the smallest ϵ-homogenous partition when $\epsilon > 0$, nor that the partition found for ϵ_1 will be smaller (or even as small) as the partition found for $\epsilon_2 < \epsilon_1$. However, it is a theorem that the partition found will be no larger than the unique smallest 0-homogenous partition.

Theorem 1 *For $\epsilon > 0$, the partition found by model ϵ-reduction using any clustering technique is coarser than, and thus no larger than the minimal model.*

Theorem 2 *For $0 < \epsilon_2 < \epsilon_1$, the smallest ϵ_1-homogenous partition is no larger than the smallest ϵ_2-homogenous partition. The model ϵ-reduction algorithm, augmented by an (impractical) search over all clustering decisions, will find these smallest partitions.*

Theorem 3 *Given a bound and an MDP whose smallest ϵ-homogenous partition is polynomial in size, the problem of determining whether there exists an ϵ-homogenous partition of size no more than the bound is NP-complete.*

These theorems imply that using an $\epsilon > 0$ can only help us, but that our methods may be sensitive to just which ϵ we choose, and are necessarily heuristic.

Currently our implementation uses a greedy clustering algorithm; in the future we hope to incorporate more sophisticated techniques from the learning and pattern recognition literature to find a smaller clustering locally within each SPLIT operation (though this does not *guarantee* a smaller final partition).

Each ϵ-homogenous partition $\mathcal{P}$ of an MDP $M = (Q, \mathcal{A}, F, R)$ induces a corresponding BMDP $\mathcal{M}_\mathcal{P} = (Q, \mathcal{A}, \hat{F}, \hat{R})$ in a straightforward manner. The states of $\mathcal{M}_\mathcal{P}$ are just the blocks of $\mathcal{P}$ and the actions are the same as those in M. The reward and transition functions are defined to give intervals bounding the possible reward and block transition probabilities within each block: for blocks B and C and action α,

$$\hat{R}(B) = [\ \min_{p \in B} R(p),\ \max_{p \in B} R(p)\ ]$$

$$\hat{F}_{B,C}(\alpha) = [\ \min_{p \in B} \sum_{q \in C} F_{p,q}(\alpha),\ \max_{p \in B} \sum_{q \in C} F_{p,q}(\alpha)\ ]$$

We can then use the methods in the next section to give intervals bounding the optimal value of each state in $\mathcal{M}_\mathcal{P}$ and select a policy which guarantees achieving at least the lower bound value at each state. The following theorem then implies the value bounds apply to the states in M, and are achieved or exceeded by following the corresponding policy in M.

We first note that any function on the blocks of $\mathcal{P}$ can be extended to a function on the states of M: for each state we return the value assigned to the block of $\mathcal{P}$ in which it falls. In this manner, we can interpret the value bounds and policies for $\mathcal{M}_\mathcal{P}$ as bounds and policies for M.

Theorem 4 *For any MDP M and ϵ-homogenous partition $\mathcal{P}$ of the states of M, sound (optimal or policy*

[8]The result is additionally sensitive to the order in which unstable blocks are split—splitting one ϵ-unstable block may make another become ϵ-stable.

specific) value bounds for $\mathcal{M}_{\mathcal{P}}$ apply also to M (by extending the policy and value functions to the state space of M according to $\mathcal{P}$).

5 Interval Value Iteration

We have developed a variant of the value iteration algorithm for computing the optimal policy for exact MDPs[Bellman, 1957] that operates on bounded parameter MDPs. A BMDP $\mathcal{M}$ represents a family of MDPs $\mathcal{F}_{\mathcal{M}}$, implying some degree of uncertainty as to which MDP in the family actions will actually be taken in. As such, there is no specific value for following a policy from a start state—rather, there is a window of possible values for following the policy in the different MDPs of the family. Similarly, for each state there is a window of possible optimal values over the MDPs in the family $\mathcal{F}_{\mathcal{M}}$. Our algorithm can compute bounds on policy specific value functions as well as bounds on the optimal value function. We have also shown how to extract from these bounds a specific "optimal" policy which is guaranteed to achieve at least the lower bound value in any actual MDP from the family $\mathcal{F}_{\mathcal{M}}$ defined by the BMDP. We call this policy $\pi_{\mathbf{pes}}$, the *pessimistic optimal policy.*

We call this algorithm, *interval value iteration* (*IVI* for optimal values, and IVI_π for policy specific values). The algorithm is based on the fact that, if we only knew the rank ordering of the states' values, we would easily be able to select an MDP from the family $\mathcal{F}_{\mathcal{M}}$ which minimized or maximized those values, and then compute the values using that MDP. Since we don't know the rank ordering of states' values, the algorithm uses the ordering of the current estimates of the values to select a minimizing (maximizing) MDP from the family, and performs one iteration of standard value iteration on that MDP to get new value estimates. These new estimates can then be used to select a new minimizing (maximizing) MDP for the next iteration, and so forth.

Bounded parameter MDPs are interesting objects and we explore them at greater length in [Givan *et al.*, 1997]. In that paper, we prove the following results about *IVI*.

Theorem 5 *Given a BMDP $\mathcal{M}$ and a specific policy π, IVI_π converges at each state to lower and upper bounds on the value of π at that state over all the MDPs in $\mathcal{F}_{\mathcal{M}}$.*

Theorem 6 *Given a BMDP $\mathcal{M}$, IVI converges at each state to lower and upper bounds on the optimal value of that state over all the MDPs in $\mathcal{F}_{\mathcal{M}}$.*

Theorem 7 *Given a BMDP $\mathcal{M}$, the policy π_{pes} extracted by assuming that states actual values are the IVI-converged lower bounds has a policy specific lower bound (from IVI_π) in $\mathcal{M}$ equal to the (non policy specific) IVI-converged lower bound. No other policy has a higher policy specific lower bound.*

6 Related Work and Discussion

This paper combines a number of techniques to address the problem of solving (factored) MDPs with very large states spaces. The definition of ϵ-homogeneity and the model reduction algorithms for finding ϵ-homogeneous partitions are new, but draw on techniques from automata theory and symbolic model checking. Burch *et al.* [1994] is the standard reference on symbolic model checking for computer-aided design. Our reduction algorithm and its analysis were motivated by the work of Lee and Yannakakis [1992] and Bouajjani *et al.* [1992].

The notion of bounded-parameter MDP is also new, but is related to aggregation techniques used to speed convergence in iterative algorithms for solving exact MDPs. Bertsekas and Castañon [1989] use the notion of aggregated Markov chains and consider grouping together states with approximately the same residuals (*i.e.*, difference in the estimated value function from one iteration to the next during value iteration).

The methods for manipulating factored representations of MDPs were largely borrowed from Boutilier *et al.* [1995b], which provides an iterative algorithm for finding optimal solutions to factored MDPs. Dean and Givan [1997] describe a model-minimization algorithm for solving factored MDPs which is asymptotically equivalent to the algorithm in [Boutilier *et al.*, 1995b].

Boutilier and Dearden [?] extend the work in [Boutilier *et al.*, 1995b] to compute approximate solutions to factored MDPs by associating upper and lower bounds with symbolically represented blocks of states. States are aggregated if they have approximately the same value rather than if they behave approximately the same behavior under all or some set of policies, though it often turns out that states with nearly the same value have nearly the same dynamics.

There are two significant differences between our approximation techniques and those of Boutilier and Dearden. First, we partition the state space and then perform interval value iteration on the resulting bounded-parameter MDP, while Boutilier and Dearden repeatedly partition the state space. Second, we use a fixed ϵ for computing a partition while Boutilier and Dearden, like Bertsekas and Castañon, repartition the state space (if necessary) on each iteration on the basis of the current residuals, and, hence, (effectively) they use different ϵ's at different times and on different portions of the state space. Despite these differences, we conjecture that the two algorithms perform asymptotically the same. Practically speaking, we expect that in some cases, repeatedly and adaptively computing partitions may provide better performance, while in other cases, performing the partition once and for all may result in a computational advantage.

We have written a prototype implementation of the model reduction algorithms described in this paper, along with the BMDP evaluation algorithms (IVI) referred to. Using this implementation we have been able to demonstrate substantial reductions in model size, and increasing reductions with increasing ϵ. However, the MDPs we have been reducing are still "toy" problems and while they were not concocted expressly to make the algorithm look good, these empirical results are still of questionable value. Further research is necessary before these techniques are adequate to handle a real-world large scale planning problem in order to give convincing empirical data.

Finally, we believe that by formalizing the notions of approximately similar behavior, approximately equivalent models, and families of closely related MDPs the mathematical entities corresponding to ϵ-homogeneous partitions, ϵ-reductions, and bounded-parameter MDPs provide valuable insight into factored MDPs and the prospects for solving them efficiently.

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.

[Bouajjani *et al.*, 1992] Bouajjani, A.; Fernandez, J.-C.; Halbwachs, N.; Raymond, P.; and Ratel, C. 1992. Minimal state graph generation. *Science of Computer Programming* 18:247–269.

[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.

[Burch *et al.*, 1994] Burch, Jerry; Clarke, Edmund M.; Long, David; McMillan, Kenneth L.; and Dill, David L. 1994. Symbolic model checking for sequential circuit verification. *IEEE Transactions on Computer Aided Design* 13(4):401–424.

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

[Dean and Kanazawa, 1989] Dean, Thomas and Kanazawa, Keiji 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.

[Dean *et al.*, 1995] Dean, Thomas; Kaelbling, Leslie; Kirman, Jak; and Nicholson, Ann 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1-2):35–74.

[Givan *et al.*, 1997] Givan, Robert; Leach, Sonia; and Dean, Thomas 1997. Bounded parameter markov decision processes. Technical Report CS-97-05, Brown University, Providence, Rhode Island.

[Hartmanis and Stearns, 1966] Hartmanis, J. and Stearns, R. E. 1966. *Algebraic Structure Theory of Sequential Machines.* Prentice-Hall, Englewood Cliffs, N.J.

[Howard and Matheson, 1984] Howard, Ronald A. and Matheson, James E. 1984. Influence diagrams. In Howard, Ronald A. and Matheson, James E., editors 1984, *The Principles and Applications of Decision Analysis.* Strategic Decisions Group, Menlo Park, CA 94025.

[Howard, 1960] Howard, Ronald A. 1960. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, Massachusetts.

[Kemeny and Snell, 1960] Kemeny, J. G. and Snell, J. L. 1960. *Finite Markov Chains.* D. Van Nostrand, New York.

[Kushmerick *et al.*, 1995] Kushmerick, Nicholas; Hanks, Steve; and Weld, Daniel 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1-2).

[Lee and Yannakakis, 1992] Lee, David and Yannakakis, Mihalis 1992. Online minimization of transition systems. In *Proceedings of 24th Annual ACM Symposium on the Theory of Computing.*

[Lin and Dean, 1995] Lin, Shieu-Hong and Dean, Thomas 1995. Generating optimal policies for high-level plans with conditional branches and loops. In *Proceedings of the Third European Workshop on Planning.* 205–218.

[Pearl, 1988] Pearl, Judea 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, California.

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

[Schweitzer *et al.*, 1985] Schweitzer, Paul J.; Puterman, Martin L.; and Kindle, Kyle W. 1985. Iterative aggregation-disaggregation procedures for discounted semi-Markov reward processes. *Operations Research* 33(3):589–605.

[Schweitzer, 1984] Schweitzer, Paul J. 1984. Aggregation methods for large Markov chains. In Iazola, G.; Coutois, P. J.; and Hordijk, A., editors 1984, *Mathemaical Computer Performance and Reliability.* Elsevier, Amsterdam, Holland. 275–302.

JAIR 2000

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 a 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 $M = (Q, \mathcal{A}, F, R)$ where 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, $\pi : \mathcal{Q} \to \mathcal{A}$. 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 $\geq_{\text{dom}}$ defined as follows: $V_1 \geq_{\text{dom}} V_2$ 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.

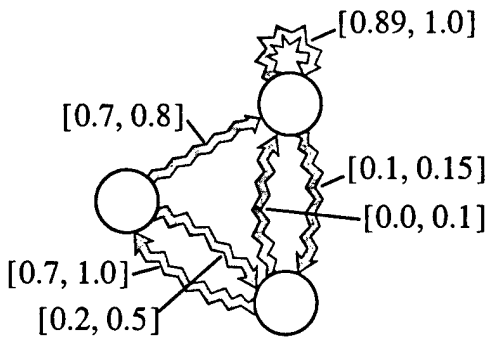


Fig. 1. The state-transition diagram for a simple bounded parameter Markov decision process with three states and a single action. The arcs indicate possible transitions and are labeled by their lower and upper bounds.

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 $\pi \in \Pi$, $VI_\pi : \mathcal{V} \to \mathcal{V}$ 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 $T : U \to U$, 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 $\hat{IVI}_\pi$ (for *interval value iteration*) which maps interval value functions to other interval value functions. We have proven that iterating $\hat{IVI}_\pi$ 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.

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

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

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 $\hat{IVI}_\pi$ 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 $1 \leq i \leq j \leq k$ then $\underline{V}(q_i) \leq \underline{V}(q_j)$ (increasing order).

Let r be the index $1 \leq r \leq k$ 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:

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

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

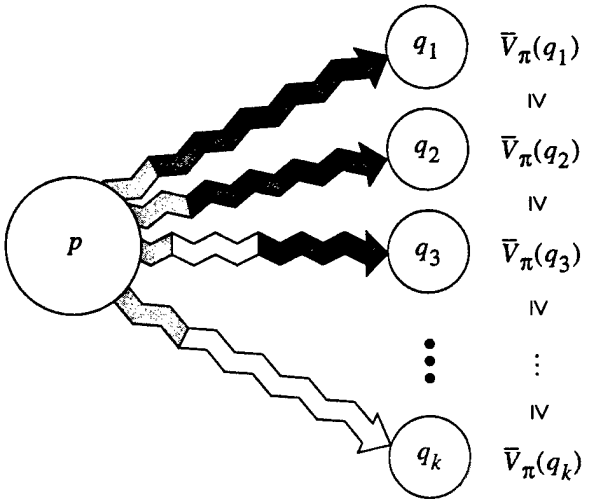


Fig. 2. An illustration of the basic dynamic programming step in computing an approximate value function for a fixed policy and bounded parameter MDP. The lighter shaded portions of each arc represent the required lower bound transition probability and the darker shaded portions represent the fraction 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 $\hat{IVI}_\pi$ 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 $\hat{F}_{B_i,B_j}(a)$ represent the interval value for the transition from B_i to B_j on action a defined as follows: $\hat{F}_{B_i,B_j}(a) = \left[\min_{p \in B_i} \sum_{q \in B_j} F_{pq}(a),\ \max_{p \in B_i} \sum_{q \in B_j} F_{pq}(a)\right]$ 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

tld@klee

/u/tld/tmp/MO3B8D30D41E26DA4.ps

2001-08-29-14:13:52

hplj2 / HP LaserJet 4050 Series

Bounded-parameter Markov Decision Processes

Robert Givan and **Sonia Leach** and **Thomas Dean**

Dept. of Electrical & Computer Engineering
Purdue University
1285 EE Building
West Lafayette, IN 47907
Phone: (765)494-9068
Email: givan@ecn.purdue.edu
Web: http://www.ece.purdue.edu/~givan

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

Abstract

In this paper, we introduce the notion of a *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* that 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* criteria for optimality, and provide a variant of value iteration [1] that we call *interval value iteration* that computes policies for a BMDP that are optimal with respect to these criteria. We show that each algorithm we present converges to the desired values in a polynomial number of iterations given a fixed discount factor.

Keywords: Decision-theoretic planning, Planning under uncertainty, Approximate planning, Markov decision processes.

1. Introduction

The theory of Markov decision processes (MDPs) [11][14][2][10][1] provides the semantic foundations for a wide range of problems involving planning under uncertainty [5][7]. Most work in the planning subarea of artificial intelligence addresses problems that can be formalized using MDP models — however, it is often the case that such models are exponentially larger than the original "intensional" problem representation used in AI work. This paper generalizes the theory of MDPs in a manner that is useful for more compactly representing AI problems

as MDPs via state-space aggregation, as we discuss below.

In this paper, we introduce a generalization of Markov decision processes called *bounded-parameter Markov decision processes* (BMDPs) that allows us to model uncertainty about 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 that 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.

Aggregation of states in very large state spaces was our original motivation for developing BMDPs. Substantial effort has been devoted in recent years within the AI community [9][6][8] to the problem of representing and reasoning with MDP problems where the state space is not explicitly listed but rather implicitly specified with a *factored representation*. In such problems, an explicit listing of the possible system states is exponentially longer than the more natural implicit problem description, and such an explicit list is often intractable to work with. Most planning problems of interest to AI researchers fit this description in that they are only representable in reasonable space using implicit representations. Recent work in applying MDPs to such problems (*e.g.*, [9], [6], and [8]) has considered state-space aggregation techniques as a means of dealing with this problem: rather than work with the possible system states explicitly, aggregation techniques work with blocks of similar or identically-behaving states. When aggregating states that have similar but not identical behavior, the question immediately arises of what transition probability holds between the aggregates: this probability will depend on which underlying state is in control, but this choice of underlying state is not modelled in the aggregate model. This work can be viewed as providing a means of addressing this problem by allowing intervals rather than point values for the aggregate transition probabilities: the interval can be chosen to include the true value for each of the underlying states present in the aggregates involved. It should be noted that under these circumstances, deriving a prior probability distribution over the true parameter values is often as expensive as simply avoiding the aggregation altogether and would defeat the purpose entirely. Moreover, assuming any particular probability

distribution could produce arbitrarily inaccurate results. As a result, this work considers parameters falling into intervals with no prior probability distribution specified over the possible parameter values in the intervals, and seeks to put bounds on how badly or how well particular plans will perform in such a context, as well as to provide means to find optimal plans under optimistic or pessimistic assumptions about the true distribution over parameter values. In Section 6, we discuss the application of our BMDP approach to state-space aggregation problems more formally. Also, in a related paper, we have shown how BMDPs can be used as part of an state-space aggregation strategy for efficiently approximating the solution of MDPs with very large state spaces and dynamics compactly encoded in a factored (or implicit) representation [10].

We also discuss later in this paper the potential use of BMDP methods to evaluate the sensitivity of the optimal policy in an exact MDP to small variations in the parameter values defining the MDP — using BMDP policy selection algorithms on a BMDP whose parameter intervals represent small variations (perhaps confidence intervals) around the exact MDP parameter values, the best and worst variation in policy value achieved can be measured.

In this paper we introduce and discuss BMDPs, the BMDP analog of value functions, called *interval value functions*, and policy selection and evaluation methods for BMDPs. 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 a 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.

Our interval policy evaluation and interval value iteration algorithms rely on iterative convergence to the desired values, and are generalizations of the standard MDP algorithms *successive approximation* and *value iteration*, respectively. We believe it is also possible to design an interval-valued variant of the standard MDP algorithm *policy iteration*, but we have not done so at this writing — however, it should be clear that our successive approximation algorithm for evaluating policies in the BMDP setting provides an essential basic building block for constructing a policy iteration method; all that need be added is a means for selecting a new action at each state based on the interval value function of the preceding policy (and a possibly difficult corresponding analysis of the properties of the algorithm). We note that there is no consensus in the decision-theoretic planning and learning

and operations-research communities as to whether value iteration, policy iteration, or even standard linear programming is generally the best approach to solving MDP problems: each technique appears to have its strengths and weaknesses.

BMDPs are an efficiently solvable specialization of the already known class of *Markov Decision Processes with Imprecisely Known Transition Probabilities* (MDPIPs) [15][17][18]. In the related work section we discuss in more detail how BMDPs relate to MDPIPs.

Here is a high-level overview of how conceptual, theoretical, algorithmic, and experimental treatments are woven together in the remainder of the paper. We begin by introducing the concept of a Bounded Parameter MDP (BMDP), and introducing and justifying BMDP analogues for optimal policies and value functions. In terms of the theoretical development, we define the basic mathematical objects, introduce notational conventions, and provide some background in MDPs. We define the objects and operations that will be useful in the subsequent theoretical and algorithmic development, e.g., composition operators on MDPs and on policies. Finally, we define and motivate the relevant notions of optimality, and then prove the existence of optimal policies with respect to the different notions of optimality.

In addition to this theoretical and conceptual development, in terms of algorithm development we describe and provide pseudo-code for algorithms for computing optimal policies and value functions with respect to the different notions of optimality, e.g., interval policy evaluation and interval value iteration. We provide an analysis of the complexity of these algorithms and prove that they compute optimal policies as defined earlier. We then describe a proof-of-concept implementation and summarize preliminary experimental results. We also provide a brief overview of some applications including sensitivity analysis, coping with parameters known to be imprecise, and support for state aggregation methods. Finally, we survey some additional related work not covered in the primary text and summarize our contributions.

Before introducing BMDPs and their algorithms in Section 4 and Section 5, we first present in the next two sections a brief review of exact MDPs, policy evaluation, and value iteration in order to establish notational conventions we use throughout the paper. Our presentation follows that of [14], where a more complete account may be found.

2. Exact Markov Decision Processes

An (exact) Markov decision process M is a four tuple $M = \langle Q, A, F, R \rangle$ where Q is a set of states, 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 $\alpha \in A$ and $p, q \in Q$

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

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

A *policy* is a mapping from states to actions, $\pi: Q \rightarrow A$. 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

$$V_{M,\pi}(p) = R(p) + \gamma \sum_{q \in Q} F_{pq}(\pi(p)) V_{M,\pi}(q) \quad (2)$$

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.

$$V^*(p) = \max_{\alpha \in A} \left(R(p) + \gamma \sum_{q \in Q} F_{pq}(\alpha) V^*(q) \right) \quad (3)$$

The value function V^* is greater than or equal to any value function V_π in the partial order $\geq_{\text{dom}}$ defined as follows: $V_1 \geq_{\text{dom}} V_2$ if and only if for all states q, $V_1(q) \geq V_2(q)$ (in this case we say that V_1 *dominates* V_2). We write $V_1 >_{\text{dom}} V_2$ to mean $V_1 \geq_{\text{dom}} V_2$ and for at least one state q, $V_1(q) > V_2(q)$.

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 argmax.

3. 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. We follow the example of [14]. In Section 5, we describe novel algorithms 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.
2. In this paper, we focus on expected discounted cumulative reward as a performance criterion, but other criteria, *e.g.*, total or average reward [14], are also applicable to bounded-parameter MDPs.

computing the interval analogs of these value functions for bounded-parameter MDPs.

We present results from the theory of exact MDPs that 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 1) to show that iterating these operators converges to unique fixed-points, V_π and V^* respectively (Theorem 3 and Theorem 4).

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

$$\|v\| = \max_{q \in Q} |v(q)| . \tag{4}$$

We use the term *convergence* to mean convergence in the norm sense. The space $\overline{\mathrm{V}}$ 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, $0 \le \lambda < 1$ such that $\|Tv - Tu\| \le \lambda \|v - u\|$ for all u and v in U.

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

$$VI(v)(p) = \max_{\alpha \in A} \left(R(p) + \gamma \sum_{q \in Q} F_{pq}(\alpha) v(q) \right), \text{ and} \tag{5}$$

$$VI_\pi(v)(p) = R(p) + \gamma \sum_{q \in Q} F_{pq}(\pi(p)) v(q) . \tag{6}$$

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 just defined, except that the transition function F is F^M. More generally, we write $VI_{M,\pi}: \overline{\mathrm{V}} \to \overline{\mathrm{V}}$ and $VI_{M,\alpha}: \overline{\mathrm{V}} \to \overline{\mathrm{V}}$ to denote operators defined on each $p \in Q$ as:

$$\begin{aligned} VI_{M,\pi}(v)(p) &= R(p) + \gamma \sum_{q \in Q} F^M_{pq}(\pi(p)) v(q) \\ VI_{M,\alpha}(v)(p) &= R(p) + \gamma \sum_{q \in Q} F^M_{pq}(\alpha) v(q) \end{aligned} \tag{7}$$

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

$$V^*(p) = VI(V^*)(p) \quad \text{and} \quad V_\pi(p) = VI_\pi(V_\pi)(p) \tag{8}$$

for all states $p \in Q$. 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. Proofs for these theorems can be found in the work of Puterman [14].

Theorem 1: For any Banach space U and contraction mapping $T: U \to U$, 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 2: VI and VI_π are contraction mappings.

Theorem 1 and Theorem 2 together prove the following fundamental results in the theory of MDPs.

Theorem 3: There exists a unique $v^* \in \bar{\mathrm{V}}$ satisfying $v^* = VI(v^*)$; furthermore, $v^* = V^*$. Similarly V_π is the unique fixed-point of VI_π.

Theorem 4: For arbitrary $v^0 \in \bar{\mathrm{V}}$, 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 4 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 $\bar{\mathrm{V}}$, and repeatedly apply the operator VI to obtain the sequence $\{v^n\}$. This algorithm is referred to as *value iteration*. Theorem 4 guarantees the convergence of value iteration to the optimal value function. Similarly, we can specify an algorithm called *policy evaluation* that finds V_π by repeatedly applying VI_π starting with an initial $v^0 \in \bar{\mathrm{V}}$.

The following theorem from [12] states a convergence rate of value iteration and policy evaluation that can be derived using bounds on the precision needed to represent solutions to a linear program of limited precision (each algorithm can be viewed somewhat nontrivially as solving a linear program).

Theorem 5: 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.

Another important theorem that is used extensively in the proofs of the succeeding sections results directly from the monotonicity of the VI_π operator with respect to the $\leq_{\mathrm{dom}}$ and $\geq_{\mathrm{dom}}$ orderings, together with the above theorems.

Theorem 6: Let $\pi \in \Pi$ be a policy and M an MDP. Suppose there exists $u \in \bar{\mathrm{V}}$ for which $u \leq_{\mathrm{dom}} (\geq_{\mathrm{dom}}) VI_{M,\pi}(u)$, then $u \leq_{\mathrm{dom}} (\geq_{\mathrm{dom}}) V_{M,\pi}$. Likewise for the orderings $<_{\mathrm{dom}}$ and $>_{\mathrm{dom}}$.

4. Bounded-parameter Markov Decision Processes

A *bounded-parameter MDP(BMDP)* is a four tuple $M_\updownarrow = \langle Q, A, F_\updownarrow, R_\updownarrow \rangle$ where

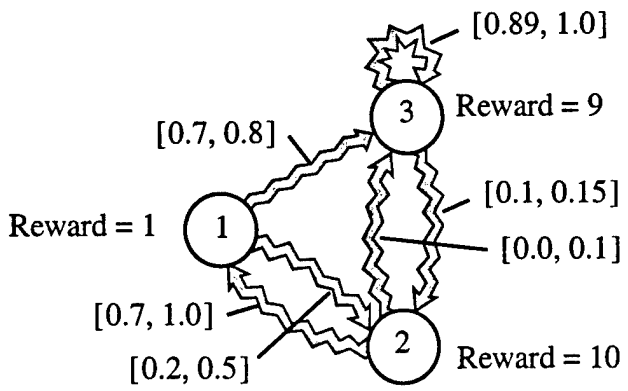


Figure 1: The state-transition diagram for a simple bounded-parameter Markov decision process with three states and a single action. The arcs indicate possible transitions and are labeled by their lower and upper bounds.

Q and A are defined as for MDPs, and $F_{\updownarrow}$ and $R_{\updownarrow}$ 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, $R_{\updownarrow}(p)$ and $F_{\updownarrow p,q}(\alpha)$ are both closed real intervals of the form $[l, u]$ for real numbers l and u with $l \leq u$, where in the case of $F_{\updownarrow}$ we require $0 \leq l \leq u \leq 1$.[3] To ensure that $F_{\updownarrow}$ admits only well-formed transition functions, we require that for any action α and state p, the sum of the lower bounds of $F_{\updownarrow p,q}(\alpha)$ 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. We use a one-action BMDP to illustrate various concepts in this paper because multi-action systems are awkward to draw, and one action suffices to illustrate the concepts. Note that a one action BMDP or MDP has only one policy available (select the only action at all states), and so represents a trivial control problem.

A BMDP $M_{\updownarrow} = \langle Q, A, F_{\updownarrow}, R_{\updownarrow} \rangle$ defines a set of exact MDPs that, by abuse of notation, we also call $M_{\updownarrow}$. For any exact MDP $M = \langle Q', A', F', R' \rangle$, we have $M \in M_{\updownarrow}$ if $Q = Q'$, $A = A'$, and for any action α and states p, q, $R'(p)$ is in the interval $R_{\updownarrow}(p)$ and $F'_{p,q}(\alpha)$ is in the interval $F_{\updownarrow p,q}(\alpha)$. We rely on context to distinguish between the tuple view of $M_{\updownarrow}$ and the set of exact MDPs view of $M_{\updownarrow}$. In the remaining definitions in this section, the BMDP $M_{\updownarrow}$ is implicit. Figure 3 shows an example of an exact MDP belonging to the family described by the BMDP in Figure 1. We use the convention that thick wavy lines represent interval valued transition probabilities and thinner straight lines represent exact transition probabilities.

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

An *interval value function* $V_{\updownarrow}$ is a mapping from states to closed real intervals. We generally use such functions to indicate that the value of a given state falls within the selected interval. Interval value functions can be specified for both exact MDPs 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 interval valued function (*e.g.*, $F_{\updownarrow}, R_{\updownarrow}, V_{\updownarrow}$, and those we define later) we define two real valued functions that take the same arguments and return the upper and lower interval bounds, respectively, denoted by the following syntactic variations: $F_{\uparrow}$, $R_{\uparrow}$, $V_{\uparrow}$ for upper bounds, and $F_{\downarrow}$, $R_{\downarrow}$, $V_{\downarrow}$ for lower bounds, respectively. So, for example, at any state q we have $V_{\updownarrow}(q) = [V_{\downarrow}(q), V_{\uparrow}(q)]$.

We note that the number of MDPs $M \in M_{\updownarrow}$ is in general uncountable. We start our analysis by showing that there is a finite subset $X_{M_{\updownarrow}} \in M_{\updownarrow}$ of these MDPs of particular interest. Given any ordering O of all the states in Q, there is a unique MDP $M \in M_{\updownarrow}$ that minimizes, for every state q and action α, the expected "position in the ordering" of the state reached by taking action α in state q — in other words, an MDP that for every state q and action α sends as much probability mass as possible to states early in the ordering O when taking action α in state q. Formally, we define the following concept:

Definition 1. Let $O = q_1, q_2, \ldots, q_k$ be an ordering of Q. We define the *order-maximizing MDP* M_O with respect to ordering O as follows.

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

$$\sum_{i=1}^{r-1} F_{\uparrow p, q_i}(\alpha) + \sum_{i=r}^{k} F_{\downarrow p, q_i}(\alpha). \tag{9}$$

The value r is the index into the state ordering $\{q_i\}$ such that below index r we assign the upper bound, and above index r we assign the lower bound, with the rest of the probability mass from p under α being assigned to q_r. Formally, we select $M_O \in M_{\updownarrow}$ by choosing $F^{M_O}_{p,q}(\alpha)$ for all $q \in Q$ as follows:

$$F^{M_O}_{pq_j}(\alpha) = \begin{cases} F_{\uparrow pq_i}(\alpha) \text{ if } j < r \\ F_{\downarrow pq_i}(\alpha) \text{ if } j > r \end{cases} \quad \text{and}$$

$$F^{M_O}_{pq_r}(\alpha) = 1 - \sum_{i=1, i \neq r}^{i=k} F^{M_O}_{pq_i}(\alpha).$$

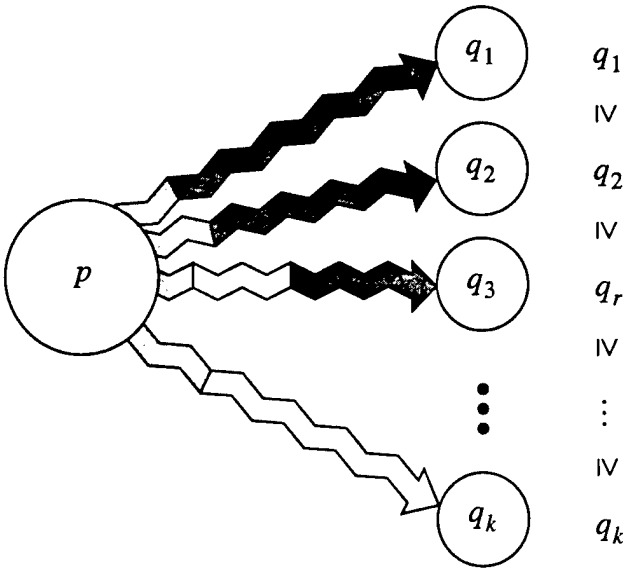


Figure 2: An illustration of the transition probabilities in the order-maximizing MDP at the state p for the order shown. The lighter shaded portions of each arc represent the required lower bound transition probability and the darker shaded portions represent the fraction of the remaining allowed transition probability assigned to the arc by T.

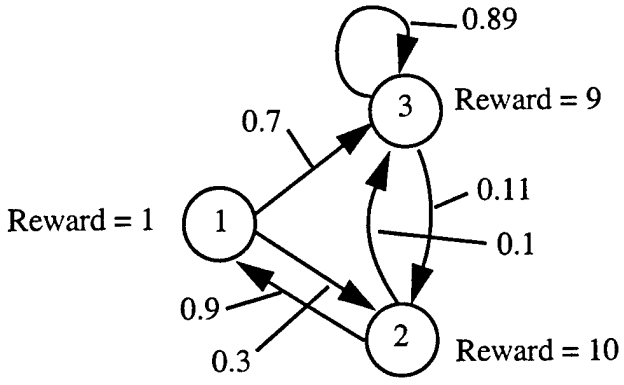


Figure 3: The order-maximizing MDP for the BMDP shown in Figure 1 using the state order $2 > 3 > 1$.

Figure 2 shows a diagrammatic representation of the order-maximizing MDP at a particular state p for the particular ordering of the state space shown. Figure 3 shows the order-maximizing MDP for the particular BMDP shown in Figure 1 using a particular state order ($2 > 3 > 1$), as a concrete example.

Definition 2. Let $X_{M_\updownarrow}$ be the set of order-maximizing MDPs M_O in $M_\updownarrow$, one for each ordering O. Note that since there are finitely many orderings of states, $X_{M_\updownarrow}$ is finite.

We now show that the set $X_{M_\updownarrow}$ in some sense contains every MDP of interest from $M_\updownarrow$. In particular, we show that for any policy π and any MDP M in $M_\updownarrow$, the value of π in M is bracketed by values of π in two MDPs in $X_{M_\updownarrow}$.

Lemma 1: For any MDP $M \in M_\updownarrow$,

(a) For any policy $\pi \in \Pi$, there are MDPs $M_1 \in X_{M_\updownarrow}$ and $M_2 \in X_{M_\updownarrow}$ such that

$$V_{M_1,\pi} \leq_{\text{dom}} V_{M,\pi} \leq_{\text{dom}} V_{M_2,\pi} \,. \tag{10}$$

(b) Also, for any value function $v \in \bar{\mathrm{V}}$, there are MDPs $M_3 \in X_{M_\updownarrow}$ and $M_4 \in X_{M_\updownarrow}$ such that

$$VI_{M_3,\pi}(v) \leq_{\text{dom}} VI_{M,\pi}(v) \leq_{\text{dom}} VI_{M_4,\pi}(v) \,. \tag{11}$$

Proof: See Appendix.

Interval Value Functions for Policies. We now define the interval analogue to the traditional MDP policy-specific value function V_π, and state and prove some of the properties of this interval value function. The development here requires some care, as one desired property of the definition is not immediate. We first observe that we would like an interval-valued function over the state space that satisfies a Bellman equation like that for traditional MDPs (as given by Equation 2). Unfortunately, stating a Bellman equation requires us to have specific transition probability distributions F rather than a range of such distributions. Instead of defining policy value via a Bellman equation, we define the interval value function directly, at each state, as giving the range of values that could be attained at that state for the various choices of F allowed by the BMDP. We then show that the desired minimum and maximum values can be achieved independent of the state, so that the upper and lower bound value functions are just the values of the policy in particular "minimizing" and "maximizing" MDPs in the BMDP. This fact enables the use of the Bellman equations for the minimizing and maximizing MDPs to give an iterative algorithm that converges to the desired values, as presented in Section 5

Definition 3. For any policy π and state q, we define the *interval value* $V_{\updownarrow\pi}(q)$ *of* π *at* q to be the interval

$$V_{M_\updownarrow,\pi}(q) = \left[\min_{M \in M_\updownarrow} V_{M,\pi}(q), \max_{M \in M_\updownarrow} V_{M,\pi}(q)\right]. \quad (12)$$

We note that the existence of these minimum and maximum values follows from Lemma 1 and the finiteness of the set $X_{M_\updownarrow}$ — because Lemma 1 implies that $V_{M_\updownarrow,\pi}(q)$ is the same as the following where the minimization and maximization are done over finite sets:

$$V_{M_\updownarrow,\pi}(q) = \left[\min_{M \in X_{M_\updownarrow}} V_{M,\pi}(q), \max_{M \in X_{M_\updownarrow}} V_{M,\pi}(q)\right]. \quad (13)$$

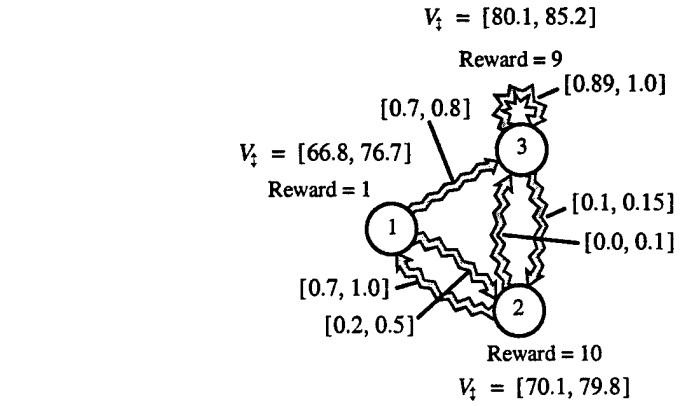
In preparation for the discussion in Section 5, we show in Theorem 7 that for any policy there is at least one specific *policy-maximizing* MDP in $M_\updownarrow$ that achieves the upper bound in Definition 3 at all states q simultaneously (and likewise a different specific *policy-minimizing* MDP that achieves the lower bound at all states q simultaneously). We formally define these terms below.

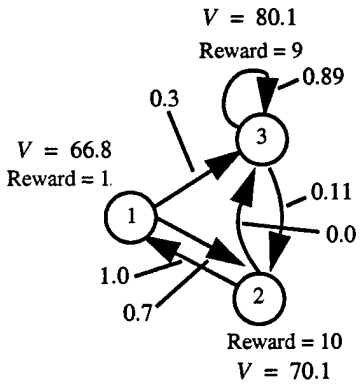
Definition 4. For any policy π, an MDP $M \in M_\updownarrow$ is *π-maximizing* if $V_{M,\pi}$ dominates $V_{M',\pi}$ for any $M' \in M_\updownarrow$, *i.e.*, for any $M' \in M_\updownarrow$, $V_{M,\pi} \geq_{\text{dom}} V_{M',\pi}$. Likewise, $M \in M_\updownarrow$ is *π-minimizing* if it is dominated by all such $V_{M',\pi}$, *i.e.*, for any $M' \in M_\updownarrow$, $V_{M,\pi} \leq_{\text{dom}} V_{M',\pi}$.

Figure 4 shows the interval value function for the only policy available in the (trivial) one-action BMDP shown in Figure 1, along with the π-maximizing and π-minimizing MDPs for that policy.

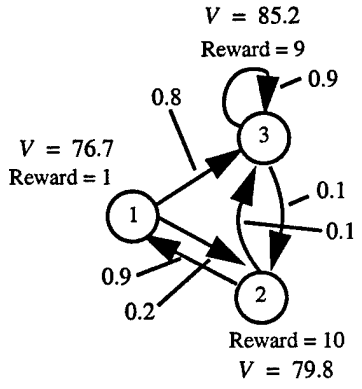
We note that Lemma 1 implies that for any single state q and any policy π we can select an MDP $M \in M_\updownarrow$ to maximize (or minimize) $V_{M,\pi}(q)$ by selecting the MDP in $X_{M_\updownarrow}$ that gives the largest value for π at q. However, we have not shown that a single MDP can be chosen to simultaneously maximize (or minimize) $V_{M,\pi}(q)$ at all states $q \in Q$ (*i.e.*, that there exist π-maximizing and π-minimizing MDPs). In order to show this fact, we show how to compose two MDPs (with respect to a fixed policy π) to construct a third MDP such that the value of π in the third MDP is not less than the value of π in either of the initial two MDPs, at every state. We can then construct a π-maximizing MDP by composing together all the MDPs that maximize the value of π at the different individual states (likewise for π-minimizing MDPs using a similar composition operator). We start by defining the just mentioned policy-relative composition operators on MDPs:

Definition 5. Let $\oplus^{\pi}_{\max}$ and $\oplus^{\pi}_{\min}$ denote composition operators on MDPs with respect to a policy $\pi \in \Pi$, defined as follows:





π–minimizing MDP



π–maximizing MDP

Figure 4: The interval value function (shown as $V_\updownarrow$ on the top subfigure), policy-minimizing MDP with state values (lower left), and policy-maximizing MDP with state values (lower right) for the one-action BMDP shown in Figure 1 under the only policy. We assume a discount factor of 0.9. Note that the lower-bound values in the interval value function are the state values under the policy-minimizing MDP, and the upper-bound values are the state values under the policy-maximizing MDP. Also, note that the policy-maximizing MDP is the order-maximizing MDP for the state order 3>2>1 and the policy-minimizing MDP is the order-maximizing MDP for the order 1>2>3—policy-minimizing and maximizing MDPs are always order-maximizing for some order (but the orders need not be reverse to one another as they are in this example).

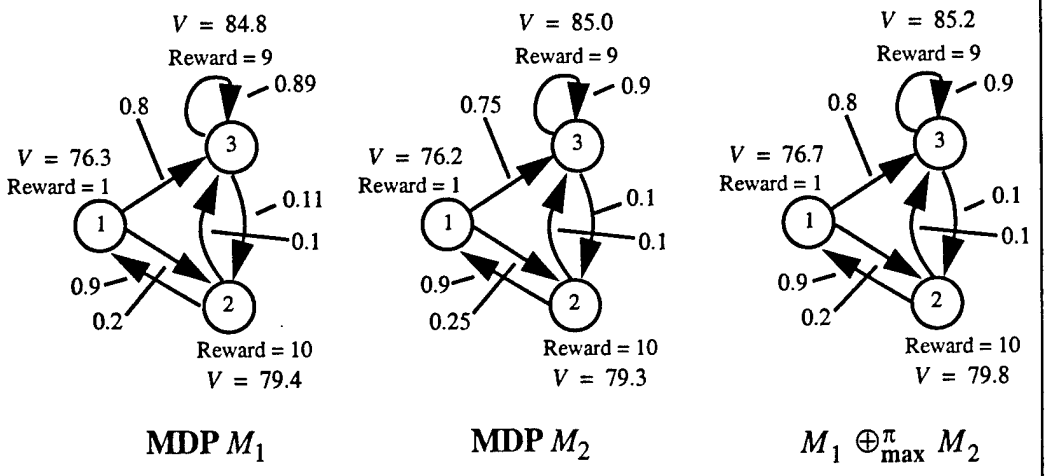


Figure 5: Two MDPs M_1 and M_2 from the BMDP shown in Figure 1, and their composition under $\oplus^{\pi}_{\max}$ where π is the only available policy in the one-action BMDP. State transition probabilities for the composition MDP are selected from the component MDP that achieves the greater value for the source state of the transition. State values are shown for all three MDPs — note that the composition MDP achieves higher value at every state, as claimed in Lemma 2.

If $M_1, M_2 \in M_{\updownarrow}$, then $M_3 = M_1 \oplus^{\pi}_{\max} M_2$ if for all states $p, q \in Q$,

$$F_{pq}^{M_3}(\alpha) = \begin{cases} F_{pq}^{M_1}(\alpha) & \text{if } V_{M_1,\pi}(p) \geq V_{M_2,\pi}(p) \text{ and } \alpha = \pi(p) \\ F_{pq}^{M_2}(\alpha) & \text{otherwise} \end{cases}$$

If $M_1, M_2 \in M_{\updownarrow}$, then $M_3 = M_1 \oplus^{\pi}_{\min} M_2$ if for all states $p, q \in Q$,

$$F_{pq}^{M_3}(\alpha) = \begin{cases} F_{pq}^{M_1}(\alpha) & \text{if } V_{M_1,\pi}(p) \leq V_{M_2,\pi}(p) \text{ and } \alpha = \pi(p) \\ F_{pq}^{M_2}(\alpha) & \text{otherwise} \end{cases}$$

We give as an example in Figure 5 two MDPs from the BMDP of Figure 1, along with their composition under the $\oplus^{\pi}_{\max}$ operator where π is the single available policy for that one-action BMDP. We now state the property claimed above for this MDP composition operator:

Lemma 2: Let π be a policy in Π and M_1, M_2 be MDPs in $M_{\updownarrow}$.

(a) For $M_3 = M_1 \oplus^{\pi}_{\max} M_2$,

$$V_{M_3,\pi} \geq_{\text{dom}} V_{M_1,\pi} \text{ and } V_{M_3,\pi} \geq_{\text{dom}} V_{M_2,\pi}, \text{ and} \qquad (14)$$

(b) for $M_3 = M_1 \oplus^{\pi}_{\min} M_2$,

$$V_{M_3,\pi} \leq_{\text{dom}} V_{M_1,\pi} \text{ and } V_{M_3,\pi} \leq_{\text{dom}} V_{M_2,\pi} \cdot \quad (15)$$

Proof: See Appendix.

These MDP composition operators can now be used to show the existence of policy-maximizing and policy-minimizing MDPs within $M_{\updownarrow}$.

Theorem 7: For any policy $\pi \in \Pi$, there exist π-maximizing and π-minimizing MDPs in $X_{M_{\updownarrow}} \subseteq M_{\updownarrow}$.

Proof: Enumerate $X_{M_{\updownarrow}}$ as a finite sequence of MDPs $M_1, \ldots, M_k$. Consider composing these MDPs together to construct the MDP M as follows:

$$M = (((M_1 \oplus^{\pi}_{\max} M_2) \oplus^{\pi}_{\max} \ldots) \oplus^{\pi}_{\max} M_k) \quad (16)$$

Note that M may depend on the ordering of $M_1, \ldots, M_k$, but that any ordering is satisfactory for this proof. It is straightforward to show by induction using Lemma 2 that $V_{M,\pi} \geq_{\text{dom}} V_{M_i,\pi}$ for each $1 \leq i \leq k$, and then Lemma 1 implies that $V_{M,\pi} \geq_{\text{dom}} V_{M',\pi}$ for any $M' \in M_{\updownarrow}$. M is thus a π-maximizing MDP. Although M may not be in $X_{M_{\updownarrow}}$, Lemma 1 implies that $V_{M,\pi}$ must be dominated by $V_{M',\pi}$ for some $M' \in X_{M_{\updownarrow}}$, which must also be π-maximizing.

An identical proof implies the existence of π-minimizing MDPs, replacing each occurrence of "max" with "min" and each $\geq_{\text{dom}}$ with $\leq_{\text{dom}}$. ❑

Corollary 1: $V_{\downarrow\pi} = \min_{M \in M_{\updownarrow}}(V_{M,\pi})$ and $V_{\uparrow\pi} = \max_{M \in M_{\updownarrow}}(V_{M,\pi})$ where the minimum and maximum are computed relative to $\leq_{\text{dom}}$ and are well-defined by Theorem 7.

We give an algorithm in Section 5 that converges to $V_{\downarrow\pi}$ by also converging to a π-minimizing MDP in $M_{\updownarrow}$ (similarly for $V_{\uparrow\pi}$, exchanging π-maximizing for π-minimizing).

Optimal Value Functions in BMDPs. We now consider how to define an optimal value function for a BMDP. First, consider the expression $\max_{\pi \in \Pi}(V_{\updownarrow\pi})$. This expression is ill-formed because we have not defined how to rank the interval value functions $V_{\updownarrow\pi}$ in order to select a maximum.[4] 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:

4. Similar issues arise if we attempt to define the optimal value function using a Bellman style equation such as Equation 3 because we must compute a maximization over a set of intervals.

$$([l_1, u_1] \leq_{\text{pes}} [l_2, u_2]) \Leftrightarrow (l_1 < l_2 \text{ or } (l_1 = l_2 \wedge u_1 \leq u_2))$$
$$([l_1, u_1] \leq_{\text{opt}} [l_2, u_2]) \Leftrightarrow (u_1 < u_2 \text{ or } (u_1 = u_2 \wedge l_1 \leq l_2)) \quad (17)$$

We extend these orderings to partial orders over interval value functions by relating two value functions $V_{\updownarrow 1} \leq_{\text{opt}} V_{\updownarrow 2}$ only when $V_{\updownarrow 1}(q) \leq_{\text{opt}} V_{\updownarrow 2}(q)$ for every state q. We can now use either of these orderings to compute $\max_{\pi \in \Pi}(V_{\updownarrow \pi})$, yielding two definitions of optimal value function and optimal policy. However, since the orderings are partial (on value functions), we prove first (Theorem 8) that the set of policies contains a policy that achieves the desired maximum under each ordering (*i.e.*, a policy whose interval value function is ordered above that of every other policy).

Definition 6. An *optimistically optimal policy* π_{opt} is any policy such that $V_{\updownarrow \pi_{\text{opt}}} \geq_{\text{opt}} V_{\updownarrow \pi}$ for all policies π. A *pessimistically optimal policy* π_{pes} is any policy such that $V_{\updownarrow \pi_{\text{pes}}} \geq_{\text{pes}} V_{\updownarrow \pi}$ for all policies π.

In Theorem 8, we prove that there exist optimistically optimal policies by induction (an analogous proof holds for pessimistically optimal policies). We develop this proof in two stages, mirroring the two-stage definition of $\geq_{\text{opt}}$ (first emphasizing the upper bound and then breaking ties with the lower bound). We first construct a policy π' for which the upper bounds of the interval value function $V_{\uparrow \pi'}$ dominate those $V_{\uparrow \pi''}$ of any other policy π''. We then show that the finite set of such policies (all tied on upper bounds) can be combined to construct a policy π_{opt} with the same upper bound values $V_{\uparrow \pi_{\text{opt}}}$ and whose lower bounds $V_{\downarrow \pi_{\text{opt}}}$ dominate those of any other policy. Each of these constructions relies on the following policy composition operator:

Definition 7. Let $\oplus_{\text{opt}}$ and $\oplus_{\text{pes}}$ denote composition operators on policies, defined as follows. Consider policies $\pi_1, \pi_2 \in \Pi$,

Let $\pi_3 = \pi_1 \oplus_{\text{opt}} \pi_2$ if for all states $p \in Q$:

$$\pi_3(p) = \begin{cases} \pi_1(p) & \text{if } V_{\updownarrow \pi_1}(p) \geq_{\text{opt}} V_{\updownarrow \pi_2}(p) \\ \pi_2(p) & \text{otherwise} \end{cases} \quad (18)$$

Let $\pi_3 = \pi_1 \oplus_{\text{pes}} \pi_2$ if for all states $p \in Q$:

$$\pi_3(p) = \begin{cases} \pi_1(p) & \text{if } V_{\updownarrow \pi_1}(p) \geq_{\text{pes}} V_{\updownarrow \pi_2}(p) \\ \pi_2(p) & \text{otherwise} \end{cases} \quad (19)$$

Our task would be relatively easy if it were necessarily true that

$$V_{\updownarrow(\pi_1 \oplus_{\text{opt}} \pi_2)} \geq_{\text{opt}} V_{\updownarrow\pi_1} \quad \text{and} \quad V_{\updownarrow(\pi_1 \oplus_{\text{opt}} \pi_2)} \geq_{\text{opt}} V_{\updownarrow\pi_2} \ . \tag{20}$$

(and likewise for the pessimistic case). However, because of the lexicographic nature of $\geq_{\text{opt}}$, these statements do not hold (in particular, the lower bound values for some states may be worse in the composed policy than in either component even when the upper bounds on those states do not change). For this reason, we prove a somewhat weaker result that must be used in a two-stage fashion as demonstrated below:

Lemma 3: Given a BMDP $M_{\updownarrow}$, and policies $\pi_1, \pi_2 \in \Pi$, $\pi_3 = \pi_1 \oplus_{\text{opt}} \pi_2$, and $\pi_4 = \pi_1 \oplus_{\text{pes}} \pi_2$,

(a) $V_{\uparrow\pi_3} \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $V_{\uparrow\pi_3} \geq_{\text{dom}} V_{\uparrow\pi_2}$
(b) If $V_{\uparrow\pi_1} = V_{\uparrow\pi_2}$ then $V_{\updownarrow\pi_3} \geq_{\text{opt}} V_{\updownarrow\pi_1}$ and $V_{\updownarrow\pi_3} \geq_{\text{opt}} V_{\updownarrow\pi_2}$
(c) $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_1}$ and $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_2}$
(d) If $V_{\downarrow\pi_1} = V_{\downarrow\pi_2}$ then $V_{\updownarrow\pi_3} \geq_{\text{pes}} V_{\updownarrow\pi_1}$ and $V_{\updownarrow\pi_3} \geq_{\text{pes}} V_{\updownarrow\pi_2}$.

Proof: See Appendix.

Theorem 8: There exists at least one optimistically (pessimistically) optimal policy.

Proof: Enumerate Π as a finite sequence of policies $\pi_1, \ldots, \pi_k$. Consider composing these policies together to construct the policy $\pi_{\text{opt, up}}$ as follows:

$$\pi_{\text{opt, up}} = (((\pi_1 \oplus_{\text{opt}} \pi_2) \oplus_{\text{opt}} \ldots) \oplus_{\text{opt}} \pi_k) \tag{21}$$

Note that $\pi_{\text{opt, up}}$ may depend on the ordering of $\pi_1, \ldots, \pi_k$, but that any ordering is satisfactory for this proof. It is straightforward to show by induction using Lemma 3 that $V_{\uparrow\pi_{\text{opt, up}}} \geq_{\text{dom}} V_{\uparrow\pi_i}$ for each $1 \leq i \leq k$. Now enumerate the subset of Π for which the value function upper bounds equal those of $\pi_{\text{opt, up}}$, *i.e.*, enumerate $\{\pi' \mid V_{\uparrow\pi'} = V_{\uparrow\pi_{\text{opt, up}}}\}$ as $\{\pi_1', \ldots, \pi_l'\}$. Consider again composing the policies π_i' together as above to form the policy π_{opt}:

$$\pi_{\text{opt}} = (((\pi_1' \oplus_{\text{opt}} \pi_2') \oplus_{\text{opt}} \ldots) \oplus_{\text{opt}} \pi_k') \tag{22}$$

It is again straightforward to show using Lemma 3 that $V_{\downarrow\pi_{\text{opt}}} \geq_{\text{dom}} V_{\downarrow\pi_i'}$ for each $1 \leq i \leq l$. It follows immediately that $V_{\updownarrow\pi_{\text{opt}}} \geq_{\text{opt}} V_{\updownarrow\pi}$ for every $\pi \in \Pi$, as desired. A similar construction using $\oplus_{\text{pes}}$ yields a pessimistically optimal policy π_{pes}. ❑

Theorem 8 justifies the following definition:

Definition 8. The *optimistic optimal value function* $V_{\updownarrow \text{opt}}$ and the *pessimistic optimal value function* $V_{\updownarrow \text{pes}}$ are given by:

$$V_{\updownarrow \text{opt}} = \max_{\pi \in \Pi}(V_{\updownarrow \pi}) \quad \text{using } \leq_{\text{opt}} \text{ to order interval value functions}$$

$$V_{\updownarrow \text{pes}} = \max_{\pi \in \Pi}(V_{\updownarrow \pi}) \quad \text{using } \leq_{\text{pes}} \text{ to order interval value functions}$$

The above two notions of optimal value can be understood in terms of a two player game in which the first player chooses a policy π and then the second player chooses the MDP M in $M_{\updownarrow}$ in which to evaluate the policy π (see Shapley's work [16] for the origins of this viewpoint). The goal for the first player is to get the highest[5] resulting value function $V_{M,\pi}$. The upper bounds $V_{\uparrow \text{opt}}$ of the optimistically optimal value function represent the best value function the first player can obtain in this game if the second player cooperates by selecting an MDP to maximize $V_{M,\pi}$ (the lower bound $V_{\downarrow \text{opt}}$ corresponds to how badly this optimistic strategy for the first player can misfire if the second player betrays the first player and selects an MDP to minimize $V_{M,\pi}$). The lower bounds $V_{\downarrow \text{pes}}$ of the pessimistically optimal value function represent the best the first player can do under the assumption that the second player is an adversary, trying to minimize the resulting value function.

We conclude this section by stating a Bellman equation theorem for the optimal interval value functions just defined. The equations below form the basis for our iterative algorithm for computing the optimal interval value functions for a BMDP. We start by stating two definitions that are useful in proving the Bellman theorem as well as in later sections. It is useful to have notation to denote the set of actions that maximize the upper bound at each state. For a given value function V, we write ρ_V for the function from states to sets of actions such that for each state p,

$$\rho_V(p) = \operatorname*{argmax}_{\alpha \in A} \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V)(p). \tag{23}$$

Likewise, for the pessimistic case, we define σ_V for the function from states to sets of actions giving the actions that maximize the lower bound. For each state p, $\sigma_V(p)$ is given by

$$\sigma_V(p) = \operatorname*{argmax}_{\alpha \in A} \min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V)(p). \tag{24}$$

Theorem 9: For any BMDP $M_{\updownarrow}$, the following Bellman-like equations hold at every state p,

5. Value functions are ranked by $\geq_{\text{dom}}$.

$$V_{\updownarrow \text{opt}}(p) = \max_{\alpha \in A,\ \leq_{\text{opt}}} \left[\min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow \text{opt}})(p), \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow \text{opt}})(p) \right], \quad (25)$$

and

$$V_{\updownarrow \text{pes}}(p) = \max_{\alpha \in A,\ \leq_{\text{pes}}} \left[\min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow \text{pes}})(p), \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow \text{pes}})(p) \right]. \quad (26)$$

Proof: See Appendix.

5. Estimating Interval Value Functions

In this section, we describe dynamic programming algorithms that operate on bounded-parameter MDPs. We first define the interval equivalent of policy evaluation $IVI_{\updownarrow \pi}$ which computes $V_{\updownarrow \pi}$, and then define the variants $IVI_{\updownarrow \text{opt}}$ and $IVI_{\updownarrow \text{pes}}$ which compute the optimistic and pessimistic optimal value functions.

5.1 Interval Policy Evaluation

In direct analogy to the exact MDP definition of VI_{π} in Section 3, we define a function $IVI_{\updownarrow \pi}$ (for *interval value iteration*) which maps interval value functions to other interval value functions. We prove that iterating $IVI_{\updownarrow \pi}$ on any initial interval value function produces a sequence of interval value functions that converges to $V_{\updownarrow \pi}$ in a polynomial number of steps, given a fixed discount factor γ.

$IVI_{\updownarrow \pi}(V_{\updownarrow})$ is an interval value function, defined for each state p as follows:

$$IVI_{\updownarrow \pi}(V_{\updownarrow})(p) = \left[\min_{M \in M_{\updownarrow}} VI_{M,\pi}(V_{\downarrow})(p), \max_{M \in M_{\updownarrow}} VI_{M,\pi}(V_{\uparrow})(p) \right] \quad (27)$$

We define $IVI_{\downarrow \pi}$ and $IVI_{\uparrow \pi}$ to be the corresponding mappings from value functions to value functions (note that for input $V_{\updownarrow}$, $IVI_{\downarrow \pi}$ does not depend on $V_{\uparrow}$ and so can be viewed as a function from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$ — likewise for $IVI_{\uparrow \pi}$ and $V_{\downarrow}$).

The algorithm to compute $IVI_{\updownarrow \pi}$ 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 $M_{\updownarrow}$ that minimizes (maximizes) the value attained. We select such an MDP by selecting a transition probability function F within the bounds specified by the $F_{\updownarrow}$ component of $M_{\updownarrow}$ to minimize (maximize) the value — each possible way of selecting F corresponds to one MDP in $M_{\updownarrow}$. 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 $\Sigma_{q \in Q}(F_{pq}(\alpha)V(q))$. This step constitutes the heart of the $IVI_{\updownarrow \pi}$ algorithm and the only significant way the algorithm differs from standard

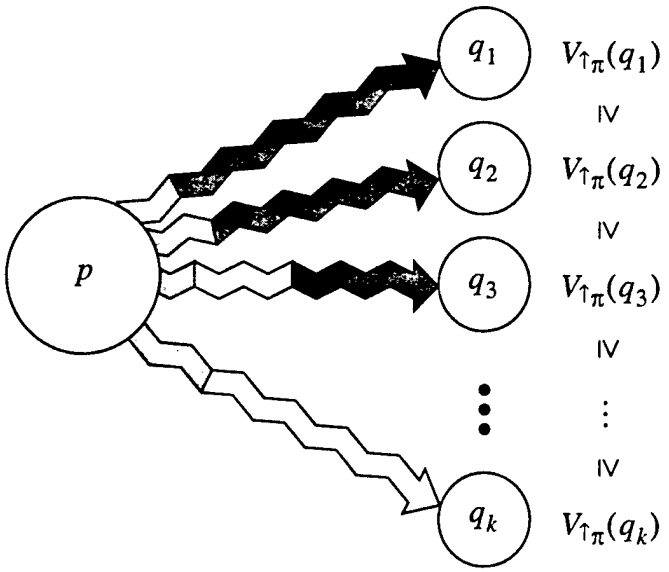


Figure 6: An illustration of the basic dynamic programming step in computing an approximate value function for a fixed policy and bounded-parameter MDP. $V_{\uparrow\pi}$ gives the upper bounds of the current interval estimates of V_π. The lighter shaded portions of each arc represent the required lower 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.

policy evaluation by successive approximation by iterating $VI_{M,\pi}$.

To compute the lower bounds $IVI_{\downarrow\pi}$ the idea is to sort the possible destination states q into increasing order according to their $V_\downarrow$ value, and then choose the transition probabilities within the intervals specified by $F_\updownarrow$ so as to send as much probability mass to the states early in the ordering (upper bounds are computed similarly, but sorting the states into decreasing order by their $V_\uparrow$ value). Let $O = q_1, q_2, \ldots, q_k$ be such an ordering of Q — so that for all i and j if $1 \le i \le j \le k$ then $V_\downarrow(q_i) \le V_\downarrow(q_j)$ (increasing order). We can then show that the order-maximizing MDP M_O is the MDP that minimizes the desired expression $\Sigma_{q \in Q}(F^M_{pq}(\alpha)V(q))$. The order-maximizing MDP for the decreasing order based on $V_\uparrow$ will maximize the same expression to generate the upper bound in Equation 27.

Figure 6 illustrates the basic iterative step in the above algorithm, for the upper bound, *i.e.* maximizing, case. The states q_i are ordered according to the value estimates in $V_\uparrow$. 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. For a more precise account of the algorithm, please refer to Figure 7 for a pseudocode description of the computation of $IVI_{\updownarrow\pi}(V_\updownarrow)$.

Techniques similar to those in Section 3 can be used to prove that iterating

```
IVI↕(V↕, π)
\\we assume that V↕ is represented as:
\\   V↓ is a vector of n real numbers giving lower-bounds for states q1 to qn
\\   V↑ is a vector of n real numbers giving upper-bounds for states q1 to qn
{  Create O, a vector of n states for holding a permutation of the states q1 to qn
   \\first, compute new lower bounds
   O = sort_increasing_order(q1,...,qn,<lb);   \\ <lb compares state lower-bounds
   Update(V↓, π, O);
   \\second, compute new upper bounds
   O = sort_decreasing_order(q1,...,qn,<ub);   \\ <ub compares state upper-bnds
   Update(V↑, π, O)}
==========================================================
\\ Update(v, π, o) updates v using the order-maximizing MDP for o
\\   o is a state ordering—a vector of states (a permutation of q1,...,qn)
\\   v is a value function—a vector of real numbers of length n
Update(v, π, o)
{  Create F', a matrix of n by n real numbers
   \\ the next loop sets F' to describe π in the order-maximizing MDP for o
   for each state p  {
      used = Σ_{state q} F↓p,q(π(p));
      remaining = 1 – used;
      \\ distribute remaining probability mass to states early in the ordering
      for i=1 to n  {                    \\ i is used to index into ordering o
         min = F↓p,o(i)(π(p));
         desired = F↑p,o(i)(π(p));
         if (desired <= remaining)
            then F'(p,o(i)) = min+desired;
            else F'(p,o(i)) = min+remaining;
         remaining = max(0,remaining-desired)}}
   \\ F' now describes π in the order-maximizing MDP w/respect to O,
   \\finally, update v using a value iteration-like update based on F'
   for each state p
      v(p) = R(p) + γ Σ_{state q} F'(p,q) v(q)   }
```

Figure 7: Pseudocode for one iteration of interval policy evaluation ($IVI_{\updownarrow}$)

$IVI_{\downarrow\pi}$ (or $IVI_{\uparrow\pi}$) converges to $V_{\downarrow\pi}$ (or $V_{\uparrow\pi}$). The key theorems, stated below, assert first that $IVI_{\downarrow\pi}$ is a contraction mapping, and second that $V_{\downarrow\pi}$ is a fixed-point of $IVI_{\downarrow\pi}$ and are easily proven.

Theorem 10: For any policy π, $IVI_{\downarrow\pi}$ and $IVI_{\uparrow\pi}$ are contraction mappings.

Proof: See Appendix.

Theorem 11: For any policy π, $V_{\downarrow\pi}$ is a fixed-point of $IVI_{\downarrow\pi}$ and $V_{\uparrow\pi}$ of $IVI_{\uparrow\pi}$, and therefore $V_{\updownarrow\pi}$ is a fixed-point of $IVI_{\updownarrow\pi}$.

These theorems, together with Theorem 1 (the Banach fixed-point theorem) imply that iterating $IVI_{\updownarrow\pi}$ on any initial interval value function converges to $V_{\updownarrow\pi}$, regardless of the starting point.

Theorem 12: For fixed $\gamma < 1$, 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 BMDP parameters.

Proof: (sketch) We provide only the key ideas behind this proof.

(a) By Theorem 10, $IVI_{\updownarrow\pi}$ is a contraction by γ on both the upper and lower bound value functions, and thus the successive estimates of $V_{\updownarrow\pi}$ produced converge exponentially to the unique fixed-point.

(b) By Theorem 11, the unique fixed-point is the desired value function.

(c) The upper bound and lower bound value functions making up the true $V_{\updownarrow\pi}$ are the value functions of π in particular MDPs (π-maximizing and π-minimizing MDPs, respectively) in $X_{M_{\updownarrow}}$.

(d) The parameters for the MDPs in $X_{M_{\updownarrow}}$ can be specified with a number of bits polynomial in the number of bits used to specify the BMDP parameters.

(e) The value function for a policy in an MDP can be written as the solution to a linear program. The precision of any such solution can be bounded in terms of the number of bits used to specify the linear program. This precision bound allows the definition of a stopping condition for $IVI_{\updownarrow\pi}$ when adequate precision is obtained.

❑ (Theorem 12).

5.2 Interval Value Iteration

As in the case of altering VI_π to obtain VI, it is straightforward to modify $IVI_{\updownarrow\pi}$ 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.

$$IVI_{\updownarrow \text{opt}}(V_{\updownarrow})(p) = \max_{\alpha \in A,\ \leq_{\text{opt}}} \left[\min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow})(p), \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow})(p) \right] \quad (28)$$

$$IVI_{\updownarrow \text{pes}}(V_{\updownarrow})(p) = \max_{\alpha \in A,\ \leq_{\text{pes}}} \left[\min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow})(p), \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow})(p) \right] \quad (29)$$

The added maximization step introduces no new difficulties in implementing the algorithm—for more details we provide pseudocode for $IVI_{\updownarrow \text{opt}}$ in Figure 8. We discuss convergence for $IVI_{\updownarrow \text{opt}}$ — the convergence results for $IVI_{\updownarrow \text{pes}}$ are similar. We first summarize our approach and then cover the same ground in more detail.

We write $IVI_{\uparrow \text{opt}}$ for the upper bound returned by $IVI_{\updownarrow \text{opt}}$, and we consider $IVI_{\uparrow \text{opt}}$ a function from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$ because $IVI_{\uparrow \text{opt}}(V_{\updownarrow})$ depends only on $V_{\uparrow}$ due to the way $\leq_{\text{opt}}$ compares intervals primarily based on their upper bound. $IVI_{\uparrow \text{opt}}$ can easily be shown to be a contraction mapping, and it can be shown that $V_{\uparrow \text{opt}}$ is a fixed point of $IVI_{\uparrow \text{opt}}$. It then follows that $IVI_{\uparrow \text{opt}}$ converges to $V_{\uparrow \text{opt}}$ (and we can argue as for $IVI_{\updownarrow \pi}$ that this convergence occurs in polynomially many steps for fixed γ). The analogous results for $IVI_{\downarrow \text{opt}}$ are somewhat more problematic. Because the action selection is done according to $\leq_{\text{opt}}$, which focuses primarily on the interval upper bounds, $IVI_{\downarrow \text{opt}}$ is not properly a mapping from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$, as the action choice for $IVI_{\downarrow \text{opt}}(V_{\updownarrow})$ depends on both $V_{\downarrow}$ and $V_{\uparrow}$. In particular, for each state, the action that maximizes the lower bound is chosen from among the subset of actions that (equally) maximize the upper bound.

To deal with this complication, we observe that if we fix the upper bound value function $V_{\uparrow}$, we can view $IVI_{\downarrow \text{opt}}$ as a function from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$ carrying the lower bounds of the input value function to the lower bounds of the output. To formalize this idea, we introduce some new notation. First, given two value functions V_1 and V_2 we define the interval value function $[V_1, V_2]$ to be the function from states p to intervals $[V_1(p), V_2(p)]$ (this notation is essentially the inverse of the $\downarrow$ and $\uparrow$ notation which extracts lower and upper bound functions from interval functions). Using this new notation, we define a family $\{IVI_{\downarrow \text{opt}, V}\}$ of functions from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$, indexed by a value function V. For each value function V, we define $IVI_{\downarrow \text{opt}, V}(V')$ to be the function from $\overline{\mathbb{V}}$ to $\overline{\mathbb{V}}$ that maps V' to $IVI_{\downarrow \text{opt}}([V', V])$. (Analogously, we define $IVI_{\uparrow \text{pes}, V}(V')$ to map V' to $IVI_{\uparrow \text{pes}}([V, V']))$. We note that $IVI_{\downarrow \text{opt}, V}$ has the following relationships to $IVI_{\updownarrow \text{opt}}$:

$$\begin{aligned} IVI_{\updownarrow \text{opt}}(V_{\updownarrow}) &= [IVI_{\downarrow \text{opt}, V_{\uparrow}}(V_{\downarrow}), IVI_{\uparrow \text{opt}}(V_{\uparrow})] \\ IVI_{\downarrow \text{opt}}(V_{\updownarrow}) &= IVI_{\downarrow \text{opt}, V_{\uparrow}}(V_{\downarrow}) \end{aligned} \quad (30)$$

In analyzing $IVI_{\updownarrow \text{opt}}$, we also use the notation defined in Section 4 for the set of actions that maximize the upper bound at each state. We restate the relevant definition here for convenience. For a given value function V, we write ρ_V for the func-

$IVI_{\updownarrow \text{opt}}(V_{\updownarrow})$

we assume that $V_{\updownarrow}$ is represented as:
\\ *$V_{\downarrow}$ is a vector of n real numbers giving lower-bounds for states q_1 to q_n*
\\ *$V_{\uparrow}$ is a vector of n real numbers giving upper-bounds for states q_1 to q_n*

{ Create O, a vector of n states for holding a permutation of the states q_1 to q_n

first, compute new lower bounds
O = sort_increasing_order(q_1,...,q_n,$<_{lb}$); \\ $<_{lb}$ *compares state lower-bounds*
VI-Update($V_{\downarrow}$, O);

second, compute new upper bounds
O = sort_decreasing_order(q_1,...,q_n,$<_{ub}$); \\ $<_{ub}$ *compares state upper-bnds*
VI-Update($V_{\uparrow}$, O)}

==========

\\ VI-Update(v, o) *updates v using the order-maximizing MDP for o*
\\ *o is a state ordering—a vector of states (a permutation of q_1,...,q_n)*
\\ *v is a value function—a vector of real numbers of length n*

VI-Update(v, o)
{ Create F_a, a matrix of n by n real numbers for each action a

\\ *the next loop sets each F_a to describe a in the order-maximizing MDP for o*
for each state p and action a {
used = $\sum_{\text{state } q} F_{\downarrow p,q}(a)$;
remaining = 1 – used;

\\ *distribute remaining probability mass to states earlier in ordering*
for i=1 to n { \\ *i is used to index into ordering o*
min = $F_{\downarrow p,o(i)}(a)$;
desired = $F_{\uparrow p,o(i)}(a)$;
if (desired <= remaining)
then $F_a(p,o(i))$ = min+desired;
else $F_a(p,o(i))$ = min+remaining;
remaining = max(0,remaining-desired)}}

\\ *F_a now describes a in the order-maximizing MDP w/respect to O,*
\\ *finally, update v using a value iteration-like update based on F'*
for each state p
$v(p) = \max_{a \in A} \left[R(p) + \gamma \sum_{\text{state } q} F_a(p,q)\, v(q) \right]$

Figure 8: Pseudocode: an iteration of optimistic interval value iteration ($IVI_{\updownarrow \text{opt}}$)

tion from states to sets of actions such that for each state p,

$$\rho_V(p) = \operatorname*{argmax}_{\alpha \in A} \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V)(p) \qquad (31)$$

Likewise, for the pessimistic case, we defined σ_V in Section 4.

Given the definition of $\leq_{\text{opt}}$, it is straightforward to show the following lemma.

Lemma 4: For any value functions V, V' and state p,

$$\begin{aligned} IVI_{\downarrow \text{opt}, V}(V')(p) &= \max_{\alpha \in \rho_V(p)} \min_{M \in M_\updownarrow} VI_{M,\alpha}(V')(p) \\ IVI_{\uparrow \text{pes}, V}(V')(p) &= \max_{\alpha \in \sigma_V(p)} \min_{M \in M_\updownarrow} VI_{M,\alpha}(V')(p) \end{aligned} \tag{32}$$

Proof: By inspection of the definitions of $IVI_{\updownarrow \text{opt}}$ and $IVI_{\updownarrow \text{pes}}$.

❑ (Lemma 4).

We now show that for each V, $IVI_{\downarrow \text{opt}, V}$ is a contraction mapping relative to the sup norm, and thus converges to a unique fixed point, as desired. Theorem 9 then implies that $V_{\updownarrow \text{opt}}$ is the unique fixed-point found. ($V_{\updownarrow \text{pes}}$ in the case of $IVI_{\updownarrow \text{pes}}$). We then show at that at any point after polynomially many iterations of $IVI_{\updownarrow \text{opt}}$, the resulting interval value function $V_\updownarrow$ has upper bounds $V_\uparrow$ that have converged to a fixed point of $IVI_{\uparrow \text{opt}}$, and thus further iteration of $IVI_{\updownarrow \text{opt}}$ is equivalent to iterating $IVI_{\uparrow \text{opt}}$ and $IVI_{\downarrow \text{opt}, V_\uparrow}$ together in parallel to generate the upper and lower bounds, respectively. We can also show that for any V, polynomially many iterations of $IVI_{\downarrow \text{opt}, V}$ suffice for convergence to a fixed point. Similar results hold for $IVI_{\updownarrow \text{pes}}$. We now give the details of these results.

Theorem 13:

(a) $IVI_{\uparrow \text{opt}}$ and $IVI_{\downarrow \text{pes}}$ are contraction mappings.

(b) For any value function V and associated action set selection function ρ_V and σ_V, $IVI_{\downarrow \text{opt}, V}$ and $IVI_{\uparrow \text{pes}, V}$ are contraction mappings.

Proof: See Appendix.

Theorem 14: For fixed γ, polynomially many iterations of $IVI_{\updownarrow \text{opt}}$ can be used to find $V_{\updownarrow \text{opt}}$, and polynomially many iterations of $IVI_{\updownarrow \text{pes}}$ can be used to find $V_{\updownarrow \text{pes}}$, with both polynomials defined relative to the problem size including the number of bits used in specifying the parameters.

Proof: (sketch)

The argument here is exactly as in Theorem 12, relying on Theorems 9 and 13, except that the iterations must be taken to convergence in two stages. Considering $IVI_{\updownarrow \text{opt}}$, we must first iterate until the upper bound has converged, with the polynomial-time bound on iterations deriving by a similar argument to the

proof of Theorem 12; then once the upper bounds have converged we must then iterate until the lower bounds have converged, again in polynomially many iterations by another argument similar to that in the proof of Theorem 12.

More precisely, let $V_{\updownarrow 1}, V_{\updownarrow 2}, \ldots$, be a sequence of interval value functions found by iterating $IVI_{\updownarrow \text{opt}}$, so that for each i greater or equal to 1 we have $V_{\updownarrow i+1}$ equal to $IVI_{\updownarrow \text{opt}}(V_{\updownarrow i})$. Then an argument similar to the proof of Theorem 12 guarantees that for some j polynomial in the size of the problem, $V_{\updownarrow j}$ must have upper bounds that are equal to the true fixed point upper bound values, up to the maximum precision of the true fixed point. We then know that truncating the upper value bounds in $V_{\updownarrow j}$ to that precision (to get an interval value function $V_{\updownarrow 1}'$) gives the true fixed point upper bound values. We can then iterate $IVI_{\updownarrow \text{opt}}$ starting on $V_{\updownarrow 1}'$ to get another sequence of value functions where the upper bounds are unchanging and the lower bounds are converging to the correct fixed point values in the same manner.

A similar argument shows polynomial convergence for $IVI_{\updownarrow \text{pes}}$.

❑ (Theorem 14).

6. Policy Selection

In this section, we consider the problem of selecting a policy based on the value bounds computed by our IVI algorithms. This section is not intended as an additional research contribution as much as a discussion of issues that arise in solving BMDP problems and of alternative approaches to policy selection (other than the optimistic and pessimistic approaches we take here). 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. We note again that in many applications it is possible to compute prior probabilities over these parameters, but that these computations are prohibitively expensive in our motivating application (solving large state-space problems by approximate state-space aggregation).

Despite the fact that a BMDP does not specify which particular MDP we are facing, 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 ultimately have to carry out the 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 maximize potential performance, we might assume that the outside process cooperates by choosing the MDP in order to help us; we can then select the policy that performs as well as possible

given that assumption. In contrast, we might minimize the risk of performing poorly by thinking in adversarial terms: we can select the policy that performs as well as possible under the assumption that an adversary chooses the MDP so that we perform as poorly as possible (in each case we assume that the MDP is chosen from the BMDP family of MDPs *after* the policy has been selected in order to minimize/maximize the value of that policy).

These choices correspond to optimistic and pessimistic optimal policies as defined above. 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 that achieve those values.

We note that it may seem unnatural to be required to take an optimistic or a pessimistic approach in order to select a policy — certainly this is not analogous to policy selection for standard MDPs. This requirement grows out of our model assumption that we have no prior probabilities on the model parameters, and we have argued that this assumption is in fact natural at very least in our motivating domain of approximate state-space aggregation. The same assumption is also natural in performing sensitivity analysis, as described in the next section. We also note that there is precedent in the related MDP literature for considering optimistic and pessimistic approaches to policy selection in the face of uncertainty about the model; see, for example, the work of Satia and Lave in [15].

Alternative approaches to selecting a policy are possible, but some approaches that seem natural at first run into trouble. For instance, we might consider placing a uniform prior probability on each model parameter within its specified interval. Unfortunately, the model parameters cannot in general be selected independently (because they must together represent a well-formed probability distribution after selection), and there may not even be any joint prior distribution over the parameters which marginalizes to the uniform distribution over the provided intervals when marginalized to each parameter. Therefore, the uniform distribution over the provided intervals does not enjoy any distinguished status — it may not even correspond to a well-formed prior over the underlying MDPs in the BMDP family.

There are other well-formed choices corresponding to other means of totally ordering real closed intervals (other than $\leq_{opt}$ and $\leq_{pes}$). For instance, we might order intervals by their midpoints, asserting a preference for states where the highest and lowest value possible in the underlying MDP family have a high mean. It is not clear when this choice might be prefered; however, we believe our methods can be naturally adapted to compute optimal policy values for other interval orderings, if desired.

A natural goal would be to find 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. Moreover, it is not at all clear how to find such a policy — our methods do not appear to generalize in this direction. As noted just above, this goal does *not* correspond to assuming a uniform prior over the model parameters, but rather a more complex joint distribution over the parameters. Also, this average case solution would not in general provide useful information in our motivating application of state-space aggregation: we would have no guarantee that the uniform prior over MDP models consistent with the BMDP had any useful correlation with the original large MDP that aggregated to the BMDP. In contrast, as discussed below, the optimistic and pessimistic bounds we compute apply directly to any MDP when the BMDP analyzed is formed by state-space aggregation of that MDP. Nevertheless, the question of how to compute the optimal average case policy for a BMDP appears to be a useful direction for future research.

7. Prototype Implementation Results and Potential Applications

In this section we discuss our intended applications for the new BMDP algorithms, and present empirical results from a prototype implementation of the algorithms for use in state-space aggregation. We note that no particular difficulties were encountered in implementing the new BMDP algorithms — implementation is more demanding than that of standard MDP algorithms, but only by the addition of a sorting algorithm.

Sensitivity Analysis. One way in which bounded-parameter MDPs might be useful in planning under uncertainty might 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 assess this possible sensitivity to the parameters, we might perturb the MDP parameters and evaluate 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. We note that this use provides our original motivation for developing BMDPs, and therefore it is this use that we give prototype empirical results for below.

In the state-aggregate interpretation of a BMDP, 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 $M_{\updownarrow}$ 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 $F_{\updownarrow B_i B_j}(\alpha)$ represent the interval value for the transition from B_i to B_j on action α defined as follows:

$$F_{\updownarrow B_i B_j}(\alpha) = \left[\min_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha), \max_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha)\right] \tag{33}$$

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.* [10] 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). In particular, the lower interval bounds computed on a given state block by $IVI_{\downarrow_{\text{pes}}}$ give lower bounds on the optimal value for states in that block in the original MDP; likewise, the upper interval bounds computed by $IVI_{\uparrow_{\text{opt}}}$ give upper bounds on the optimal value in the original MDP.

Empirical Results. We constructed a prototype implementation of our BMDP algorithms, interval value iteration and interval policy evaluation. We then used this implementation in conjunction with implementations of our previously presented approximate state-space aggregation algorithms [10] in order to compute lower and upper bounds on the values of individual states in large MDP problems.

The MDP problems used were derived by partially modelling air campaign planning problems using implicit MDP representations. These problems involve selecting tasks for a variety of military aircraft over time in order to maximize the utility of their actions, and require modeling many aspects of the aircraft capabilities, resources, crew, and tasks. Modeling the full problem as an MDP is still out of reach — the MDP models used in these experiments were constructed by representing the problem at varying degrees of (extremely coarse) abstraction so that the resulting problem would be within reach of our prototype implementation.

We show in Table 15 the original problem state-space size, the state-space size

Table 15: Model Size after Approximate Minimization

#State Vars	# States	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.1$	$\varepsilon = 0.3$	$\varepsilon = 0.5$	$\varepsilon = 0.8$
9	512	114	114	72	24	11	8
10	1024	131	122	85	55	21	21
13	8192	347	347	272	148	66	63
14	16384			442	153	67	63
15	32768			520	152	88	69
IVI Inaccuracy:		0%	0.2%	10%	40%	58%	62%

of the BMDP that results from our aggregation algorithm, and the quality of the resulting state-value bounds for several different sized MDP problems. Each row in the table corresponds to a specific explicit MDP that we solved (approximately and/or exactly) using state-space aggregation. We note that one parameter (ε) of our aggregation method is the degree of approximation tolerated in transition probability — this corresponds to the interval width in the BMDP parameter intervals. As this parameter is given larger and larger values across the columns of the table, the aggregate BMDP model has fewer and fewer states — in return, the value bounds obtained are less and less tight. The quality of the resulting state-value bounds is given by showing "IVI Inaccuracy" — this percentage is the average width of the value intervals computed as a percentage of the difference between the lowest possible state value and the highest possible state value (these are defined by assuming a repeated occurence of the lowest/highest reward available for an infinite time period and computing the total discounted reward obtained). Our prototype aggregation code was incapable of handling the exact and near-exact analysis of the largest models tried, and those entries in the table are therefore missing.

We note that IVI inaccuracies of much greater than 25% may not represent very useful bounds on state value (we have not yet conducted experiments to evaluate this question). For this reason, the last three columns of the table are shown primarily for completeness and to satisfy curiosity. However, an inaccuracy of 10% can be expected to yield useful information in selecting between different control actions — we can think of this level of inaccuracy as allowing us to rate each state on a scale of one to ten as to how good its value is. Such ratings should be very useful in designing control policies.

We note that our prototype code is not optimized in its handling of either space or time. Similar prototype code for explicit MDP problems can handle no more than a few hundred states. Production versions of explicit MDP code today can handle as many as a million or so states. Our aggregation and BMDP algorithms, even in this unoptimized form, are able to obtain nontrivial bounds on state value for state-space sizes involving thousands of states. We believe that a production

version of these algorithms could derive near-optimal policies for MDP planning problems involving hundreds of millions of states.

8. 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 of the closest such ideas. First, BMDPs specialize the MDPs with imprecisely known parameters (MDPIPs) described and analyzed in the operations research literature by White and Eldeib [17], [18], and Satia and Lave [15]. The more general MDPIPs described in these papers require more general and expensive algorithms for solution. For example, [17] 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 as we do here. [15] mentions the restriction to BMDPs but gives 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 [15] in defining optimistic and pessimistic optimal policies. In summary, when uncertainty about MDP parameters is such that a BMDP model is appropriate, the MDPIP literature does not provide an approach that exploits the restricted structure to achieve an efficient method (we note appealing to linear programming at each iteration can be very expensive).

Shapley [16] introduced the notion of *stochastic games* to describe two-person games in which the transition probabilites are controlled by the two players. MDPIPs, and therefore BMDPs, are a special case of *alternating* stochastic games in which the first player is the decision-making agent and the second player, often considered as either an adversary or advocate, makes its move by choosing from the set of possible MDPs consistent with having seen the agent's move.

Bertsekas and Castañon [3] 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 [13] describes their use in solving partially observable MDPs. Puterman [14] provides an excellent introduction to Markov decision processes and techniques involving bounding value functions.

Boutilier, Dean and Hanks [5] provide a careful treatment of MDP-related methods demonstrating how they provide a unifying framework for modeling a

wide range of problems in AI involving planning under uncertainty. This paper also describes such related issues as state space aggregation, decomposition and abstraction as these ideas pertain to work in AI. We encourage the reader unfamiliar with the connection between classical planning methods in AI and Markov decision processes to refer to this paper.

Boutilier and Dearden [6] and Boutilier *et. al.* [8] describe methods for solving implicitly described MDPs using dynamic aggregation — in their methods the state space aggregates vary over the iterations of the dynamic programming algorithm. This work can be viewed as using a compact representation of both policies and value functions in terms of state aggregates to perform the familiar dynamic programming algorithms. Dean and Givan [9] reinterpret this work in terms of computing explicitly described MDPs with aggregate states corresponding to the aggregates that the above compactly represented value functions use when they have converged. Dean, Givan, and Leach [10] discuss relaxing these aggregation techniques to construct approximate aggregations — it is from this work that the notion of BMDP emerged in order to represent the resulting aggregate models.

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.

9. Acknowledgements

Many thanks to Michael Littman for useful conversation and insights concerning the proofs that the algorithms herein run in polynomial time.

10. References

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

[2] Bertsekas, D. P. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[3] Bertsekas, D. P. and Castañon, D. A., "Adaptive Aggregation for Infinite Horizon Dynamic Programming," *IEEE Transactions on Automatic Control*, 1989, 34(6):589-598.

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

[5] Boutilier, C., Dean, T. L., Hanks, S. "Decision Theoretic Planning: Structural Assumptions and Computational Leverage," *Journal of Artificial Intelligence Research*, 1999, 11:1–94.

[6] Boutilier, C. and Dearden, R., "Using Abstractions for Decision Theoretic Planning with Time Constraints," *Proceedings of AAAI-94,* Seattle, Washington, 1994, pp. 1016-1022.

[7] Boutilier, Craig, Thomas Dean, and Steve Hanks. "Planning Under Uncertainty: Structural Assumptions and Computational Leverage," *Proceedings of the Third European Workshop on Planning*. Assisi, Italy, 1995.

[8] Boutilier, C., Dearden, R., and Goldszmidt, M., "Exploiting Structure in Policy Construction," *Proceedings of IJCAI 14*, Montreal, Canada, 1995, pp. 1104-1111.

[9] Dean, T. and Givan, R., "Model Minimization in Markov Decision Processes," *Proceedings of AAAI-97*, Providence, RI, 1997.

[10] Dean, T., Givan, R., and Leach, S. "Model Reduction Techniques for Computing Approximately Optimal Solutions for Markov Decision Processes," *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence,* Providence, RI, Aug. 1997, pp. 124-131.

[11] Howard, Ronald A. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, Massachusetts, 1960.

[12] Littman, M. L., Dean, T. L., and Kaelbling, L. P., "On the complexity of solving Markov decision problems," *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montreal, Québec, Canada, 1995.

[13] Lovejoy, William S., "Computationally Feasible Bounds for Partially Observed Markov Decision Processes," *Operations Research,* 1991, 39(1):162-175.

[14] Puterman, M. *Markov Decision Processes — Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, 1994.

[15] Satia, J. K. and Lave, R. E., "Markovian Decision Processes with Uncertain Transition Probabilities," *Operations Research*, 1973, 21:728-740.

[16] Shapley, L. S. "Stochastic Games," *Proceedings of the National Academy of Sciences of the United States of America,* 39:1095-1100, 1953.

[17] White, C. C., and Eldeib, H. K., "Parameter Imprecision in Finite State, Finite Action Dynamic Programs," *Operations Research*, 1986, 34: 120-129.

[18] White, C. C., and Eldeib, H. K., "Markov Decision Processes with Imprecise Transition Probabilities," *Operations Research*, 1994, 43:739-749.

11. Appendix — Proofs Omitted Above for Readability

Lemma 1: For any policy $\pi \in \Pi$, MDP $M \in M_{\updownarrow}$, and value function $v \in \bar{V}$,

(a) there are MDPs $M_1 \in X_{M_{\updownarrow}}$ and $M_2 \in X_{M_{\updownarrow}}$ such that

$$V_{M_1,\pi} \leq_{\text{dom}} V_{M,\pi} \leq_{\text{dom}} V_{M_2,\pi} \,. \tag{34}$$

(b) Also, there are MDPs $M_3 \in X_{M_{\updownarrow}}$ and $M_4 \in X_{M_{\updownarrow}}$ such that

$$VI_{M_3,\pi}(v) \leq_{\text{dom}} VI_{M,\pi}(v) \leq_{\text{dom}} VI_{M_4,\pi}(v) \,. \tag{35}$$

Proof: To show the existence of M_1, let $O = q_1, \ldots, q_k$ be an ordering on states such that for all i and j if $1 \leq i \leq j \leq k$ then $V_{M,\pi}(q_i) \leq V_{M,\pi}(q_j)$ (increasing order). Note that ties in state values permit different orderings; for the proof, it is sufficient to chose one ordering arbitrarily. Consider $M_O \in X_{M_{\updownarrow}}$, the order-maximizing MDP of O. M_O is constructed so as to send as much probability mass as possible to states earlier in the ordering O, i.e. to those states q with lower value $V_{M,\pi}(q)$. It follows that for any state p,

$$\sum_{q \in Q} \left(F^{M_O}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \leq \sum_{q \in Q} \left(F^{M}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \tag{36}$$

Thus, for any state p,

$$V_{M,\pi}(p) = R(p) + \gamma \sum_{q \in Q} \left(F^{M}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \tag{37}$$

$$\geq R(p) + \gamma \sum_{q \in Q} \left(F^{M_O}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \tag{38}$$

$$= VI_{M_O,\pi}(V_{M,\pi})(p) \tag{39}$$

By Theorem 6, these lines imply $V_{M_O,\pi} \leq_{\text{dom}} V_{M,\pi}$, as desired.

The existence of M_2 can be shown in the same except that O is chosen to order the states by increasing value. Thus M_O is constructed so that

$$\sum_{q \in Q} \left(F^{M}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \leq \sum_{q \in Q} \left(F^{M_O}_{pq}(\pi(p)) V_{M,\pi}(q) \right) \,. \tag{40}$$

Part (b) is shown in the same manner as part (a) except that we replace each occurrence of $V_{M,\pi}(p)$ with $VI_{M,\pi}(v)(p)$ and each occurrence of $V_{M,\pi}(q)$ with $v(q)$.

❑ (Lemma 1)

Lemma 2: Let π be a policy in Π and M_1, M_2 be MDPs in $M_\updownarrow$.

(a) For $M_3 = M_1 \oplus^{\pi}_{\max} M_2$,

$$V_{M_3,\pi} \geq_{\text{dom}} V_{M_1,\pi} \text{ and } V_{M_3,\pi} \geq_{\text{dom}} V_{M_2,\pi}, \text{ and} \tag{41}$$

(b) for $M_3 = M_1 \oplus^{\pi}_{\min} M_2$,

$$V_{M_3,\pi} \leq_{\text{dom}} V_{M_1,\pi} \text{ and } V_{M_3,\pi} \leq_{\text{dom}} V_{M_2,\pi}. \tag{42}$$

Proof: Part (a): We construct a value function v such that $v \geq_{\text{dom}} V_{M_1,\pi}$, $v \geq_{\text{dom}} V_{M_2,\pi}$, and $v \leq_{\text{dom}} V_{M_3,\pi}$, as follows. For each $p \in Q$, let

$$v(p) = \max(V_{M_1,\pi}(p), V_{M_2,\pi}(p)) \tag{43}$$

Note that this implies $v \geq_{\text{dom}} V_{M_1,\pi}$ and $v \geq_{\text{dom}} V_{M_2,\pi}$. We now show using Theorem 6 that $v \leq_{\text{dom}} V_{M_3,\pi}$. By Theorem 6 it suffices to prove that $v \leq_{\text{dom}} VI_{M_3,\pi}(v)$, which we now do by showing $v(p) \leq VI_{M_3,\pi}(v)(p)$ for arbitrary $p \in Q$.

Case 1. We suppose $V_{M_1,\pi}(p) \geq V_{M_2,\pi}(p)$.

From Equation 43 we then have that $v(p) = V_{M_1,\pi}(p)$. By the definition of $\oplus^{\pi}_{\max}$, we know $F^{M_3}_{pq}(\pi(p)) = F^{M_1}_{pq}(\pi(p))$ when $V_{M_1,\pi}(p) \geq V_{M_2,\pi}(p)$ as in this case. This fact, together with the definitions of VI, $V_{M_1,\pi}$, $\oplus^{\pi}_{\max}$, and v allow the following chain of equations to conclude the proof of case 1:

$$\begin{aligned} v(p) &= V_{M_1,\pi}(p) \\ &= R(p) + \gamma \sum_{q \in Q} F^{M_1}_{pq}(\pi(p)) V_{M_1,\pi}(q) \\ &\leq R(p) + \gamma \sum_{q \in Q} F^{M_1}_{pq}(\pi(p)) v(q) \\ &= R(p) + \gamma \sum_{q \in Q} F^{M_3}_{pq}(\pi(p)) v(q) \\ &= VI_{M_3,\pi}(v)(p) \end{aligned} \tag{44}$$

Case 2: Suppose $V_{M_1,\pi}(p) < V_{M_2,\pi}(p)$.

We then have $F_{pq}^{M_3}(\pi(p)) = F_{pq}^{M_2}(\pi(p))$ by the definition of $\oplus_{\max}^{\pi}$, and $v(p) = V_{M_1,\pi}$ by the definition of v, and Equation 44 holds with M_1 replaced by M_2, as desired, concluding the proof of part (a).

Part (b): The proof is exactly dual to part (a) by replacing "max" with "min, $\leq$ with $\geq$ (and vice versa), and $<$ with $>$.

❑ (Lemma 2).

Lemma 3: Given a BMDP $M_\updownarrow$, and policies $\pi_1, \pi_2 \in \Pi$, $\pi_3 = \pi_1 \oplus_{\text{opt}} \pi_2$, and $\pi_4 = \pi_1 \oplus_{\text{pes}} \pi_2$,

(a) $V_{\uparrow\pi_3} \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $V_{\uparrow\pi_3} \geq_{\text{dom}} V_{\uparrow\pi_2}$
(b) If $V_{\uparrow\pi_1} = V_{\uparrow\pi_2}$ then $V_{\updownarrow\pi_3} \geq_{\text{opt}} V_{\updownarrow\pi_1}$ and $V_{\updownarrow\pi_3} \geq_{\text{opt}} V_{\updownarrow\pi_2}$
(c) $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_1}$ and $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_2}$
(d) If $V_{\downarrow\pi_1} = V_{\downarrow\pi_2}$ then $V_{\updownarrow\pi_4} \geq_{\text{pes}} V_{\updownarrow\pi_1}$ and $V_{\updownarrow\pi_4} \geq_{\text{pes}} V_{\updownarrow\pi_2}$.

Proof: Part (a): We prove part (a) of the lemma by constructing a value function v such that $v \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $v \geq_{\text{dom}} V_{\uparrow\pi_2}$. We then show that $v \leq_{\text{dom}} V_{\uparrow\pi_3}$ using Theorem 6. We construct v as follows. Let $v(p) = \max(V_{\uparrow\pi_1}(p), V_{\uparrow\pi_2}(p))$ for each $p \in Q$.

This construction implies that $v \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $v \geq_{\text{dom}} V_{\uparrow\pi_2}$. We now show $v \leq_{\text{dom}} V_{\uparrow\pi_3}$ by giving an MDP M_3 for which $V_{M_3,\pi_3} \geq_{\text{dom}} v$. Using Theorem 6 it suffices to show that $VI_{M_3,\pi_3}(v) \geq_{\text{dom}} v$.

Let $M_1 \in M_\updownarrow$ be a π_1-maximizing MDP, and $M_2 \in M_\updownarrow$ be a π_2-maximizing MDP. Note that this implies that $V_{\uparrow\pi_1} = V_{M_1,\pi_1}$ and $V_{\uparrow\pi_2} = V_{M_2,\pi_2}$.

We now construct $M_3 \in M_\updownarrow$ as follows: for each p, q, α,

$$F_{pq}^{M_3}(\alpha) = \begin{cases} F_{pq}^{M_1}(\alpha) & \text{if } V_{\updownarrow\pi_1}(p) \geq_{\text{opt}} V_{\updownarrow\pi_2}(p) \\ F_{pq}^{M_2}(\alpha) & \text{otherwise} \end{cases}$$

It remains to show that $VI_{M_3,\pi_3}(v)(p) \geq v(p)$ for all $p \in Q$. Now fix an arbitrary $p \in Q$.

Case 1: Suppose $V_{\updownarrow\pi_1}(p) \geq_{\text{opt}} V_{\updownarrow\pi_2}(p)$.

Then by the definition of $\oplus_{\text{opt}}$, $\pi_3(p) = \pi_1(p)$. Also, by the definition of $\geq_{\text{opt}}$, $V_{\uparrow\pi_1}(p) \geq V_{\uparrow\pi_2}(p)$, and so $v(p) = V_{M_1,\pi_1}(p)$ is true, and by the definition of M_3, $F_{pq}^{M_3}(\pi_3(p)) = F_{pq}^{M_1}(\pi_3(p))$. The following inequations thus hold:

$$v(p) = V_{\uparrow\pi_1}(p) \tag{45}$$

$$= R(p) + \gamma \sum_{q \in Q} (F_{pq}^{M_1}(\pi_1(p)) V_{\uparrow \pi_1}(q)) \tag{46}$$

$$= R(p) + \gamma \sum_{q \in Q} (F_{pq}^{M_3}(\pi_3(p)) V_{\uparrow \pi_1}(q)) \tag{47}$$

$$\leq R(p) + \gamma \sum_{q \in Q} (F_{pq}^{M_3}(\pi_3(p)) v(q)) \tag{48}$$

$$= VI_{M_3, \pi_3}(v)(p) \tag{49}$$

Case 2: Suppose $V_{\updownarrow \pi_1}(p) <_{\text{opt}} V_{\updownarrow \pi_2}(p)$.

Then by the definition of $\oplus_{\text{opt}}$, $\pi_3(p) = \pi_2(p)$. Also, by the definition of $\geq_{\text{opt}}$, $V_{\uparrow \pi_1}(p) \leq V_{\uparrow \pi_2}(p)$, and so $v(p) = V_{M_2, \pi_2}(p)$ is true, and by the definition of M_3, $F_{pq}^{M_3}(\pi_3(p)) = F_{pq}^{M_2}(\pi_3(p))$. Then Equation 45 thru Equation 49 hold with M_2 and π_2 in place of M_1 and π_1 respectively, yielding again that $v(p) \leq VI_{M_3, \pi_3}(v)(p)$, as desired.

Case 1 and Case 2 together imply that $v(p) \leq VI_{M_3, \pi_3}(v)(p)$ for all $p \in Q$, which with Theorem 6 implies part (a) of the lemma.

Proof: Part (b): Supposing that $V_{\uparrow \pi_1} = V_{\uparrow \pi_2}$, we show $V_{\updownarrow \pi_3} \geq_{\text{opt}} V_{\updownarrow \pi_1}$ and $V_{\updownarrow \pi_3} \geq_{\text{opt}} V_{\updownarrow \pi_2}$. From part (a) of the theorem, we know that $V_{\uparrow \pi_3} \geq_{\text{dom}} V_{\uparrow \pi_1}$ and $V_{\uparrow \pi_3} \geq_{\text{dom}} V_{\uparrow \pi_2}$. It suffices to prove in addition that $V_{\downarrow \pi_3} \geq_{\text{dom}} V_{\downarrow \pi_1}$ and $V_{\downarrow \pi_3} \geq_{\text{dom}} V_{\downarrow \pi_2}$. We show both by defining $v(p) = \max(V_{\downarrow \pi_1}(p), V_{\downarrow \pi_2}(p))$ for each state $p \in Q$, observing that $v \geq_{\text{dom}} V_{\downarrow \pi_1}$ and $v \geq_{\text{dom}} V_{\downarrow \pi_2}$, and then showing that $V_{\downarrow \pi_3} \geq_{\text{dom}} v$.

We can show $V_{\downarrow \pi_3} \geq_{\text{dom}} v$ by showing that for arbitrary $M \in M_{\updownarrow}$, $V_{M, \pi_3} \geq_{\text{dom}} v$. By Theorem 6 it suffices to show that for arbitrary state $p \in Q$, $VI_{M, \pi_3}(v)(p) \geq v$. We divide now into two cases:

Case 1: Suppose $V_{\downarrow \pi_1}(p) \geq V_{\downarrow \pi_2}(p)$.
With the part (b) assumption ($V_{\uparrow \pi_1} = V_{\uparrow \pi_2}$), this implies $V_{\updownarrow \pi_1}(p) \geq_{\text{opt}} V_{\updownarrow \pi_2}(p)$.
Then by the definition of $\oplus_{\text{opt}}$, $\pi_3(p) = \pi_1(p)$. Also by definition in this case $v(p) = V_{\downarrow \pi_1}(p)$. Let M_1 be a π_1-minimizing MDP. The following inequation chain gives the desired conclusion:

$$v(p) = V_{\downarrow \pi_1}(p) \tag{50}$$

$$= R(p) + \gamma \sum_{q \in Q} F_{pq}^{M_1}(\pi_1(p)) V_{\downarrow \pi_1}(q) \tag{51}$$

$$\leq R(p) + \gamma \sum_{q \in Q} F^{M}_{pq}(\pi_1(p)) V_{\downarrow \pi_1}(q) \tag{52}$$

$$\leq R(p) + \gamma \sum_{q \in Q} F^{M}_{pq}(\pi_3(p)) v(q) \tag{53}$$

$$\leq VI_{M, \pi_3}(v)(p) \tag{54}$$

Line 52 requires some justification. Consider an MDP M_1' defined to agree with M_1 everywhere except that $F^{M_1'}_{pq} = F^{M}_{pq}$ for every $q \in Q$. If Line 52 did not hold, we would have $VI_{M_1', \pi_1}(V_{\downarrow \pi_1}) <_{\text{dom}} V_{\downarrow \pi_1}$ and then Theorem 6 could be used to show that $V_{M_1', \pi_1} <_{\text{dom}} V_{\downarrow \pi_1}$, contradicting the definition of $V_{\downarrow \pi_1}$.

Case 2: Suppose $V_{\downarrow \pi_1}(p) < V_{\downarrow \pi_2}(p)$.
With the part (b) assumption this implies that $V_{\updownarrow \pi_1}(p) <_{\text{opt}} V_{\updownarrow \pi_2}(p)$.
Then by the definition of $\oplus_{\text{opt}}$, $\pi_3(p) = \pi_2(p)$. Also $v(p) = V_{\downarrow \pi_2}(p)$. Let M_2 be a π_2-minimizing MDP. Equations 50 through 54 now hold with M_1 and π_1 replaced by M_2 and π_2, respectively.

We have now shown in both cases that $v(p) \leq VI_{M, \pi_3}(v)(p)$, as desired, concluding the proof of part (b) of the theorem.

Proof: Part (c): We prove part (c) of the lemma by constructing a value function v such that $v \geq_{\text{dom}} V_{\downarrow \pi_1}$ and $v \geq_{\text{dom}} V_{\downarrow \pi_2}$. We then show that $v \leq_{\text{dom}} V_{\downarrow \pi_4}$ using Theorem 6. We construct v as follows. Let $v(p) = \max(V_{\downarrow \pi_1}(p), V_{\downarrow \pi_2}(p))$ for each $p \in Q$. This implies $v \geq_{\text{dom}} V_{\downarrow \pi_1}$ and $v \geq_{\text{dom}} V_{\downarrow \pi_2}$. We now show $v \leq_{\text{dom}} V_{\downarrow \pi_4}$ by showing that for arbitrary $M \in M_{\updownarrow}$, $V_{M, \pi_4} \geq_{\text{dom}} v$. Using Theorem 6 it suffices to show that $VI_{M, \pi_4}(v) \geq_{\text{dom}} v$.

Let $M_1 \in M_{\updownarrow}$ be a π_1-minimizing MDP, and $M_2 \in M_{\updownarrow}$ be a π_2-minimizing MDP. Note that this implies that $V_{\downarrow \pi_1} = V_{M_1, \pi_1}$ and $V_{\downarrow \pi_2} = V_{M_2, \pi}$.

Now fix an arbitrary $p \in Q$, and show that $VI_{M, \pi_4}(v)(p) \geq v(p)$.

Case 1: Suppose $V_{\updownarrow \pi_1}(p) \geq_{\text{pes}} V_{\updownarrow \pi_2}(p)$.

Then by the definition of $\oplus_{\text{pes}}$, $\pi_4(p) = \pi_1(p)$. Also, by the definition of $\geq_{\text{pes}}$, $V_{\downarrow \pi_1}(p) \geq V_{\downarrow \pi_2}(p)$, and so $v(p) = V_{M_1, \pi_1}(p)$ is true. Equations 50 through 54 now hold with π_4 in place of π_3, giving the desired result.

Case 2: Suppose $V_{\updownarrow \pi_1}(p) <_{\text{pes}} V_{\updownarrow \pi_2}(p)$.

Then by the definition of $\oplus_{\text{pes}}$, $\pi_4(p) = \pi_2(p)$. Also, by the definition of $\geq_{\text{pes}}$, $V_{\downarrow \pi_1}(p) \leq V_{\downarrow \pi_2}(p)$, and so $v(p) = V_{M_2, \pi_2}(p)$ is true. Then Equations 50 through 54 hold with M_2, π_2, and π_4 in place of M_1, π_1, and π_3, respectively, yielding again that $v(p) \leq VI_{M, \pi_4}(v)(p)$, as desired.

Case 1 and Case 2 together imply that $v(p) \leq VI_{M,\pi_4}(v)(p)$ for all $p \in Q$, which with Theorem 6 implies part (c) of the theorem.

Proof: Part (d): Supposing that $V_{\downarrow\pi_1} = V_{\downarrow\pi_2}$, we show $V_{\updownarrow\pi_4} \geq_{\text{pes}} V_{\updownarrow\pi_1}$ and $V_{\updownarrow\pi_4} \geq_{\text{pes}} V_{\updownarrow\pi_2}$. From part (c) of the theorem, we know that $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_1}$ and $V_{\downarrow\pi_4} \geq_{\text{dom}} V_{\downarrow\pi_2}$. It suffices to prove in addition that $V_{\uparrow\pi_4} \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $V_{\uparrow\pi_4} \geq_{\text{dom}} V_{\uparrow\pi_2}$. We show both by defining $v(p) = \max(V_{\uparrow\pi_1}(p), V_{\uparrow\pi_2}(p))$ for each state $p \in Q$, observing that $v \geq_{\text{dom}} V_{\uparrow\pi_1}$ and $v \geq_{\text{dom}} V_{\uparrow\pi_1}$, and then showing that $V_{\uparrow\pi_4} \geq_{\text{dom}} v$ by giving an MDP M_4 for which $V_{M_4,\pi_4} \geq_{\text{dom}} v$. Using Theorem 6 it suffices to show that $VI_{M_4,\pi_4}(v) \geq_{\text{dom}} v$.

Let $M_1 \in M_{\updownarrow}$ be a π_1-maximizing MDP, and $M_2 \in M_{\updownarrow}$ be a π_2-maximizing MDP. Note that this implies that $V_{\uparrow\pi_1} = V_{M_1,\pi_1}$ and $V_{\uparrow\pi_2} = V_{M_2,\pi_2}$.

We now construct $M_4 \in M_{\updownarrow}$ as follows: for each p, q, α,

$$F_{pq}^{M_4}(\alpha) = \begin{cases} F_{pq}^{M_1}(\alpha) & \text{if } V_{\updownarrow\pi_1}(p) \geq_{\text{pes}} V_{\updownarrow\pi_2}(p) \\ F_{pq}^{M_2}(\alpha) & \text{otherwise} \end{cases}$$

It remains to show that $VI_{M_4,\pi_4}(v)(p) \geq v(p)$ for all $p \in Q$. Now fix an arbitrary $p \in Q$.

Case 1: Suppose $V_{\uparrow\pi_1}(p) \geq_{\text{pes}} V_{\uparrow\pi_2}(p)$.
With the part (d) assumption this implies that $V_{\updownarrow\pi_1}(p) \geq_{\text{pes}} V_{\updownarrow\pi_2}(p)$.
Then by the definition of $\oplus_{\text{pes}}$, $\pi_4(p) = \pi_1(p)$. Also by definition in this case $v(p) = V_{\uparrow\pi_1}(p)$. Also, by the definition of M_4, $F_{pq}^{M_4}(\pi_4(p)) = F_{pq}^{M_1}(\pi_4(p))$. Equations 45 through 49 with π_3 and M_3 replaced by π_4 and M_4 complete the argument.

Case 2: Suppose $V_{\uparrow\pi_1}(p) < V_{\uparrow\pi_2}(p)$.
With the part (d) assumption this implies that $V_{\updownarrow\pi_1}(p) <_{\text{opt}} V_{\updownarrow\pi_2}(p)$.
Then by definition $\pi_4(p) = \pi_2(p)$. Also $v(p) = V_{\uparrow\pi_2}(p)$. Equations 45 through 49 now hold with M_1, π_1, and π_3 replaced by M_2, π_2, and π_4, respectively.

We have now shown in both cases that $v(p) \leq VI_{M_4,\pi_4}(v)(p)$, as desired, concluding the proof of part (d) of the theorem.

❑ (Lemma 3).

Theorem 9: For any BMDP $M_{\updownarrow}$, at every state p,

$$V_{\updownarrow\,\text{opt}}(p) = \max_{\alpha \in A,\ \leq_{\text{opt}}} \left[\min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow\text{opt}})(p), \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow\text{opt}})(p) \right], \quad (55)$$

and

$$V_{\updownarrow \text{pes}}(p) = \max_{\alpha \in A,\ \leq_{\text{pes}}} \left[\min_{M \in M_\updownarrow} VI_{M,\alpha}(V_{\downarrow \text{pes}})(p), \max_{M \in M_\updownarrow} VI_{M,\alpha}(V_{\uparrow \text{pes}})(p) \right]. \quad (56)$$

Proof: We consider the $V_{\updownarrow \text{opt}}$ version only. Throughout this proof we assume π_{opt} is an optimistically optimal policy for $M_\updownarrow$, which exists by Theorem 8. We suppose Equation 55 is false and show a contradiction. We have two cases:

Case 1: Suppose the upper bounds are not equal at some state p:

$$V_{\uparrow \text{opt}}(p) \neq \max_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M,\alpha}(V_{\uparrow \text{opt}})(p). \quad (57)$$

There are two ways this can happen:

Subcase 1a: Suppose there exist some MDP $M \in M_\updownarrow$ and action $\alpha \in A$ such that

$$V_{\uparrow \text{opt}}(p) < VI_{M,\alpha}(V_{\uparrow \text{opt}})(p) \quad (58)$$

We show how to construct a policy π whose interval value $V_{\updownarrow \pi}$ dominates $V_{\updownarrow \text{opt}}$ under $\leq_{\text{opt}}$, contradicting the definition of $V_{\updownarrow \text{opt}}$. Define π to be the same as π_{opt} except that $\pi(p) = \alpha$. By the definition of $V_{\updownarrow \pi_{\text{opt}}}$, there must exist $M' \in M_\updownarrow$ such that $V_{\uparrow \text{opt}} = V_{\uparrow \pi_{\text{opt}}} = V_{M', \pi_{\text{opt}}}$. From the theory of exact MDPs, we then have that:

$$V_{\uparrow \text{opt}} = V_{M', \pi_{\text{opt}}} = VI_{M', \pi_{\text{opt}}}(V_{M', \pi_{\text{opt}}}) = VI_{M', \pi_{\text{opt}}}(V_{\uparrow \text{opt}}). \quad (59)$$

Our subcase assumption implies

$$V_{\uparrow \text{opt}}(p) < VI_{M,\pi}(V_{\uparrow \text{opt}})(p). \quad (60)$$

Consider the MDP $M_3 \in M_\updownarrow$ with the same parameters as M' except at state p where the parameters are given by M. More formally,

$$F^{M_3}_{p'q'} = \begin{cases} F^{M}_{p'q'} & \text{when } p' = p \\ F^{M'}_{p'q'} & \text{otherwise} \end{cases} \quad (61)$$

This construction of M_3, together with Equation 59 and Equation 60, guarantees the following property of $V_{\uparrow \text{opt}}$:

$$V_{\uparrow \text{opt}} <_{\text{dom}} VI_{M_3,\pi}(V_{\uparrow \text{opt}}) \quad (62)$$

Equation 62 along with Theorem 6 implies that $V_{M_3,\pi} >_{\text{dom}} V_{\uparrow\text{opt}}$ and thus that $V_{\updownarrow\pi} >_{\text{opt}} V_{\updownarrow\text{opt}}$, contradicting the definition of $V_{\updownarrow\text{opt}}$ and concluding Subcase 1a.

Subcase 1b. Suppose that for every choice of $\alpha \in A$ and $M \in M_{\updownarrow}$

$$V_{\uparrow\text{opt}}(p) > VI_{M,\alpha}(V_{\uparrow\text{opt}})(p). \tag{63}$$

We obtain a contradiction directly by exhibiting α and $M \in M_{\updownarrow}$ in violation of this supposition. Let α be $\pi_{\text{opt}}(p)$. Let M be a π_{opt}-maximizing MDP in $M_{\updownarrow}$, which exists by Theorem 7. Our selection of π_{opt} guarantees that $V_{\uparrow\pi_{\text{opt}}} = V_{\uparrow\text{opt}}$, and our choice of M guarantees that $V_{M,\pi_{\text{opt}}} = V_{\uparrow\pi_{\text{opt}}}$. Equations 7 and 8 from the theory of exact MDPs then ensure that $V_{\uparrow\text{opt}}(p) = VI_{M,\alpha}(V_{\uparrow\text{opt}})(p)$, concluding case 1.

Case 2. Suppose at every state q the upper bounds are equal but at some state p the lower bounds are not equal:

$$\begin{aligned} \text{for all } q,\ V_{\uparrow\text{opt}}(q) &= \max_{\alpha \in A} \max_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\uparrow\text{opt}})(q), \text{ and} \\ V_{\downarrow\text{opt}}(p) &\neq \max_{\alpha \in \rho_{V_{\uparrow\text{opt}}}(p)} \min_{M \in M_{\updownarrow}} VI_{M,\alpha}(V_{\downarrow\text{opt}})(p) \end{aligned} \tag{64}$$

Note that the action selection in the second line of Equation 64 is restricted to range over those actions in $\rho_{V_{\uparrow\text{opt}}}(p)$ because those are the only actions that can be selected in Equation 55 due to the emphasis of $\leq_{\text{opt}}$ on upper bounds (the upper bounds achievable by an action primarily determine whether it is selected by the outer maximization in Equation 55, and only if the action is tied for the maximum upper bound, *i.e.* in $\rho_{V_{\uparrow\text{opt}}}(p)$, does its lower bound affect the maximization).

Again, there are two ways the second line of Equation 64 can hold.

Subcase 2a. Suppose $V_{\downarrow\text{opt}}(p)$ is too small, *i.e.*, there exists some action $\alpha \in \rho_{V_{\uparrow\text{opt}}}(p)$ such that for every MDP $M \in M_{\updownarrow}$, we have

$$V_{\downarrow\text{opt}}(p) < VI_{M,\alpha}(V_{\downarrow\text{opt}})(p). \tag{65}$$

We show a contradiction by giving a policy π whose interval value function is greater than $V_{\updownarrow\text{opt}}$ under the $\leq_{\text{opt}}$ ordering. Define π to be the same as π_{opt} except that $\pi(p) = \alpha$. By the definition of $V_{\updownarrow\pi_{\text{opt}}}$, there must exist $M' \in M_{\updownarrow}$ such that $V_{\uparrow\text{opt}} = V_{\uparrow\pi_{\text{opt}}} = V_{M',\pi_{\text{opt}}}$. As in Subcase 1a, we then have that:

$$V_{\uparrow\text{opt}} = V_{M',\pi_{\text{opt}}} = VI_{M',\pi_{\text{opt}}}(V_{M',\pi_{\text{opt}}}) = VI_{M',\pi_{\text{opt}}}(V_{\uparrow\text{opt}}). \tag{66}$$

From Equation 64 and $\alpha \in \rho_{V_{\uparrow\text{opt}}}(p)$ it follows that for some $M \in M_{\updownarrow}$,

$$V_{\uparrow\text{opt}}(p) = VI_{M,\alpha}(V_{\uparrow\text{opt}})(p), \tag{67}$$

and thus for $M_3 \in M_\updownarrow$ defined as in Subcase 1a to be equal to M' everywhere except at state p where M_3 is equal to M, we have

$$V_{\uparrow \mathrm{opt}} = VI_{M_3, \pi}(V_{\uparrow \mathrm{opt}}). \tag{68}$$

Therefore $V_{M_3, \pi} = V_{\uparrow \mathrm{opt}}$, and by the definitions of $V_{\updownarrow \mathrm{opt}}$ and $V_{\uparrow \pi}$, we then have that $V_{\uparrow \mathrm{opt}} \geq_{\mathrm{dom}} V_{\uparrow \pi} \geq_{\mathrm{dom}} V_{M_3, \pi} = V_{\uparrow \mathrm{opt}}$, and so $V_{\uparrow \pi}$ is equal to $V_{\uparrow \mathrm{opt}}$. We must now show that $V_{\downarrow \pi} >_{\mathrm{dom}} V_{\downarrow \mathrm{opt}}$ to conclude Subcase 2a. We show this by showing that for every MDP $M_4 \in M_\updownarrow$, $V_{\downarrow \mathrm{opt}} <_{\mathrm{dom}} VI_{M_4, \pi}(V_{\downarrow \mathrm{opt}})$ and using Theorem 6 to conclude $V_{M_4, \pi} >_{\mathrm{dom}} V_{\downarrow \mathrm{opt}}$ and thus $V_{\downarrow \pi} >_{\mathrm{dom}} V_{\downarrow \mathrm{opt}}$ as desired.

To conclude Subcase 2a, then, we must show $V_{\downarrow \mathrm{opt}} <_{\mathrm{dom}} VI_{M_4, \pi}(V_{\downarrow \mathrm{opt}})$. We show this by contradiction. Suppose this is false — then either $V_{\downarrow \mathrm{opt}} = VI_{M_4, \pi}(V_{\downarrow \mathrm{opt}})$, which our Subcase 2a assumption rules out at state p, or there must be some state q for which $V_{\downarrow \mathrm{opt}}(q) > VI_{M_4, \pi}(V_{\downarrow \mathrm{opt}})(q)$. Again our Subcase assumption rules this out for state p, so we know that q is not equal to p, and therefore by our choice of π we have that $\pi(q) = \pi_{\mathrm{opt}}(q)$, and thus that $V_{\downarrow \mathrm{opt}}(q) > VI_{M_4, \pi_{\mathrm{opt}}}(V_{\downarrow \mathrm{opt}})(q)$. We can now derive a contradiction by combining M_4 at state q with a π_{opt}-minimizing MDP M_5 at all other states to get an MDP $M_6 \in M_\updownarrow$ for which $V_{\downarrow \mathrm{opt}}$ strictly dominates $VI_{M_6, \pi_{\mathrm{opt}}}(V_{\downarrow \mathrm{opt}})$, showing that $V_{\downarrow \mathrm{opt}} >_{\mathrm{dom}} V_{M_6, \pi_{\mathrm{opt}}}$ (by Theorem 6) contradicting the fact that $V_{\downarrow \pi_{\mathrm{opt}}} = V_{\downarrow \mathrm{opt}}$. (The combination of M_4 and M_5 to get M_6 is analogous to the construction in Line 61 above).

Subcase 2b. Suppose $V_{\downarrow \mathrm{opt}}(p)$ is "too big" in Line 64, *i.e.,* for every action $\alpha \in \rho_{V_{\uparrow \mathrm{opt}}}(p)$ there is some MDP $M_\alpha \in M_\updownarrow$ such that $VI_{M_\alpha, \alpha}(V_{\downarrow \mathrm{opt}})(p) < V_{\downarrow \mathrm{opt}}(p)$.

Consider $\alpha = \pi_{\mathrm{opt}}(p)$. The definition of "optimistically optimal" along with the theory of exact MDPs guarantees us that there is some MDP M such that

$$V_{\uparrow \mathrm{opt}} = V_{\uparrow \pi_{\mathrm{opt}}} = V_{M, \pi_{\mathrm{opt}}} = VI_{M, \pi_{\mathrm{opt}}}(V_{M, \pi_{\mathrm{opt}}}) = VI_{M, \pi_{\mathrm{opt}}}(V_{\uparrow \mathrm{opt}}) \tag{69}$$

By our case 2 assumption,

$$V_{\uparrow \mathrm{opt}}(p) = \max_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M, \alpha}(V_{\uparrow \mathrm{opt}})(p), \tag{70}$$

and this, together with Line 69 and $\alpha = \pi_{\mathrm{opt}}(p)$ implies

$$VI_{M, \pi_{\mathrm{opt}}}(V_{\uparrow \mathrm{opt}})(p) = \max_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M, \alpha}(V_{\uparrow \mathrm{opt}})(p), \tag{71}$$

and therefore that

$$\pi_{\mathrm{opt}}(p) \in \operatorname*{argmax}_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M, \alpha}(V_{\uparrow \mathrm{opt}})(p), \tag{72}$$

which implies that $\alpha = \pi_{opt}(p) \in \rho_{V_{\uparrow opt}}(p)$. We can then use our subcase assumption that there must be an MDP $M_\alpha \in M_\updownarrow$ such that $VI_{M_\alpha, \pi_{opt}}(V_{\downarrow opt})(p) < V_{\downarrow opt}(p)$.

Let M_7 be a π_{opt}-minimizing MDP, as per Theorem 7. Then $V_{M_7, \pi_{opt}} = V_{\downarrow \pi_{opt}} = V_{\downarrow opt}$ by expanding definitions. So $VI_{M_7, \pi_{opt}}(V_{\downarrow opt}) = V_{\downarrow opt}$. We can now create a new MDP M_8 by copying M_7 at every state except p, where M_8 copies M_α, following the construction used to define M_3 in Subcase 1a. By construction we then have

$$VI_{M_8, \pi_{opt}}(V_{\downarrow opt}) <_{dom} V_{\downarrow opt}, \tag{73}$$

which by Theorem 6 implies $V_{\downarrow \pi_{opt}} <_{dom} V_{\downarrow opt}$, contradicting our choice of π_{opt} and concluding Subcase 2b, Case 2, and the proof of Theorem 9.

❑ (Theorem 9).

Theorem 10: For any policy π, $IVI_{\downarrow \pi}$ and $IVI_{\uparrow \pi}$ are contraction mappings.

Proof: We first show that $IVI_{\uparrow \pi}$ is a contraction mapping on $\overline{V}$, the space of value functions. Strictly speaking, $IVI_{\uparrow \pi}$ is a mapping from an interval value function $V_\updownarrow$ to a value function V. However, the specific values $V(p)$ only depend on the upper bounds $V_\uparrow$ of $V_\updownarrow$. Therefore, the mapping $IVI_{\uparrow \pi}$ is isomorphic to a function that maps value functions to value functions and with some abuse of terminology, we can consider $IVI_{\uparrow \pi}$ to be such a mapping. The same is true for $IVI_{\downarrow \pi}$, which depends only on the lower bounds $V_\downarrow$.

Let $\hat{u}$ and $\hat{v}$ be interval value functions, fix $p \in Q$, and assume that $IVI_{\uparrow \pi}(\hat{v})(p) \geq IVI_{\uparrow \pi}(\hat{u})(p)$. Let M be an MDP $M \in M_\updownarrow$ that maximizes the expression $VI_{M, \pi}(v_\uparrow)(p)$ (Lemma 1 implies that there is such an MDP in the finite set $X_{M_\updownarrow}$, guaranteeing the existence of M in spite of the infinite cardinality of $M_\updownarrow$).

Then,

$$0 \leq IVI_{\uparrow \pi}(\hat{v})(p) - IVI_{\uparrow \pi}(\hat{u})(p) \tag{74}$$

$$= \max_{M \in M_\updownarrow} VI_{M, \pi}(v_\uparrow)(p) - \max_{M \in M_\updownarrow} VI_{M, \pi}(u_\uparrow)(p) \tag{75}$$

$$\leq R(p) + \gamma\left(\sum_{q \in Q} F^M_{pq}(\pi(p)) v_\uparrow(q)\right) - R(p) - \gamma\left(\sum_{q \in Q} F^M_{pq}(\pi(p)) u_\uparrow(q)\right) \tag{76}$$

$$= \gamma\left(\sum_{q \in Q} F^M_{pq}(\pi(p))[v_\uparrow(q) - u_\uparrow(q)]\right) \tag{77}$$

$$\leq \gamma\Big(\sum_{q \in Q} F^{M}_{pq}(\pi(p)) \|v_{\uparrow} - u_{\uparrow}\| \Big) \tag{78}$$

$$= \gamma \|v_{\uparrow} - u_{\uparrow}\| . \tag{79}$$

Line 75 expands the definition of $IVI_{\uparrow_\pi}$. Line 76 follows by expanding the definition of VI and from the fact that M maximizes $VI_{M,\pi}(v_{\uparrow})(p)$ by definition. In Line 77, we simplify the expression by cancelling the immediate reward terms and factoring out the coefficients F^{M}_{pq}. In Line 78, we introduce an inequality by replacing the term $v_{\uparrow}(q) - u_{\uparrow}(q)$ with the maximum difference over all states, which by definition is the sup norm. The final step Line 79 follows from the fact that F is a probability distribution that sums to 1 and $\|v_{\uparrow} - u_{\uparrow}\|$ does not depend on q.

Repeating this argument interchanging the roles of $\hat{u}$ and \hatv in the case that $IVI_{\uparrow_\pi}(\hat{v})(p) \leq IVI_{\uparrow_\pi}(\hat{u})(p)$ implies

$$\left| IVI_{\uparrow_\pi}(\hat{v})(p) - IVI_{\uparrow_\pi}(\hat{u})(p) \right| \leq \gamma \|v_{\uparrow} - u_{\uparrow}\| \tag{80}$$

for all $p \in Q$. Taking the maximum over p in the above expression gives the result.

The proof that $IVI_{\downarrow_\pi}$ is a contraction mapping is very similar, replacing $IVI_{\uparrow_\pi}$ with $IVI_{\downarrow_\pi}$ throughout, replacing maximization with minimization in Line 74, and selecting MDP M to minimize the expression $VI_{M,\pi}(u_{\uparrow})(p)$ when $IVI_{\downarrow_\pi}(\hat{v})(p) \geq IVI_{\downarrow_\pi}(\hat{u})(p)$.

❑ (Theorem 10).

Theorem 11: For any policy π, $V_{\downarrow_\pi}$ is a fixed-point of $IVI_{\downarrow_\pi}$ and $V_{\uparrow_\pi}$ of $IVI_{\uparrow_\pi}$, and therefore $V_{\updownarrow_\pi}$ is a fixed-point of $IVI_{\updownarrow_\pi}$.

Proof: We prove the theorem for $IVI_{\downarrow_\pi}$; the proof for $IVI_{\uparrow_\pi}$ is similar. We show

(a) $IVI_{\downarrow_\pi}(V_{\updownarrow_\pi}) \leq_{\text{dom}} V_{\downarrow_\pi}$, and

(b) $IVI_{\downarrow_\pi}(V_{\updownarrow_\pi}) \geq_{\text{dom}} V_{\downarrow_\pi}$,

from which we conclude that $IVI_{\downarrow_\pi}(V_{\updownarrow_\pi}) = V_{\downarrow_\pi}$. Throughout both cases we take M^* to be a π-minimizing MDP, so that $V_{\downarrow_\pi} = V_{M^*,\pi}$. By Theorem 7 M^* must exist.

We first prove (a). From Theorem 3, we know that $V_{M^*,\pi}$ is a fixed point of $VI_{M^*,\pi}$. Thus, for any state $q \in Q$,

$$V_{\downarrow_\pi}(q) = V_{M^*,\pi}(q) = VI_{M^*,\pi}(V_{M^*,\pi})(q) = VI_{M^*,\pi}(V_{\downarrow_\pi})(q) . \tag{81}$$

Using this fact and expanding the definition of $IVI_{\downarrow\pi}$, we have, at every state q,

$$\begin{aligned} IVI_{\downarrow\pi}(V_{\updownarrow\pi})(q) &= \min_{M \in M_{\updownarrow}} VI_{M,\pi}(V_{\downarrow\pi})(q) \\ &\leq VI_{M^*,\pi}(V_{\downarrow\pi})(q) \\ &= V_{\downarrow\pi}(q). \end{aligned} \tag{82}$$

This implies that $IVI_{\downarrow\pi}(V_{\updownarrow\pi}) \leq_{\text{dom}} V_{\downarrow\pi}$ as desired.

To prove (b), suppose for sake of contradiction that for some state p, $IVI_{\downarrow\pi}(V_{\updownarrow\pi})(p) < V_{\downarrow\pi}(p)$. Let $M_1 \in M_{\updownarrow}$ be an MDP that minimizes[6] the expression $VI_{M,\pi}(V_{\downarrow\pi})(p)$.

Then, substituting M_1 into the definition of $IVI_{\downarrow\pi}$,

$$IVI_{\downarrow\pi}(V_{\updownarrow\pi})(p) = VI_{M_1,\pi}(V_{\downarrow\pi})(p) < V_{\downarrow\pi}(p). \tag{83}$$

We can then construct an MDP M_2 by copying M^* at every state except p, where M_2 copies M_1 (see the proof of Theorem 9, Case 1a for the details of a similar construction). Because M_2 is a copy of M^* at every state but p, Equation 81 must hold with M_2 replacing M^* at every state but p. Because M_2 is a copy of M_1 at state p, Equation 83 with M_2 replacing M_1 must hold at state p. These two facts together imply

$$VI_{M_2,\pi}(V_{\downarrow\pi}) <_{\text{dom}} V_{\downarrow\pi} \tag{84}$$

Then by Theorem 6 $V_{M_2,\pi} <_{\text{dom}} V_{\downarrow\pi}$, contradicting the definition of $V_{\downarrow\pi}$.

❑ (Theorem 11).

Theorem 13:

(a) $IVI_{\uparrow\text{opt}}$ and $IVI_{\downarrow\text{pes}}$ are contraction mappings.

(b) For any value function V and associated action set selection function ρ_V and σ_V, $IVI_{\downarrow\text{opt},V}$ and $IVI_{\uparrow\text{pes},V}$ are contraction mappings.

Proof: We first prove (a). The proof that $IVI_{\uparrow\text{opt}}$ is a contraction mapping is an extension of the proof of Theorem 10. Let $\hat{u}$ and $\hat{v}$ be interval value functions, fix $p \in Q$, and assume that $IVI_{\uparrow\text{opt}}(\hat{v})(p) \geq IVI_{\uparrow\text{opt}}(\hat{u})(p)$. Select $M \in M_{\updownarrow}$ and $\alpha \in A$ to maximize the expression $VI_{M,\alpha}(v_{\uparrow})(p)$ (again, Lemma 1 implies that

6. Such an MDP exists by Lemma 1, which implies that there must be such an MDP in the finite set $X_{M_{\updownarrow}} \subseteq M_{\updownarrow}$.

there is such an MDP in the finite set $X_{M_\updownarrow}$, guaranteeing the existence of M in spite of the infinite cardinality of $M_\updownarrow$).

Then,

$$0 \le IVI_{\uparrow_{\text{opt}}}(\hat{v})(p) - IVI_{\uparrow_{\text{opt}}}(\hat{u})(p) \tag{85}$$

$$= \max_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M,\alpha}(v_\uparrow)(p) - \max_{\alpha \in A} \max_{M \in M_\updownarrow} VI_{M,\alpha}(u_\uparrow)(p) \tag{86}$$

$$\le R(p) + \gamma\left(\sum_{q \in Q} F^M_{pq}(\alpha) v_\uparrow(q)\right) - R(p) - \gamma\left(\sum_{q \in Q} F^M_{pq}(\alpha) u_\uparrow(q)\right) \tag{87}$$

$$\le \gamma \| v_\uparrow - u_\uparrow \| . \tag{88}$$

Line 86 expands the definition of $IVI_{\uparrow_{\text{opt}}}$, noting that maximizing using $\le_{\text{opt}}$ selects interval upper bounds based only on the upper bounds of the input intervals. Line 87 follows from our choice of M and α to maximize $VI_{M,\alpha}(v_\uparrow)(p)$. Line 88 follows from Line 87 in the same manner that Line 79 followed from Line 76 in the proof of Theorem 10, and the desired result for $IVI_{\updownarrow_{\text{opt}}}$ for part (a) of the theorem also follow in the same manner as the remainder of Theorem 10 followed from Line 79.

To prove that $IVI_{\downarrow_{\text{pes}}}$ is a contraction mapping, we again fix a state p and assume $IVI_{\downarrow_{\text{pes}}}(\hat{v})(p) \ge IVI_{\downarrow_{\text{pes}}}(\hat{u})(p)$. We then use $v_\downarrow$ to choose an action α that maximizes $\min_{M \in M_\updownarrow}(VI_{M,\alpha}(v_\downarrow)(p))$ and $u_\downarrow$ to choose an MDP M that minimizes $VI_{M,\alpha}(u_\downarrow)(p)$ (again, Lemma 1 implies that there is such an MDP in the finite set $X_{M_\updownarrow}$, guaranteeing the existence of M). Using α and M as defined above, we have

$$0 \le IVI_{\downarrow_{\text{pes}}}(\hat{v})(p) - IVI_{\downarrow_{\text{pes}}}(\hat{u})(p) \tag{89}$$

$$= \max_{\alpha \in A} \min_{M \in M_\updownarrow} VI_{M,\alpha}(v_\downarrow)(p) - \max_{\alpha \in A} \min_{M \in M_\updownarrow} VI_{M,\alpha}(u_\downarrow)(p) \tag{90}$$

$$\le \min_{M \in M_\updownarrow} VI_{M,\alpha}(v_\downarrow)(p) - \min_{M \in M_\updownarrow} VI_{M,\alpha}(u_\downarrow)(p) \tag{91}$$

$$\le VI_{M,\alpha}(v_\downarrow)(p) - VI_{M,\alpha}(u_\downarrow)(p) \tag{92}$$

Line 90 expands the definition of $IVI_{\downarrow_{\text{pes}}}$, using the fact that maximizing over $\le_{\text{pes}}$ selects lower bounds based only on the lower bounds of the intervals being maximized over. Line 91 substitutes the action α, which introduces the inequality since α was chosen to guarantee

$$\min_{M \in M_\updownarrow} VI_{M,\alpha}(v_\downarrow)(p) = \max_{\alpha \in A} \min_{M \in M_\updownarrow} VI_{M,\alpha}(v_\downarrow)(p), \tag{93}$$

and the meaning of maximization guarantees that

$$\min_{M \in M_\updownarrow} VI_{M,\alpha}(u_\downarrow)(p) \leq \max_{\alpha \in A} \min_{M \in M_\updownarrow} VI_{M,\alpha}(u_\downarrow)(p). \tag{94}$$

Line 92 follows similarly because M was chosen to guarantee

$$VI_{M,\alpha}(u_\downarrow)(p) = \min_{M \in M_\updownarrow} VI_{M,\alpha}(u_\downarrow)(p), \tag{95}$$

and the meaning of minimization guarantees that

$$VI_{M,\alpha}(v_\downarrow)(p) \geq \min_{M \in M_\updownarrow} VI_{M,\alpha}(v_\downarrow)(p). \tag{96}$$

The desired result for $IVI_{\updownarrow \text{pes}}$ in part (a) of the theorem then follows directly from Line 92 in the same manner as the result for $IVI_{\updownarrow \text{opt}}$ followed from Line 86, concluding the proof of part (a) of the theorem.

For part (b), the proof for $IVI_{\downarrow \text{opt}, V}$ follows exactly as the proof for $IVI_{\downarrow \text{pes}}$, except that the set of actions considered in the maximization over actions at each state p is restricted to $\rho_V(p)$. Likewise, proving $IVI_{\uparrow \text{pes}, V}$ is the same as proving $IVI_{\uparrow \text{opt}}$ where the set of actions is restricted to $\sigma_V(p)$.

❑ (Theorem 13).

ECP-97

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 a 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, $\pi : \mathcal{Q} \to \mathcal{A}$. 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 $\geq_{\text{dom}}$ defined as follows: $V_1 \geq_{\text{dom}} V_2$ 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.

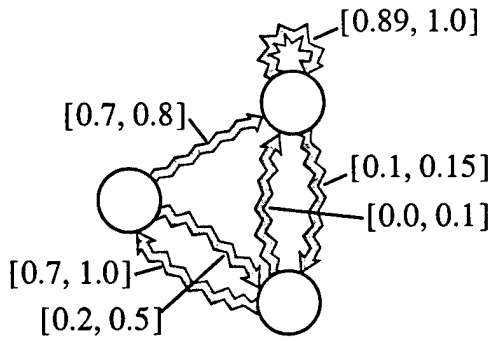


Fig. 1. The state-transition diagram for a simple bounded parameter Markov decision process with three states and a single action. The arcs indicate possible transitions and are labeled by their lower and upper bounds.

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 $\pi \in \Pi$, $VI_\pi : \mathcal{V} \to \mathcal{V}$ 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 $p \in Q$. 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 $T : U \to U$, 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 $\hat{IVI}_\pi$ (for *interval value iteration*) which maps interval value functions to other interval value functions. We have proven that iterating $\hat{IVI}_\pi$ 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.

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

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

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 $\hat{IVI}_\pi$ 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 $1 \leq i \leq j \leq k$ then $\underline{V}(q_i) \leq \underline{V}(q_j)$ (increasing order).

Let r be the index $1 \leq r \leq k$ 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:

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

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

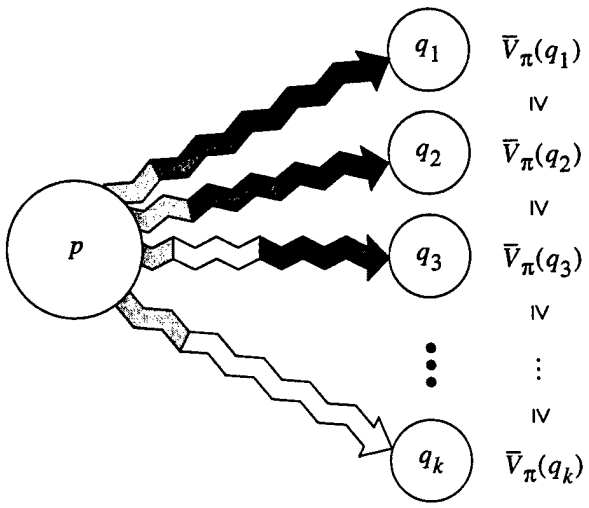


Fig. 2. An illustration of the basic dynamic programming step in computing an approximate value function for a fixed policy and bounded parameter MDP. The lighter shaded portions of each arc represent the required lower bound transition probability and the darker shaded portions represent the fraction 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 $\hat{IVI}_\pi$ 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 $\hat{F}_{B_i,B_j}(\alpha)$ represent the interval value for the transition from B_i to B_j on action α defined as follows: $\hat{F}_{B_i,B_j}(\alpha) = \left[\min_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha),\ \max_{p \in B_i} \sum_{q \in B_j} F_{pq}(\alpha)\right]$ 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

Adaptive Importance Sampling for Estimation in Structured Domains

Luis E. Ortiz
Computer Science Department
Brown University
Box 1910
Providence, RI 02912 USA
leo@cs.brown.edu

Leslie Pack Kaelbling
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139 USA
lpk@ai.mit.edu

Abstract

Sampling is an important tool for estimating large, complex sums and integrals over high-dimensional spaces. For instance, importance sampling has been used as an alternative to exact methods for inference in belief networks. Ideally, we want to have a sampling distribution that provides optimal-variance estimators. In this paper, we present methods that improve the sampling distribution by systematically adapting it as we obtain information from the samples. We present a stochastic-gradient-descent method for sequentially updating the sampling distribution based on the direct minimization of the variance. We also present other stochastic-gradient-descent methods based on the minimization of typical notions of *distance* between the current sampling distribution and approximations of the target, optimal distribution. We finally validate and compare the different methods empirically by applying them to the problem of action evaluation in influence diagrams.

1 INTRODUCTION

Often, we are interested in computing quantities involving large sums, such as expectations in uncertain, structured domains. For instance, belief inference in *Bayesian networks (BNs)* requires that we sum or marginalize over the remaining variables that are not of interest. Similarly, in order to solve the problem of action selection in *influence diagrams*, we sum over the variables that are not observed at the time of the decision in order to compute the value of different action choices.

We can represent the uncertainty in structured environments using a BN. A BN allows us to compactly define a joint probability distribution over the relevant variables in a domain. It provides a graphical representation of the distribution by means of a directed acyclic graph (DAG). It defines locally a conditional probability distribution for each relevant variable, represented as a node in the graph, given the state of its parents in the graph. This decomposition can help in the evaluation of the sums. However, due to factors regarding the connectivity of the graph, in general this is not sufficient to allow an efficient computation of the exact value of the sums of interest.

Sampling provides an alternative tool for approximately computing these sums. Sampling methods have been proposed as an alternative to exact methods for such problems. In particular, *importance sampling* (see Geweke [1989], and the references therein) has been applied to the problem of belief inference in BNs [Fung and Chang, 1989, Shachter and Peot, 1989] and action selection in IDs (see Charnes and Shenoy [1999] and the references therein, and Ortiz and Kaelbling [2000]). In its simpler form, the importance-sampling distribution used is the "prior" distribution of the BN resulting from setting the value of the evidence. It has been noted early on that this sampling distribution is far from optimal in the sense that it provides estimates with larger variance than necessary [Shachter and Peot, 1989]. For instance, the optimal sampling distribution in the case of belief inference is to sample the unobserved variables from the posterior distribution over them given the observed evidence. If we knew this distribution we would know the answer to the belief inference problem.

Several modifications have been proposed to improve the estimation of the simple importance sampling distribution discussed above, based on information obtained from the samples [Fung and Chang, 1989, Shachter and Peot, 1989, Shwe and Cooper, 1991]. In this paper, we propose methods to systematically and sequentially update the importance-sampling distribution. We view the updating process as one of learning a separate BN just for sampling. The learning objective is to minimize some error criterion. A stochastic-gradient method results from the direct minimization of the variance of the estimator with respect to the importance sampling distribution as an error function. Other stochastic-gradient methods result from minimizing

error functions based on typical measures of the notion of *distance* between the current sampling distribution and approximations of the optimal sampling distribution.

2 DEFINITIONS

We begin by introducing some notation used throughout the paper. We denote one-dimensional random variables by capital letters and denote multi-dimensional random variables by bold capital letters. For instance, we denote a multi-dimensional random variable by $\boldsymbol{X}$ and denote all its components by $(X_1, \ldots, X_n)$ where X_i is the i^{th} one-dimensional random variable. We use small letters to denote assignments to random variables. For instance, $\boldsymbol{X} = \boldsymbol{x}$ means that for each component X_i of $\boldsymbol{X}$, $X_i = x_i$. We denote the set of possible values that X_i can take by Ω_{X_i} and the set of possible values that $\boldsymbol{X}$ can take by $\Omega_{\boldsymbol{X}} = \bigtimes_{i=1}^{n} \Omega_{X_i}$. We also denote by capital letters the nodes in a graph. We denote by $\mathrm{Pa}(Y)$ the parents of node Y in a directed graph.

We now introduce notation that will become useful during the description of the methods presented in this paper. We denote by the operator $\sum_{\boldsymbol{Z}}$ the sum over the possible values of the individual variables forming $\boldsymbol{Z}$, $\sum_{Z_1} \sum_{Z_2} \cdots \sum_{Z_{n_1}}$. For any function h with variables $\boldsymbol{Z}$ and $\boldsymbol{O}$, the expression $h(\boldsymbol{Z}, \boldsymbol{O})|_{\boldsymbol{O}=\boldsymbol{o}}$ stands for a function f' over variables $\boldsymbol{Z}$ that results from setting the values of $\boldsymbol{O}$ in h with assignment $\boldsymbol{o}$ while letting the values for $\boldsymbol{Z}$ remain unassigned. In other words, $f'(\boldsymbol{Z}) = h(\boldsymbol{Z}, \boldsymbol{O})|_{\boldsymbol{O}=\boldsymbol{o}} = h(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o})$. The notation $\boldsymbol{X} = (\boldsymbol{Z}, \boldsymbol{O})$ means that the variable $\boldsymbol{X}$ is formed by all the variables that form $\boldsymbol{Z}$ and $\boldsymbol{O}$. That is, $\boldsymbol{X} = (X_1, \ldots, X_n) = (Z_1, \ldots, Z_{n_1}, O_1, \ldots, O_{n_2}) = (\boldsymbol{Z}, \boldsymbol{O})$, where $n = n_1 + n_2$. Note that we are assuming that the set of variables forming $\boldsymbol{Z}$ and those forming $\boldsymbol{O}$ are disjoint. The notation $\boldsymbol{Z} \sim f$ means that the random variable $\boldsymbol{Z}$ is distributed according to probability distribution f.

A *Bayesian network (BN)* is a graphical probabilistic model used to represent uncertainty in structured domains. It compactly represents the joint probability distribution over the relevant variables of the system of interest. It uses a *directed acyclic graph (DAG)* to represent the relationship between the relevant variables. A node in the graph represents a variable. The model defines a local conditional distribution $P(X_i \mid \mathrm{Pa}(X_i))$ for each node or variable X_i given its parents $\mathrm{Pa}(X_i)$ in the graph. The joint distribution is then

$$P(\boldsymbol{X}) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i)).$$

For instance, we can define a BN on the graph given in Figure 1(a).

The inference problem in BNs is that of computing the posterior probability of an assignment to a subset of variables

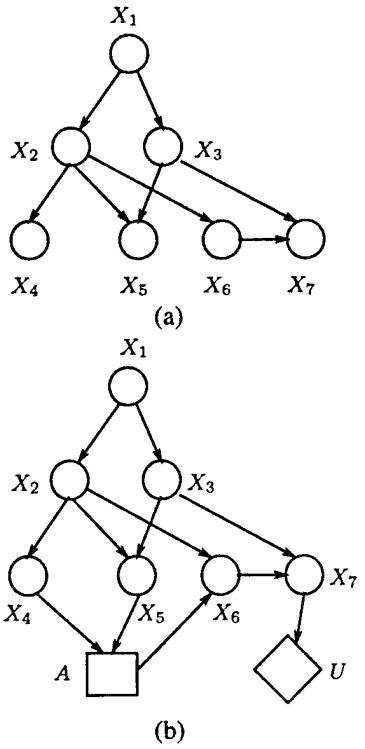


Figure 1: Example of (a) Bayesian network and (b) influence diagram.

given evidence about another subset of variables in the system. Assume that the variables are discrete and their *sample spaces* or the possible values each variable can take are finite. In general, let $\boldsymbol{X} = (\boldsymbol{Z}, \boldsymbol{O})$ where $\boldsymbol{O}$ is the set of variables of interest, $\boldsymbol{o}$ is an assignment to it and $\boldsymbol{Z}$ are the remaining variables. For this problem we want to compute probabilities of the kind

$$P(\boldsymbol{O} = \boldsymbol{o}) = \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o}).$$

Often, the local decomposition of the joint distribution still leads to the evaluation of sums over a large number of variables. In general, this problem is intractable [Cooper, 1990].

An *influence diagram (ID)* is a probabilistic model for decision-making under uncertainty. We can think of an ID as a BN with decision and utility nodes added. For instance, we can use our example BN to build an ID as shown in Figure 1(b). The square is a decision node. The diamond is a utility node. We now have potentially different joint distributions over the variables, for each action choice available. Assume for simplicity that there is a single decision node in the graph. The joint distribution over the variables, given the action choice a assigned to the decision variable, is

$$P(\boldsymbol{X} \mid A = a) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i))|_{A=a}.$$

The decision associated with a decision node is a function of its parent nodes in the graph. We will have access to

the value of these variables at the time of making the decision. Similarly, the utility associated with a utility node is a function of its parent nodes in the graph.

Assume that we have a finite number of discrete action choices. Then, one problem is to select the *best strategy* or function π^* mapping each possible value of the parents of the decision node to an action choice. The best strategy is the strategy with highest expected utility. Let $\boldsymbol{X} = (\boldsymbol{Z}, \boldsymbol{O})$ where the variables in $\boldsymbol{O}$ are parents of the decision node and $\boldsymbol{Z}$ are the remaining variables. The problem of obtaining an optimal strategy reduces to obtaining, for each assignment $\boldsymbol{O} = \boldsymbol{o}$, the action that maximizes the value associated with the action and the assignment:

$$V_{\boldsymbol{o}}(a) = \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o} \mid A = a) U(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o}, A = a).$$

Note once again that computing this value requires the evaluation of a sum. For the same reasons as in the previous problem of belief inference in BNs, the exact computation of this value is intractable in general.

3 IMPORTANCE SAMPLING

Importance sampling provides an alternative to the exact methods for evaluating sums. Let the quantity of interest be $G = \sum_{\boldsymbol{Z}} g(\boldsymbol{Z})$ for some real function g. We can turn the sum into an expectation by expressing $G = \sum_{\boldsymbol{Z}} f(\boldsymbol{Z}) \left(g(\boldsymbol{Z})/f(\boldsymbol{Z})\right)$, where f is a probability distribution over $\boldsymbol{Z}$ satisfying, for all $\boldsymbol{Z}$, $g(\boldsymbol{Z}) \neq 0 \Rightarrow f(\boldsymbol{Z}) \neq 0$. We call f the *importance-sampling distribution*. We define the *weight function* $\omega(\boldsymbol{Z}) = g(\boldsymbol{Z})/f(\boldsymbol{Z})$ which allows us to express $G = \sum_{\boldsymbol{Z}} f(\boldsymbol{Z})\omega(\boldsymbol{Z})$. Hence, we can obtain an unbiased estimate of G by obtaining N samples $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ from $\boldsymbol{Z} \sim f$ and computing the estimate

$$\hat{G} = \frac{1}{N} \sum_{l=1}^{N} \omega(\boldsymbol{z}^{(l)}). \tag{1}$$

We can apply this technique to the problem of belief inference in BNs. Typically, we let

$$\begin{aligned} g(\boldsymbol{Z}) =& P(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o}) \\ =& \prod_{i=1}^{n_1} P(Z_i \mid \mathrm{Pa}(Z_i)) \prod_{j=1}^{n_2} P(O_j \mid \mathrm{Pa}(O_j))\Big|_{\boldsymbol{O}=\boldsymbol{o}}, \\ f(\boldsymbol{Z}) =& \prod_{i=1}^{n_1} P(Z_i \mid \mathrm{Pa}(Z_i))|_{\boldsymbol{O}=\boldsymbol{o}}, \text{ which implies} \\ \omega(\boldsymbol{Z}) =& \prod_{j=1}^{n_2} P(O_j \mid \mathrm{Pa}(O_j))\Big|_{\boldsymbol{O}=\boldsymbol{o}}. \end{aligned}$$

Note that we are defining the importance sampling distribution to be the "prior" distribution of the BN. We obtain samples from this distribution by sampling the variables in the (partial) order defined by the DAG and according to the local conditional distribution of the original BN for each variable. As we obtain samples from each variable by traversing the nodes in the graph and sampling the variable corresponding to it, if we get to a node or variable that is in the evidence set $\boldsymbol{O}$, we do not sample it. Instead, we assign to it the value given by the evidence assignment $\boldsymbol{o}$. Therefore, the resulting samples will be assignments to those variables that are not in the evidence set according to the "prior" distribution of the BN. We call the method resulting from this importance-sampling distribution the *traditional method*. In the context of belief inference, this method is called *likelihood-weighting (LW)* since the weight function is a "likelihood" and thus each sample is weighted by its "likelihood."

We can similarly apply this technique in the context of action selection in IDs to evaluate $V_{\boldsymbol{o}}(a)$. In general, we let

$$\begin{aligned} g(\boldsymbol{Z}) &= P(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o} \mid A = a) U(\boldsymbol{Z}, \boldsymbol{O} = \boldsymbol{o}, A = a), \\ f(\boldsymbol{Z}) &= \prod_{i=1}^{n_1} P(Z_i \mid \mathrm{Pa}(Z_i))|_{\boldsymbol{O}=\boldsymbol{o}, A=a}, \\ \omega(\boldsymbol{Z}) &= \prod_{j=1}^{n_2} P(O_j \mid \mathrm{Pa}(O_j)) U(\boldsymbol{Z}, \boldsymbol{O}, A)\Big|_{\boldsymbol{O}=\boldsymbol{o}, A=a}. \end{aligned}$$

In particular, for our example,

$$\begin{aligned} g(\boldsymbol{Z}) &= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1) \times \\ &\quad P(X_6 \mid X_2, A = a)P(X_7 \mid X_3, X_6) \times \\ &\quad P(X_4 = x_4 \mid X_2)P(X_5 = x_5 \mid X_2, X_3) \times \\ &\quad U(X_7, A = a), \\ f(\boldsymbol{Z}) &= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1) \times \\ &\quad P(X_6 \mid X_2, A = a)P(X_7 \mid X_3, X_6), \\ \omega(\boldsymbol{Z}) &= P(X_4 = x_4 \mid X_2)P(X_5 = x_5 \mid X_2, X_3) \times \\ &\quad U(X_7, A = a). \end{aligned}$$

An important property of the estimator $\hat{G}$ is the variance of the weights associated with the importance-sampling distribution. This is

$$\mathrm{Var}[\omega(\boldsymbol{Z})] = \sum_{\boldsymbol{Z}} f(\boldsymbol{Z})\omega(\boldsymbol{Z})^2 - G^2.$$

Recall that $G = \sum_{\boldsymbol{Z}} g(\boldsymbol{Z})$ by definition and assume that g is a positive function. From this we can derive that the optimal or minimum-variance importance-sampling distribution is proportional to $g(\boldsymbol{Z})$:

$$f^*(\boldsymbol{Z}) = g(\boldsymbol{Z}) / \sum_{\boldsymbol{Z}} g(\boldsymbol{Z}). \tag{2}$$

The weights will have zero variance in that case, since the weight function will always output our value of interest G. We also note that we need to avoid letting $f(\boldsymbol{Z})$ be too small with respect to $g(\boldsymbol{Z})$, since this will increase the variance. As a matter of fact, $\mathrm{Var}[\omega(\boldsymbol{Z})] \rightarrow \infty$ as $f(\boldsymbol{Z}) \rightarrow 0$ for at least one value of $\boldsymbol{Z}$. This implies that we should use importance-sampling distributions with sufficiently "fat tails."

4 ADAPTIVE IMPORTANCE SAMPLING

The traditional method presented above uses as the importance-sampling distribution the "prior" distribution

of the BN which can be far from optimal in the sense that it can have higher variance than necessary. In the case of evaluating actions in IDs, it also completely ignores potentially useful information about the utility values. Therefore, we try to learn the optimal importance-sampling distribution by adapting the current sampling distribution as we obtain samples from it.

We view the adaptive process as one of learning a distribution over the variables the sum is over to use specifically as an importance-sampling distribution. In particular, we can view this process as learning BNs from the samples just for sampling. From the expression of the optimal importance-sampling distribution given in equation 2 (and, in particular, from the factorization of the function g for the different estimation problems), we can deduce that in order to be able to represent this distribution graphically using a BN we need to add arcs that connect every pair of nodes that are parents of observations and/or utility nodes, if they are not already connected. However, doing so can increase the size of the model, particularly in cases where the local conditional probabilities and the utilities have a smaller, more compact parametric representation (i.e., noise-or's). In this paper, we do not deal with this issue and instead concentrate on the problem of learning a BN with the same structure as the original BN (or ID). Hence, we only need to update the local conditional probability distributions as we obtain samples.

We can parameterize the importance-sampling distribution using a set of parameters Θ. Let the indicator function $I(Z_i = k, \mathrm{Pa}(Z_i) = j \mid \boldsymbol{Z}) = 1$ if the condition $Z_i = k$ and $\mathrm{Pa}(Z_i) = j$ agrees with the value assigned to $\boldsymbol{Z}$; 0 otherwise. Then, we can express the importance-sampling distribution as

$$f(\boldsymbol{Z} \mid \Theta) = \prod_{i=1}^{n} \prod_{j \in \Omega_{\mathrm{Pa}(Z_i)}} \prod_{k \in \Omega_{Z_i}} \theta_{ijk}^{I(Z_i=k, \mathrm{Pa}(Z_i)=j \mid \boldsymbol{Z})}, \tag{3}$$

where for each i, j, k, $\theta_{ijk} = P(Z_i = k \mid \mathrm{Pa}(Z_i) = j, \Theta)$. Hence, for all i, j, $\sum_k \theta_{ijk} = 1$, and for all k, $\theta_{ijk} > 0$. Note that this representation uses the assumptions of *global* and *local parameter independence* typically used in BNs. The weight function is also parameterized and defined as $\omega(\boldsymbol{Z} \mid \Theta) = g(\boldsymbol{Z})/f(\boldsymbol{Z} \mid \Theta)$.

4.1 LEARNING CRITERIA AND UPDATE RULES

In the following subsections we present different methods for updating the sampling distribution. The update rules are all based on gradient-descent. Hence, at each time t, we update the parameters as follows:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha(t) \nabla^p e(\theta^{(t)}). \tag{4}$$

In the update rule above, $\alpha(t)$ denotes the learning rate or the step size rule and $\nabla^p e(\Theta)$ denotes the gradient of error function e, appropriately projected to satisfy the constraints on Θ. The methods differ in how they define $\nabla^p e(\theta^{(t)})$.

In the discussion below we denote the $N(t)$ i.i.d. samples as $\boldsymbol{z}^{(t,1)}, \ldots, \boldsymbol{z}^{(t,N(t))}$ drawn according to $\boldsymbol{Z} \sim f(\boldsymbol{Z} \mid \theta^{(t)})$. If we gather samples to estimate G using many different sampling distributions, how can we combine them to get an unbiased estimate? It is sufficient to weight them using any weighting function that is independent of the sub-estimates obtained by using just the samples for one sampling distribution. For instance, the estimator

$$\hat{G}^{(T)} = \sum_{t=1}^{T} W(t) \hat{G}(\theta^{(t)}), \tag{5}$$

where $\sum_{t=1}^{T} W(t) = 1$ and $W(t) \geq 0$, for all t, and

$$\hat{G}(\theta^{(t)}) = \frac{1}{N(t)} \sum_{l=1}^{N(t)} \omega(\boldsymbol{z}^{(t,l)} \mid \theta^{(t)}), \tag{6}$$

is unbiased as long as $W(t)$ and $\hat{G}(\theta^{(t)})$ are independent for each t. Letting $W(t) = 1/T$ will produce an unbiased estimate. This is the weight we use in the experiments. In general, we would like to give more weight to importance-sampling distributions with smaller variances. Assuming that the variance decreases with t, we would like $W(t)$ to be an increasing sequence of t. Note that using $W(t) \propto 1/\hat{\sigma}_t^2$, where $\hat{\sigma}_t^2$ is the sample variance at time t, though appealing, does not necessarily lead to an unbiased estimator since $W(t)$ and $\hat{G}(\theta^{(t)})$ are not independent.

We will consider three general strategies: minimizing variance directly, minimizing distance to *global approximations* of the optimal sampling distribution, and minimizing distance to the empirical distribution of the optimal sampling distribution based on *local approximations*. For the first two strategies, we will find that we can express the partial derivatives that form the gradient as, for all i, j, k,

$$\frac{\partial e(\Theta)}{\partial \theta_{ijk}} = \sum_{\boldsymbol{Z}} f(\boldsymbol{Z} \mid \Theta) \left[\frac{-I(Z_i=k, \mathrm{Pa}(Z_i)=j \mid \boldsymbol{Z})}{\theta_{ijk}} \times \varphi(\boldsymbol{Z}, \Theta) \right],$$

where $\varphi(\boldsymbol{Z}, \Theta)$ is a function that depends on the error functions. Note that this is an expectation. Then, the methods update the parameters by estimating the value of the partial derivatives evaluated at the current setting of the parameters $\theta^{(t)}$ as

$$\frac{\partial \hat{e}(\theta^{(t)})}{\partial \theta_{ijk}} = \frac{1}{N(t)} \sum_{l=1}^{N(t)} \left[\frac{-I(Z_i=k, \mathrm{Pa}(Z_i)=j \mid \boldsymbol{Z}=\boldsymbol{z}^{(t,l)})}{\theta_{ijk}^{(t)}} \times \varphi(\boldsymbol{z}^{(t,l)}, \theta^{(t)}) \right].$$

4.1.1 Minimizing Variance Directly

As we noted above, the optimal importance-sampling distribution for estimating G is that which minimizes the variance of ω. Using that as our objective, we derive a stochastic-gradient update rule for the parameters of the

importance-sampling distribution. Let the error function be

$$\begin{aligned} e_{\mathrm{Var}}(\Theta) &= \mathrm{Var}(\omega(Z \mid \Theta)) \\ &= \textstyle\sum_Z f(Z \mid \Theta)\omega(Z \mid \Theta)^2 - G^2 \end{aligned}$$

The corresponding function for the gradient is

$$\varphi_{\mathrm{Var}}(Z, \Theta) = \omega(Z \mid \Theta)^2. \quad (7)$$

Note that using this definition of φ yields an unbiased estimate of the gradient. This is because the gradient is the expectation of a particular function and, in this case, we can always evaluate the function exactly. Hence, we can obtain an unbiased estimate by sampling from $f(Z \mid \Theta)$.

4.1.2 Minimizing Variance Indirectly via Approximate Global Minimization

Recall the optimal importance-sampling distribution f^* for estimating G given in equation 2. The update rules of the following subsection are all motivated by the idea of reducing some notion of *distance* between the current sampling distribution and this optimal sampling distribution. Note that we cannot really compute the values of the optimal distribution since that requires knowing the normalizing constant $\sum_Z g(Z) = G$ which is exactly the value we want to estimate. We approximate the optimal distribution using the current estimate of G as follows

$$\hat{f}^t(Z) = g(Z)/\hat{G}^{(t)}. \quad (8)$$

In the following, we will consider four error functions, one based on the sum-squared-error and three based on versions of the *Kullback-Leibler divergence.*

If we use the L_2 norm or sum-squared-error function as a notion of distance between the distributions, then the error function is

$$e_{\mathrm{L}_2}(\Theta) = \tfrac{1}{2}\textstyle\sum_Z \left(f(Z \mid \Theta) - f^*(Z)\right)^2.$$

The corresponding function for the gradient is

$$\begin{aligned} \varphi_{\mathrm{L}_2}(Z, \Theta) &= f^*(Z) - f(Z \mid \Theta) \\ &\approx f(z^{(t,l)} \mid \theta^{(t)}) \times \\ &\quad \left(\omega(z^{(t,l)} \mid \theta^{(t)})/\hat{G}^{(t)} - 1\right), \end{aligned} \quad (9)$$

where the approximation results from using $\hat{f}^t(Z)$ as defined in equation 8 as an approximation to $f^*(Z)$.

An alternative, commonly-used notion of *distance* between two probability distributions is given by the *Kullback-Leibler (KL) divergence.* This measure is not symmetric. One version of the KL divergence in this context is given by the error function

$$e_{\mathrm{KL}_1}(\Theta) = \textstyle\sum_Z f^*(Z) \log\left(f^*(Z)/f(Z \mid \Theta)\right).$$

The corresponding function for the gradient is

$$\begin{aligned} \varphi_{\mathrm{KL}_1}(Z, \Theta) &= f^*(Z)/f(Z \mid \Theta) \\ &\approx \omega(z^{(t,l)} \mid \theta^{(t)})/\hat{G}^{(t)}. \end{aligned} \quad (10)$$

Another version of the KL divergence is given by the error function

$$e_{\mathrm{KL}_2}(\Theta) = \textstyle\sum_Z f(Z \mid \Theta) \log\left(f(Z \mid \Theta)/f^*(Z)\right).$$

The corresponding function for the gradient is

$$\begin{aligned} \varphi_{\mathrm{KL}_2}(Z, \Theta) &= \log\left(f^*(Z)/f(Z \mid \Theta)\right) - 1 \\ &\approx \log\left(\omega(z^{(t,l)} \mid \theta^{(t)})/\hat{G}^{(t)}\right) - 1. \end{aligned} \quad (11)$$

A "symmetrized" version of KL sometimes used is given by the error function

$$e_{\mathrm{KL}_s}(\Theta) = \tfrac{1}{2}e_{\mathrm{KL}_1}(\Theta) + \tfrac{1}{2}e_{\mathrm{KL}_2}(\Theta).$$

We can obtain the partial derivatives for this error function and their approximation accordingly.

4.1.3 Heuristic Local Minimization Based on Empirical Distribution

The update methods in this subsection are motivated by the idea of minimizing different notions of distance between the current sampling distribution and an empirical distribution of the optimal importance-sampling distribution that we build from the samples. The hope is that the empirical distribution is a good approximation of the optimal sampling distribution. We define the empirical distribution, parameterized by $\hat{\Theta}$ locally as follows: for all i, j, k,

$$\hat{\theta}^{(t)}_{ijk} = \frac{\sum_{l=1}^{N(t)} I(Z_i=k, \mathrm{Pa}(Z_i)=j \mid Z=z^{(t,l)})\omega(z^{(t,l)} \mid \theta^{(t)})}{\sum_{l=1}^{N(t)} I(\mathrm{Pa}(Z_i)=j \mid Z=z^{(t,l)})\omega(z^{(t,l)} \mid \theta^{(t)})}, \quad (12)$$

if $\sum_{l=1}^{N(t)} I(\mathrm{Pa}(Z_i) = j \mid Z = z^{(t,l)})\omega(z^{(t,l)} \mid \theta^{(t)}) \neq 0$; $\hat{\theta}^{(t)}_{ijk} = \theta^{(t)}_{ijk}$ otherwise. We are essentially defining the empirical distribution using the samples if there are samples that can be used to define it; otherwise, we revert to the current distribution. We try to minimize the distance between the current sampling distribution and the empirical distribution locally.

Similar to the case of the previous strategies, we will find that we can express the partial derivatives that form the gradient of the error functions discussed in this subsection as, for all i, j, k,

$$\frac{\partial e'(\Theta)}{\partial \theta_{ijk}} = -\varphi'(\hat{\theta}_{ijk}, \theta_{ijk}),$$

where $\varphi'(\hat{\theta}_{ijk}, \theta_{ijk})$ is a function that depends on the error functions. Then, the methods update the parameters by estimating the value of the partial derivatives evaluated at the current setting of the parameters $\theta^{(t)}$ as

$$\frac{\partial \hat{e}'(\theta^{(t)})}{\partial \theta_{ijk}} = -\varphi'(\hat{\theta}^{(t)}_{ijk}, \theta^{(t)}_{ijk}).$$

We define the *local* L_2-norm error function as

$$e'_{\mathrm{L}_2}(\Theta) = \frac{1}{2}\sum_{i,j,k}\left(\theta_{ijk} - \hat{\theta}_{ijk}\right)^2,$$

the error function for one version of KL as

$$e'_{\mathrm{KL}_1}(\Theta) = \sum_{i,j,k}\hat{\theta}_{ijk}\log\left(\hat{\theta}_{ijk}/\theta_{ijk}\right),$$

and the other as

$$e'_{\mathrm{KL}_2}(\Theta) = \sum_{i,j,k}\theta_{ijk}\log\left(\theta_{ijk}/\hat{\theta}_{ijk}\right).$$

From this we obtain the corresponding functions for the gradient:

$$\begin{aligned}
\varphi'_{\mathrm{L}_2}(\hat{\theta}_{ijk}, \theta_{ijk}) &= \hat{\theta}_{ijk} - \theta_{ijk}, \\
\varphi'_{\mathrm{KL}_1}(\hat{\theta}_{ijk}, \theta_{ijk}) &= \hat{\theta}_{ijk}/\theta_{ijk}, \\
\varphi'_{\mathrm{KL}_2}(\hat{\theta}_{ijk}, \theta_{ijk}) &= \log\left(\hat{\theta}_{ijk}/\theta_{ijk}\right) - 1.
\end{aligned}$$

We can obtain an update rule based on the "symmetrized" version of KL accordingly.

4.2 DISCUSSION OF UPDATE RULES

First, note that of all the update rules, only the one derived for e_{Var} clearly uses an unbiased estimate of the gradient. It is not immediately apparent whether the update rules based on e_{L_2}, e_{KL_1} and e_{KL_2} use unbiased estimates.

Note also that the magnitude of the components of the resulting gradients are different, as suggested by their respective φ functions. The function φ_{Var} has magnitude proportional to the squares of the weights. The magnitudes of φ_{L_2} and φ_{KL_1} are linear in the weights. However, the magnitude of φ_{L_2} is potentially smaller since it has the probability of the sample as a factor. The magnitude of φ_{KL_2} is logarithmic in the weights.

Because we assume that g is positive, the weights are positive. Hence, φ_{Var} and φ_{KL_1} are always positive. The function φ_{L_2} is positive if $\omega(\boldsymbol{Z} \mid \Theta)/G > 1$. Similarly, the function φ_{KL_2} is positive if $\log(\omega(\boldsymbol{Z} \mid \Theta)/G) > 1$. If $\omega(\boldsymbol{Z} \mid \Theta) > G$ then the sampling distribution underestimates the value of g while if $\omega(\boldsymbol{Z} \mid \theta) < G$ then it overestimates the value. Therefore, the sign of φ_{L_2} and φ_{KL_2} depends on whether we under- or over-estimated the value of g. Similarly, the magnitudes of φ_{Var}, φ_{L_2}, φ_{KL_1}, and φ_{KL_2} are related to the amount of under- or overestimation. For φ_{Var}, φ_{L_2} and φ_{KL_1} the magnitude is larger when the sampling distribution underestimates than when it overestimates. For φ_{KL_2}, the logarithm brings the amount of over- and underestimation to the same scale. Note that for the approximations of φ_{L_2}, φ_{KL_1}, and φ_{KL_2}, $\hat{G}$ cannot be zero, and in addition for φ_{KL_2}, $\omega(\boldsymbol{Z} \mid \theta)$ cannot be zero. These conditions hold from the assumption that g is positive. Note that unless we constrain the importance-sampling distribution, all the functions φ_{Var}, φ_{L_2}, φ_{KL_1} and φ_{KL_2} will be unbounded even if g is bounded.

The local L_2 error function, e'_{L_2}, leads to an update rule for which the step size has a very intuitive interpretation as a weighting between the current importance-sampling distribution and the empirical distribution. In the case of e'_{KL_1}, the update direction is proportional to the ratio of the empirical distribution with respect to the current importance-sampling distribution. On the other hand, for e'_{KL_2}, the update direction is proportional to the logarithm of the same ratio. Note φ_{KL_2} is not defined if at least one $\hat{\theta}^{(t)}_{ijk} = 0$. We can fix this by letting, for each i, j, k, $\hat{\theta}^{(t)}_{ijk} = \frac{\left(\sum_{l=1}^{N(t)} I(Z_i=k, \mathrm{Pa}(Z_i)=j \mid \boldsymbol{Z}=\boldsymbol{z}^{(t,l)})\omega(\boldsymbol{z}^{(t,l)} \mid \boldsymbol{\theta}^{(t)})\right) + \theta^{(t)}_{ijk}}{\left(\sum_{l=1}^{N(t)} I(\mathrm{Pa}(Z_i)=j \mid \boldsymbol{Z}=\boldsymbol{z}^{(t,l)})\omega(\boldsymbol{z}^{(t,l)} \mid \boldsymbol{\theta}^{(t)})\right)+1}$. This is essentially imposing a Dirichlet prior with parameters equal to the current probability values on the empirical distribution parameters.

We can interpret the update rules based on local KL-divergence as adding weights to the elements of the domain of the importance-sampling distribution and renormalizing. For the version of KL-divergence with respect to the empirical distribution, we are always adding weights. We add values relative to the amount we underestimated or overestimated the magnitude of the distribution for a particular state. If we underestimated, we add weights larger than one. If we overestimated, we add weights smaller than one. For the other version of KL-divergence, due to the logarithm function, we add weight if we underestimated while we subtract weight if we overestimated. Therefore, the logarithm brings the amount of underestimation and overestimation to the same scale and adds or subtracts weight accordingly.

Note that when approximating the gradients for e_{Var}, e_{L_2}, e_{KL_1} and e_{KL_2}, we can use as little as one sample to obtain an estimate of the gradient (i.e., $N(t) = 1$). This is not advisable for the method based on the local heuristic since the empirical distribution of the optimal sampling distribution will be highly inaccurate. Hence, the update rules based on the empirical distribution will work better when we take a larger number of samples between updates. Finally, note that when $t = 1$ and $N(t) = 1$, $\varphi_{\mathrm{L}_2} = 0$, and therefore, the parameters will not change in the first iteration.

5 RELATED WORK

Different variations of importance sampling have been used for the problems discussed in this paper (See Lin and Druzdzel [1999] and the references therein). Our methods belong to the class of *forward samplers* since they sample from a distribution based on the original structure of the BN. Of these, *self-importance sampling* [Shachter and Peot, 1989, Shwe and Cooper, 1991] is the method closest to the methods proposed in this paper since it also updates the sampling distribution as it obtains information from the samples. This method has an update rule that is very similar to the one derived for e'_{L_2}. It updates the distribution

after obtaining the empirical distribution, but the update is a weighting between the empirical distribution and the first sampling distribution used [Shwe and Cooper, 1991]. The update rule is

$$\begin{aligned}\theta_{ijk}^{(t+1)} &\leftarrow (1-\alpha(t))\hat{\theta}_{ijk}^{(t)} + \alpha(t)\theta_{ijk}^{(0)} \\ &= \theta_{ijk}^{(t)} - \\ &\quad \alpha(t)\left(\frac{\theta_{ijk}^{(t)}}{\alpha(t)} - (1-\alpha(t))\frac{\hat{\theta}_{ijk}^{(t)}}{\alpha(t)} - \theta_{ijk}^{(0)}\right).\end{aligned}$$

In our framework, we can think of this update rule as resulting from the error function

$$e'_{SIS}(\Theta, t) = \frac{1}{2\alpha(t)}\sum_{ijk}\left(\theta_{ijk} - \left((1-\alpha(t))\hat{\theta}_{ijk} + \alpha(t)\theta_{ijk}^{(0)}\right)\right)^2.$$

Annealed importance sampling [Neal, 1998] is a related technique in that it tries to obtain samples from the optimal sampling distribution. As we understand it, the user sets up a sequence of distributions, the last distribution being the optimal distribution, typically defined by Markov chains. We move from one distribution to another as we "anneal" and the sequence converges to the optimal sampling distribution. The hope is that we can get an independent sample from that distribution, then we restart the process to try to obtain another independent sample, and so on. Finally, it uses those independent samples to obtain an estimate. Notice that each "traversal" of the sequence of distributions (or Markov chains) produces a single sample. The technique is very general and we are unaware of whether it has been applied to the problems considered in this paper. We are currently investigating possible connections between our methods and this technique.

6 EMPIRICAL RESULTS

We implemented all of the adaptive importance-sampling methods described above. We let the learning rate $\alpha(t) = \beta/t$, where β is a value that depends on the updating method. We need different values of β for the different methods because of the differences in magnitude of their gradients. We impose an additional constraint on the parameters which we call the *ϵ-boundary*. We require that for all i, j, k, $\theta_{ijk} \geq \epsilon(|\Omega_{X_i}|) = \gamma/|\Omega_{X_i}|$, where γ is a constant factor. In our experiments, we let $\gamma = 0.1$. We do this so that our sampling distribution has "fat tails", avoiding extrema in probability and hence the possibility of infinite variance. We initialize the parameters $\theta^{(0)}$ such that the starting importance-sampling distribution is the "prior" probability distribution of the original BN. However, if one of the local conditional probability values does not satisfy the ϵ-boundary constraint, we change the distribution so that it does.

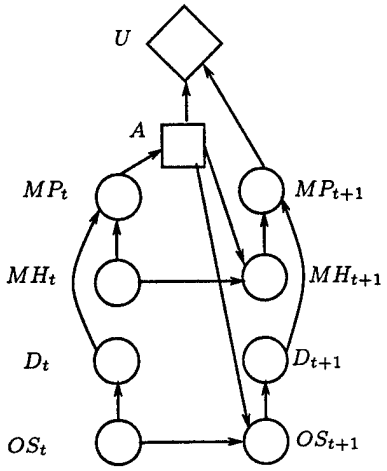


Figure 2: Graphical representation of the ID for the computer mouse problem.

In order to satisfy the constraint that for all i, j, $\sum_k \theta_{ijk} = 1$, we project the approximation of the gradients onto the simplex of the local conditional probability distribution. We do so by letting, for all i, j, k,

$$\frac{\partial^p \hat{e}(\theta)}{\partial \theta_{ijk}} \leftarrow \frac{\partial \hat{e}(\theta)}{\partial \theta_{ijk}} - \frac{1}{|\Omega_{X_i}|}\sum_{k=1}^{|\Omega_{X_i}|}\frac{\partial \hat{e}(\theta)}{\partial \theta_{ijk}}. \qquad (13)$$

Note that this is not enough to guarantee that after taking a step in the projected direction, the parameters will remain in the constraint space. If, when updating a local conditional probability distribution, its respective parameters do not satisfy the constraint, we find the minimum step α' that will allow them to remain inside the constraint space and take a step of size $\alpha'/2$ along the gradient direction (i.e., half the distance between the current position of the parameter we are updating in the simplex and the closest point on the ϵ-boundary along the gradient direction).

We tested the methods on the *computer mouse problem* [Ortiz and Kaelbling, 2000], a simple made-up ID shown in Figure 2. We added one to all the utility values presented in Ortiz and Kaelbling [2000] to make g positive. We will consider the problem of obtaining the value $V_{MP_t}(A)$ for the action $A = 2$ and the observation $MP_t = 1$.

We evaluated each method by computing the *mean-squared-error (MSE)* between the true value of the expectation of interest ($V_{MP_t}(A)$) and the estimate generated using the adaptive sampling method. The first results show how the methods achieve better MSEs with fewer samples for this problem. We only show results for those methods that were the most competitive. We denote by "Var" the method based on the minimization of the variance, and by "L2 ", "KL1", and "KLS" the methods based on the global minimization of L_2, KL_1 and KL_s respectively. For the update methods we use $N(t) = 1$ for all t. We take into

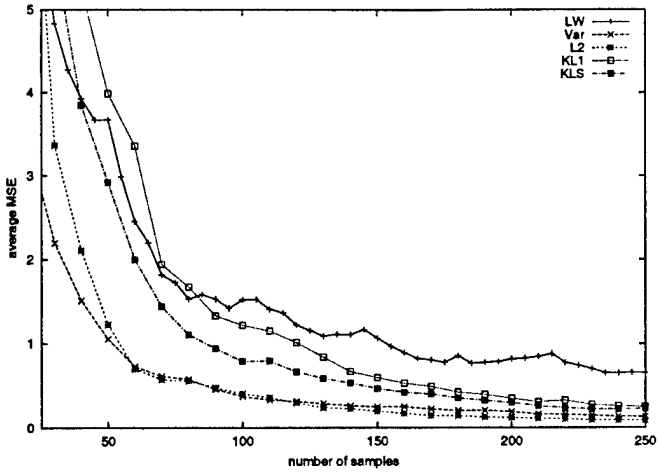


Figure 3: Average mean squared error, over 40 runs, as a function of the number of samples taken. We allow LW twice as many samples.

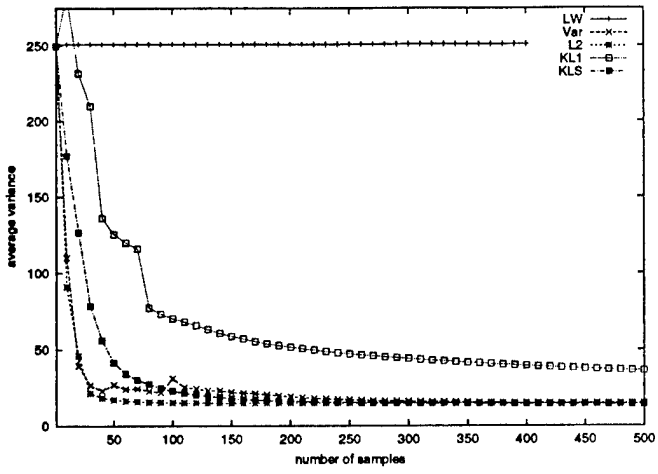


Figure 4: Average of the true variance of the weight function, over 40 runs, as a function of the total number of samples taken.

account that the update methods have to traverse the graph once every iteration to update the parameters relevant to the sample taken. To compensate for this time, we allow the estimate based on LW to use twice as many samples. Figure 3 shows the results. The graph shows the average MSE over 40 runs as a function of the total number of samples taken (times 2 for LW) by the methods. We note that Var and L2 achieve better MSEs than LW and converge to them faster. With significance level 0.005 we can state (individually) for each total number of samples $N = 50, 150, 250$, that Var and L2 (individually) are better with respect to MSE than LW. Also, for $N = 250$, KLS is better than LW.

We also ran the methods with $N(t) = 50$, including the local heuristic methods. They were only competitive after a larger total number of samples ($N > 150$). Although further analysis is necessary, we would like to convey some general observations. We believe that in general there is a tradeoff in the setting of $N(t)$ and β. We note that, of the updates based on the two KL versions, KL1 typically performs better than KL2. We believe this is because the error function e_{KL_1} is defined with respect to the optimal sampling distribution while e_{KL_2} is with respect to the current sampling distribution. KLS seems to perform better than both. L2 is more stable than any of the other methods, suggesting further theoretical analysis which we are currently undertaking. Several possible reasons for this behavior are (1) the variance of the gradient might be smaller than in other cases, (2) the error function is bounded, and/or (3) the error surface might be smoother than in other cases. We conjecture that L2 converges to a stationary point of e_{L_2}.

The second result shows that the update methods indeed lead to importance-sampling distributions with smaller variance relatively quickly for this problem. Figure 4 shows a graph of the true variance of the sampling distribution learned using the different update methods as a function of the total number of samples used. The horizontal line shows the variance associated with the sampling distribution used by LW (i.e., the "prior" distribution of the original BN).

These experiments are all carried out on a single problem. Although they must clearly be extended to a variety of larger problems, they indicate that adaptive importance-sampling methods, particularly those that minimize variance and the L_2 norm, can lead to significant improvements in the efficiency of sampling as a method for computing large expectations.

Acknowledgments

The dynamic weighting scheme and the $1/\sigma^2$ recommendation in Section 4.1 and the ϵ-boundary in Section 6 were independently developed by Jian Cheng and Marek Druzdzel. Both heuristics are reported in a manuscript that the first author saw while he was working on this paper.

We would like to thank Milos Hauskrecht, Thomas Hofmann, Kee-Eung Kim and Thomas Dean for many discussions and feedback. Also, our implementation uses some of the functionality of the *Bayes Net Toolbox for Matlab* [Murphy, 1999], for which we thank Kevin Murphy. We would also like to thank the anonymous reviewers for their insightful comments.

Luis E. Ortiz was supported in part by an NSF Graduate Fellowship and in part by NSF IGERT award SBR 9870676. Leslie Pack Kaelbling was supported in part by a grant from NTT and in part by DARPA Contract #DABT 63-99-1-0012.

References

John M. Charnes and Prakash P. Shenoy. A forward Monte Carlo method for solving influence diagrams using local computation. School of Business, University of Kansas, Working Paper No. 273, August 1999.

Gregory F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

Robert Fung and Kuo-Chu Chang. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 112–117, 1989.

John Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6): 1317–1339, November 1989.

Yan Lin and Marek Druzdzel. Stochastic sampling and search in belief updating algorithms for very large Bayesian networks. In *Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information*, pages 77–82, Stanford, California, March 1999. Stanford University. Available from `http://www.pitt.edu/~druzdzel/publ.html`.

Kevin P. Murphy. Bayes net toolbox for Matlab, 1999. Available from `http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html`.

Radford M. Neal. Annealed importance sampling. Technical Report 9805, Department of Statistics, University of Toronto, Toronto, Ontario, Canada, September 1998. Available from `http://www.cs.utoronto.ca/~radford/`.

Luis E. Ortiz and Leslie Pack Kaelbling. Sampling methods for action selection in influence diagrams. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 2000. Forthcomming.

Ross D. Shachter and Mark A. Peot. Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 311–318, 1989.

Michael Shwe and Gregory Cooper. An empirical analysis of likelihood-weighting simulation on a large, multiply connected medical belief network. *Computers and Biomedical Research*, 24:453–475, 1991.

Sampling Methods for Action Selection in Influence Diagrams

Luis E. Ortiz
Computer Science Department
Brown University
Box 1910
Providence, RI 02912 USA
leo@cs.brown.edu

Leslie Pack Kaelbling
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139 USA
lpk@ai.mit.edu

Abstract

Sampling has become an important strategy for inference in belief networks. It can also be applied to the problem of selecting actions in influence diagrams. In this paper, we present methods with probabilistic guarantees of selecting a near-optimal action. We establish bounds on the number of samples required for the traditional method of estimating the utilities of the actions, then go on to extend the traditional method based on ideas from sequential analysis, generating a method requiring fewer samples. Finally, we exploit the intuition that equally good value estimates for each action are not required, to develop a heuristic method that achieves major reductions in required sample size. The heuristic method is validated empirically.

Introduction

The problem of decision-making involves the selection of an *optimal strategy*. A strategy determines how we should act based on observations or available information about the variables of the system relevant to the decision problem. Posed in the framework of decision theory, an optimal strategy is one that maximizes our utility. The utility defines our notion of value associated with the execution of actions and the states of the system. The states result from the combination of the state of the individual variables in the system. In the case of decision-making under uncertainty, we are uncertain about both the state of the system and the result of the actions we take. We express this uncertainty as probabilities. Therefore, in this context an *optimal strategy* is one that *maximizes our expected utility*.

In this paper our main interest is in decision problems under uncertainty formulated as *influence diagrams (ID)*. An *influence diagram* is a graphical model that provides a compact representation of (1) the probability distribution governing the states, (2) the structural strategy model representing how we make decisions, and (3) a utility model defining our notion of value associated with actions and states. We study the problem of selecting an optimal strategy in an influence diagram, concentrating on the case in which there is only one decision to be made. This is because we can decompose the problem of multiple decisions into many subproblems involving single decisions (i.e., by using the technique presented by Charnes & Shenoy (1999)). We note that we can apply methods developed to solve IDs of this kind to obtain methods to solve finite-horizon Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs) expressed as dynamic Bayesian networks (DBNs) (i.e., by modifying the technique presented by Kearns, Mansour, & Ng (1999)).

The problem of strategy selection involves the subproblem of selecting an *optimal action*, from the set of action choices available for that decision, *for each possible observation* available at the time of making the decision. Therefore, we want to select the action that maximizes the expected utility for each observation. One way to do action selection is to compute, exactly or approximately, the probabilities of the sub-states of the system directly relevant to our utility in order to evaluate the expected utility or *value* of each action. A sub-state is formed from the state of a subset of variables in the system. We believe this approach fails to take advantage of an important intuition: it only matters which action is best. Therefore, the problem of action selection is primarily one of comparing the values of the actions. We combine this with the intuition that actions that are close to optimal are also good. In this paper, we present methods for action selection in IDs that take advantage of these intuitions to make major gains in efficiency.

Notation

Before we present the definition of the ID model, we introduce some notation used throughout the paper. We denote one-dimensional random variables by capital letters and denote multi-dimensional random variables by bold capital letters. For instance, we denote a multi-dimensional random variable by $\boldsymbol{X}$ and denote all its components by $(X_1, \ldots, X_n)$ where X_i is the i^{th} one-dimensional random variable. We use small letters to denote assignments to random variables. For instance, $\boldsymbol{X} = \boldsymbol{x}$ means that for each component X_i of $\boldsymbol{X}$, $X_i = x_i$. We also denote by capital letters the nodes in a graph. We denote by $Pa(Y)$ the parents of node Y in a directed graph.

We now introduce notation that will become useful during the description of the methods presented in this paper. For any function h with variables $\boldsymbol{X}$ and $\boldsymbol{Z}$, the expression

$$h(\boldsymbol{X}, \boldsymbol{Z})|_{\boldsymbol{Z}=\boldsymbol{z}}$$

stands for a function f' over variables $\boldsymbol{X}$ that results from setting the values of $\boldsymbol{Z}$ in h with assignment $\boldsymbol{z}$ while letting the values for $\boldsymbol{X}$ remain unassigned. In other words,

$$f'(\boldsymbol{X}) = h(\boldsymbol{X}, \boldsymbol{Z})|_{\boldsymbol{Z}=\boldsymbol{z}} = h(\boldsymbol{X}, \boldsymbol{Z} = \boldsymbol{z}).$$

The notation $\boldsymbol{Z} = (\boldsymbol{S}, \boldsymbol{S}')$ means that the variable $\boldsymbol{Z}$ is formed by all the variables that form $\boldsymbol{S}$ and $\boldsymbol{S}'$. That is, $\boldsymbol{Z} = (Z_1, \ldots, Z_{n'}) = (S_1, \ldots, S_{n_1}, S'_1, \ldots, S'_{n_2}) = (\boldsymbol{S}, \boldsymbol{S}')$, where $n' = n_1 + n_2$. Note that we are assuming that the set of variables forming $\boldsymbol{S}$ and those forming $\boldsymbol{S}'$ are disjoint. The notation $\boldsymbol{Z} \sim f$ means that the random variable $\boldsymbol{Z}$ is distributed according to probability distribution f. We denote a sequence of samples from $\boldsymbol{Z}$ by $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots$, where $\boldsymbol{z}^{(i)}$ is the i^{th} sample. In this paper, we assume that the samples are *independent*.

Definitions

An influence diagram (ID) is a graphical model for decision-making (See Jensen (1996) for additional information and references). It consists of a directed acyclic graph along with a structural strategy model, a probabilistic model and a utility model. The graph represents the decomposition used to compactly define the different models. Figure 1 shows an example of a general graphical representation of an ID. The vertices of the graph consist of three types of nodes: decision nodes, chance nodes and utility nodes. Decision nodes are square and represent the decisions or action choices in the decision problem. Chance nodes are circular and represent the variables of the system relevant to the decision problem. Utility nodes are diamonds and represent the utility associated with actions and *states*. A *state* is an assignment to the variables associated with the chance nodes of the ID.

Structural strategy model The structural strategy model defines locally the form of a decision rule for each decision node A_i. This rule is a function of (a subset of) the information available at the time of making that decision, which is contained in its parents $\mathrm{Pa}(A_i)$ in the graph, the decision nodes that are predecessors of decision node A_i in the graph and their respective parents. The example ID of Figure 1 has only one decision node. Denote a strategy for our example model by π, the *state space* or set of possible assignments for the parents of the action node by $\Omega_{\mathrm{Pa}(A)}$ and the set of possible actions Ω_A. Then, a policy $\pi : \Omega_{\mathrm{Pa}(A)} \rightarrow \Omega_A$.

Probability model The probability model compactly defines the joint probability distribution of the relevant variables given the actions taken using a Bayesian network (BN) (See Jensen (1996) for additional information and references). The model defines locally a conditional probability distribution $P(X_i \mid \mathrm{Pa}(X_i))$ for each variable X_i given its parents $\mathrm{Pa}(X_i)$ in the graph. This defines the following joint probability distribution over the n variables of the system, given that a particular action $\boldsymbol{a}$ is taken:

$$P(X_1, \ldots, X_n \mid \boldsymbol{A} = \boldsymbol{a}) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i))|_{\boldsymbol{A}=\boldsymbol{a}}.$$

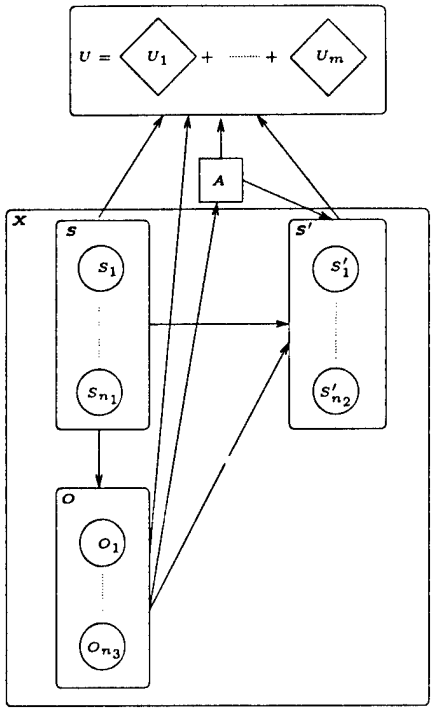


Figure 1: General structure of ID we consider.

In our example ID, $\boldsymbol{X} = (\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O})$ and, since there is only one decision node, we can express $P(\boldsymbol{X} \mid \boldsymbol{A} = \boldsymbol{a})$ as

$$\begin{aligned} P(\boldsymbol{X} \mid \boldsymbol{A} = \boldsymbol{a}) &= P(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} \mid \boldsymbol{A} = \boldsymbol{a}) \\ &= P(\boldsymbol{S}) P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O}, \boldsymbol{A} = \boldsymbol{a}) P(\boldsymbol{O} \mid \boldsymbol{S}), \end{aligned}$$

where

$$P(\boldsymbol{S}) = \prod_{i=1}^{n_1} P(S_i \mid \mathrm{Pa}(S_i)), \quad (1)$$

$$P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O}, \boldsymbol{A} = \boldsymbol{a}) = \prod_{i=1}^{n_2} P(S'_i \mid \mathrm{Pa}(S'_i))|_{\boldsymbol{A}=\boldsymbol{a}} \quad (2)$$

$$P(\boldsymbol{O} \mid \boldsymbol{S}) = \prod_{i=1}^{n_3} P(O_i \mid \mathrm{Pa}(O_i)). \quad (3)$$

Utility model Finally, the utility model defines the utility associated with actions resulting from the decisions made and states of the variables in the system. The total utility function U is the sum of local utility functions associated with each utility node. For each utility node U_i, the utility function provides a utility value as a function of its parents $\mathrm{Pa}(U_i)$ in the graph. The total utility can be expressed as

$$U(\boldsymbol{X}, \boldsymbol{A}) = \sum_{i=1}^{m} U_i(\mathrm{Pa}(U_i)). \quad (4)$$

Note that we are using the label of the utility node to also denote the utility function associated with it.

In this paper we assume that the variables and the decisions are discrete and the local utilities are bounded. In addition, we concentrate on IDs with one decision node and the general structure shown in Figure 1. The results in this paper are still valid for more general structural decompositions of the probability distribution. We use the structure given by the ID in the figure to simplify the presentation. Also, the results allow random utility functions.

Value of a strategy The *value* V^π *of a strategy* π is the expected utility of the strategy:

$$\begin{aligned} V^\pi &= \sum_{\boldsymbol{X}} P(\boldsymbol{X} \mid A = \pi(\boldsymbol{O}))U(\boldsymbol{X}, A = \pi(\boldsymbol{O})) \\ &= \sum_{\boldsymbol{O}} \sum_{\boldsymbol{S}} \sum_{\boldsymbol{S}'} P(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} \mid A = \pi(\boldsymbol{O})) \\ &\qquad U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} \mid A = \pi(\boldsymbol{O})). \end{aligned}$$

The optimal strategy π^* is that which maximizes V^π over all π. We denote the value of the optimal strategy by V^*.

Note that we can decompose this maximization into maximizations over the set of actions for each observation. For each assignment to the observations $\boldsymbol{o}$, we define the *value of an action* a by

$$\begin{aligned} V_{\boldsymbol{o}}(a) = \sum_{\boldsymbol{S}} \sum_{\boldsymbol{S}'} P(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o} \mid A = a) \\ U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o} \mid A = a). \quad (5) \end{aligned}$$

Hence, the value of a strategy is $V^\pi = \sum_{\boldsymbol{O}} V_{\boldsymbol{O}}(\pi(\boldsymbol{O}))$. Note that this is not the traditional definition of the value of an action. We discuss below why we do not use the traditional definition.

If we denote by $a^* = \pi^*(\boldsymbol{o})$ the action that maximizes $V_{\boldsymbol{o}}(a)$ over all actions a, then the value of the optimal strategy is $V^* = \sum_{\boldsymbol{O}} V_{\boldsymbol{O}}(\pi^*(\boldsymbol{O})) = \sum_{\boldsymbol{O}} \max_a V_{\boldsymbol{O}}(a)$. Hence, the problem of strategy selection reduces to that of action selection for each observation.

Exact methods exist for computing the optimal strategy in an ID (See Charnes & Shenoy (1999) and Jensen (1996) for short descriptions and a list of references). However, this problem is hard in general. In this paper, we concentrate on obtaining approximations to the optimal strategy with certain guarantees. Our objective is to find policies that are close to optimal with high probability. That is, for a given accuracy parameter ϵ^* and confidence parameter δ^*, we want to obtain a strategy $\hat{\pi}$ such that $V^* - V^{\hat{\pi}} < \epsilon^*$ with probability at least $1-\delta^*$. Note that given the decomposition described above, if we obtain actions for each observation such that their value is *sufficiently* close to optimal with *sufficiently* high probability, then we obtain a near-optimal strategy with high probability. That is, let l be the number of possible assignments to the observations. If for each observation $\boldsymbol{o}$ we select action $\hat{a}$ such that $V_{\boldsymbol{o}}(a^*) - V_{\boldsymbol{o}}(\hat{a}) < 2\epsilon$ with probability at least $1-\delta$, where $\epsilon = \epsilon^*/(2l)$ and $\delta = \delta^*/l$, then we obtain a strategy that is within ϵ^* of the optimal with probability at least $1-\delta^*$. Therefore, we concentrate on finding a *good* action for each observation.

Typically the value of an action is defined as the *conditional* expected utility of the action *given* an assignment of the observations. If we denote this value by $V(a \mid \boldsymbol{o})$, we can express the value of a policy as $V^\pi = \sum_{\boldsymbol{O}} P(\boldsymbol{O}) V(\pi(\boldsymbol{O}) \mid \boldsymbol{O})$. We do not use this definition because it is harder to obtain estimates for $V(a \mid \boldsymbol{o})$ with guaranteed confidence bounds than it is to obtain estimates for $V_{\boldsymbol{o}}(a)$.

Multiple Comparisons with the Best: Results

There are two important results from the field of *multiple comparisons* and in particular from the field of *multiple comparisons with the best* that we take advantage of in this paper. These results are based on the work of Hsu (1981) (See Hsu (1996) for more information). Before we present the results we introduce the following notation: denote $x^+ = \max(x, 0)$ and $-x^- = \min(0, x)$. The first result is known as *Hsu's single-bound lemma*, which is presented as Lemma 1 by Matejcik & Nelson (1995).

Lemma 1 *Let* $\mu_{(1)} \leq \mu_{(2)} \leq \cdots \leq \mu_{(k)}$ *be the (unknown) ordered performance parameters of* k *systems, and let* $\hat{\mu}_{(1)}, \hat{\mu}_{(2)}, \ldots, \hat{\mu}_{(k)}$ *be any estimators of the parameters. If*

$$\begin{aligned} Pr\{\hat{\mu}_{(k)} - \hat{\mu}_{(i)} - (\mu_{(k)} - \mu_{(i)}) > -w, i = 1, \ldots, k-1\} \\ = 1 - \alpha, \quad (6) \end{aligned}$$

then

$$\begin{aligned} Pr\{\mu_i - \max_{j \neq i} \mu_j \in [-(\hat{\mu}_i - \max_{j \neq i} \hat{\mu}_j - w)^-, \\ (\hat{\mu}_i - \max_{j \neq i} \hat{\mu}_j + w)^+], \text{ for all } i\} \geq 1 - \alpha. \quad (7) \end{aligned}$$

If we replace the $=$ *in* (6) *with* $\geq$, *then* (7) *still holds.*

In our context, we let for each action a, the true value $\mu_a = V_{\boldsymbol{o}}(a)$ and the estimate $\hat{\mu}_a = \hat{V}_{\boldsymbol{o}}(a)$. Also, the i^{th} smallest true value corresponds to $\mu_{(i)}$. That is, if $V_{\boldsymbol{o}}(a_1) \leq V_{\boldsymbol{o}}(a_2) \leq \cdots \leq V_{\boldsymbol{o}}(a_k)$, then for all i, $\mu_{(i)} = V_{\boldsymbol{o}}(a_i)$. Note that in practice, we do not know which action has the largest value. In order to apply Hsu's single-bound lemma, we obtain the bound $Pr\{\hat{\mu}_j - \hat{\mu}_i - (\mu_j - \mu_i) > -w, \text{ for all } i \neq j\} \geq 1 - \alpha$, for each action j, individually. This implies that $Pr\{\hat{\mu}_{(k)} - \hat{\mu}_{(i)} - (\mu_{(k)} - \mu_{(i)}) > -w, i = 1, \ldots, k-1\} \geq 1 - \alpha$, which allow us to apply the lemma. Figure 2 graphically describes this practical interpretation of the lemma. For each action i, individually, the upper bounds on the true differences, drawn on the left-hand side, $V_{\boldsymbol{o}}(i) - V_{\boldsymbol{o}}(j) < \hat{V}_{\boldsymbol{o}}(i) - \hat{V}_{\boldsymbol{o}}(j) + w$, for each $j \neq i$, hold simultaneously with probability at least $1 - \alpha$. The confidence intervals, drawn on the right-hand side, $V_{\boldsymbol{o}}(i) - \max_{j \neq i} V_{\boldsymbol{o}}(j) \in [-(\hat{V}_{\boldsymbol{o}}(i) - \max_{j \neq i} \hat{V}_{\boldsymbol{o}}(j) - w)^-, (\hat{V}_{\boldsymbol{o}}(i) - \max_{j \neq i} \hat{V}_{\boldsymbol{o}}(j) + w)^+]$, for each action i, hold simultaneously with probability at least $1 - \alpha$.

The second result allows us to assess joint confidence intervals on the difference between the value of each action from the value of the best action when we have estimates of the differences between value of each pair of actions with different degrees of accuracy. The result is known as *Hsu's multiple-bound lemma*. It is presented as Lemma 2 by Matejcik & Nelson (1995), and credited to Chang & Hsu (1992).

Lemma 2 *Let* $\mu_{(1)} \leq \mu_{(2)} \leq \cdots \leq \mu_{(k)}$ *be the (unknown) ordered performance parameters of* k *systems. Let* T_{ij} *be a point estimator of the parameter* $\mu_i - \mu_j$. *If for each* i *individually*

$$Pr\{T_{ij} - (\mu_i - \mu_j) > -w_{ij}, \text{for all } j \neq i\} = 1 - \alpha, \quad (8)$$

then we can make the joint probability statement

$$Pr\{\mu_i - \max_{j \neq i} \mu_j \in [D_i^-, D_i^+], \text{for all } i\} \geq 1 - \alpha, \quad (9)$$

where $D_i^+ = (\min_{j \neq i}[T_{ij} + w_{ij}])^+$, $\mathcal{G} = \{l : D_l^+ > 0\}$, *and*

$$D_i^- = \begin{cases} 0 & \text{if } \mathcal{G} = \{i\} \\ -(\min_{j \in \mathcal{G}, j \neq i}[-T_{ji} - w_{ji}])^- & \text{otherwise.} \end{cases}$$

If we replace the $=$ *in* (8) *with* $\geq$, *then* (9) *still holds.*

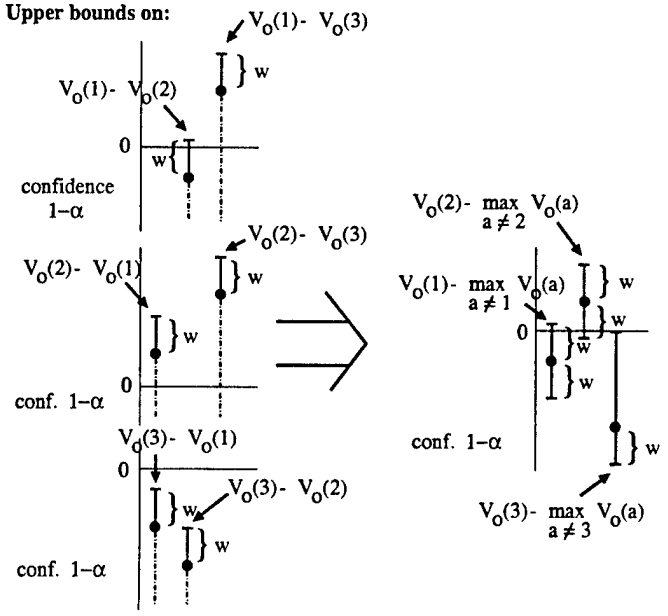


Figure 2: Graphical description for practical application of Hsu's single-bound lemma. Note that the "lower bounds" on the left-hand side are $-\infty$.

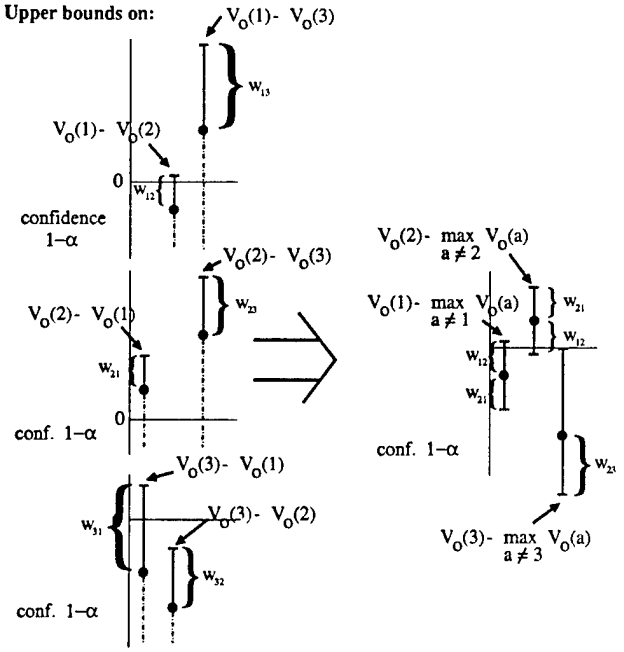


Figure 3: Graphical description of Hsu's multiple-bound lemma. Note that the "lower bounds" on the left-hand side are $-\infty$.

Figure 3 presents a graphical description of this lemma. Let, for all actions i, D_i^- and D_i^+, be as defined in Hsu's multiple-bound lemma, with $\mu_i = V_o(i)$ and for all $j \neq i$, $T_{ij} = \hat{V}_o(i) - \hat{V}_o(j)$. For each action i, individually, the upper bounds on the true differences, drawn on the left-hand side, $V_o(i) - V_o(j) < T_{ij} + w_{ij}$, for each $j \neq i$, hold simultaneously with probability at least $1-\alpha$. The confidence intervals, drawn on the right-hand side, $V_o(i) - \max_{j \neq i} V_o(j) \in [D_i^-, D_i^+]$, for each action i, hold simultaneously with probability at least $1-\alpha$. Also, in this example, $\mathcal{G} = \{1, 2\}$. In our context, $\mathcal{G}$ is the set of all the actions that could potentially be the best with probability at least $1-\alpha$. That is, for each action a in $\mathcal{G}$, the upper bound D_a^+ on the difference of the true value of action a and the best of *all* the other actions, including those in $\mathcal{G}$, is positive.

Estimation-based methods

One approach to selecting the best action is to obtain estimates of $V_o(a)$ for each a by sampling, using the probability model of the ID conditioned on a, then select the action with the largest estimated value.

We can apply the idea of *importance sampling* (See Geweke (1989) and the references therein) to this estimation problem by using the probability distribution defined by the ID as *the importance function* or *sampling distribution*. This is essentially the same idea as *likelihood-weighting* in the context of probabilistic inference in Bayesian networks (Shachter & Peot, 1989; Fung & Chang, 1989). We present this method in the context of our example ID.

First, we present definitions that will allow us to rewrite $V_o(a)$ more clearly. First, let $\boldsymbol{Z} = (\boldsymbol{S}, \boldsymbol{S}')$. Define the *target function* (in our case, the *weighted utilities*)

$$\begin{aligned} g_{a,o}(\boldsymbol{Z}) &= g_{a,o}(\boldsymbol{S}, \boldsymbol{S}') \\ &= P(\boldsymbol{S})P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O} = \boldsymbol{o}, A = a) \cdot \\ &\quad P(\boldsymbol{O} = \boldsymbol{o} \mid \boldsymbol{S})U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o}, A = a). \end{aligned}$$

Note that $V_o(a) = \sum_{\boldsymbol{Z}} g_{a,o}(\boldsymbol{Z})$. Define the *importance function* as

$$f_{a,o}(\boldsymbol{Z}) = P(\boldsymbol{S})P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O} = \boldsymbol{o}, A = a). \quad (10)$$

Define the *weight function* $\omega_{a,o}(\boldsymbol{Z}) = g_{a,o}(\boldsymbol{Z})/f_{a,o}(\boldsymbol{Z})$. Note that in this case,

$$\omega_{a,o}(\boldsymbol{Z}) = P(\boldsymbol{O} = \boldsymbol{o} \mid \boldsymbol{S})U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o}, A = a). \quad (11)$$

Finally, note that $V_o(a) = \sum_{\boldsymbol{Z}} f_{a,o}(\boldsymbol{Z})(g_{a,o}(\boldsymbol{Z})/f_{a,o}(\boldsymbol{Z}))$. The idea of the sampling methods described in this section is to obtain independent samples according to $f_{a,o}$, use those samples to estimate the value of the actions, and finally select an approximately optimal action by taking the action with largest value estimate. Denote the *weight of a sample* $\boldsymbol{z}^{(i)}$ from $\boldsymbol{Z} \sim f_{a,o}$ as $\omega_{a,o}^{(i)} = \omega_{a,o}(\boldsymbol{z}^{(i)})$. Then an unbiased estimate of $V_o(a)$ is $\hat{V}_o(a) = \frac{1}{N_{a,o}} \sum_{i=1}^{N_{a,o}} \omega_{a,o}^{(i)}$.

Traditional Method

We can obtain an estimate of $V_o(a)$ using the straightforward method presented in Algorithm 1; it requires parameters $N_{a,o}$ that will be defined in Theorem 1.

This is the traditional sampling-based method used for action selection. However, we are unaware of any result regarding the number of samples needed to obtain a near-optimal strategy with high probability using this method.

Algorithm 1 Traditional Method

1. Obtain independent samples $z^{(1)}, \ldots, z^{(N_{a,o})}$ from $Z \sim f_{a,o}$.
2. Compute the weights $\omega_{a,o}^{(1)}, \ldots, \omega_{a,o}^{(N_{a,o})}$.
3. Output $\hat{V}_o(a)$ = average of the weights.

Theorem 1 *If for each possible action $i = 1, \ldots, k$, we estimate $V_o(i)$ using the traditional method, the weight function satisfies $l_{i,o} \leq \omega_{i,o}(Z) \leq u_{i,o}$, and the estimate uses*

$$N_{i,o} = \left\lceil \frac{(u_{i,o} - l_{i,o})^2}{2\epsilon^2} \ln \frac{k}{\delta} \right\rceil$$

samples, then the action with the largest value estimate has a true value that is within 2ϵ of the optimal with probability at least $1 - \delta$.

Proof sketch. The proof goes in three basic steps. First, we apply *Hoeffding bounds* (Hoeffding, 1963) to obtain a bound on the probability that each estimate deviates from its true mean by some amount ϵ. Then, we apply the *Bonferroni inequality (Union bound)* to obtain joint bounds on the probability that the difference of each estimate from all the others deviates from the true difference by 2ϵ. Finally, we apply Hsu's single bound lemma to obtain our result.

Note that we can compute $l_{i,o}$ and $u_{i,o}$ efficiently from information local to each node in the graph. Assuming that we have non-negative utilities, we can let

$$u_{i,o} = \left[\prod_{j=1}^{n_3} \max_{\mathrm{Pa}(O_j)} P(O_j \mid \mathrm{Pa}(O_j))|_{O=o}\right] \cdot \left[\sum_{j=1}^{m} \max_{\mathrm{Pa}(U_j)} U_j(\mathrm{Pa}(U_j))|_{O=o,A=i}\right], \quad (12)$$

$$l_{i,o} = \left[\prod_{j=1}^{n_3} \min_{\mathrm{Pa}(O_j)} P(O_j \mid \mathrm{Pa}(O_j))|_{O=o}\right] \cdot \left[\sum_{j=1}^{m} \min_{\mathrm{Pa}(U_j)} U_j(\mathrm{Pa}(U_j))|_{O=o,A=i}\right]. \quad (13)$$

However, these bounds can be very loose.

Sequential Method

The sequential method tries to reduce the number of samples needed by the traditional method, using ideas from sequential analysis. The idea is to first obtain an estimate of the variance and then use it to compute the number of samples needed to estimate the mean. The method, presented in Algorithm 2, requires parameters $N'_{a,o}$ and $N''_{a,o}$ that will be defined in Theorem 2.

Note that given the sequential nature of the method, the total number of samples is now a random variable. We also note that while multi-stage procedures of this kind are commonly used in the statistical literature, we are only aware of results based on restricting assumptions on the distribution of the random variables (i.e., parametric families like normal and binomial distributions) (Bechhofer, Santner, & Goldsman, 1995).

Theorem 2 *If, for each possible action $i = 1, \ldots, k$, we estimate $V_o(i)$ using the sequential method, the weight function satisfies $l_{i,o} \leq \omega_{i,o}(Z) \leq u_{i,o}$, $\sigma^2_{i,o} = Var[\omega_{i,o}(Z)]$,*

$$N'_{i,o} = \left\lceil \frac{(u_{i,o} - l_{i,o})^{4/3}}{2\, 2^{2/3} \epsilon^{4/3}} \ln \frac{2k}{\delta} \right\rceil,$$

and

$$N''_{i,o}(\hat{\sigma}^2_{i,o}) = \left\lceil \left(\frac{2\hat{\sigma}^2_{i,o} + 2(u_{i,o} - l_{i,o})\epsilon/3}{\epsilon^2} + 2^{1/3} \frac{(u_{i,o} - l_{i,o})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{2k}{\delta} \right\rceil,$$

then the action with the largest value estimate has a true value that is within 2ϵ of the optimal with probability at least $1 - \delta$. Also,

$$\begin{aligned} N_{i,o} &< \left(\frac{2\sigma^2_{i,o} + 2(u_{i,o} - l_{i,o})\epsilon/3}{\epsilon^2} + \frac{5}{2^{2/3}} \frac{(u_{i,o} - l_{i,o})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{2k}{\delta} + 1 \\ &= O\left(\max\left(\frac{\sigma^2_{i,o}}{\epsilon^2}, \frac{(u_{i,o} - l_{i,o})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{k}{\delta} \right), \end{aligned}$$

with probability at least $1 - \delta/(2k)$, and

$$\begin{aligned} E[N_{i,o}] &= 2N'_{i,o} + N''_{i,o}(\sigma^2_{i,o}) \\ &= O\left(\max\left(\frac{\sigma^2_{i,o}}{\epsilon^2}, \frac{(u_{i,o} - l_{i,o})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{k}{\delta} \right). \end{aligned}$$

Algorithm 2 Sequential Method

1. Obtain independent samples $z^{(1)}, \ldots, z^{(2N'_{a,o})}$ from $Z \sim f_{a,o}$.
2. Compute the weights $\omega_{a,o}^{(1)}, \ldots, \omega_{a,o}^{(2N'_{a,o})}$.
3. For $j = 1, \ldots, N'_{a,o}$, let $y_j = (\omega_{a,o}^{(2j-1)} - \omega_{a,o}^{(2j)})^2/2$.
4. Compute $\hat{\sigma}^2_{a,o}$ = average of y_j's.
5. Let $N_{a,o} = 2N'_{a,o} + N''_{a,o}(\hat{\sigma}^2_{a,o})$.
6. Obtain $N''_{a,o}(\hat{\sigma}^2_{a,o})$ new independent samples $z^{(2N'_{a,o}+1)}, \ldots, z^{(N_{a,o})}$ from $Z \sim f_{a,o}$.
7. Compute the new weights $\omega_{a,o}^{(2N'_{a,o}+1)}, \ldots, \omega_{a,o}^{(N_{a,o})}$.
8. Output $\hat{V}_o(a)$ = average of the new weights.

Proof sketch. The only difference from the proof of Theorem 1 is the first step. Instead of using Hoeffding bounds to bound the probability that each estimate deviates from its true mean, we use a combination of *Bernstein's inequality* (as presented by Devroye, Györfi, & Lugosi (1996) and credited to Bernstein (1946)) and Hoeffding bounds as follows. We first use the Hoeffding bound to bound the probability that the estimate of the variance after taking some number of samples $2N'$ deviates from the true variance by some amount ϵ'. We then use Bernstein's inequality to bound the probability that the estimate we obtain after taking some number of samples N'' deviates from its true mean by

ϵ *given* that the true variance is no larger than our estimate of the variance plus ϵ'. We then find the value of ϵ' (in terms of ϵ) that minimizes the total number of samples $N'' + 2N'$. The results on the number of samples follow by substituting the minimizing ϵ' back into the expressions for N'' and N'. Steps 2 and 3 are as in Theorem 1.

The sequential method is particularly more effective than the traditional method when $\sigma^2_{i,o} \ll (u_{i,o} - l_{i,o})^2$.

Comparison-based Method

Using the results from MCB, we can compute simultaneous or joint confidence intervals on the difference between the value of $V_o(a)$ and the best of all the others for all actions a. Therefore, MCB allows us to select the best action choice or an action with value close to it, within a confidence level.

In the previous section we presented methods that require that we have estimates with the same precision in order to select a good action. Hsu's multiple-bound lemma applies when we do not have estimates of $V_o(a)$ for each a with the same precision. Based on this result, we propose the method presented in Algorithm 3 for action selection.

Algorithm 3 Comparison-based Method

1. Obtain an *initial number of samples* for each action a.
2. Compute *MCB confidence intervals* on the difference in value of each action from the best of the other actions using those samples.

while *not able to select a good action with high certainty* **do**

3(a). Obtain *additional samples*.

3(b). Recompute MCB confidence intervals using total samples so far.

We compute the MCB confidence intervals heuristically. To do this, we approximate the precisions that satisfy the conditions required by Hsu's multiple-bound lemma (Equation 8) using Hoeffding bounds (Hoeffding, 1963). Using this approach, for each pair of actions i and j, and values $l_{ij,o}$ and $u_{ij,o}$ such that $l_{ij,o} \leq \omega_{i,o}(\boldsymbol{Z}) \leq u_{ij,o}$ and $l_{ij,o} \leq \omega_{j,o}(\boldsymbol{Z}) \leq u_{ij,o}$, we approximate w_{ij} as

$$w_{ij} = (u_{ij,o} - l_{ij,o}) \sqrt{\frac{1}{2}\left(\frac{1}{N_{i,o}} + \frac{1}{N_{j,o}}\right) \ln \frac{k-1}{\delta}}, \quad (14)$$

where $N_{i,o}$ is the number of samples taken for action i thus far. We then use these approximate precisions and the value-difference estimates to compute the MCB confidence intervals (as specified by Equation 9). There are alternative ways of heuristically approximating the precisions but, in this paper, we use the one above for simplicity.

Once we compute the intervals, the stopping condition is as follows. If at least one of the lower bounds of the MCB confidence intervals is greater than -2ϵ, then we stop and select the action that attains this lower bound. Otherwise, we continue taking additional samples.

We define the value of *initial number of samples* in our experiments as 40. When taking additional samples, we use a sampling schedule that is somewhat selective in that it takes more samples from more promising actions as suggested by the MCB confidence intervals. We find the action whose corresponding MCB confidence interval has an upper bound greater than 0 (i.e., from the set $\mathcal{G}$ as defined in Hsu's multiple bound lemma) and whose lower bound is the largest. We take 40 additional samples from this action and 10 from all the others. We understand that these sample sizes are very arbitrary. Potentially, other setting of these sample sizes can be more effective but we did not try to optimize them for our experiments. Algorithm 4 presents a detailed description of the instance of the method we used in the experiments.

Algorithm 4 Algorithmic description of the instance of the comparison-based method used in the experiments.

for each observation o **do**
  $l \leftarrow 1$
  for each action $i = 1, \ldots, k$ **do**
    Compute $u_{i,o}$ and $l_{i,o}$ using equations 12 and 13, respectively.
    $D_i^- \leftarrow -\infty$; $N_{i,o}^{(l)} \leftarrow 40$; $N_{i,o} \leftarrow 0$; $\hat{V}_o(i) \leftarrow 0$.
  for each pair of actions (i, j), $i \neq j$ **do**
    $u_{ij,o} \leftarrow \max(u_{i,o}, u_{j,o})$; $l_{ij,o} \leftarrow \max(l_{i,o}, l_{j,o})$.
  while there is no action i such that $D_i^- > -2\epsilon$ **do**
    for each action i **do**
      Obtain $N_{i,o}^{(l)}$ samples $\boldsymbol{z}^{(N_{i,o}+1)}, \ldots, \boldsymbol{z}^{(N_{i,o}+N_{i,o}^{(l)})}$ from $\boldsymbol{Z} \sim f_{i,o}$, as in equation 10.
      Compute weights $\omega_{i,o}^{(N_{i,o}+1)}, \ldots, \omega_{i,o}^{(N_{i,o}+N_{i,o}^{(l)})}$.
      $\hat{V}_o(i) \leftarrow (N_{i,o}\hat{V}_o(i) + \sum_{j=1}^{N_{i,o}^{(l)}} \omega_{i,o}^{(N_{i,o}+j)}) / (N_{i,o} + N_{i,o}^{(l)})$.
      $N_{i,o} \leftarrow N_{i,o} + N_{i,o}^{(l)}$.
    for each pair of actions (i, j), $i \neq j$ **do**
      $T_{ij} \leftarrow \hat{V}_o(i) - \hat{V}_o(j)$; $T_{ji} \leftarrow -T_{ij}$.
      Compute w_{ij} using equation 14; $w_{ji} \leftarrow w_{ij}$.
    for each action i **do**
      Compute D_i^+, $\mathcal{G}$, and D_i^- using Hsu's multiple-bound lemma.
    for each action i **do**
      if $D_i^- == \max_{j \in \mathcal{G}} D_j^-$ **then** $N_{i,o}^{(l+1)} \leftarrow 40$
      else $N_{i,o}^{(l+1)} \leftarrow 10$.
    $l \leftarrow l + 1$.
  $\hat{\pi}(o) \leftarrow \text{argmax}_i D_i^-$.

Although this method may seem well-grounded, we are not convinced that the bounds hold rigorously. The precisions are correct if the samples obtained so far for each action are independent. However, this might not be the case, since the number of samples gathered on each round depends on a property of the previous set of samples (that is, that the lower-bound condition did not hold). It is not yet clear to us whether the fact that the *number* of samples depends on the values of the samples implies that the samples must be considered dependent.

Related Work

Charnes & Shenoy (1999) present a Monte Carlo method similar to our "traditional method." One difference is that they use a heuristic stopping rule based on a normal approximation (i.e., the estimates have an *asymptotically* normal distribution). Their method takes samples until all the estimates achieve a required standard error to provide the correct confidence interval on each value under the assumption that the estimates are normally distributed and the estimate of the variance is equal to the true variance. They do not give bounds on the number of samples needed to obtain a near-optimal action with the required confidence. We refer the reader to Charnes & Shenoy (1999) for a short description and references on other similar Monte Carlo methods for IDs.

Bielza, Müller, & Insua (1999) present a method based on Markov-Chain Monte Carlo (MCMC) for solving IDs. Although their primary motivation is to handle continuous action spaces, their method also applies to discrete action spaces. Because of the typical complications in analyzing MCMC methods, they do not provide bounds on the number of samples needed. Instead, they use a heuristic stopping rule which does not guarantee the selection of a near-optimal action. Other MCMC-based methods have been proposed (See Bielza, Müller, & Insua (1999) for more information).

Empirical results

We tried the different methods on a simple made-up ID. Given space restrictions we only describe it briefly (See Ortiz (2000) for details). Figure 4 gives a graphical representation of the ID for the *computer mouse problem.* The idea is to select an optimal strategy of whether to *buy* a new mouse ($A = 1$), *upgrade* the operating system ($A = 2$), or take *no action* ($A = 3$). The observation is whether the mouse pointer is working ($MP_t = 1$) or not ($MP_t = 0$). The variables of the problem are the status of the operating system (OS), the status of the driver (D), the status of the mouse hardware (MH), and the status of the mouse pointer (MP), all at the current and future time (subscripted by t and $t+1$). The variables are all binary.

The probabilistic model encodes the following information about the system. The mouse is old and somewhat unreliable. The operating system is reliable. It is very likely that the mouse pointer will not work if either the driver or the mouse hardware has failed. Table 1 shows the utility function $U(MP_{t+1}, A)$ and the values of the actions and observations $V_O(A)$ computed using an exact method. From Table 1 we conclude that the optimal strategy is: buy a new mouse ($A = 1$) if the mouse pointer is not working ($MP_t = 0$); take no action ($A = 3$) if the mouse pointer is working ($MP_t = 1$). This strategy has value 26.50.

Table 2 presents our results on the effectiveness of the sampling methods for this problem. We set our final desired accuracy for the output strategy to $\epsilon^* = 5$ and confidence level $\delta^* = 0.05$. This leads to the individual accuracy $2\epsilon = 2.5$ and confidence level $\delta = 0.025$ for each subproblem. We executed the sequential method and the comparison-based method 100 times. The comparison-based method produces major reductions in the number of samples. When we observe the mouse pointer not working, The comparison-based method always selects the optimal action of buying a new mouse. When we observe the mouse pointer working, The comparison-based method failed to select the optimal action of *taking no action* 4 times out of the 100. In those cases, it selected the next-to-optimal action of upgrading the operating system ($A = 2$). This action is within our accuracy requirements since the difference in value with respect to the optimal action is 0.91.

The comparison-based method is highly effective in cases where there is a clear optimal action to take. For instance, in the computer mouse problem, buying a new mouse when we observe the mouse not working is clearly the best option. The differences in value between the optimal action and the rest are not as large as when we observe the mouse working.

In this problem, the results for the sequential method should not fully discourage us from its use, because the variances are still relatively large. We have seen major reductions in problems where the variance is significantly smaller than the square of the range of the variable whose mean we are estimating.

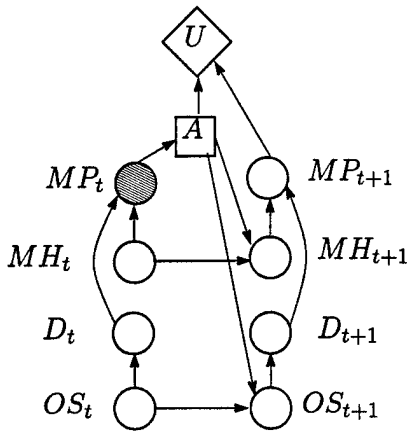


Figure 4: Graphical representation of the ID for the computer mouse problem.

	U		V	
	MP_{t+1}		MP_t	
A	0	1	0	1
1	0	40	**18.20**	6.60
2	5	45	7.54	7.39
3	10	50	10.57	**8.30**

Table 1: This table presents the utility function and the (exact) value of actions and observations for the computer mouse problem.

Summary and Conclusion

The methods presented in this paper are an alternative to exact methods. While the running time of exact methods depends on aspects of the structural decomposition of the

A	MP_t	Method		
		Traditional	Sequential	Comp-based
1	0	2403	3802 (188)	335 (151)
2	0	3007	2266 (142)	115 (37)
3	0	3679	2426 (129)	118 (39)
1	1	2213	2508 (178)	521 (216)
2	1	2794	2969 (201)	695 (421)
3	1	3443	3468 (202)	1361 (560)
Total		17539	17438 (434)	**3145** (809)

Table 2: Number of samples taken by the different methods for each action and observation. For the sequential and the comparison-based methods, the table displays the average number of samples over 100 runs. The values in parenthesis are the sample standard deviations.

ID, the running time of the methods presented in this paper depends primarily on the range of the weight functions, the variance of the value estimators and the amount of separation between the value of the best action and that of the rest (in addition to the natural dependency on the number of action choices, and the precision and confidence parameters). In some cases, we can know in advance whether they will be faster or not. The methods presented in this paper can be a useful alternative in those cases where exact methods are intractable. How useful depends on the particular characteristics of the problem.

Sampling is a promising tool for action selection. Our empirical results on a small ID suggest that sampling methods for action selection are more effective when they take advantage of the intuition that action selection is primarily a comparison task. We look forward to experimenting with IDs large enough that sampling methods are the only potentially efficient alternative. Also, our work leads to the study of adaptive sampling as a way to improve the effectiveness of sampling methods (Ortiz & Kaelbling, 2000).

Acknowledgments We would like to thank Constantine Gatsonis for suggesting the MCB literature; Eli Upfal, Milos Hauskrecht, Thomas Hofmann, Thomas Dean and Kee Eung Kim for useful discussions and suggestions; and the anonymous reviewers for their useful comments. Our implementations use the *Bayes Net Toolbox for Matlab* (Murphy, 1999), for which we thank Kevin Murphy. Luis E. Ortiz was supported in part by an NSF Graduate Fellowship and by NSF IGERT award SBR 9870676. Leslie Pack Kaelbling was supported in part by a grant from NTT and by DARPA Contract #DABT 63-99-1-0012.

References

Bechhofer, R. E.; Santner, T. J.; and Goldsman, D. M. 1995. *Design and analysis of experiments for statistical selection, screening and multiple comparisons*. Wiley.

Bernstein, S. 1946. *The Theory of Probabilities*. Gastehizdat Publishing House, Moscow.

Bielza, C.; Müller, P.; and Insua, D. R. 1999. Monte Carlo methods for decision analysis with applications to influence diagrams. *Management Science*. Forthcoming.

Chang, J. Y., and Hsu, J. C. 1992. Optimal designs for multiple comparisons with the best. *Journal of Statistical Planning and Inference* 30:45–62.

Charnes, J. M., and Shenoy, P. P. 1999. A forward Monte Carlo method for solving Influence diagrams using local computation. School of Business, University of Kansas, Working Paper No. 273.

Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A Probabilistic Theory of Pattern Recognition*. Springer.

Fung, R., and Chang, K.-C. 1989. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, 112–117.

Geweke, J. 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57(6):1317–1339.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Hsu, J. C. 1981. Simultaneous confidence intervals for all distances from the "best". *Annals of Statistics* 9(5):1026–1034.

Hsu, J. C. 1996. *Multiple Comparisons: Theory and Methods*. Chapman and Hall.

Jensen, F. V. 1996. *An Introduction to Bayesian Networks*. UCL Press.

Kearns, M.; Mansour, Y.; and Ng, A. Y. 1999. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1324–1331. Menlo Park, Calif.: International Joint Conference on Artificial Intelligence, Inc.

Matejcik, F. J., and Nelson, B. L. 1995. Two-stage multiple comparisons with the best for computer simulation. *Operations Research* 43(4):633–640.

Murphy, K. P. 1999. Bayes net toolbox for Matlab. Available from `http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html`.

Ortiz, L. E., and Kaelbling, L. P. 2000. Adaptive importance sampling for estimation in structured domains. Under review.

Ortiz, L. E. 2000. Selecting approximately-optimal actions in complex structured domains. Technical Report CS-00-05, Computer Science Department, Brown University.

Shachter, R. D., and Peot, M. A. 1989. Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, 311–318.

Computing Global Strategies for Multi-Market Commodity Trading

Milos Hauskrecht, Luis Ortiz, Ioannis Tsochantaridis and Eli Upfal
Computer Science Department, Box 1910
Brown University
Providence, RI 02912
{*milos,leo,it,eli*}*@cs.brown.edu*

Abstract

The focus of this work is the computation of efficient strategies for commodity trading in a multi-market environment. In today's "global economy" commodities are often bought in one location and then sold (right away, or after some storage period) in different markets. Thus, a trading decision in one location must be based on expectations about future price curves in all other relevant markets, and on current and future storage and transportation costs. Investors try to compute a strategy that maximizes expected return, usually with some limitations on assumed risk.

With standard stochastic assumptions on commodity price fluctuations, computing an optimal strategy can be modeled as a Markov decision process (MDP). However, in general such a formulation does not lead to efficient algorithms. In this work we propose a model for representing the multi-market trading problem and show how to obtain efficient structured algorithms for computing optimal strategies for a number of commonly used trading objective functions (Expected NPV, Mean-Variance, and Value at Risk).

Introduction

Investment is the act of incurring immediate cost in the expectation of future reward. Investment options represent various tradeoffs between risk and expected profit. Investors try to maximize their expected return subject to the risk level that they are willing to assume. Modern economics theory models the uncertainty of future rewards as a stochastic process defining future price curves. The process is typically Markovian, thus investment decision can be modeled as a Markov decision process (MDP) (Bellman 1957; Howard 1960; Puterman 1994) where a state of the underlying process needs only to include the current investment portfolio and current prices. While the MDP gives a succinct formalization of the investment decision processes it does not necessarily imply efficient algorithms for computing optimal strategies. A challenging goal in this research area is to characterize special cases of the general investment paradigm that are interesting enough from the application point of view while simple enough to allow efficiently computable analytic solutions.

We focus in this paper on commodity trading. Past work has mainly dealt with single market trading problems (see (Dixit & Pindyck 1994; Hauskrecht, Pandurangan, & Upfal 1999) and the references there), where commodity is bought, stored and eventually sold at the same location. Here we address a more realistic scenario in today's "global economy", that of a multi-site trading problem where a commodity can be bought in one location, stored at a second location and eventually sold at a third market. Prices at different locations may be different, and they may have different future price curves. Transportation costs also vary in time. While there can be large gaps in spot prices in different locations, future prices are more correlated - the future price of the commodity at site X cannot be larger than the price at site Y plus the cost of transportation between Y and X. Trading in a "global economy" is significantly more complex, since a local trading decision must be based on expectations about future price curves in all other relevant markets, as well as transportation and storage costs.

Modeling the multi-site commodity trading as a Markov decision process leads to a large state space, and a large action space. Nevertheless, we show in this work that under several commonly used trading utility functions an optimal strategy can still be computed efficiently.

A standard assumption in mathematical economics is that commodity prices (e.g., oil and copper) are best modeled as a *mean reverting* stochastic process (Dixit & Pindyck 1994). In our case, prices in all locations follow the mean reverting process but with different set of parameters for different sites. To solve the trading problem we first consider the *expected net present value (ENPV)* objective function, where the goal is to maximize expected gain with no consideration to risk. Under this objective function the optimization problem becomes myopic and can be computed by considering only current and next step prices. This allows us to design global optimal portfolio allocation algorithms that are polynomial in the number of sites in each trading step.

Building on the myopic property of the ENPV objective function we extend the result to two commonly

used objective functions that combine ENPV maximization with limits on assumed risk at any one step. In the *Mean-Variance* function the goal is to maximize a weighted difference of the expected gain and the variance. The *Value at Risk* function maximizes expected gain subject to a (probabilistic) limit on the possibility of a large loss at any one step. Since both functions include a term that is linear in the variance of the process, the optimization problems in both cases lead to a constrained quadratic optimization problem. However, the computational complexities of the two problems are different. The mean-variance function has a particular structure that allows for polynomial time solution. The complexity of the optimization problem for the value at risk function varies, some special cases have polynomial time solutions. To improve the computational efficiency of both methods even further we present structure-based algorithms exploiting the special structure and regularities of the problem.

The Model

We consider investment problems with one type of commodity that is traded at n different sites. Once the commodidity is bought it can be either stored in each of the locations or transported between any two locations.

Price model

We assume that trading occurs at discrete time steps. To model commodity price fluctuations we adopt a discrete time version of the mean-reverting model (Dixit & Pindyck 1994):

$$p^{(t+1)} = \mu - e^{-\eta}(\mu - p^{(t)}) + \epsilon^{(t)}, \qquad (1)$$

where μ is the long term average price of the commodity i.e., a value to which the process reverts, η is the speed of reversion and $\epsilon^{(t)}$ is a sequence of independent random variables following normal distribution $N(0, \sigma_\epsilon)$.[1]

Commodity prices at all locations follow mean reverting processes, each with different parameters and with possible correlations between their random components ϵ's. Their combined fluctuations are fully described by a multivariate normal distribution $N(\mathbf{0}, \Sigma)$, with a zero mean vector and a covariance matrix Σ. We assume that price movements are independent of our trading activities. Also, there is no fee for trading and buy and sell prices are the same.[2]

There are natural capacity constraints on the number of commodity units we can transport (store) between the two locations at any time step. However, there are no constraints on buy and sell activities.

Valuation

Profit is measured by the standard *expected net present value (ENPV)* (see e.g. (Brealey & Myers 1991; Trigeorgis 1996)):

$$V^\pi(s) = E(\sum_{t=0}^{T} \gamma^t m^{(t)} | \pi, s) \qquad (2)$$

where s denotes an initial state, π is the trading strategy, $\gamma = \frac{1}{1+r}$ is a discount factor, with r denoting the interest rate (present cost of money), T is the decision horizon, and $m^{(t)}$ is the cash flow at time t. We focus primarily on problems with infinite horizon ($T \to \infty$).

Markov decision process formulation of the problem

A *Markov decision process (MDP)* (Bellman 1957; Howard 1960; Puterman 1994) describes a stochastic controlled process represented by a 4-tuple (S, A, T, R), where S is a set of process states; A is a set of actions; $T : S \times A \times S \to [0, 1]$ is a probabilistic transition model describing the dynamics of the modeled system; and $R : S \times A \times S \to \mathcal{R}$ models rewards assigned to transitions.

In the multi-site commodity trading problem the state of a process is determined by a price vector

$$\mathbf{p} = \{p_1, p_2, \cdots, p_i, \cdots, p_n, p_{11}, p_{12}, \cdots, p_{nn}\},$$

where the p_i's give the commodity price at location i, the $p_{i,j}$'s give the transportation price from i to j, and the $p_{i,i}$'s give the storage price at site i. Actions represent trading activities at a specific time step, and are defined as

$$\mathbf{a} = \{a_{11}, a_{12}, \cdots, a_{ij}, \cdots a_{nn}\},$$

where a_{ij} is the amount of commodity to be transported between i and j, or stored at location i if $j = i$. Thus, actions define allocations of commodity to different transportation (storage) edges. [3]

The transition model is defined by a set of mean-reverting price functions (Equation 1), one for each location. For example, the price movements for location i is

$$p_i' = \mu_i - e^{-\eta_i}(\mu_i - p_i) + \epsilon_i,$$

where p_i and p_i' is the current and next step price, η_i and μ_i are the parameters of the mean-reverting process and ϵ_i is the random component.

[1] We note that normally distributed random components of the price process may lead to negative prices. One way to deal with this issue is to use a geometric version of the mean reverting process, where the logarithm of the price follows the mean reverting model. However, the behavior of such a model is quite different, and price curves of the standard model are more realistic.

[2] In the more general setting (not considered here) prices can also fluctuate based on our demand and supply for the commodity or transportation service.

[3] It is easy to see that the number of units to be transported between different locations is sufficient to define all trading activities. Simply, the number of units to buy and sell at different locations can be obtained by comparing the number of units currently held and the number of units to be transported from that location in the next step.

Rewards represent partial profits from applying the strategy and are modeled in terms of *step-wise gains*. The gain for transporting one unit of commodity between location i and j is defined by

$$g_{ij}(\mathbf{p}) = -p_i - p_{ij} + \gamma p'_j,$$

where p_i is the current price of the commodity in location i, p_{ij} is the cost of transportation and p'_j is the price of the commodity in location j in the next step. The gain for an action $\mathbf{a}$ that allocates commodity to different transportation edges is the combination of partial gains

$$g_{\mathbf{a}}(\mathbf{p}) = g(\mathbf{p}).\mathbf{a} = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}(\mathbf{p})a_{ij}.$$

Using our model, a sequence of cash flows for any strategy can be expressed in terms of step-wise gains (rewards) rather than actual money inflow and outflow. Intuitively, we can replicate payoffs from any strategy by buying the commodity at the beginning of a decision step and selling it at the end of that step. Therefore, the expected NPV model from Equation 2 for a strategy π can be expressed in terms of gains as

$$V^{\pi}(\mathbf{p}) = \lim_{T\to\infty} E(\sum_{t=0}^{T} \gamma^t g^{(t)} | \pi, \mathbf{p}), \qquad (3)$$

where $g^{(t)}$ is the gain at time t. This is exactly the discounted, infinite-horizon criterion used commonly in MDPs (Puterman 1994). Thus, our multi-site investment problem for expected NPV model can be expressed and solved as a Markov decision problem.

The optimal trading strategy for the discounted, infinite horizon Markov decision problem is stationary (see (Bellman 1957; Puterman 1994)) and maps states of the process to actions. Therefore, the optimal strategy for our problem is $\pi^* : R^n \times R^{n^2} \to R^{n^2}$, mapping the current commodity and transportation prices to amounts of units to be allocated to different transportation/storage edges.

Solving the expected NPV problem

Using the MDP formulation, Equation 3 for the expected NPV model and a fixed policy π can be rewritten in Bellman's form (Bellman 1957) as

$$V^{\pi}(\mathbf{p}) = E(g_{\pi(\mathbf{p})}(\mathbf{p})) + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^{\pi}(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}', \qquad (4)$$

where $E(g_{\pi(\mathbf{p})}(\mathbf{p}))$ is the expected one-step gain for $\pi(\mathbf{p})$ and $f(\mathbf{p}'|\mathbf{p})$ is the conditional probability density function of the next step prices.

Myopic property

We see that $V^{\pi}(\mathbf{p})$ is hard to compute exactly. However, despite this difficulty the optimal strategy that maximizes ENPV can be computed efficiently. A key feature of our model is that prices change independently of our trading decisions (see Equation 4). Thus, the optimal policy is *myopic* (a greedy one-step policy is globally optimal) and can be easily computed (see (Hauskrecht, Pandurangan, & Upfal 1999)).

Theorem 1 *The optimal trading strategy for the expected NPV model is myopic.*

Proof The value of the optimal trading strategy is obtained from Equation 4 by maximizing over all possible actions

$$V^*(\mathbf{p}) = \max_{\mathbf{a}} \left[E(g_{\mathbf{a}}(\mathbf{p})) + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}' \right].$$

As the next step prices are independent of the action choice, the value can be rewritten as

$$V^*(\mathbf{p}) = \max_{\mathbf{a}} [E(g_{\mathbf{a}}(\mathbf{p}))] + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}'.$$

We see that in order to get the optimal solution for $\mathbf{a}$ it is sufficient to optimize $\mathbf{a}$ only with regard to $E(g_{\mathbf{a}}(\mathbf{p}))$. Thus the optimal strategy is myopic. □

The myopic property of the optimal investment strategy is critical for computing the solution for the commodity problem. The complete optimal investment strategy $\pi : R^n \times R^{n^2} \to R^{n^2}$ allocates the commodity units to different transportation edges for every price vector $\mathbf{p}$. As the number of possible prices and corresponding allocations is very large, it is not feasible to represent and store the optimal policy.

One way to avoid the computation of the complete policy is to compute individual price-specific allocations on-line. The on-line algorithm is invoked repeatedly in every step. In the general case, the on-line phase may be very time consuming as it may require to examine multiple price trajectories spanning multiple time steps. The myopic property of the decision process (Theorem 1) assures that we can obtain the optimal solution just by looking on what can happen in the next step. Simply, in order to decide the best allocation of investment for some price vector $\mathbf{p}$ it is sufficient to choose the allocation with the best one-step expected gain, and it is not necessary to consider more distant future and possible later price movements.

Optimal allocation

To find the optimal trading strategy for the expected NPV model it is sufficient to optimize expected one-step gains. Let $\mathbf{a}$ be some allocation of units to different transportation edges. The expected gain for $\mathbf{a}$ is

$$E(g_{\mathbf{a}}(\mathbf{p})) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} E(g_{ij}(\mathbf{p})).$$

To maximize the expectation we need to maximize the components of the sum. Assuming that C_{ij} is the constraint on the number of units we can transport between location i and j, the optimal allocation of a_{ij} is easy:

$$a^*_{ij} = \begin{cases} C_{ij} & \text{if } E(g_{ij}(\mathbf{p}) > 0; \\ 0 & \text{otherwise.} \end{cases}$$

Simply, we invest the limit to every edge with a positive expected gain.

Objective functions with one-step risk models

Once risk is taken into account, the above strategy of investing the limit on all edges with positive expected gains may not be optimal anymore.

Investment risk can be incorporated into the model in various ways. We focus here on objective functions that penalize or bound risk in any single step. In particular, we investigate:

- Mean-Variance model (Markowitz 1991; Alexander & Francis 1986; Bodie, Kane, & Marcus 1992) that explicitly relates expected one-step gain and the gain variance;
- Value at Risk (VaR) model (Jorion 1996) which maximizes the expected present value of the investment, but at the same time limits possible step losses.

The important property of both models is that their value function is time-decomposable and can be expressed in the form similar to the expected NPV model

$$V^*(\mathbf{p}) = \tag{5}$$
$$= \max_{\mathbf{a}} \left[h(g_{\mathbf{a}}(\mathbf{p})) + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}' \right]$$
$$= \max_{\mathbf{a}} [h(g_{\mathbf{a}}(\mathbf{p}))] + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}'.$$

Here, $h(g_{\mathbf{a}}(\mathbf{p}))$ is a function of a one-step gain (a random variable), not just its expectation. Different risk models use different forms of h. Note that the optimal policies must be myopic for this formalization.

Mean-Variance (MV) model

The mean-variance model (Markowitz 1991; Alexander & Francis 1986; Bodie, Kane, & Marcus 1992) quantifies the risk in terms of the gain volatility. The model is additive and combines the expected one-step gain and the gain volatility into a single objective function $h_{\mathbf{a}}(\mathbf{p})$:

$$h_{\mathbf{a}}(\mathbf{p}) = \alpha E(g_{\mathbf{a}}(\mathbf{p})) - \beta Var(g_{\mathbf{a}}(\mathbf{p})), \tag{6}$$

where $\alpha, \beta \geq 0$. Intuitively the function reflects the fact that investors like the mean to be large but dislike the variance. Parameters α, β quantify this relation. We note that this valuation corresponds to the quadratic utility function (Markowitz 1991).

Using the valuation function from equation 6, our goal is to find the allocation of commodity maximizing it. That is:

$$\pi^*(\mathbf{p}) = \arg\max_{\mathbf{a}} [\alpha E(g_{\mathbf{a}}(\mathbf{p})) - \beta Var(g_{\mathbf{a}}(\mathbf{p}))], \tag{7}$$

subject to constraints $C_{ij} \geq a_{ij} \geq 0$ for all a_{ij}. The variance of the gain for $\mathbf{a}$ is:

$$Var(g_{\mathbf{a}}(\mathbf{p})) = \mathbf{a}^T \mathbf{\Sigma}' \mathbf{a},$$

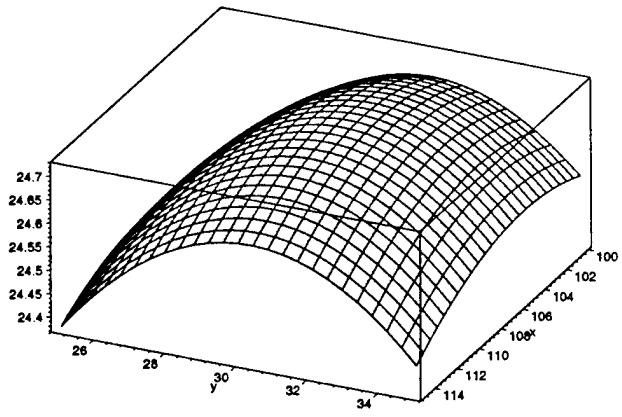


Figure 1: An example of a concave quadratic function for two dimensions.

where $\mathbf{\Sigma}'$ is the gain covariance matrix obtained from the price covariance matrix $\mathbf{\Sigma}$ as:

$$\mathbf{\Sigma}'_{(ij)(kl)} = Cov(g_{ij}(\mathbf{p}), g_{kl}(\mathbf{p})) = \gamma^2 Cov(\epsilon_j, \epsilon_l) = \gamma^2 \mathbf{\Sigma}_{jl}.$$

The allocation weights in $\mathbf{a}$ must be non-negative since there is no meaning in our model to negative investment.[4] Also, weights a_{ij} should have only integer values. However, to simplify the problem and its solution we approximate the integer problem by allowing continuous allocation weights.

Solution for the model Equation 7 defines a quadratic optimization problem with linear constraints. The important property of this problem is that the h function has a unique global optimum solution. We can observe this from the fact that the Hessian of our function is a constant negative definite matrix (equal to $-2\beta\mathbf{\Sigma}'$).[5] Therefore, the function is concave. Figure 1 illustrates the shape of the function for the 2-dimensional case. This special case of Quadratic Programming is known to have a polynomial time solution (Vavasis 1991).

Exploiting the structure Solving the optimization problem requires to optimize all n^2 possible allocation weights. We show that this optimization can be carried out more efficiently by taking advantage of the problem structure and by solving a sequence of optimizations of smaller complexity.

The idea of our solution is to exploit the regularities of the covariance matrix $\mathbf{\Sigma}'$ of one-step gains for all transportation edges, in particular the fact that random

[4] We note that in some of the problems in finance, similar to our problem (e.g. portfolio optimization), constraints on weights can be lifted. This is the case when short-selling of an asset or security is possible. In that case, negative weights in the portfolio will reflect a short position.

[5] Recall that the covariance matrix Σ' is symmetric, positive definite.

components of transportation links leading to the same location are fully correlated. Combining this property with the MV criterion makes it possible to find the optimal allocation incrementally. The idea of the approach is based on the following theorem.

Theorem 2 *Let* $\mathbf{a}^*$ *be the optimal allocation of commodity maximizing expected gains (returns) and penalizing risk (volatility). Let,* (i,j) *and* (k,j) *be two different transportation links ending in the same target location* j *such that* $-p_k - p_{kj} < -p_i - p_{ij}$ *holds. Then* $a^*_{kj} > 0$ *only if* $a^*_{ij} = C_{ij}$*, otherwise* $a^*_{kj} = 0$.

Proof Gains from transporting one unit of commodity from i to j and k to j are

$$g_{ij}(\mathbf{p}) = -p_i - p_{ij} + \gamma\left[\mu_j - e^{\eta_j}(p_j - \mu_j) + \epsilon_j\right]$$
$$g_{kj}(\mathbf{p}) = -p_k - p_{kj} + \gamma\left[\mu_j - e^{\eta_j}(p_j - \mu_j) + \epsilon_j\right]$$

As the two gains share the same stochastic component and their difference is always deterministic

$$g_{ij}(\mathbf{p}) - g_{kj}(\mathbf{p}) = -p_i - p_{ij} - [-p_k - p_{kj}].$$

Moreover their covariance terms in Σ are the same. Thus, if $p_k - p_{kj} > -p_i - p_{ij}$, there is no value in allocating the commodity to the transport link choice from k before we allocate the maximum, C_{ij}, to a_{ij}. Therefore if $a^*_{kj} > 0$, a^*_{ij} must be saturated ($a^*_{ij} = C_{ij}$). By similar argument, $a^*_{ij} < C_{ij}$ implies $a^*_{kj} = 0$. □

By using this result we can perform the allocation of commodity to different transportation edges incrementally by allocating commodity to edges according to their expected gains, i.e. edges with higher expected gains for the same target location are allocated first. This approach translates to a sequence of quadratic optimization problems with at most n variables.

The algorithm works as follows: the optimization starts by considering only transportation choices with the highest expected gains, one for each target location. We refer to these edges as *active edges*. The optimization procedure for the MV model is then applied to active edges. The solution gives an allocation of units to all active edges. During the optimization a transportation edge can reach its maximum capacity; we say that the edge becomes *saturated*. Once an edge is saturated it is removed and no longer considered as a choice. After the removal, the transportation edge with the next highest expected gain (and the same target location) becomes active and the optimization process continues with the next step. This is repeated until all edges have been exhausted or when none of the edges were saturated in the last step.

The optimization steps are not independent. In particular, every optimization step must take into consideration results of all previous (partial) allocations. The dependencies between the current and previous steps are summarized by:

- a vector of target allocations $\mathbf{s} = \{s_1, s_2, \cdots, s_n\}$, reflecting, for each target location, the number of units of commodity already allocated to edges incident to that location;
- adjusted capacity constraints $\{D_{11}, \cdots, D_{nn}\}$ representing the remaining capacity of all edges, i.e., the original capacity less the capacity already allocated in all previous solutions.

To find the optimal allocation of commodity to active set of edges we solve a quadratic program (with n variables). Let $\tilde{\mathbf{a}} = \{\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n\}$ denote a vector of allocations for the current set of active edges and $E(g_j(\mathbf{p}))$ be the expected gain for the active edge for target j. Then the optimization task corresponds to:

$$\max_{\tilde{\mathbf{a}}} \left\{ \alpha E(g_{\tilde{\mathbf{a}}}(\mathbf{p})) - \beta \left[(\mathbf{s} + \tilde{\mathbf{a}})^T \tilde{\mathbf{\Sigma}} (\mathbf{s} + \tilde{\mathbf{a}}) \right] \right\} \quad (8)$$

subject to constraints:

$$\tilde{D}_j \geq \tilde{a}_j \geq 0 \text{ for all } \tilde{a}_j,$$

where $E(g_{\tilde{\mathbf{a}}}(\mathbf{p})) = \sum_{j=1}^{n} E(g_j(\mathbf{p}))\tilde{a}_j$ is the expected gain for the portfolio of active edges and $\tilde{\mathbf{\Sigma}}$ is the *reduced gain covariance matrix*, an $n \times n$ matrix of the gain fluctuations for target locations ($\tilde{\mathbf{\Sigma}}_{kl} = \gamma^2 \mathbf{\Sigma}_{kl}$). $\tilde{D}_j$ denotes an adjusted capacity constraint corresponding to the transportation link for a target location j which is subject to optimization (is active).

During the computation process we keep track of the number of units allocated to each transportation link (staring from zero allocations at the beginning). That is, after every optimization step we apply the following update:

$$a^*_{ij} \leftarrow \begin{cases} a^*_{ij} + \tilde{a}^*_j & \text{if link } (i,j) \text{ is active;} \\ a^*_{ij} & \text{if not active.} \end{cases}$$

This allows us to recover the optimal allocation $\mathbf{a}^*$ at the end. In addition, we update $\mathbf{s}$ quantities and adjust dynamic capacity constraints:

$$D_{ij} \leftarrow \begin{cases} D_{ij} - \tilde{a}^*_j & \text{if link } (i,j) \text{ is active;} \\ D_{ij} & \text{if not active.} \end{cases}$$

Example Figures 2, 3 and Table 1 illustrate and compare the performance of strategies for different optimality criteria (the Value at Risk criterion is discussed in the next section) on a problem with 5 trading sites. Figure 2 shows the actual step-wise gains obtained for these criteria using a fixed 50-step trajectory of prices' fluctuations; each price following a mean-reverting process. Table 1 summarizes the results in Figure 2 by showing real gain averages and their standard deviations. Finally, Figure 3 compares expectations of gains under different strategies. We see that ENPV always leads to the maximum expected gain and it also achieves higher real gains on average. However, step-wise gains for ENPV are also subject to higher fluctuations. On the other hand, Mean-Variance (MV) criterion yields gains that fluctuate less, but at the same time lead to considerable lower expected gains and also real gains on average.

Besides the experiments shown here, we have tested the performance of the MV model for different combinations of parameters α and β. As expected, higher

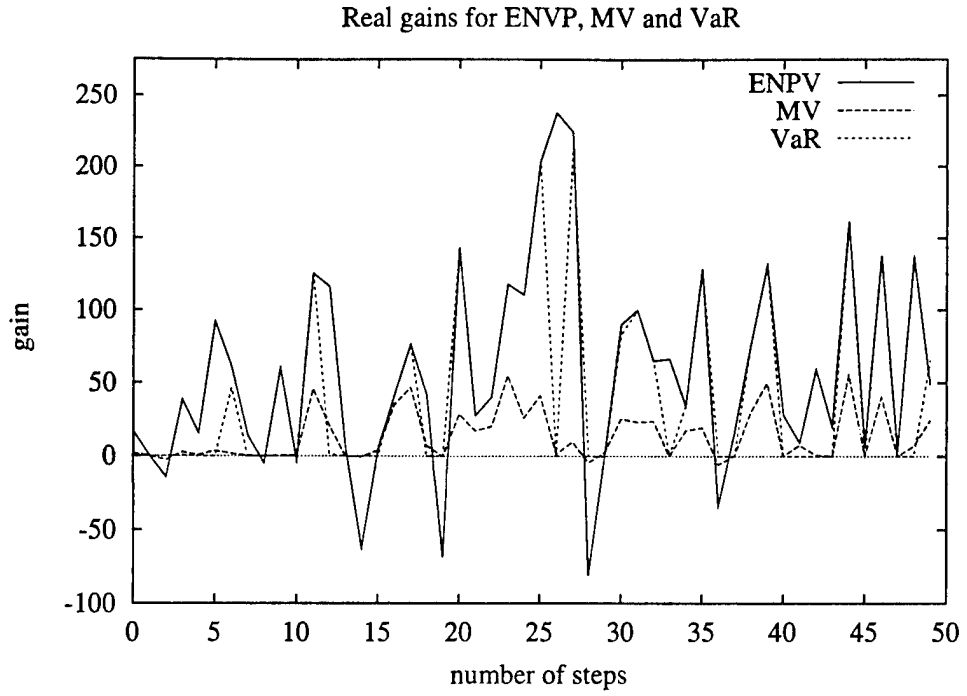


Figure 2: Comparison of three different optimization criteria: Expected NPV (ENPV), Mean-Variance (MV) and Value at Risk (VaR) on a problem with 5 trading sites and 50-step trajectory of prices' fluctuations. each following a mean-reverting process. For each step we plot the real gains for that step. The parameters of the MV model we use are $\alpha = 1$ and $\beta = 0.01$. We use $K = 0$ and $\delta = 0.0005$ for the VaR model.

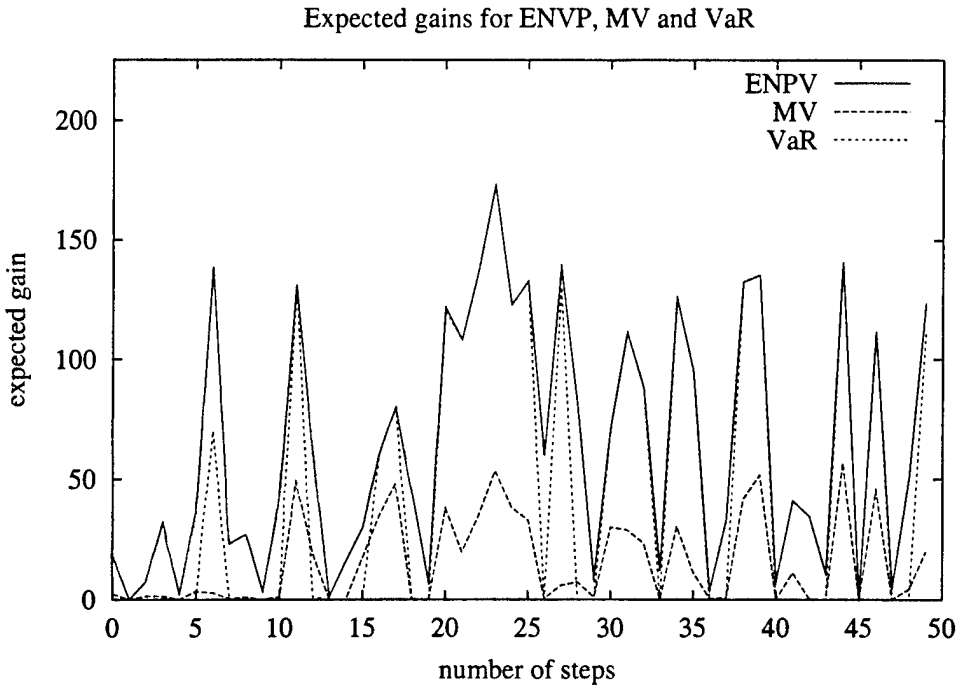


Figure 3: Comparison of expected gains for three different optimization criteria: Expected NPV (ENPV), Mean-Variance (MV) and Value at Risk (VaR) on a problem with 5 trading sites and 50 step long prices' trajectories.

	ENPV	MV	VaR
average real gains	57.31	13.69	42.47
standard deviation	70.64	17.49	60.54

Table 1: Average of the real gains and their standard deviation for ENPV, MV and VaR criteria and data from Figure 2.

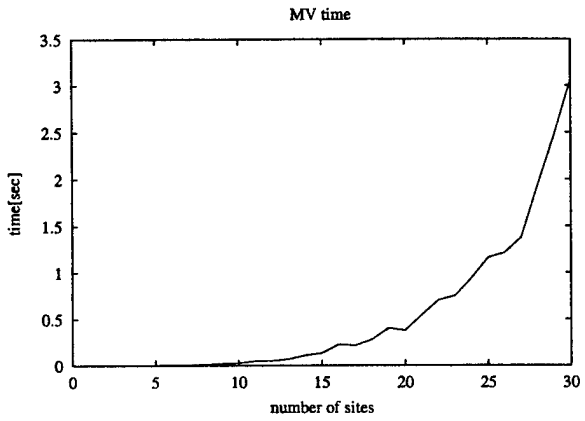


Figure 4: Average running times for markets with varying number of trading sites.

values of β lead to smaller average gains and smaller gain fluctuations. Simply, for higher values of β we penalize the variance more and thus we are likely to sacrifice the opportunity to capture higher gains.

One concern in applying our approach is that the optimization is carried on-line in every step, and thus it may lead to large reaction delays for larger problems (with many trading sites). To see the effect of the size of the multi-site market on the actual running time of the optimization problem we ran a set of experiments, varying the number of trading sites. For each market size we ran 1000 different parameter settings and averaged them. To solve the quadratic optimization problem we use ISML C/Math/Library implementation based on (Goldfarb & Idnani 1983). Figure 4 shows average running times, obtained for different market sizes. The running time (in seconds) increases moderately with the number of sites. In particular, the solution for 30 different trading sites, which is about the practical limit, can be obtained very quickly (in about 3 seconds on a SUN Ultra-10).

Value at Risk (VaR) model

Let K be a loss threshold and δ the maximum probability of losing K or more units. The value of K is called the *value at risk* for δ (see (Jorion 1996)).

This optimization problem has the form of Equation 5, where we maximize

$$h(g_{\mathbf{a}}(\mathbf{p})) = E(g_{\mathbf{a}}(\mathbf{p}))$$

subject to

$$C_{ij} \geq a_{ij} \geq 0 \text{ for all } a_{ij}$$
$$P(g_{\mathbf{a}}(\mathbf{p}) \leq -K) \leq \delta. \qquad (9)$$

This is a linear optimization problem with linear and quadratic constraints. Inequality 9 reduces to a quadratic constraint by the properties of the normal distribution. Let x be a normally distributed random variable with mean μ and variance σ^2. Let k be a value such that $P(x \leq \mu - k\sigma) \leq \delta$ holds. The value of k measures the distance from the mean in terms of a standard deviation σ, such that values smaller than $\mu - k\sigma$ occur with probability less than δ. In the case of a normal distribution, k is only a function of δ, and it is independent of μ and σ. Therefore, in order to limit the losses of more than K units with probability $1-\delta$, we set the value of k_δ such that it satisfies $\mu - k_\delta\sigma \geq -K$.

Therefore, the constraint 9 can be rewritten as

$$[E(g_{\mathbf{a}}(\mathbf{p})) + K]^2 - k_\delta^2 Var(g_{\mathbf{a}}(\mathbf{p})) \geq 0 \qquad (10)$$

which is quadratic in allocation weights $\mathbf{a}$. We can rewrite the constraint in terms of mean one-unit gains (vector $\boldsymbol{\mu}$) and covariances ($\boldsymbol{\Sigma}'$) as:

$$\mathbf{a}^T \left[\boldsymbol{\mu}\boldsymbol{\mu}^T - k_\delta^2 \boldsymbol{\Sigma}'\right] \mathbf{a} + 2K\boldsymbol{\mu}^T\mathbf{a} + K^2 \geq 0. \qquad (11)$$

Let $W = \left[\boldsymbol{\mu}\boldsymbol{\mu}^T - k_\delta^2 \boldsymbol{\Sigma}'\right]$ be the $n^2 \times n^2$ matrix defining the quadratic term. We note that if the matrix W is negative definite, the problem corresponds to the linear optimization over the convex space. Thus, it can be solved efficiently in polynomial time (Papadimitriou & Stieglitz 1998). However, when the matrix W is not negative definite we have a non-convex space over which we optimize. To solve this problem we can apply standard augmented Lagrangian techniques (see e.g. (Bertsekas 1995)).

Using structure to solve the VaR model The optimization of VaR criterion can be performed more efficiently by solving a sequence of optimization problems of smaller complexity. This is the same idea as used for the structured solution of the Mean-Variance model and Theorem 2 also applies to this case. Simply, the only sources of stochasticity are price fluctuations at different target locations. Thus, if two different transportation edges share the same target location, their stochastic component is the same and for the rational and risk averse investor the transportation choice with better expected gain should be chosen first. Therefore, under transportation capacity constraints, the global optimization can be carried incrementally by solving a sequence of optimization problems with n variables, instead of the optimization with n^2 variables. The globally optimal solution is then constructed from results of partial solutions.

To solve the problem, we optimize repeatedly the (reduced) problem with n variables:

$$\max_{\bar{\mathbf{a}}} E(g_{\bar{\mathbf{a}}}(\mathbf{p}))$$

subject to,

$$\tilde{D}_j \geq \tilde{a}_j \geq 0, \text{ for all } \tilde{a}_j;$$

$$[SM + E(g_{\tilde{\mathbf{a}}}(\mathbf{p})) + K]^2 - k_\delta^2 \left[(\mathbf{s}+\tilde{\mathbf{a}})^T \tilde{\mathbf{\Sigma}} (\mathbf{s}+\tilde{\mathbf{a}})\right] \geq 0.$$

The notation used and the basic algorithm applied are the same as in the Mean-Variance case. The only difference is that for the VaR criterion we have to add constant SM which represents the sum of expected gains for all previous solution. This quantity is updated dynamically after every step and is needed to assure that the non-linear constraint is not violated during the optimization process.

Example Figures 2, 3 and Table 1 compare the VaR criterion to ENPV and MV criteria on a problem with 5 sites. We note that the VaR choices do not penalize a large variance when expectation is also high. Instead, it only tries to limit the probability of losses. Thus the real gains obtained for the VaR model vary more than those of the MV model and also tend to achieve higher gains (both under expectation and on average). From the graphs we observe that in many instances the allocations for the VaR criterion replicate exactly the ENPV choices. However, in some instances, when a chance of losses exceeds the confidence threshold, the approach is more conservative and the allocation it chooses is different. For example, in 50 simulation steps in Figure 2 the VaR approach (with threshold gain 0) never lead to the negative gain, while there are seven different cases of negative gains for ENPV and two for the MV criterion.

Conclusion

We addressed the complex problem of finding optimal strategies for trading commodity in a multi-market environment. We investigated various objective criteria based on expected net present value (ENPV) and risk preferences of the investor. Different criteria can lead to optimization problems of different complexity. We showed that under the assumption of equal buy and sell prices, a number of criteria lead to the myopic portfolio optimization problem. This is very important as the computation of the optimal strategy needs to take into account only the current and next step prices and not all possible future price trajectories.

We analyzed and solved the problem for the expected NPV criterion and two commonly used risk-based criteria: Mean-Variance and Value at Risk models. We showed that in both risk-based models the optimization problem reduces to some form of the quadratic optimization problem. To further improve the efficiency of the solution we exploited the structure of the covariance matrix, in particular the fact that gains for the same target locations are fully correlated. This allowed us to reduce a large optimization problem for both risk-based criteria into a sequence of problems of smaller complexity. The empirical results obtained for the mean-variance and value at risk models support the feasibility of the solution and its practical applicability.

We note that our results and algorithms can be applied directly to any multi-site model in which the next-step price fluctuations are normally distributed, and thus not necessarily mean-reverting. The current model can be extended in a number ways. For example, interesting issues will arise if we refine the market models and extend them to include price spreads, trading (buy, sell) constraints, prices sensitive to supply and demands, etc. Another interesting direction is the investigation and application of more complex risk models, reflecting different preferences of an investor.

Acknowledgement

We wish to thank Oliver Frankel of Goldman Sachs for introducing us to this problem and for valuable technical discussions.

References

Alexander, G. J., and Francis, J. C. 1986. *Portfolio Analysis.* Prentice Hall.

Bellman, R. E. 1957. *Dynamic Programming.* Princeton: Princeton University Press.

Bertsekas, D. P. 1995. *Nonlinear programming.* Belmont, MA: Athena Scientific.

Bodie, Z.; Kane, A.; and Marcus, A. J. 1992. *Investments.* Richard D. Irwin.

Brealey, R. A., and Myers, S. C. 1991. *Principles of Corporate Finance.* McGraw-Hill.

Dixit, A. K., and Pindyck, R. S. 1994. *Investment under Uncertainty.* Princeton: Princeton University Press.

Goldfarb, D., and Idnani, A. 1983. A numerical stable dual method for solving strictly convex quadratic programs. *Mathematical Programming* 27:1–33.

Hauskrecht, M.; Pandurangan, G.; and Upfal, E. 1999. Computing near optimal strategies for stochastic investment planning problems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1310–1315.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes.* Cambridge: MIT Press.

Jorion, P. 1996. *Value at Risk: The New Benchmark for Controlling Market Risk.* Irwin Professional Pub.

Markowitz, H. M. 1991. *Portfolio Selection.* Cambridge: Basil Blackwell.

Papadimitriou, C., and Stieglitz, K. 1998. *Combinatorial Optimization: Algorithms and Complexity.* Dover.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York: Wiley.

Trigeorgis, L. 1996. *Real Options.* Cambridge: MIT Press.

Vavasis, S. A. 1991. *Nonlinear optimization: Complexity issues.* Oxford: Oxford University Press.