Mixed Probabilistic and Nondeterministic Factored Planning through Markov Decision Processes with Set-Valued Transitions

Felipe W. Trevizan
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, USA
fwt@cs.cmu.edu

Fábio G. Cozman
Escola Politécnica
Universidade de São Paulo
Av. Prof. Mello Moraes, 2231
São Paulo, SP, Brazil
fgcozman@usp.br

Leliane N. de Barros
Instituto de Matemática e Estatística
Universidade de São Paulo
Rua do Matão, 1010
São Paulo, SP, Brazil
leliane@ime.usp.br

Abstract

This paper focuses on factored planning problems with probabilistic and nondeterministic elements. We first show that problems expressed in the nondeterministic extensions of PPDDL used in the 5th planning competition, yield Markov Decision Processes with Set-Valued Transitions (MDPSTs). We present a generalization of the language that still yields MDPSTs, and examine the solution of these MDPSTs using real time dynamic programming.

Introduction

Although the benefits of recent techniques for planning under uncertainty seems ”highly contentious”, the research in this field has brought promissing results in a short time (compared to the time taken to reach the state-of-the-art of classical planning). We believe that after establishing a common foundation on reasoning about planning under uncertainty, the community will be ready to cope with real planning applications. In particular, we claim that the two different forms of uncertainties, *probabilistic* and *nondeterministic*, should be unified in a common framework.

In this paper we present an analysis of planning problems that are described in factored form and that contain *probabilistic* and *nondeterministic* actions. As usual in the literature on planning, a “probabilistic” action is represented by a probability distribution over states, while a “nondeterministic” action is just specified through a subset of states — any one of these states can be selected by some unspecified procedure. We are interested in minimax policies; that is, in policies that minimize the maximum expected cost. The minimax character of the solutions is needed to handle the presence of nondeterminism. Depending on the situation, the focus may be on the minimax expected discounted cost for infinite horizon, or on the minimax expected cost to reach goal states with indefinite horizon. The experiments described in the paper focus on the latter case (that is, goal states and indefinite horizon) because this is the situation commonly found in the literature, and particularly in the Planning Competitions held so far.

Our analysis focuses on planning problems that can be expressed in existing extensions of the PDDL language. We adopt PPDDL 1.0 (Younes and Littman 2004; Younes et al. 2005), a probabilistic extension of PDDL 2.1 (Fox and Long 2003), plus the extensions made by Bonet and Givan (2005) in order to handle non-determinism. To simplify the text, we refer to the resulting language as *Planning Definition Language version 1.0*, or simply as PDL_1. Even though other variants and extensions of PDDL might have worked as our starting point, PDL_1 is a relatively mature language for dealing with uncertainty that has proven its worth in several Planning Competitions.

One interesting (and perhaps not widely known) fact about PDL_1 is that it allows probabilistic actions and nondeterministic actions in the same domain, as discussed in the next section. In fact it does not seem that PDL_1 has a complete semantics at this point, in the sense that every allowed domain description can be put in correspondence with known mathematical structures.

Our first result is simple to state: Every planning problem expressed in PDL_1 defines a *Markov Decision Process with Set-Valued Transitions (MDPSTs)*, as discussed in the next section. That is, MDPSTs can be taken as the underlying semantics of PDL_1 (Proposition 1).

We then ask: What happens if we allow probabilistic and nondeterministic features to be used *together* in the description of the effects of the *same* action? Our second result is Theorem 1, which shows that any planning problem that mixes probabilistic and nondeterministic effects, with a restriction on their order, yields an MDPST. We call the resulting description language PDL_2.

The consequences of Proposition 1 and Theorem 1 are nontrivial. We use results from the theory of MDPSTs to derive a special version of the Bellman equation with excellent computational properties. We discuss the importance of these facts for value/policy iteration and for real time dynamic programming (RTDP) approaches to policy construction, and present empirical evaluation of real time dynamic programming in standard problems in the field.

The results in this paper can be read from two different perspectives. First, the results are statements about the PDL_x languages: they show that an elegant semantics for PDL_1 can be built with MDPSTs, and they indicate how far one can go in extending the syntax before one finds a computational cliff. Second, the results clarify some of the prospects for merging probabilistic and nondeterministic planning — they show that MDPSTs are indeed a valid foundation even for factored models, and they suggest how far one can go with MDPSTs before one switches to more complex mathematical structures.

PDL_1 and MDPSTs

As described by Bonet and Givan (2005), PDL_1 is a relatively simple language syntactically. Every planning problem is expressed in two parts: the `domain` contains directives, constants, and descriptions of actions; the `problem` basically contains a description of the initial state and the desired goal. We have no comment on the syntax and semantics of `problem`s, so the reader may consult the paper by Bonet and Givan (2005) for more information on that. We wish to focus on the syntax and semantics of `domain`s, so we present the relevant pieces of the syntax here. The basic BNF for `domain`s is:

```
<domain> ::= (define (domain <NAME>)
    (:requirements :adl)
     [<types>][<constants>][<predicates>]
      <action>*)
<action> ::= (:action <NAME>
      [<param>] [<prec>] [<effect>])
<prec> ::= (:precondition <p-formula>)
<effect> ::= (:effect {<nd-eff>|<det-eff>})
<nd-eff> ::= <prob>|<one-of>
<prob> ::= (probabilistic <p-eff>+)
<p-eff> ::= <RATIONAL> <det-eff>
<one-of> ::= (oneof <det-eff>+),
```

where: `<types>`, `<constants>`, `<predicates>` and `<param>` refer to lists of names or logical variables (possibly typed); `<RATIONAL>` denotes a rational number; `<p-formula>` is a formula containing either atoms, or conjunction of `p-formula`s, or universal quantification over `p-formula`s, or inequality of two given names as `(not (= <NAME> <NAME>))`; and a `<det-eff>` is a formula containing either atoms, or negation of atoms, or conjunction of `det-eff`s, or universal quantification over `det-eff`s, or the *conditional* operator `when`. This conditional operator has syntax
`(when <p-formula> <simple-eff>)`,
where `simple-eff` is a formula containing either atoms, or negations of atoms, or conjunction of `det-eff`s, or universal quantification over `simple-eff`s.

For the purposes of this paper the important point is that any action in PDL_1 may contain *either* a probabilistic effect *or* a nondeterministic effect. For instance, a domain may contain two actions, one with probabilistic effects, and the other with nondeterministic effects. What is not allowed in PDL_1 is the mixture of probabilistic and nondeterministic effects *in the same action*.[1]

[1]All planning problems written in PDL_1 so far for relevant tracks of Planning Competitions are either entirely probabilistic or entirely nondeterministic. Perhaps the intention of the creators of PDL_1 was to deal only with such cases. The flexibility of PDL_1 may be a small oversight on the part of the creators of PDL_1; here we wish to exploit this feature of PDL_1 positively rather than take it as a flaw.

What is the semantics of complete PDL_1? It is not hard to imagine what the semantics should be: (1) the semantics of a probabilistic action is inherited from PPDDL 1.0 (that is, a probabilistic action is to be understood as a probabilistic transition given by a Bayesian network (Younes and Littman 2004; Younes et al. 2005)); (2) the semantics of a nondeterministic action should be given by a direct reading of the `oneof` element (where a nondeterministic choice is made and one of the effects listed in the scope of the `oneof` element is selected and pursued). We call these conventions the *standard semantics* of PDL_1; we believe there is nothing controversial about it.

Note that the standard semantics of PDL_1 takes us beyond Markov Decision Processes (MDPs) given the presence of nondeterminism. Thus the interesting question is, What is the right mathematics in which to embed the standard semantics of PDL_1? Besides, what can be derived concerning existence and stationarity of optimal policies, and algorithms for finding optimal policies?

Our first contribution is to indicate the mathematical model for the standard semantics of PDL_1:

Proposition 1 *Any planning problem expressed in* PDL_1 *defines an MDPST.*

Before we show the proof of this proposition, we review the theory of MDPSTs. We use the framework proposed by Trevizan et al (2007), who have employed MDPSTs to analyze *non-factored* planning problems with probabilistic and nondeterministic uncertainty. After the main facts about MDPSTs and their relation to planning are reviewed, we hope the proposition should become almost self-evident.

An MDPST consists of:
(M1) a state space $\mathcal{S}$ (assumed finite),
(M2) a nonempty set of initial states $\mathcal{S}_0$,
(M3) a nonempty set of actions $\mathcal{A}(s)$ (assumed finite) for each state $s \in \mathcal{S}$,
(M4) a cost $c(a, s)$ for taking action $a \in \mathcal{A}(s)$ in state s,
(M5) a state transition function $F(a, s)$ that maps state s and action $a \in \mathcal{A}(s)$ into a *set of subsets* of $\mathcal{S}$,
(M6) a mass assignment $m(k|a, s)$ that assigns a number $p \in [0, 1]$ to every set in $F(a, s)$ such that they add up to one: $\sum_k m(k|a, s) = 1$.
We may also have a set of goal states (if the problem of interest prescribes them) or a discount factor (if we are interested in infinite horizon with discounted cost).

The intuition is that any MDPST is an MDP where transitions may happen probabilistically from a given state into a *set* of states. Once a transition is made into a set of states, the resolution of any remaining uncertainty is nondeterministic. As discussed by Trevizan et al (2007), deterministic planning is obtained when $F(a, s)$ yields a singleton $\{s'\}$ for every (a, s). Nondeterministic planning is obtained when $F(a, s)$ yields a single subset k of $\mathcal{S}$, where $|k| \geq 1$. Probabilistic planning is obtained when $F(a, s)$ is a set of singletons, and then the mass assignment $m(k|a, s)$ is the usual transition probability $P(r|a, s)$ for each $\{r\} \in F(a, s)$.

MDPSTs are clearly more expressive than MDPs, and the latter are special cases of the former. But MDPSTs are not the only variant of MDPs that generalize the transition probabilities. *Markov Decision Process with Imprecise Probabilities (MDPIPs)* are even more general (Satia and Lave Jr. 1970; White III and El-Deib 1994): an MDPIP is an MDP with a *set of transition probabilities* amongst states. Any MDPST is an MDPIP; this is a consequence of the fact that a mass assignment over subsets of a set defines a *Choquet capacity of infinite order* (Dempster 1967) because it prescribes a probability distribution (the mass assignment) followed by a multivalued mapping (the state transition function).[2] We also note that BMDPs as proposed by Givan, Leach, and Dean (2000) are MDPIPs as the transition probabilities are encoded by probability intervals, but MDPSTs and BMDPs define classes of models that are not contained in each other (Trevizan, Cozman, and de Barros 2007).

The important fact about MDPSTs is that a very attractive version of the Bellman equation can be written for their minimax policies. By minimax we mean a policy that minimizes the maximum expected cost — minimization with respect to policies, and maximization with respect to the allowed transition probabilities.[3] First, note that results from the theory of MDPIPs show that backward induction yields:

$$V(s) = \min_a \max_P \left(c(a,s) + \gamma \sum_r P(r|a,s) V'(r) \right), \quad (1)$$

where again the minimization is over actions and the maximization is over transition probabilities. This equation applies to a specific time (V is the value function at the next time point and V' is the value function at the current time point) or to infinite time (V and V' both refer to the optimal value function); existence and stationarity of the optimal policy for infinite horizon can be shown (Satia and Lave Jr. 1970; White III and El-Deib 1994). The search for optimal policies in MDPIPs (and in BMDPs) is based on the solution of Equation (1). Now for MDPSTs the backward induction argument leads to a simplified equation:

$$V(s) = \min_a \left(c(a,s) + \gamma \sum_{k \in F(a,s)} m(k|a,s) \max_{r \in k} V'(r) \right), \quad (2)$$

Trevizan et al (2007) show the validity of this simplified Bellman equation. A concise proof can be produced by noting that Choquet capacities of infinite order always satisfy $\max_P E[f(x)] = E[\max_{y \in Z} f(y)]$ where the first expectation is with respect to the whole space of X while the second expectation is with respect to the space of mass assignments, encoded here by Z (Wasserman and Kadane 1992). Thus the maximum over summation in Equation (1) becomes a summation over maxima in Equation (2).

```
(define (domain tire) (:requirements :adl) (:types location)
  (:predicates (road ?from - location ?to - location)
      (vehicle-at ?loc - location) (spare-in ?loc - location)
          (not-flattire) (hasspare))
  (:action load-tire
    (:parameters (?loc - location))
    (:precondition (and (vehicle-at ?loc) (spare-in ?loc)))
    (:effect (and (hasspare) (not (spare-in ?loc))))
  )
  (:action move-car
    (:parameters (?from - location ?to - location))
    (:precondition (and (vehicle-at ?from)
                        (road ?from ?to) (not-flattire)))
    (:effect (and (vehicle-at ?to) (not (vehicle-at ?from))
                  (probabilistic 2/5 (not (not-flattire)))))
  )
  (:action change-tire
    (:precondition (hasspare))
    (:effect (oneof (and (not (hasspare)) (not-flattire)) (and)))
  )
)
```

Figure 1: The `tire` domain in PDL_1, with actions `load-tire` (deterministic), `move-car` (probabilistic), and `change-tire` (nondeterministic) together. This domain defines an MDPST (Proposition 1).

The simplified Bellman equation represented by Equation (2) leads to important computational simplifications, as the maximization required by Equation (1) becomes a search for maxima inside the sets k, presumably containing significantly less elements than the whole space. In fact, the theory of MDPSTs is much closer to the theory of MDPs than to the theory of MDPIPs. Trevizan et al explore these facts to produce efficient value/policy iteration schemes for *non-factored* MDPST; here "efficient" means "as efficient as for an MDP of identical size". We return to issues of computational efficiency in a later section.

At this point we can present a short proof for Proposition (1):
Proof of Proposition (1). Given a planning problem in PDL_1, construct an MDPST where the state space is the product space of predicates, the initial states, costs and goal are as encoded in PDL_1, and actions are as follows. For a probabilistic action a' in PDL_1, define $F(a', s)$ as a set of singletons where each singleton is in the support of the transition probabilities, and the mass assignment is just identical to the probabilities expressed in PDL_1 (following the PPDDL 1.0 semantics (Younes and Littman 2004; Younes et al. 2005)). For a nondeterministic action a'' in PDL_1, define $F(a'', s)$ as a single set containing as many elements as there are deterministic effects in the `<one-of>` construct. □

Example 1 Consider the following example, based on versions of the `tire` problem used in Planning Competitions (Bonet and Givan 2005; Bryce and Buffet 2008). The problem is to plan a trip by car from one city to another. One may have to deal with flat tires during the trip. The original description of `tire` included probabilisties for getting a flat tire and for finding difficulty in changing a flat tire.
Figure 1 shows a description of `tire` in valid PDL_1 syntax with probabilities on `move-car` and nondeterministic effects on `change-tire`. Suppose there is vast statistical evidence that the probability of a flat tire during an interme-

[2]Choquet capacities of infinite order are also called *belief functions* (Shafer 1976) and *random sets* (Molchanov 2005). The construction of a set of probability distributions out of a Choquet capacity of infinite order is standard (Cozman 2000): the set is the convex hull of all distributions produced by selecting atomic events inside each subset k (in $F(a,s)$ for each a and s).

[3]There are other criteria that apply when probabilities are not precisely specified (Kikuti, Cozman, and de Campos 2005; Seidenfeld 2004; Troffaes 2004; Utkin and Augustin 2005); we focus on the minimax criterion because nondeterministic planning itself generally adopts a minimax approach.

diate trip is 2/5. The original version of the `tire` domain represented this situation by assigning 2/5 to the probability of a flat tire. However, suppose there is really no evidence on the behavior of drivers concerning the change of tires. The original version of the `tire` domain presumably addressed this situation by assigning 1/2 to the probability that a driver actually succeeds in changing a tire. In Figure 1 we leave indeterminate the probability that one succeeds in changing tires.
The optimal policy for this example is discussed in a later section. □

PDL_2, MDPSTs and MDPIPs

We have seen that PDL_1 allows probabilistic and nondeterministic actions to coexist. However, the mix is not terribly expressive when it comes to representing uncertainty. Consider two unsatisfying aspects of Example 1:

1. In a real trip, a flat tire may cause the trip to halt and the car to return to its origin. A more realistic description of the action `move-car` should consider the possibility that, when a tire is flat, the car may not reach its intermediate goal. There is thus probability 3/5 that a car reaches its intermediate goal with intact tires, and probability 2/5 that a car, having a flat tire, *either* reaches its intermediate goal *or* not (without any probability on this latter disjunction).

2. The nondeterminism of the action `change-tire` is too extreme, for it would imply that, *in the worst case*, the probability of actually succeeding in changing a tire is zero. But we might expect that any driver would eventually succeed in changing a tire, given enough trials, or perhaps even by calling upon professional assistance.

The following example avoids such difficulties:

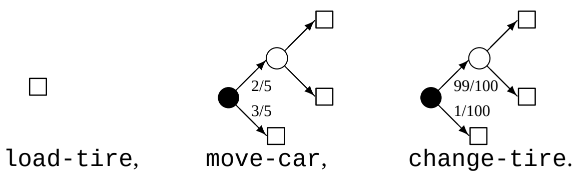
Example 2 Figure 2 describes the `tire` domain discussed in Example 1, with two modifications. Action `move-car` now expresses the fact that a car may or may not reach its intermediate goal when a tire is flat. And action `change-tire` now reflects the fact that any driver can change a tire at least with some small probability (in this case, at least 1/100). The semantics is to be presented shortly, but the meaning of this example should be easy to grasp.
The optimal policy for this example is discussed in a later section. □

Clearly the ability to mix probabilistic and nondeterministic effects within actions dramatically increases expressivity. But does it keep us within the boundaries of MDPSTs, so that we are still quite close to MDPs as far as computations go?

Start with Example 2. Under the obvious semantics for actions, this example does define an MDPST. To see why, consider a represention for actions that uses trees with deterministic nodes (boxes), probabilistic nodes (filled circles) and nondeterministic nodes (void circles):

```
(define (domain tire) (:requirements :adl) (:types location)
  (:predicates (vehicle-at ?loc - location)
     (road ?from - location ?to - location)
        (spare-in ?loc - location) (not-flattire) (hasspare))
  (:action load-tire
    (:parameters (?loc - location))
    (:precondition (and (vehicle-at ?loc) (spare-in ?loc)))
    (:effect (and (hasspare) (not (spare-in ?loc))))
  )
  (:action move-car
    (:parameters (?from - location ?to - location))
    (:precondition (and (vehicle-at ?from)
                        (road ?from ?to) (not-flattire)))
    (:effect (probabilistic
      2/5 (oneof (and (vehicle-at ?to) (not (vehicle-at ?from))
                      (not (not-flattire)))
                 (not (not-flattire)))
      3/5 (and (vehicle-at ?to) (not (vehicle-at ?from))
               (not-flattire))))
   )
  (:action change-tire
    (:precondition (hasspare))
    (:effect (probabilistic
               99/100 (oneof (and (not (hasspare)) (not-flattire))
                             (and))
                1/100 (and (not (hasspare)) (not-flattire)))
  )
)
```

Figure 2: The `tire` domain in PDL_1, with actions `load-tire` (purely deterministic), `change-tire` and `move-car` (mixtures of probabilistic/nondeterministic) together. This domain defines an MDPST (Theorem 1).



`load-tire`, `move-car`, `change-tire`.

In the last two drawings, a probabilistic node defines a mass assignment over sets of states, and each one of these sets is defined by a nondeterministic node. Thus we can simply extract the mass assignments m and the state transition functions F from the drawings.

We can generalize as follows. Consider an action a with `probabilistic` and `oneof` elements, and build a tree T_a that encodes the structure of a. A `probabilistic` element in a produces a probabilistic node with as many children as the `probabilistic` element has effects. An edge in T_a leaving a probabilistic node is labelled with the probability that the effect pointed to by the edge obtains. And a `nondeterministic` element in a produces a nondeterministic node with as many children as the `nondeterministic` element has effects. The leaves of T_a are the deterministic effects in a. We call a *path* in T_a a maximal sequence of consecutive probabilistic nodes starting from the root (and obviously stopping either before a nondeterministic node or before a deterministic node). A path may be empty.

Suppose any path from the root of T_a to a leaf meets probabilistic effects before nondeterministic effects. We can then easily define reasonable semantics for the actions using the semantics of PPDDL 1.0. The *probability of a path* is defined as the product of all probabilities attached to the edges in the path; if the path is empty its probability is 1 by definition. This is a direct application of the rule used to combine probabilities in PPDDL 1.0 (Younes and Littman 2004, page

9):

$$p_1(p'_1e'_1|\ldots|p'_ke'_k)|p_2e_2|\ldots|p_ne_n \equiv$$
$$(p_1p'_1)e'_1|\ldots|(p_1p'_k)e'_k)|p_2e_2|\ldots|p_ne_n.$$

The *transition of a path* is defined as the set of all deterministic descendants of the last probabilistic node in the path; if the path is empty its transition is the root of the tree by definition.

Now these definitions do produce an MDPST: the mass assignment $m(k|a, s)$ for an action a are the path probabilities in T_a, and the state transition functions $F(a, s)$ are the transitions in T_a.

We emphasize this nice construction by defining an extension of PDL_1:

Definition 1 *The Planning Definition Language version 2.0 (*PDL_2*) has the same syntax as* PDL_1 *but allows nesting of* `probabilistic` *and* `oneof` *elements, provided that all* `probabilistic` *elements come before the* `oneof` *elements in a single action.*

The semantics of PDL_2 inherits the semantics of PDL_1 plus the interpretation of actions using path probabilities and transitions as discussed before Definition 1. Given that discussion, it should be clear that:

Theorem 1 *Any planning problem described in* PDL_2 *defines an MDPST.*

Can we generalize even further? The next obvious step would be to let a probabilistic node to appear after a nondeterministic node in T_a. But it is not possible to do so and stay within the confines of MDPSTs. This can be seen considering an action as follows:

```
(:action dummy
   (:parameters ...)
   (:precondition (...))
   (:effect (oneof
     (probabilistic $p_{11}e_{11}\ldots p_{1n}e_{1n}$)
     ⋱
     (probabilistic $p_{m1}e_{m1}\ldots p_{mn}e_{mn}$)))
).
```

With such an action we can describe any finite set of distributions, so we are beyond Choquet capacities of infinite order and consequently of MDPSTs. As far as minimax policies are concerned, we have moved to the realm of MDPIPs modeled by general sets of transition probabilities with finitely many vertices (Walley 2000). This is so because there is no difference between a set of probabilities and its vertices:

Proposition 2 *In Equation (1) any maxima with respect to probabilities must be attained at vertices of the sets of transition probabilities.*

Proof. Suppose the maxima is attained at an interior point of the set for $P(r|a, s)$ for some action a. This interior point must necessarily be a convex combination of vertices, and given the linearity of summation over states, the maximum value must be a convex combination of the values of the summation at the vertices. Thus either both vertices also attain the maximum, or else the vertices do not attain the maximum and the interior point (being a convex combination of the vertices) fails to do so as well, contradicting the hypothesis that it attains the maximum. □

It should be noted that the resulting factored MDPIPs have a very interesting property: all vertices of the set of probabilities distribution are explicitly represented by possible paths in trees representing actions. Due to Proposition 2, we can exploit this structural property to solve factored MDPIPs using the minimax criterion because $P^* = \mathrm{argmax}_P\, E[f(x)]$ is always a vertex of P. We leave this investigation for the future.

We now turn to the problem of computing optimal policies for planning problems expressed in PDL_2.

Solving factored MDPSTs

Given the form of the Bellman equation for MDPSTs (2), most of the MDPs algorithms can be adapted to solve MDPSTs (Trevizan, Cozman, and de Barros 2007). For instance, suppose one runs the value iteration algorithm (Puterman 1994), but uses Equation (2) instead of the usual Bellman equation. This kind of value iteration has the same theoretical guarantees of the original value iteration (e.g., convergence to a minimax stationary policy for the discounted cost with infinite horizon).

The more interesting case for planning problems is goal reachability with indefinite horizon. This situation can also be tackled for MDPSTs by adapting existing algorithms. Before presenting the actual algorithm, it is worth pausing to specify a common assumption made when solving goal-driven problems, the *reachability assumption* (Barto, Bradtke, and Singh 1995). For MDPs, this assumption can be stated simply as: a goal state *must be reachable* from every state. Clearly the reachability assumption aims at ensuring that the problem being solved does not have dead ends so that it can be safely explored by greedy algorithms.

For MDPSTs, the reachability assumption requires some refinement because we do not know probability distributions over effects. The assumption must hold for all probability distributions defined by the mass assignment function m. Thus the reachability assumption for MDPSTs can be stated as: for all possible probability distributions P induced by m, a goal state must be reachable from every state if P is adopted as the probability distribution for state transitions. If we interpret an MDPST M as a special encoding of a (infinite and uncountable) set of MDPs, then the reachability assumption for M just ensures that every MDP M_i encoded by M respects the usual reachability assumption for MDPs. Although this assumption is harder to be proved for MDPSTs, if, $\forall$ state s, $s' \in \mathcal{S}$, action $a \in A(s)$ and $k \subseteq \mathcal{S}$, $m(k \cup \{s'\}|a, s) > 0$ implies $P(s'|a, s) > 0$, then it suffices to prove the MDP reachability assumption for any MDP M_i encoded by M (that is for any valid choice of P).

When the reachability assumption is satisfied by an MDPST, we can apply an adapted version of any algorithm based on real time dynamic programming (RTDP) (Barto, Bradtke, and Singh 1995), such as Labeled RTDP (Bonet and Geffner 2003), Bounded RTDP (McMahan, Likhachev, and Gordon 2005) and Heuristic Dynamic Programming

(Bonet and Geffner 1998). The adaptation of these algorithms is straightforward: just replace the Bellman equation for MDPs by its MDPST counterpart. Due to space restrictions, we present a high level description of the adapted RTDP, depicted in Figure 3.

Two other points are worth mentioning about algorithms based on RTDP: (1) the termination proof for RTDP; and (2) the implementation of the auxiliary method `Simulate` that is required by RTDP.

The termination proof for the RTDP is based on the fact that, if the reachability assumption holds, then in a finite number of steps (calls to `Simulate`) the goal will be reached (Barto, Bradtke, and Singh 1995). Using the reachability assumption for MDPSTs, the termination proof holds, and the RTDP (and derivated methods) are guaranteed to stop.

The second point that should be mentioned is the auxiliary method `Simulate` used by RTDP (Figure 4). This method simulates the environment and returns one of the possible resulting states when action a is applied in state s. Because the exact probability distribution is not known *a priori*, we can use any probability measure P defined by m to simulate the environment. The minimax probability distribution P_{max} obtained by the $\max$ operator in Equation (1) could be used in this step. However, P_{max} does not guarantee that a closed policy π with respect to $\mathcal{S}_0$ (that is a policy defined over all states reachable by following π from $s_0 \in \mathcal{S}_0$) can be found. This is so because for each $k \in F(a, s)$, P_{max} assigns all probability mass of k to exactly one state s'; thus the states in $k \setminus \{s'\}$ may not be visited. If the states in $k \setminus \{s'\}$ are reachable only through s, then a starvation scenario may happen. In such a case a closed policy cannot be found because it has to be defined for all $s \in k$. To guarantee that a closed policy with respect to $\mathcal{S}_0$ is found, we need to assign a non-zero probability to every transition that $P_{\text{max}} > 0$, that is, for all $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}(s)$ such that $\sum_{k \subseteq \mathcal{S}} m(k \cup \{s'\}|s, a) > 0$, then $P(s'|a, s)$ should be greater than 0 too. Because this approach allows any state $s' \in k$ to be chosen, eventually all states in k are visited, what is the condition needed to define a closed policy π.

In order to make the algorithms based on RTDP more efficient, we need to provide an admissible heuristic (that is, a heuristic that never overestimates the optimal cost). One example of admissible heuristic is the zero-heuristic: $h_0(s) = 0$ for all $s \in \mathcal{S}$. Another possible admissible heuristic is the *min-min* (Bonet and Geffner 2005). For MDPs, the value of this heuristic for a state s is defined as the optimal cost from s to the goal in the *weak relaxation* of the original problem. The weak relaxation of a probabilistic planning problem is a deterministic problem computed by transforming all the probabilistic actions in a set of deterministic actions. For MDPSTs, the same procedure can be employed, by replacing each action a of the form:

$$\langle pred; (p_1(e_1^1 \oplus \cdots \oplus e_{n_1}^1)| \cdots |p_r(e_1^r \oplus \cdots \oplus e_{n_r}^r))\rangle$$

by $\prod_{i=1}^{r} n_i$ deterministic actions of the form $\langle pred; e_{j_1}^1 \wedge \cdots e_{j_r}^r\rangle$, where $e_{j_i}^k$ is a deterministic effect and $j_i \in \{1, \cdots, n_k\}$. This weak relaxation for MDPSTs give us a lower bound of the optimal cost because the (deterministic) planner can choose exactly the effect to be applied, instead of relying on the uncertainty of the environment.

Experiments

The potential gain of any mixture of probabilistic and nondetermistic planning lies in the increased expressivity; however, this gain is lost if the resulting complexity is too high. We have argued throughout the paper that MDPSTs not only increase expressivity, but also have nice computational properties as they are close cousins of MDPs. These statements, although based on similarity of theoretical properties, must be tested empirically before acceptance. Such an empirical evaluation is what we report in this section. With this in mind, we have implemented optimized versions of Labeled RTDP (LRTDP) schemes that can take planning problems

```
Input: an MDPST $q = \langle \mathcal{S}, \mathcal{S}_0, \mathcal{A}, F, C\rangle$, an admissible
       heuristic $H$ and $\epsilon$ the maximum estimation error.
Output: The $\epsilon$-optimal closed (with relation to $s_0$)
        policy $\pi$ for $q$.
begin
    repeat
        $H \leftarrow$ RTDP-Trial-SPST$(q, H)$
        $\pi_H \leftarrow$ Greedy-Policy$(q, s_0, H)$
    until $\delta_{max} < \epsilon$
    return $\pi_H$
end
```

Figure 3: Adapted version of the RTDP algorithm for solving MDPSTs. The definition of δ_{max} is $\max_{s \in S_{\pi_H}^{s_0}} |H(s) - \min_{a \in \mathcal{A}(s)}\{C(a, s) + \sum_{k \in F(a,s)} m(k|a, s) \max_{s' \in k} H(s')\}|$. The `RTDPTrialMDPST` is depicted in Figure 4.

```
Input: an MDPST $q = \langle \mathcal{S}, \mathcal{S}_0, \mathcal{A}, F, C\rangle$ and an
       admissible heuristic $H$.
Output: an admissible heuristic, tighter or equal to $H$
begin
    while $s \notin S_G$ do
        $a \leftarrow \operatorname{argmin}_{a' \in A(s)}\{C(a', s) + \sum_{k \in F(a',s)} m(k|a', s) \max_{s' \in k} H(s')\}$
        $H(s) \leftarrow C(a, s) + \sum_{k \in F(a,s)} m(k|a, s) \max_{s' \in k} H(s')$
        $s \leftarrow$ Simulate$(q, s, a)$
    return $H$
end
```

Figure 4: Auxiliary method for the MDPST version of RTDP. For each iteration of this method, H is used to find a path from the initial state to a goal state. The method `Simulate` is discussed in the text.

expressed in PDL$_2$ and produce plans efficiently — again, "efficient" means "as efficient as MDPs".

The set of experiments we have resorted to the following mixture of probabilistic and nondeterministic effects. Every probabilistic effect in the original domain definition is replaced by a convex mixture of the original distribution and a nondeterministic choice; that is, if P_0 is the original distribution, the set of distributions we consider is

$$(1 - \epsilon)P_0 + \epsilon Q$$

where Q is an arbitrary distribution (thus encoding the nondeterminism). This kind of model is in fact widely employed in robust statistics under the name of ϵ*-contaminated* class (Berger 1985; Huber 1980). For our experiments, we have 0.1-contaminated versions of the blocks world and the exploding blocks world generated by replacing every probabilistic action of the form $\langle pred; (p_1 e_1 | \cdots | p_r e_r) \rangle$ by the following new action $\langle pred; (0.9 \times p_1 e_1 | \cdots | 0.9 \times p_r e_r | 0.1 (e_1 \oplus \cdots \oplus e_r)) \rangle$. We have also tested the tire domain described in Figure 2.

Figure 5 summarizes the running time of the LRTDP for these three domains. For the MDP case, we split the probability mass $m(k|a, s)$ uniformly between the states in k, resulting in the MDP version of the tested problems. We also initialized the LRTDP with two different heuristics, the zero-heuristic and the min-min heuristic. As one may notice, the running time for MDPSTs is near to a constant factor distance from the running time for MDPs. This corroborates the theoretical result, since the computation of policies for MDPSTs consists in adding the overhead of the $\max$ operator in (2). However, this small running time difference is compensated by the quality of the solution, since the policy generated by MDPSTs offers an extra theoretical guarantee – it has the best performance in the worst case (minimax).

Conclusion

In this section we would like to revisit the remarks made in the last paragraph of the Introduction. As noted there, the results in this paper contribute: (1) to the quest for a planning description language that balances expressivity and complexity; and (2) to a better understanding of the interface between probabilistic and nondeterministic planning.

The results in this paper suggest that we should at least leave the syntax of PDL$_1$ as relaxed as it is now concerning mixtures of probabilistic and nondeterministic actions. There is a small cost to be paid in dealing with MDPSTs, as the theoretical and computational tools to use are rather close to their MDP counterparts. And the results of this paper suggest that we could go even further in relaxing the language. PDL$_2$ seems to be a viable relaxation, because it increases expressivity considerably, by allowing true mixtures of probabilistic/nondeterministic effects, while paying a small cost.

With regard to the interface between probabilistic and nondeterministic planning, the results in this paper suggest the theory of MDPSTs to be a solid starting point when dealing with *factored* planning problems. This observation adds to the results by Trevizan et al (2007) for the non-factored

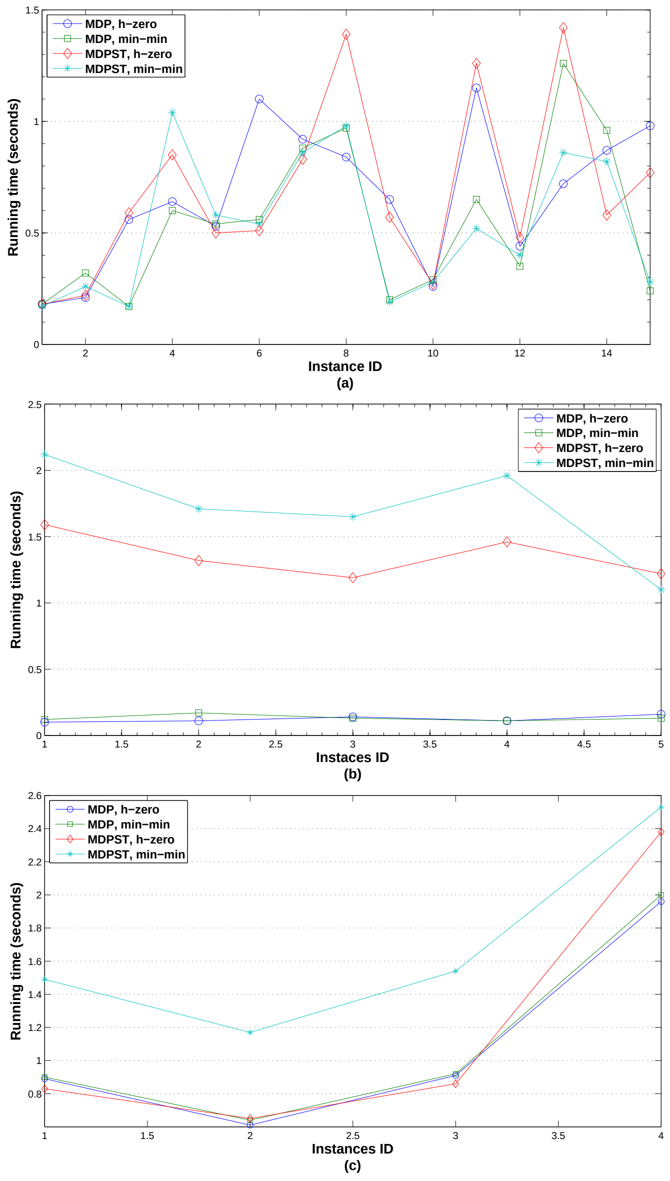


Figure 5: Running time of the LRTDP for the final problems of the 5th International Planning Competition. The chosen domains are: (a) tire world, (b) blocks world, and (c) exploding blocks world.

case. Again, the main point is that the added flexibility costs little in conceptual and computational terms.

In short: instead of MDPs, one can easily use MDPSTs; instead of PDL_1, one can use PDL_2. However, it is apparent that MDPSTs have their own limitations concerning expressivity; it seems that more general combinations of `probabilistic` and `oneof` constructions will take us directly into the theory of MDPIPs. Even though MDPIPs are harder to handle, their factored form presents the interesting property of explicitly decode all the vertexes of the set of probabilities distributions. This structure can be exploited in order to develop efficient planning algorithms for factored MDPIPs. As additional future research directions, we highlight the translation to MDPSTs (and MDPIPs) of important concepts for MDPs, such as the *strong relaxation* and heuristics based on this concept, for instance the atom-min-forward, atom-min-backward and ff heuristics.

References

Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 72(1-2):81–138.

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

Bonet, B., and Geffner, H. 1998. HSP: Heuristic search planner. *AIPS-98 Planning Competition, Pittsburgh, PA.*

Bonet, B., and Geffner, H. 2003. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *Proc. of the 13th ICAPS*, 12–21. Trento, Italy: AAAI Press.

Bonet, B., and Geffner, H. 2005. mGPT: A probabilistic planner based on heuristic search. *Journal of Artificial Intelligence Research* 24:933–944.

Bonet, B., and Givan, R. 2005. 5th international planning competition: Non-deterministic track — call for participation.

Bryce, D., and Buffet, O. 2008. 6th international planning competition: Uncertainty part.

Cozman, F. G. 2000. Credal networks. *Artificial Intelligence* 120:199–233.

Dempster, A. P. 1967. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics* 38:325–339.

Fox, M., and Long, D. 2003. PDDL2.1: An extension of PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

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

Huber, P. J. 1980. *Robust Statistics*. New York: Wiley.

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

McMahan, H.; Likhachev, M.; and Gordon, G. 2005. Bounded real-time dynamic programming: RTDP with monotone upper bounds and performance guarantees. *ACM International Conference Proceeding Series* 119:569–576.

Molchanov, I. 2005. *Theory of Random Sets*. Springer.

Puterman, M. L. 1994. *Markov Decision Processes*. John Wiley and Sons.

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

Seidenfeld, T. 2004. A contrast between two decision rules for use with (convex) sets of probabilities: γ-maximin versus e-admissibility. *Synthese* 140(1-2):69–88.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press.

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

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

Utkin, L. V., and Augustin, T. 2005. Powerful algorithms for decision making under prior information and general ambiguity attitudes. In *Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications*, 349–358.

Walley, P. 2000. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning* 24:125–148.

Wasserman, L., and Kadane, J. B. 1992. Computing bounds on expectations. *Journal of the American Statistical Association* 87(418):516–522.

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

Younes, H. L. S., and Littman, M. L. 2004. PDDL1.0: An extension to PDDL for expressing domains with probabilistic effects. Technical Report CMU-CS-04-167, Carnegie Mellon University, Pittsburgh, PA 15213.

Younes, H. L. S.; Littman, M. L.; Weissman, D.; and Asmuth, J. 2005. The first probabilistic track of the international planning competition. *Journal of Artificial Intelligence Research* 24:851–887.