Multilinear and Integer Programming for Markov Decision Processes with Imprecise Probabilities

Ricardo Shirota Filho
Escola Politécnica,
Universidade de São Paulo, SP, Brazil
ricardo.shirota@poli.usp.br

Fabio Gagliardi Cozman
Escola Politécnica,
Universidade de São Paulo, SP, Brazil
fgcozman@usp.br

Felipe Werndl Trevizan
Departamento de Tecnología,
Universitat Pompeu Fabra, Barcelona, Spain
felipe.trevizan@upf.edu

Cassio Polpo de Campos
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo, SP, Brazil
cassiopc@usp.br

Leliane Nunes de Barros
Instituto de Matemática e Estatística
Universidade de São Paulo, SP, Brazil
leliane@ime.usp.br

Abstract

Markov Decision Processes (MDPs) are extensively used to encode sequences of decisions with probabilistic effects. Markov Decision Processes with Imprecise Probabilities (MDPIPs) encode sequences of decisions whose effects are modeled using sets of probability distributions. In this paper we examine the computation of Γ-maximin policies for MDPIPs using multilinear and integer programming. We discuss the application of our algorithms to "factored" models and to a recent proposal, Markov Decision Processes with Set-valued Transitions (MDPSTs), that unifies the fields of probabilistic and "nondeterministic" planning in artificial intelligence research.

Keywords. Markov Decision Processes with Imprecise Probabilities, Γ-maximin criterion, multilinear and integer programming.

1 Introduction

In this paper we are concerned with the computation of *policies*, or *plans*, that aim at maximizing reward over a possibly countably infinite sequence of *stages*. At each stage, our decision maker finds herself in a *state* and she must select an *action*. As a result of this decision, she gets a *reward*, and she moves to a new state. The process is then repeated. We focus on situations where transitions between states are modeled by *credal sets*; that is, by sets of probability distributions. Thus we focus on Markov Decision Processes with Imprecise Probabilities (MDPIPs), following a sizeable literature that has steadily grown in the last few decades. We review the basic concepts on MDPIPs in Section 2; we offer a relatively long review as we attempt to capture, in a somewhat organized form, various concepts dispersed in the literature.

There are several possible criteria that we might use to evaluate policies in an MDPIP. The term *optimal policy* is used in this paper in connection with Γ-maximin expected total discounted reward; that is, highest expected total discounted reward under the worst possible selection of probabilities.

We show how to reduce the generation of optimal policies for an MDPIP to *multilinear/integer programming* in Section 3. We also discuss in that section the practical reasons to pursue such a programming solution. We comment on the relationship between multilinear programming and "factored" models in Section 4. We then move, in Section 5, to a recently proposed special type of MDPIP that has particularly pleasant properties and important applications, the Markov Decision Process with Set-valued Transitions (MDPSTs).

2 Background

In this section we review basic facts about MDPs, MDPIPs, evaluation criteria, and algorithms.

2.1 MDPs

Markov Decision Processes (MDPs) are used in many fields to encode possibly infinite sequences of decisions under uncertainty. For historical review, basic technical development, and substantial reference to related literature, the reader may consult books by Puterman [29] and Bertsekas [5]. In this paper we consider MDPs that are described by:

- a countable set $\mathcal{T}$ of *stages*; a decision is made at each stage.
- a finite set $\mathcal{S}$ of *states.*
- a finite set of *actions* $\mathcal{A}$; the set of actions may be indexed by states, but we simplify notation here by assuming a single set of actions for all states.
- a conditional probability distribution P_t that specifies the probability of transition from state s to state r given action a at stage t. We assume that probabilities are stationary (do no depend on t) and write $P(r|s,a)$.
- a *reward* function R_t that indicates how much is gained (or lost, by using a negative value) when action a is selected in state s at stage t. We assume the reward function to be stationary and write $R(s,a)$.

We refer to the state obtained at stage t, in a particular realization of the process, as s_t; likewise, the action selected at stage t is referred to as a_t.

The history h_t of an MDP at stage t is the sequence of states and actions visited by the process, $[s_1, a_1, \ldots, a_{t-1}, s_t]$. The *Markov assumption* that is adopted for MDPs is that $P(s_t|h_{t-1}, a_t) = P(s_t|s_{t-1}, a_t)$; consequently:

$$\begin{aligned} P(h_t|s_1) &= P(s_t|s_{t-1}, a_{t-1})P(s_{t-1}|s_{t-2}, a_{t-2}) \\ &\quad \ldots \times P(s_3|s_2, a_2)P(s_2|s_1, a_1). \end{aligned} \tag{1}$$

A *decision rule* $d_t(s,t)$ indicates the action that is to be taken in state s at stage t. A *policy* π is a sequence of decision rules, one for each stage. A policy may be *deterministic* or *randomized*; that is, it may prescribe actions with certainty, or rather it may just prescribe a probability distribution over the actions. A policy may also be *history-dependent* or not; that is, it may depend on all states and actions visited in previous stages, or just on the current state. A policy that is not history-dependent is called *Markovian.* A Markovian policy induces a probability distribution over histories through Expression (1).

We also assume that an MDP with infinite horizon (that is, with infinite $\mathcal{T}$) may always stop with some probability. In fact, we assume that the process stops with geometric probability: the process stops at stage t with probability $(1-\gamma)\gamma^{t-1}$ (independently of all other aspects of the process). Then γ is called the *discount* factor of the MDP [29, p. 125].

2.2 MDPIPs

Additional realism and flexibility can be attached to MDPs by allowing imprecision and indeterminacy in the assessment of transition probabilities. A decision process with states, actions, stages and rewards as described before, but where a set of probability distributions is associated with each transition, has been called a *Markov Decision Process with Imprecise Probabilities* (MDPIP) by White III and Eldeib [44], a name we adopt in this paper. Satia and Lave Jr. use instead the name *MDP with Uncertain Transition Probabilities* [31], in what may be the first thorough analysis of this model in the literature; Harmanec uses the term *generalized MDP* to refer to MDPIPs [21].

MDPIPs can represent incomplete and ambiguous beliefs about transitions between states; conflicting assessments by a group of experts; and situations where one wishes to investigate the effect of perturbations in a “base” model. MDPIPs have also been investigated as representations for abstracted processes, where details about transition probabilities are replaced by an enveloping set of distributions [17, 20]. Similar models are encoded by the *controlled Markov set-chains* by Kurano et al [26, 24]. Slightly less related are the vector-valued MDPs by Wakuta [41]. Some of these efforts have also adopted *interval-valued rewards*; in this paper we focus on imprecision/indeterminacy only in transition probabilities.

Thus an MDPIP is composed of a set of stages $\mathcal{T}$, a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function R_t and sets of probability distributions, each containing transition probabilities P_t. We assume $\mathcal{T}$ to be the non-negative integers, $\mathcal{S}$ and $\mathcal{A}$ to be finite, and $\mathcal{A}$ to be constant for all states. We assume R_t to be a stationary function $R(s,a)$. We also assume stationarity for the sets $K(r|s,a)$ of probability distributions. Note, however, that now we have to distinguish two situations. First, the sets of transition probabilities may be identical across stages, while a history of the process may be associated with different draws within these sets (that is, probabilities are selected from sets that do not depend on t, but the selection depends on t). We might refer to these MDPIPs as *set-stationary.* Alternatively, it may be that each history h_t is associated with stationary probability distributions $P(s_t|s_{t-1}, a_{t-1})$ that themselves satisfy the Markov condition (and of course $P(s_t|s_{t-1}, a_{t-1}) \in K(s_t|s_{t-1}, a_{t-1})$). We might refer to the second MDPIPs as *elementwise-stationary* or simply *stationary.* In this paper we only deal with elementwise-stationary MDPIPs; in fact it does not seem that set-stationary MDPIPs have received any attention in the literature.

In the remainder of this paper we will use the following notation and terminology regarding sets of probability distributions. A set of probability distributions is called a *credal set* [27]. The credal set $K(X)$ contains distributions for variable X, and the conditional credal set $K(X|A)$ contains conditional distributions for variable X given event A. Conditioning is elementwise: $K(X|A)$ is obtained from $K(X)$ by conditioning every distribution in $K(X)$ on the event A. The notation $K(X|Y)$ represents a *set* of credal sets: there is a credal set $K(X|Y = y)$ for each nonempty event $\{Y = y\}$. A set of credal sets $K(X|Y)$ is *separately specified* if the joint credal set $K(X, Y)$ is such that, whenever $P(X|Y = y_1) \in K(X|Y = y_1)$, $P(X|Y = y_2) \in K(X|Y = y_2)$, then $P(X|Y = y_1)$ and $P(X|Y = y_2)$ are conditional distributions obtained from a single $P(X, Y)$ in $K(X, Y)$. That is, $K(X|Y)$ is separately specified if we can select conditional distributions independently from its sets, an assumption we make throughout for our credal sets. We loosely use $K(r|s, a)$ to indicate a separately specified collection of credal sets, for a given action a, where r and s refer to states.

Given a credal set $K(X)$, we can compute *lower* and *upper* probabilities respectively as $\underline{P}(A) = \inf_{P\in K} P(A)$ and $\overline{P}(A) = \sup_{P\in K} P(A)$. We can also compute *lower* and *upper* expectations for any bounded function $f(X)$ as $\underline{E}[f] = \inf_{P\in K} E[f]$ and $\overline{E}[f] = \sup_{P\in K} E[f]$, and likewise for conditional lower/upper probabilities/expectations. We assume all credal sets to be closed, so infima and suprema can be replaced by minima and maxima.

2.3 Evaluation criteria and algorithms

Given an MDP that starts at state s, we might evaluate a policy π by its expected reward:

$$V_\pi(s) = E_{s,\pi}\left[E_T\left[\sum_{t=1}^{T} R(s_t, a_t)\right]\right]; \qquad (2)$$

that is, the expectation of the expected reward assuming the process stops at stage T. Now if the process has a geometric probability of stopping at T, with parameter γ, we have [29, p. 126]:

$$V_{\pi,\gamma}(s) = E_{s,\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R(s_t, a_t)\right]. \qquad (3)$$

We refer to $V_{\pi,\gamma}(s)$ as the expected total discounted reward. There are other criteria to evaluate policies in MDPs; for example, the expected total reward $E_{s,\pi}[\sum_{t=1}^{\infty} R(s_t, a_t)]$, and the average reward $\lim_{T\to\infty}(1/T)E_{s,\pi}\left[\sum_{t=1}^{T} R(s_t, a_t)\right]$ [5, 29]. These criteria may be useful in specific problems but they are usually less realistic than Expression (2) and the associated discounted reward (3). We focus on the latter in this paper.

When we move to MDPIPs, we find that several criteria may be used to evaluate policies, even if we adopt total discounted reward. Three possible criteria are:

- Select the policy that yields the largest value of $\min V_\pi(s)$, where the minimum applies to all transition probabilities, subject to the fact that these probabilities must belong to given credal sets [4]. That is, the optimal policy produces the highest expected total discounted reward even when probabilities are most unfavorable. This is the Γ-maximin total discounted reward, where an optimal policy starting from state s must yield
$$\max_\pi \min_P V_{\pi,\gamma}(s),$$
where we append a subscript P in the minimization operator, to emphasize that it applies with respect to all transition probabilities that are imprecise/indeterminate.
- Select the policy that yields, when starting from state s,
$$\max_\pi \max_P V_{\pi,\gamma}(s).$$
That is, both decisions and probabilities can be selected so as to maximize expected total discounted reward. This criterion is referred to as Γ-maximax total discounted reward.
- Select any policy (or perhaps select all of those policies) that maximizes $V_{\pi,\gamma}(s)$ for at least one choice of transition probabilities. This is the criterion of E-admissibility [27].

Note that Γ-maximin and Γ-maximax create a complete order over policies, while E-admissibility is content to explore the partial order of policies induced by credal sets in any convenient way. To date, most authors have adopted the Γ-maximin criterion. An exception is Harmanec's algorithm [21] which employs *interval dominance* (Harmanec presents his algorithm as providing maximal policies, however [14, 38] argue that in fact is adopts interval dominance). Several other criteria can be found in the literature [14, 37, 38].

In this paper we focus on Γ-maximin total discounted reward; we refer to it as ΓETDR (for Expected Total Discounted Reward)[1]. The work of Satia and Lave

[1] It is not our goal to discuss here the adequacy of the Γ-maximin criterion; it is investigated in this paper because of its wide application in MDP problems. Other criteria will be investigated by the authors in the future. For discussions on the different criterions see [4, 25, 34, 32, 37, 42].

Jr. has derived several important results for this situation [31]. First, there exists a deterministic stationary policy that is optimal. Second, the optimal policy induces a value function that is the unique solution of

$$V^*(s) = \sup_a \inf_P \left(R(s,a) + \gamma \sum_r P(r|s,a) V^*(r) \right). \quad (4)$$

We can take maximum and minimum in this equation whenever the set of actions $\mathcal{A}$ is finite and the credal sets $K(r|s,a)$ have finitely many vertices. We assume this to be true in the remainder of this paper.

Expression (4) can be compactly written as $V^* = \mathbf{V}V^*$, by lumping the supremum, infimum, and summation into the operator $\mathbf{V}$. Whenever the transition probabilites are fixed (or are precisely specified) at some value P, we indicate it through the operator $\mathbf{V}_P$ (where the infimum is either suppressed or unnecessary). In fact, for an MDP with transition probabilities P, the optimal policy satisfies $V^* = \mathbf{V}_P V^*$, the *Bellman equation*.

2.4 Algorithms for MDPs and MDPIPs

Consider now algorithms that solve the Bellman equation. There are three "classic" algorithms for generating optimal policies in MDPs: value iteration, policy iteration, and reduction to linear programming [5, 29]. Most of the literature focuses on value or policy iteration. However, there are at least three reasons to pay attention to linear programming solutions to MDPs. First, a linear program produces an exact solution without the need to specify any stopping criteria (as needed for value and policy iteration). This property is useful in practice and particularly important while testing other algorithms. Second, several algorithms based on approximating the value function by lower dimensional functions are based on linear programming [19, 22, 33]. Third, and perhaps more importantly, linear programs seem to offer the only organized way to deal with problems where maximization of expected total discounted reward is subject to additional constraints on expected rewards [1, 29].

The linear programming algorithm for MDPs solves the equation $V^* = \mathbf{V}_P V^*$ for the precisely specified transition probabilities as follows [16]:

$$\begin{aligned} \min_{V^*} \quad & \sum_s V^*(s) & (5) \\ \text{s.t.} \quad & V^*(s) \geq R(s,a) + \gamma \sum_r P(r|s,a) V^*(r), \end{aligned}$$

where each pair (s, a) corresponds to a constraint.

Policy and value iteration have known counterparts for ΓETDR. Satia and Lave Jr. presented a policy iteration algorithm for ΓETDR. The results by Satia and Lave Jr., and by Denardo [15], produce a value iteration algorithm as indicated by White III and Eldeib [44]; the same algorithm was later derived in the special case of Bounded-parameter Markov Decision Processes (BMDPs) [17]. The value iteration algorithm starts with a candidate value function $V'_0(s)$ and iterates:

$$V'_{i+1} = \mathbf{V}V'_i \quad (6)$$

until $||V'_{i+1} - V'_i||$ is sufficiently small.[2] Convergence of this procedure is based on the fact that the operator $\mathbf{V}$ is a contraction mapping.[3]

3 A multilinear/integer solution for ΓETDR

Expression (5) describes the linear program for solving MDPs with precisely specified probabilities. It does not seem possible to produce a linear programming solution for ΓETDR; however, as we show in this section, it is possible to generate solutions using well known programming problems. We do not attempt to produce algorithms that surpass value/policy iteration in execution time; rather, our reasons to pursue a programming solution mirror the reasons why others have investigated linear programming for MDPs (summarized in Section 2.4). First, the results produced by multilinear and integer programming, and in particular the latter, depend on combinatorial properties of credal sets, and can be produced exactly; this is useful, for instance, while evaluating other algorithms that only promise ϵ-optimal policies. Second, several approximate algorithms for MDPs that can possibly be extended to MDPIPs depend on linear programming; we conjecture that these potential extensions to MDPIPs will depend on the results in this section. In fact, it seems that multilinear programming is unavoidable in factored models, as we discuss in Section 4. Third, solutions based on optimization seem to be the only way to handle constraints on expected rewards, a topic we wish to pursue in connection with planning (Section 5).

Our main result is, in essence, simple. We start from Expression (4), and note that its solution can be found by solving the following optimization problem:

$$\begin{aligned} \min_{V^*} \quad & \sum_s V^*(s) & (7) \\ \text{s.t.} \quad & V^*(s) \geq R(s,a) + \gamma \min_P \sum_r P(r|s,a) V^*(r). \end{aligned}$$

[2] The norm $||V|| = \max_s V(s)$ is typically used in the literature.

[3] A mapping $\mathbf{V} : U \to U$, where U is a complete normed linear space, is a contraction mapping iff $||\mathbf{V}u_1 - \mathbf{V}u_2|| \leq \gamma ||u_1 - u_2||$ for some $\gamma \in [0, 1)$.

This can be shown to be an instance of bilevel programming [8, 40]. Similar problems have been tackled before in connection with linear programming with uncertainty, with obvious application to ΓETDR [2, 3]. Current algorithms for bilevel programming are complex, and convergence guarantees are not as sharp as one would like. It would be interesting to reduce Program (7) to a form that were closer to existing, well studied optimization problems. We do this by reducing Program (7) to multilinear and then to integer programming.

The multilinear program we consider is:

$$\begin{aligned} \min_{V^*,P} \quad & \sum_s V^*(s) \\ \text{s.t.} \quad & V^*(s) \geq R(s,a) + \gamma \sum_r P(r|s,a)V^*(r). \end{aligned} \tag{8}$$

Denote by (V_R^*, P_R^*) a solution of Program (7) and by (V_G^*, P_G^*) a solution of Program (8). In order to use Program (8), we must prove that V_G^* and V_R^* are identical.

Theorem 1 $V_G^* = V_R^*$

Proof. Let Ω_R and Ω_G be the solution spaces for Programs (7) and (8) respectively. We prove that Ω_R is a subset of Ω_G. Then, we show that no solution in $\Omega_G \setminus \Omega_R$ can have better performance than one in Ω_R. We have:

$$\Omega_R = \{(V,P) : V \in \mathcal{V}, P = \arg\min_{P \in \mathcal{P}} \sum_r P(r|s,a)V(r)\},$$

$$\Omega_G = \{(V,P) : V \in \mathcal{V}, P \in \mathcal{P}\}.$$

Given that the solution space in the second case is the whole space $\mathcal{V} \times \mathcal{P}$, while in the first case P can only be in a subspace $\mathcal{V} \times \mathcal{P}_R$ of $\mathcal{V} \times \mathcal{P}$ (hence restricted), Program (8) produces a value function at least as low as Program (7). So, $V_G^* \leq V_R^*$, because $\Omega_G \supset \Omega_R$. Now suppose $V_G^* < V_R^*$. For a state $s \in \mathcal{S}$ we have $V_G^*(s) = R(s,a) + \gamma \sum_r P_G^*(r|s,a)V_G^*(r)$, with $P_G^*(r|s,a) \neq \arg\min_P \sum_r P(r|s,a)V(r)$. If we take $P'(r|s,a) = \arg\min_P \sum_r P(r|s,a)V(r)$, then $V'(s) = R(s,a) + \gamma \sum_r P'(r|s,a)V_G^*(r) < V_G^*(s)$ and V_G^* is not optimal. Since V_G^* *is* optimal (given that it considers the whole state space), then $V_G^* \not< V_R^*$. This implies that $V_G^* = V_R^*$. •

Apparently we have moved from a difficult problem (bilevel programming) to another difficult problem (multilinear programming). However, the significance of this result is that multilinear programming is a widely studied field, with close connections to geometric and linear programming [18, 23, 28, 35, 39]. Implementations can deal with hundreds of variables; in our tests we resort to Sherali and Adams' algorithm [35], a branch-and-bound scheme based on linear programming relaxations. Our implementation is an optimized version of this algorithm, that has been used to solve a variety of large and challenging multilinear programs [10, 11, 13, 12]. The examples presented later in this section were solved using this implementation.

An even more interesting result obtains if we assume that the vertices of credal sets $K(r|s,a)$ are known. Consider a list of vertices (each vertex is a distribution over $\mathcal{S}$) for a credal set $K(r|s,a)$, $\{p_1, \ldots, p_M\}$. Every distribution in this credal set can be expressed as a convex combination $\sum_{i=1}^M \alpha_i p_i$ where $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$. We can then write our goal as:

$$\begin{aligned} \min_{V^*,\alpha_{i,s,a}} \quad & \sum_s V^*(s) \\ \text{s.t.} \quad & V^*(s) \geq R(s,a) + \\ & \qquad \gamma \sum_r \sum_i \alpha_{i,s,a} p_i(r|s,a) V^*(r), \\ & \alpha_{i,s,a} \geq 0, \quad \sum_i \alpha_{i,s,a} = 1, \end{aligned} \tag{9}$$

where we explicitly indicate that $\alpha_{i,s,a}$ depends on (s,a).

We now use the fact that a multilinear program has a maximum at the vertices of the credal sets; thus we necessarily have $\alpha_{i,s,a} \in \{0,1\}$ at a solution. We then resort to the following transformation to produce an *integer* program out of the multilinear program (9), just assuming that we can bound V^* from above and below (such bounds can be produced quite generally using results by White III and Eldeib [44]). First, we replace $V^*(r) \in [l,u]$ by $l + (V^*(r) - l)$, and create a new variable $\beta_r = V^*(r) - l \in [0, u-l]$. Each $\alpha_{i,s,a} p_i(r|s,a) V^*(r)$ is thus replaced by $\alpha_{i,s,a} p_i(r|s,a) l + \alpha_{i,s,a} p_i(r|s,a) \beta_r$. Note that $\alpha_{i,s,a} p_i(r|s,a) l$ is easy to evaluate. As $\alpha_{i,s,a}$ can be restricted to 0 or 1, we take each term $\alpha_{i,s,a} p_i(r|s,a) \beta_r$ and replace $\alpha_{i,s,a} \beta_r$ by a new variable $\beta_{i,r,s,a}$. To ensure that this replacement does not change the original problem, we introduce linear restrictions:

$$0 \leq \beta_{i,r,s,a} \leq \beta_r,$$

$$\beta_{i,r,s,a} \leq \alpha_{i,s,a}(u-l),$$

$$\beta_r - (u-l) + \alpha_{i,s,a}(u-l) \leq \beta_{i,r,s,a}.$$

The first and second restrictions are obvious (limitations on β_r and $\alpha_{i,s,a}$. The last restriction imposes the following. When $\alpha_{i,s,a} = 1$, $\beta_r \leq \beta_{i,s,a}$. However, since from the first restriction $\beta_{i,s,a} \leq \beta_r$, then $\beta_{i,s,a} = \beta_r$, and the full $V^*(r)$ will be considered. If $\alpha_{i,s,a} = 0$, then $\beta_r - (u-l) \leq \beta_{i,r,s,a}$, but

$\beta_r - (u-l) < 0$ (since $\beta_r \leq (u-l)$), so $\beta_{i,r,s,a} = 0$, and this non-optimal pair state-action will not be considered.

We end up with the following integer program:

$$\begin{aligned}
\min_{V^*, \alpha_{i,s,a}} \quad & \sum_s V^*(s) && (10)\\
\text{s.t.} \quad & V^*(s) \geq R(s,a) + \\
& \quad \gamma \sum_r \sum_i [\alpha_{i,s,a} p_i(r|s,a) l + \\
& \qquad\qquad p_i(r|s,a)\beta_{i,r,s,a}] \\
& \alpha_{i,s,a} \geq 0, \quad \sum_i \alpha_{i,s,a} = 1 \\
& \beta_r = V^*(r) - l \\
& 0 \leq \beta_r \leq u - l \\
& 0 \leq \beta_{i,r,s,a} \leq \beta_r \\
& \beta_{i,r,s,a} \leq \alpha_{i,s,a}(u-l) \\
& \beta_r - (u-l) + \alpha_{i,s,a}(u-l) \leq \beta_{i,r,s,a}
\end{aligned}$$

We close this section with two examples of MDPIPs. We focus on multilinear programming solutions; later we will consider examples where integer programming is used.

3.1 A small MDPIP

This is a very simple, abstract example. Consider two states, s_1 and s_2. In each state, the decision maker can choose between two actions. In s_1 the transition probability for both actions are imprecisely specified, while transition probabilities in s_2 are precisely specified. Probabilities and rewards are presented in Table 1 (left). The transition probabilities are defined from the states in the first column (origin states) to the states on the first row under P (destination states). The solution given by multilinear programming leads to the optimal solution; the value function V^* is shown in Table 1 (right).

3.2 Planning airplane maintenance through MDPIPs

This example is based on a problem described by White [43, p. 171]:

> An airline classifies the condition of its planes into three categories, viz. excellent, good and poor. The annual running costs for each category are 0.25×10^6, 10^6 and 2×10^6 [monetary units] respectively. At the beginning of each year the airline has to decide whether or not to overhaul each plane individually. With no overhaul a plane in excellent condition has probabilities of 0.75 and 0.25 of its condition being excellent or good, respectively, at the beginning of the next year. A plane in good condition has probabilities of 0.67 and 0.33 of its condition being good or poor, respectively, at the beginning of the next year. A plane in poor condition will remain in a poor condition at the beginning of the next year. An overhaul costs 2×10^6 and takes no significant time to do. It restores a plane in any condition to an excellent condition with probability 0.75, and leaves it in its current condition with probability 0.25. The airline also has an option of scrapping a plane and replacing it with a new one at a cost of 5×10^6. Such a new plane will be in excellent condition initially. There is an annual discount factor of $\gamma = 0.5$.

We consider a variant of this problem where probabilities are specified as in Table 2 (left). Multilinear programming produces the value function in Table 2 (right).

4 Factored MDPs

The specification of transitions between states is particularly burdensome in large MDPs. One strategy that has been often employed is to encode transition probabilities in *factored* form; usually this means that transition probabilities are encoded by *Bayesian networks* [7]. Here the state space is defined by the configurations of variables $\{X_1, \ldots, X_n\}$. We denote by $X_{i,t}$ the ith variable at stage t. For each action a, we specify a bipartite directed acyclic graph containing $2n$ nodes denoted by X_i^+ and X_i^-; node X_i^- and X_i^+ represent respectively $X_{i,t-1}$ and $X_{i,t}$ for any $t > 0$. One layer of the graph contains nodes X_i^- for all i, and no edge between them. The other layer contains nodes X_i^+ for all i, and edges between them. Edges are allowed *from* nodes in the first layer *into* the second layer, and also between nodes in the second layer. We denote by $\text{pa}(X_i^+)$ the *parents* of X_i^+ in the graph. The graph is assumed endowed with the following Markov condition: a variable X_i^+ is conditionally independent of its nondescendants given its parents. This implies the following factorization of transition probabilities:

$$P(X_1^+, \ldots, X_n^+) = \prod_{i=1}^n P(X_i^+ | \text{pa}(X_i^+)). \qquad (11)$$

Now suppose that conditional probability distributions $P(X_i^+|\text{pa}(X_i^+))$, or a subset of them, are not known precisely, but rather up to inclusion in credal sets $K\big(X_i^+|\text{pa}(X_i^+)\big)$. We assume the Markov condition to operate over all combinations of distributions from these credal sets, thus producing a possibly large set of joint distributions, each one of them satisfying

$\mathcal{S}$	$\mathcal{A}$	P		$R(s,a)$
		s_1	s_2	
s_1	$a_{1,1}$	[0,0.5]	[0.5,1]	7
	$a_{1,2}$	[0,0.2]	[0.8,1]	3
s_2	$a_{2,1}$	0.3	0.7	-1
	$a_{2,2}$	0.6	0.4	9

$V^*(s_1)$	21.486474
$V^*(s_2)$	18.108099
$\sum_s V^*(s)$	39.594573

Table 1: Specification of simple MDPIP example (left), and value function V^* (right).

$\mathcal{S}$	$\mathcal{A}$	P			$R(s,a)$
		s_1	s_2	s_3	
s_1	$a_{1,1}$	[0.5,1]	[0,0.4]	[0,0.1]	-0.25×10^6
	$a_{1,2}$	1	0	0	-2×10^6
	$a_{1,3}$	1	0	0	-5×10^6
s_2	$a_{2,1}$	0	[0.67,1]	[0,0.33]	-10^6
	$a_{2,2}$	[0.75,1]	[0,0.25]	0	-2×10^6
	$a_{2,3}$	1	0	0	-5×10^6
s_3	$a_{3,1}$	0	0	1	-2×10^6
	$a_{3,2}$	[0,0.25]	[0.5,0.8]	[0,0.25]	-2×10^6
	$a_{3,3}$	1	0	0	-5×10^6

$V^*(s_1)$	-1265664.1604
$V^*(s_2)$	-2496240.6015
$V^*(s_3)$	-4000000.0
$\sum_s V^*(s)$	-7761904.7619

Table 2: Specification of MDPIP for plane maintenance (left), and value function V^* (right).

the factorization in Expression (11) — the resulting structure is a *credal network* for each action [9].

The main point of this section is to indicate that Expression (11) defines a multilinear product for the probabilities that appear in Program (8). Thus, the multilinear character of Program (8) is left unchanged: the computation of Γ-maximin policies is still a matter of multilinear programming. The development of algorithms that produce optimal policies and that exploit the factorization in Expression (11) is left for the future; this is a promising avenue of research as the most advanced algorithms for factored MDPs do use all available structure encoded in the factorization [19, 22].

5 MDPSTs

In this section we explore the properties of a class of MDPIPs that have an important application in the field of artificial intelligence planning. Roughly speaking, planning in artificial intelligence focuses on sequential decision making problems that are specified using high-level languages. There are many variants of AI planning, depending on the properties of the specification language; for example, we have *deterministic* planning, where actions have deterministic effects; *probabilistic* planning, where actions have probabilistic effects; and *nondeterministic* planning, where an action may cause a transition to a set of states without any clue at to what state will be moved into [30]. The latter name is somewhat unfortunate as "nondeterminism" is an overloaded term, but it is the usual terminology in the field. Typically deterministic and nondeterministic planning are tackled by search through state spaces, while probabilistic planning is tackled by generation of equivalent MDPs.

There has been considerable effort in the field of AI planning to develop general algorithms that can be instantiated for different types of planning problems [6]. However, until recently no model considered actions with simultaneously "probabilistic" and "nondeterministic" effects. In response to this situation, Trevizan et al. have proposed a jointly probabilistic/nondeterministic framework, based on MDPIPs [36]. Their proposal is based on a class of MDPIPs, called Markov Decision Processes with Set-valued Transitions (MDPSTs), defined as follows.

An MDPST is composed by a set of stages $\mathcal{T}$, a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function R, a state transition function $F(s,a)$ mapping states s and actions $a \in \mathcal{A}$ into reachable sets of $\mathcal{S}$, i.e., into nonempty subsets of $\mathcal{S}$, and a set of mass assignments $m(k|s,a)$ for all s, $a \in \mathcal{A}$, and $k \in F(s,a)$. Here we also assume $\mathcal{T}$ to be the non-negative integers, $\mathcal{S}$ and $\mathcal{A}$ to be finite, $\mathcal{A}$ to be constant for all states, and $R(s,a)$ to be a stationary function. The state transition function $F(s,a)$ and mass assignments $m(k|s,a)$ are also stationary. MDPSTs satisfy a simplified version of Expression (4) [36]:

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{k \in F(s,a)} m(k|s,a) \min_{r \in k} V^*(r) \right). \tag{12}$$

MDPSTs form a strict subset of MDPIPs [36]; thus Programs (8) or (10) can be used to solve MDPSTs. These solutions require an enumeration on mass assignments $m(k|s,a)$. However we can produce simpler programs if we study Expression (12) carefully.

Given any action $a \in \mathcal{A}$, we can collect all feasible $k \in F(s,a)$, and define a binary vector $I(s,a)$ with as many elements as sets of states in $F(s,a)$, such that $I_i(s,a) \in \{0,1\}$ for $i \in \{1,\ldots,N\}$, and $\sum_i I_i(s,a) = 1$. Because each $I_i(s,a)$ can only be equal to 0 or 1, and their sum is equal to one, only an unique $I_i(s,a)$ can be equal to one at a time. We now write Expression (12) as:

$$\begin{aligned} V^*(s) \quad = \quad & \max_{a \in \mathcal{A}} R(s,a) + \\ & \gamma \sum_{k \in F(s,a)} m(k|s,a) \sum_{i=1}^{k} I_i(s,a) V^*(r_i). \end{aligned} \tag{13}$$

We now transform each product $I_i(s,a)V^*(r_i)$ into a new variable, following the procedure outlined in Section 3. We first replace $V^*(r_i)$ by $l + (V^*(r_i) - l)$, where $V^*(r_i) \in [l,u]$; we then define $\beta_i = V^*(r_i) - l$, with $\beta_i \in [0, u-l]$. We define a variable $\beta_{i,s,a} = I_i(s,a)\beta_i$, and add the necessary constraints to the optimization problem. The final integer program is very similar to the Program (10):

$$\begin{aligned} \min_{V^*,I} \quad & \sum_s V^*(s) \\ \text{s.t.} \quad & V^*(s) \geq R(s,a) + \\ & \quad \gamma \sum_k \sum_i [I_i(s,a) m(k|s,a) l + \\ & \qquad m(k|s,a) \beta_{i,s,a}] \\ & I_i(s,a) \geq 0, \sum_i I_i(s,a) = \\ & \beta_i = V^*(r_i) - l \\ & 0 \leq \beta_i \leq u - l \\ & 0 \leq \beta_{i,s,a} \leq \beta_i \\ & \beta_{i,s,a} \leq I_i(s,a)(u-l) \\ & \beta_i - (u-l) + I_i(s,a)(u-l) \leq \beta_{i,s,a}. \end{aligned} \tag{14}$$

This is a very useful transformation, once integer programming is much simpler than multilevel programming. There are many powerful integer program solvers that guarantee global optimal solutions, where multilevel program solvers only achieve global optimals in certain specific cases.

5.1 A small MDPST

Consider 3 states, s_1, s_2 and s_3. At state s_i, there are actions $a_{i,1}$ and $a_{i,2}$. All actions define probabilistic transitions from one state to itself or to the set composed by the other 2 states, however with different assignments of rewards and transition probabilities. The values assigned to each state and action can be found in Table 3. The optimal solution was obtained by solving an integer program.

5.2 Probabilistic/nondeterministic planning of airplane maintenance

Consider the example of airplane maintenance in Section 3. Suppose that transition probabilities follow Table 4 (left); a transition that "fills" more than a column is a nondeterministic one. The optimal solution obtained can be seen in Table 4 (right).

6 Conclusion

We have reviewed the basic theory of MDPIPs under the criterion of Γ-maximin expected total discounted reward, and we have shown how to produce policies using multilinear and integer programming. This type of solution may be useful to handle problems with further constraints on expected rewards, and to deal with factored models and factored approximations. We plan to continue the present work by exactly addressing such constraints and factorizations.

We have then looked into the recently proposed MDPSTs. We have briefly reviewed the application of these processes as a unifying language for "probabilistic" and "nondeterministic" planning, and then showed how these processes nicely lead to integer programming solutions. As indicated previously, one of the reasons to investigate a programming solution for MDPIPs is the promise it holds for treating problems with constraints on policy. For instance, it may be required that a policy, besides maximizing minimum expected total discounted reward, also guarantees the probability of some set of states to be higher than some value (in practice: maximization of profit for a company, subject to the probability that a client is left unattended being smaller than a given value). Markov decision processes subject to such constraints are called *constrained MDPs* [1, 29], and the main method of solution there is linear programming. We conjecture that constrained MDPIPs will require solutions based on multilinear/integer programming. This will be even more important in the context of MDPSTs, because "nondeterministic" planning is usually associated with contraints on policies.

$\mathcal{S}$	$\mathcal{A}$	P		$R(s,a)$
		s_i	$\mathcal{S} \setminus \{s_i\}$	
s_1	$a_{1,1}$	0.8	0.2	5
	$a_{1,2}$	0.1	0.9	-1
s_2	$a_{2,1}$	0.8	0.2	4
	$a_{2,2}$	0.3	0.7	7
s_3	$a_{3,1}$	0.7	0.3	3
	$a_{3,2}$	0.25	0.75	9

$V^*(s_1)$	17.670251
$V^*(s_2)$	19.820789
$V^*(s_3)$	22.153796
$\sum_s V^*(s)$	59.644836

Table 3: Specification of small MDPST (left), and value function V^* (right).

$\mathcal{S}$	$\mathcal{A}$	P			$R(s,a)$
		s_1	s_2	s_3	
s_1	$a_{1,1}$	0.5	0.5		-0.25×10^6
	$a_{1,2}$	1	0	0	-2×10^6
	$a_{1,3}$	1	0	0	-5×10^6
s_2	$a_{2,1}$	0	1		-10^6
	$a_{2,2}$	0.75	0.25	0	-2×10^6
	$a_{2,3}$	1	0	0	-5×10^6
s_3	$a_{3,1}$	0	0	1	-2×10^6
	$a_{3,2}$	0.8		0.2	-2×10^6
	$a_{3,3}$	1	0	0	-5×10^6

$V^*(s_1)$	-1666666.6666
$V^*(s_2)$	-3000000.0
$V^*(s_3)$	-4000000.0
$\sum_s V^*(s)$	-8666666.666

Table 4: Specification of MDPST for plane maintenance (left), and value function V^* (right).

Acknowledgements

This work has been supported by FAPESP grant 2004/09568-0; the first author is supported by FAPESP grant 05/58090-9; the second author is partially supported by CNPq grant 3000183/98-4; the third author is supported by CAPES grant BEX 4248-05-8; the fifth author is partially supported by CNPq grant 308530/03-9.

Tests were run in a multilinear programming solver that calls AMPL and CPLEX in its internal loops.

References

[1] E. Altman. *Constrained Markov decision processes.* Chapman & Hall, Boca Raton, Florida, 1999.

[2] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90(2):263–272, 2001.

[3] A. Ben-Tal and A. Nemirovski. Robust solutions of undertain linear programs. *Operations Research Letters*, 25(1):1–13, 1993.

[4] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, 1985.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control (Vol. 1, 2).* Athena Scientific, Belmont, Massachusetts, 1995.

[6] B. Bonet and H. Geffner. Learning Depth-First Search: A unified approach to heuristic search in deterministic and non-deterministic settings, and its application to MDPs. In *Proc. of the 16th ICAPS*, 2006.

[7] C. Boutilier, S. Hanks, and T. Dean. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[8] B. Colson, P. Marcotte and G. Savard. Bilevel programming: A survey. *Quaterly Journal of Operations Research*, 3(2):87–107, 2005.

[9] F. G. Cozman. Graphical models for imprecise probabilities. *International Journal of Approximate Reasoning*, 39(2-3):167–184, 2005.

[10] F. G. Cozman, C. P. de Campos, J. S. Ide, and J. C. F. da Rocha. Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In *Uncertainty in Artificial Intelligence*, pages 104–111. AUAI Press, 2004.

[11] C. P. de Campos and F. G. Cozman. Inference in credal networs using multilinear programming. In *Second Starting AI Researchers' Symposium (STAIRS)*, pages 50–61, Amsterdam, IOS Press, 2004.

[12] C. P. de Campos and F. G. Cozman. Belief updating and learning in semi-qualitative probabilistic networks. In *Uncertainty in Artificial Intelligence*, pages 153–160, Edinburgh, United Kingdom, 2005.

[13] C. P. de Campos and F. G. Cozman. Computing lower and upper expectations under epistemic independence. In *Fourth International Symposium on*

Imprecise Probabilities and Their Applications, pages 78–87, Dulles, Virginia, 2005.

[14] G. de Cooman and M. C. M. Troffaes. Dynamic programming for deterministic discrete-time systems with uncertain gain. *International Journal Approximate Reasoning*, 39(2-3):257–278, 2005.

[15] E. V. Denardo. Contraction mappings in the theory underlying dynamic programming. *SIAM Review*, 9(2):165–177, 1967.

[16] F. D'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.

[17] R. Givan, S. M. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1-2):71–109, 2000.

[18] W. Gochet and Y. Smeers. A branch-and-bound method for reversed geometric programming. *Operations Research*, 27(5):983–996, September/October 1979.

[19] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[20] V. Ha and P. Haddawy. Theoretical foundations for abstraction-based probabilistic planning. In *Uncertainty in Artificial Intelligence*, pages 291–298, San Francisco, California, United States, 1996. Morgan Kaufmann.

[21] D. Harmanec. Generalizing Markov decision processes to imprecise probabilities. *Journal of Statistical Planning and Inference*, 105:199–213, 2002.

[22] M. Hauskrecht and B. Kveton. Linear program approximations for factored continuous-state Markov decision processes. In *Advances in Neural Information Processing Systems 16*, pages 895–902. 2004.

[23] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, 1995.

[24] M.Hosaka, J. Nakagami, and M. Kurano. Controlled Markov set-chains with set-valued rewards – the average case. *International Transactions in Operations Researach*, 9:113–123, 2002.

[25] D. Kikuti, F. G. Cozman, and C. P. de Campos. Partially ordered preferences in decision trees: computing strategies with imprecision in probabilities. In *IJCAI Workshop on Advances in Preference Handling*, pages 118–123, Edinburgh, United Kingdom, 2005.

[26] M. Kurano, J. Song, M. Hosaka, and Y. Huang. Controlled Markov set-chains with discounting. *Journal Applied Probability*, 35:293–302, 1998.

[27] I. Levi. *The Enterprise of Knowledge*. MIT Press, Cambridge, Massachusetts, 1980.

[28] C.D. Maranas and C.A. Floudas. Global optimization in generalized geometric programming. *Computers and Chemical Engineering*, 21(4):351–370, 1997.

[29] M. L. Puterman. *Markov Decision Processes*. John Wiley and Sons, New York, 1994.

[30] S. J. Russell and P. Norvig. *Artificial Intelligence: a Modern Approach*. Prentice Hall, New Jersey, 1995.

[31] J. K. Satia and R. E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21:728–740, 1970.

[32] M. Schervish, T. Seidenfeld, J. Kadane, and I. Levi. Extensions of expected utility theory and some limitations of pairwise comparisons. In *Third International Symposium on Imprecise Probabilities and their Applications*, pages 496–510. Carleton Scientific, 2003.

[33] P. J. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

[34] T. Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: γ-maximin versus e-admissibility. *Synthese*, 140(1-2), 2004.

[35] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publishers, 1999.

[36] F. W. Trevizan, F. G. Cozman, and L. N. de Barros. Planning under risk and Knightian uncertainty. In *20th IJCAI*, pages 2023–2028, 2007.

[37] M. C. M. Troffaes. Decision making with imprecise probabilities: A short review. *SIPTA Newsletter*, pages 4–7, 2004.

[38] M. C. M. Troffaes. Learning and optimal control of imprecise Markov decision processes by dynamic programming using the imprecise Dirichlet model. pages 141–148, Berlin, 2004. Springer.

[39] H. Tuy. *Convex Analysis and Global Optimization*, volume 22 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1998.

[40] L. N. Vicente and P. H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306, 1994.

[41] K. Wakuta. Vector-valued Markov decision processes and the system of linear inequalities. *Stochastic Processes and their Applications*, 56:159–169, 1995.

[42] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[43] D. J. White. *Markov Decision Processes*. John Wiley and Sons, 1993.

[44] C. C. White III and H. K. El-Deib. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, July-August 1994.